BLOOD VESSEL DETECTION IN RETINAL IMAGES AND ITS APPLICATION IN DIABETIC RETINOPATHY SCREENING

A Dissertation

by

MING ZHANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2008

Major Subject: Computer Science

BLOOD VESSEL DETECTION IN RETINAL IMAGES
AND ITS APPLICATION IN DIABETIC RETINOPATHY SCREENING

A Dissertation

by

MING ZHANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Jyh-Charn Liu
Committee Members,	Dezhen Song
	Yoonsuck Choe
	Christopher Quick
	Michael Freckleton
Head of Department,	Valerie Taylor

August 2008

Major Subject: Computer Science

ABSTRACT

Blood Vessel Detection in Retinal Images
and Its Application in Diabetic Retinopathy Screening. (August 2008)

Ming Zhang, B.S., Southeast University (China)
Chair of Advisory Committee: Dr. Jyh-Charn (Steve) Liu

In this dissertation, I investigated computing algorithms for automated retinal blood vessel detection. Changes in blood vessel structures are important indicators of many diseases such as diabetes, hypertension, etc. Blood vessel is also very useful in tracking of disease progression, and for biometric authentication. In this dissertation, I proposed two algorithms to detect blood vessel maps in retina. The first algorithm is based on integration of a Gaussian tracing scheme and a Gabor-variance filter. This algorithm traces the large blood vessel in retinal images enhanced with adaptive histogram equalization. Small vessels are traced on further enhanced images by a Gabor-variance filter. The second algorithm is called a *radial contrast transform* (RCT) algorithm, which converts the intensity information in spatial domain to a high dimensional radial contrast domain. Different feature descriptors are designed to improve the speed, sensitivity, and expandability of the vessel detection system. Performances comparison of the two algorithms with those in the literature shows favorable and robust results. Furthermore, a new performance measure based on central line of blood vessels is proposed as an alternative to more reliable assessment of detection schemes for small vessels, because the significant variations at the edges of small vessels need not be considered.

The proposed algorithms were successfully tested in the field for early diabetic retinopathy (DR) screening. A highly modular code library to take advantage of the parallel processing power of multi-core computer architecture was tested in a clinical trial. Performance results showed that our scheme can achieve similar or even better performance than human expert readers for detection of micro-aneurysms on difficult images.

To My Parents

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Dr. Jyh-Charn Liu. He has been more than an advisor to me for the past 6 years of my life. Dr. Liu always gave me trust, inspiration and encouragement, not only in my research, but also my career development and many other things in my life. His deep thoughts on my research topic guided me through the hardest time in my research and gave me the courage to conquer the most challenging problems. In every talk and discussion with Dr. Liu, I learned something new, and the time we spent in finding new ideas, in debate, or even in a warm casual talk would be the most pleasant time in my Ph.D study. Besides the guidance in research, Dr. Liu also helped me through a grim time in my life. I would like to say, from the bottom of heart, "Thank you" to Dr. Liu for his years of support.

I am immensely grateful to Dr. Dezhen Song, Dr. Yoonsuck Choe, Dr. Christopher Quick, Dr. Michael Freckleton, and Dr. Ricardo Gutierrez-Osuna for their support in my research. They gave me a lot of input to my research, and showed me many new things, either in a computer science aspect or a medical science aspect. I really appreciate their time spent with me on my research.

I would like to express my appreciation to all of my colleagues, and in particular to Mr. Huajun Ying. My cooperation with Mr. Ying has been very pleasant and productive. We worked together to get many important results, which are presented in this dissertation. I thank Dr. Di Wu, Dr. Yong Xiong, Dr. Tak-Cheung Lam, Mr. Hong Lu, Mr. Pu Duan, Mr. Cheng-Chung Tan, Mr. Shengya Lin and Mr. Oscar Almeida, who were members of a great team in the RTDS lab. I also would like thank all my friends at Texas A&M University. During my time here, every one brought me tons of joy and encouragement. Thank you all for being with me when I needed your help.

Finally, my deepest gratitude goes to my dear parents and my younger brother, who provide unreserved support to me.. Without them, I would not have studied abroad and finished my Ph.D.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

I. INTRODUCTION

The retinal blood vessel network provides nutrition to the eyes and removes waste from the retinal systems. While doing this, it must not interfere with the light transmission to the photoreceptors in the neural layer. Human eyes have evolved a two layer circulation systems to accommodate this need [1, 2]. The two circulation systems are uveal blood circulation and retinal blood circulation. The retina needs both systems to survive [3]. The uveal blood circulation supplies the outer layers of the eyes including the iris, ciliary body and the choroidal tissue. And the retinal blood circulation supplies the inner portion of the eye as far as the neural layer. The two circulation systems don't connect to each other and complement each other to provide full blood support to the eyes. In this work, we focus on the retinal circulation system. The nutrition and oxygen rich blood are delivered by highly elastic, thick walled arteries to the retina. Arteries and veins extend to smaller arterioles and venules, respectively and are connected with the capillary bed [4]. After nutrition and waste exchange in the capillary bed, the blood is returned by highly expandable, thin walled veins to the heart. The arterioles, capillaries and venules are collectively called the microcirculation of the retina (See Figure 1).

This dissertation follows the style of *IEEE Transactions on Medical Imaging*.

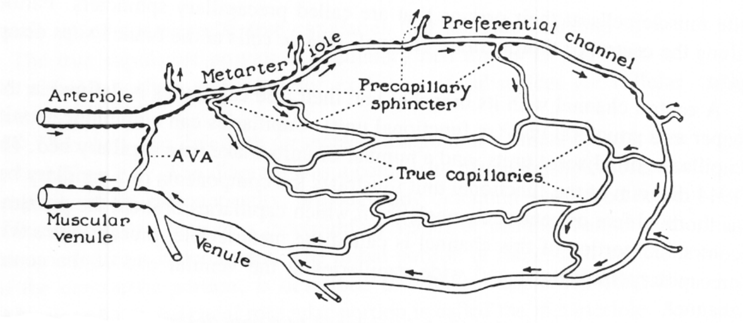


Figure 1 The schematic representation of a capillary bed. (Source: [4])

A complete vessel network illustration can be found in Figure 2. Vessels which can be seen in (A) are large vessels including arteries and veins; those in (B) are arterioles and venules, they connect to the capillaries, deliver oxygen and nutrition to the tissue, as illustrated in (C). The arterioles and venules in retinal circulation usually have diameter less than 20 μm, and capillaries less than 6 μm [1]. When discussing the microcirculatory vessels, it’s important and more accurate to identify them with their physical characteristics, not solely with their diameter. The diameter can change as a result of disease or blood flow. The relevant diameters of the microcirculation vessels are listed in Table 1.

Table 1 Relevant vessel diameters of the retina.

Vessel	Diameter
Central Retinal Artery (Retinal Circulation)	.3mm
Arterioles (Retinal Circulation)	<20µm
Venules (Retinal Circulation)	<20µm
Capillaries (Retinal Circulation)	<6µm

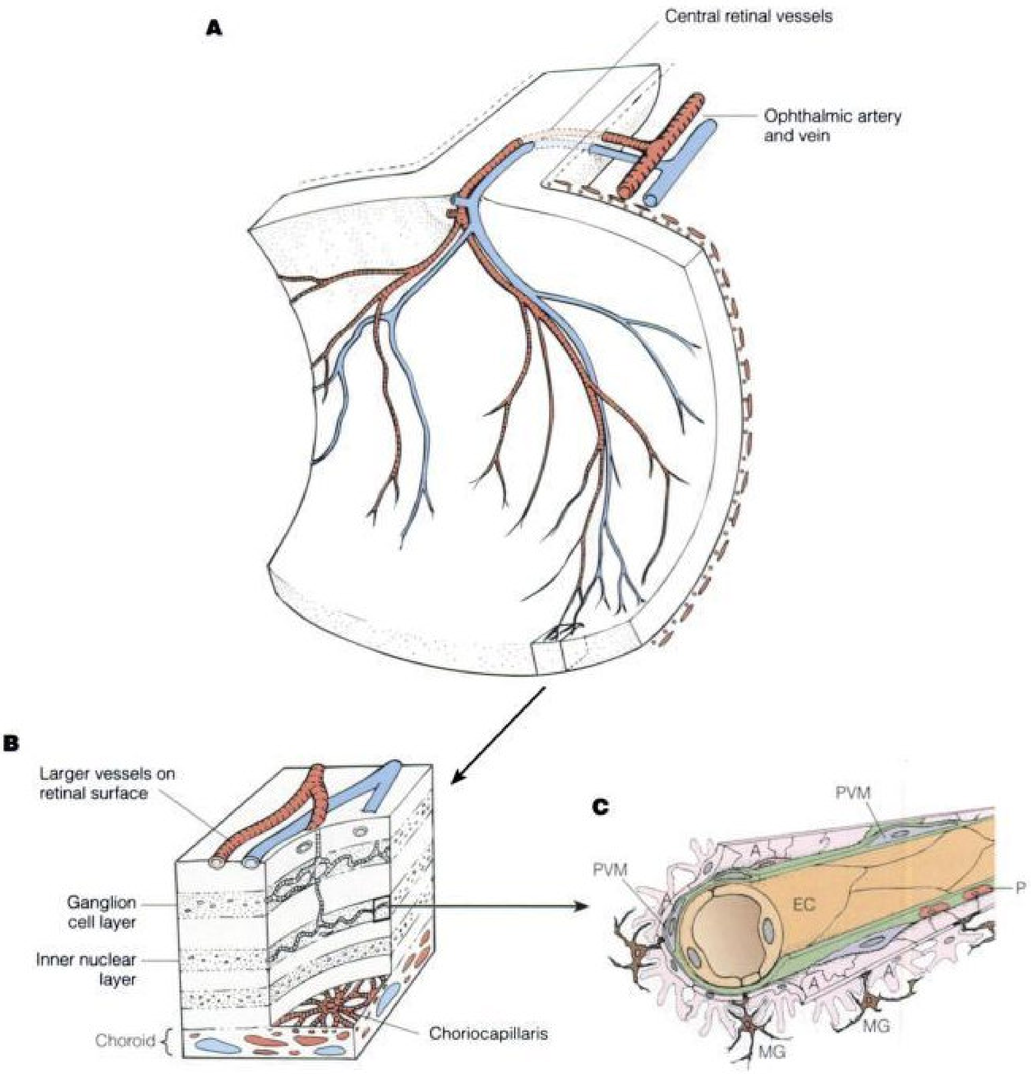


Figure 2 Blood vessels in retinal images. (Source: [5] . Courtesy of Dr. J.V. Forrester)

Several retinal implications have been related to vascular anomalies and structure change, which includes diabetic retinopathy, glaucoma, retinal artery occlusion and macular degeneration [6, 7]. These implications may disturb the blood supply in the retina and increase the risk of visual impairment [8-11]. The characteristics of the diseases that are related to our Early Diabetic Retinopathy Screening system (EDRSS) will be discussed later when necessary.

Many automated blood vessel detection algorithms have been proposed in decades of study. A blood vessel map is the basis of many applications. Many diseases such as diabetic retinopathy, hypertension and arteriosclerosis, are associated with anomalies of blood vessels. By monitoring the change in diameter, location and tortuosity of blood vessels, severe diseases may be predicted in the early stages and thus increase the possibility of a cure. As the only obvious structure existing in every retinal image, the blood vessel map can be used as landmarks of image registration and biometric authentication.

In the early years, vessel detection was mostly applied to images shot with the help of a contrast enhancer, such as fluorescein angiography and indocyanine green angiography images, or special instruments, e.g., scanning laser ophthalmoscope images, in which blood vessels have high contrast against the background (See Appendix A for more detail). In recent years, color fundus images have received more attention for its rich color representation, low cost and non-invasive nature. Fundus images are taken through the patient’s pupils. In the past, images were produced for 35mm film with old film cameras. More recently, digital cameras have been widely used and the resolution has been dramatically improved to 1024x1024, which will capture most of the structural details of the retina. However, this resolution will not resolve the fine capillary details. Nowadays, higher resolution cameras can capture rich fine details in the retina. Several things can affect the image quality: eye movement as well as the optical systems of the eye or even the lens system of the instrument itself can cause distortion of the image; dirt/dust speckles may be sources of noise in the image; or whether the pupils are dilated. Pupil dilation is an option in fundus photography. With dilated pupils, more lights can enter the retina and clearer images can be obtained. Dilation can cause medical complications for some patients. Therefore, it should be avoided whenever possible. Both mydriatic and non-mydriatic cameras are available. Figure 3 is an example fundus photograph of a normal retina from the National Institutes of Health.

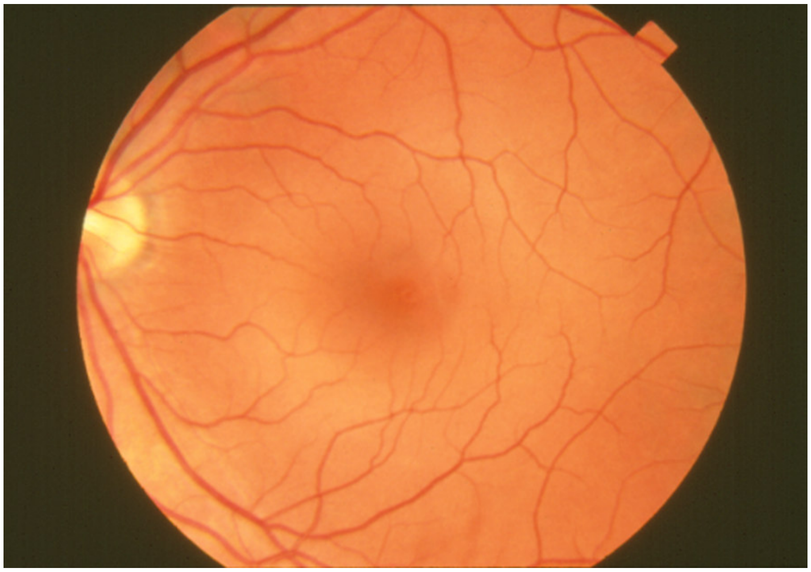


Figure 3 An example of fundus photography. (Source: National Eye Institute, National Institutes of Health)

The operator of the camera can target various anomalies by focusing on the specific layer of the retina during the imaging process. A table of identifiable abnormal features in different retinal layers can be found in [12]. Fundus photography's non-invasive nature and the relatively easy operation to take quality images makes it a major choice for large scale screening of disease.

Color fundus images also have it's own drawbacks. Without the help of a contrast enhancer, the contrast of objects such as blood vessels and some lesions in the fundus images are usually not as high as those in angiography images. Also, the illumination conditions, such as flash level, and room lighting, may dramatically affect the quality of fundus images. Additionally, different races of people have retinas of different pigments. For example, Caucasians usually have red color retinas and Asians and Hispanics have brown color retinas while Africa Americas usually have dark color retinas. This causes the problem of parameter adjustment if the detection algorithm is sensitive to the overall intensity. Also, dark pigment of retina will further reduce the vessel contrast against the background. Existence of diseases often causes false detections, e.g., the area between two bright hard exudates that are close to each other may be incorrectly recognized as part of blood vessel since its characteristics are pretty similar to those of vessels.

In [1], it was summarized the width of central retina artery is about 0.3mm (300μm), the arteriole/venule is less than 20μm and the capillary is less than 6μm. According to the camera specification from Canon Inc., the resolution of the camera we used, Canon 20D, is 6.4μm/pixel in the largest image size (3504x2336 pixels). So, in the largest image size, width of the arteries are about 50 pixels, the arterioles and venules is larger than 3 pixels and the capillaries are about 1 pixel. For computational efficiency, full resolution images are often downsized In a downsized image with size 720x480 (4.87 times smaller), the diameter of the artery is about 10 pixels, the arterioles and venules are about 1 pixel, and the capillaries are basically invisible. Our discussion below will be based on downsized images if not particularly specified. The diameter of FOV (field of view) in downsized images taken by Canon CR-DGi is about 600 pixels and that of STARE [13], a popular vessel detection benchmark image database, is about 650 pixels. And the diameters of optic disc in these two are both 80-90 pixels. So the diameters of different kinds of vessels in STARE images are about the same as those in Canon images. Thus, we can say the diameters of arterioles and venules in STARE are also between 1-10 pixels. Vessels with width less than or equal to 3 pixels are usually shallow while other with larger widths usually have much higher contrast against the background. We divided the whole vessel maps in STARE into two categories: those with diameter less than or equal to 3 pixels, called small vessels (arteriole or venule) and those with larger diameters, called large vessels. STARE database will be used to test the proposed algorithms later.

The large vessel detection is relatively easy because of their high contrast against the background. The small vessels are usually very shallow and detection of them is pretty difficult. Small vessels deliver important information on health condition of the eyes. Appearance of neovascular, which are very fine small vessels, is an indictor of late stage diabetic retinopathy. The tortuosity of large vessels usually won't change greatly, while that happens a lot on small vessels if some disease is in development. Without accurate small vessel detection, the detection of all these indicators is not possible. The length of small vessels in a retinal image takes 42% of the whole vessel network in an image database used in this work. The detection performance on small vessels would affect the overall performance dramatically. Most algorithms in the literature are intensity based algorithms and don't perform well on them. Also they have the problem of parameter adjustment and insufficient sampling data.

In this work, two detection algorithms are discussed. One is a tracing algorithm containing image preprocessing, vessel direction prediction and tracing. Small vessels are detected on the result image of a novel Gabor-variance filter. The other algorithm is a radial contrast based algorithm. It transforms the intensity information into a high dimensional radial contrast domain. Feature descriptors for blood vessels are designed to introduce the human intuition in the vessel pixel classification and reduce the computational cost. The detection of other objects in the retinal images with the radial contrast based algorithm is also discussed.

The later algorithm, radial contrast based algorithm, is used in the Early Diabetic Retinopathy Screening System (EDRSS) project. Images are collected in several primary care clinics in the State of Texas. The detail design of the system is presented. In literatures, the benchmark images used in algorithm performance test are bright, well focused and of higher contrast. But images obtained in the real field are usually of poor quality, because of insufficient flash light, dark pigment of patient's retina, ophthalmological diseases, inadequate maintenance of the equipment, or inexperienced operators. These problems lead directly to the increased sensitivity and performance requirement of the detection algorithm. Our algorithm shows good ability to identify weak signals in the low quality images.

We will also present a code framework that can handle the image processing task in parallel on multiple CPU(s). The framework makes adding new filters easy. All task parallelization and flow control jobs are done in the upper level abstractions. The time cost of the same algorithm is reduced linearly with the increase of the processor number. Also, the framework improves research productivity. With shorter computational time, more experiments can be done and more time can be used on theoretical analysis.

The detail of the blood vessel detection is introduced in Section II, and the diabetic screening system is covered in Section III. Section IV concludes this work.

II. BLOOD VESSEL DETECTION

The objective of computer-based blood vessel mapping is to extract blood vessel (BV) pixels from a digital image. Similar to many other medical applications, BV mapping is an important basis for various (computer-based) analyses of retinal images. Computer-aided blood vessel mapping aims at accurate extraction of vascular structure at different *generation levels* [14]. Automated vessel detection has been an open problem and has been studied for decades. A BV network has a self-similar geometric structure among its branches on the two dimensional image. *Morphological profiling*, i.e., diameters, length, tortuosity, angle of branches, of BV provide basic measurements for anatomic and pathological studies. Growth, death and deformation of BV segments derived from such studies can assess the general condition of the retina, and diseases such as diabetic retinopathy, hypertension or other cardiovascular complications. Early detection and treatment of DR increases the chance of early intervention. Retinal images can be taken in regular eye exams, and a computer algorithm can register them with vessel networks and track the development of disease and can give an alert when something serious may happen. Since no two retinal vessels are exactly the same, retinal vessel maps can be used for biometric authentication.

Numerous algorithms have been developed for blood vessel mapping for retinal images (see the literature survey). In this section, two new algorithms are reported to automatically detect the vessel network. One of them is a tracking method based on Gaussian filter and Gabor-variance filter. The other is called radial contrast based detection algorithm, which first transforms the intensity information in spatial space into a high dimensional radial contrast (RC) space and then extracts features of vessel pixels in the RC space and classifies them from the background, based on these features.

A. EXISTING BLOOD VESSEL DETECTION ALGORITHMS

A major stream of the existing blood vessel detection algorithms are kernel based algorithms [15-18]. They use a predefined kernel to convolute to the image intensity map and do analysis on the result response map. Some tracing algorithms have been developed based on the matched filter response [19-26]. Chaudhuri *et al* proposed in [17] the matched filter response (MFR) algorithm. It modeled the vessels with a Gaussian function and the parameters of the Gaussian kernel are set to maximize the signal-noise-ratio of the output image. It has been extensively studied and expanded in the following years. Many algorithms were developed based on matched filter. Liu and Sun [20] proposed a detection-deletion scheme for blood vessel detection based on matched filter. Starting from some manually selected initial points the algorithm does an extrapolation along current vessel direction at a look-ahead distance. The path is corrected by the one-dimensional matched filter. The algorithm assumes that the signal-noise-ratio is high enough to differentiate the vessel and background, which is often violated on the area containing small vessels even in angiography images. Also, the algorithm is semi-automatic; users must specify the starting point manually. Chutatape *et al* [21] assumes the vessel is considered as a set of connected linear segments with fixed width and density inside the segment. The detection starts from the optical disc. Initial points are selected by finding local minimal responses of MFR on the starting circle. On MFR of the whole image, the detection of the vessel is done with second order derivative Gaussian filter. The position of the point on the central line, moving-ahead distance and direction of the vessel segments are estimated and corrected by an Extended Kalman Filter.

Most of the algorithms use a Gaussian function to model the vessel cross-section profile. Gao *et al* [27] proposed a bi-Gaussian model that can describe the characteristics of veins with central reflection. In [28], Gang *et al* proposed an amplitude-modified second-order Gaussian filter to measure the width of the vessel. They found that the vessel width can be found by analyzing the "spreading factor" of the matched filter and a predefined calibration line. The method works well on large vessels but fails in most cases on small vessels.

Some other algorithms used the region growing technique [13, 29, 30]. Hoover *et al* [13] developed an algorithm to recognize vessels in retina images called piece-wise thresholding on matched filter response. The original image is applied MFR and the result image is threshold and thinned to get the skeleton of major vessels. The skeleton is then broken into pieces by removing the bifurcation points. All end points of the remaining segments are listed as initial points. Based on these initial points an iterative thresholding procedure with the threshold decreasing gradually is performed. The starting pixel is labeled and pixels with higher intensity connected with any labeled pixel are labeled and a test is run to check if the formed region is a valid vessel. The procedure continues with a decreased intensity threshold until the test fails or the threshold reaches zero. The algorithm will stop when starting point list is empty. An important contribution of this paper is that it establishes a benchmark of performance evaluation. It publics a set of images containing both normal and abnormal retinal images and two set of manually labeled ground truth, called STARE. Many researchers use these images to test their algorithms, report and compare the performance with that of others. 10 sets of parameters of this algorithm are tested and the average performance is reported. A fast algorithm proposed by Jiang in [30] maps vessels from multiple localized thresholding results based on the geometric properties of vessels. The algorithm starts with thresholding the whole image with a particular threshold *T* to get vessel pixel candidates and then for every vessel candidate, the Euclidean distance map is calculated. On the resulted map, every vessel candidate has its nearest background neighbor's position and the distance from it. The values of two measures are calculated to identify the curvilinear pattern. Thresholds for these two measures and two other ones on size and contrast are used to remove false detections and get the final binary results for this intensity threshold *T*. All possible thresholds are tested, and the final detection result is reconstructed by a logical operation OR on all result images. This method is effective for large vessel detection but less accurate for small vessels. And the parameter settings in the paper are empirical. Martinez-Perez *et al* [29] analyzed the first and second derivatives of the intensity map, capture features of the result maps and use a region growing method to get the final vessel pixels.

Sim and Ves [31] proposed an algorithm that model the distribution of vessel pixels with a Markov random field. Simulated annealing and iterated conditional modes are used to estimate the vessel labels.

Some algorithms extract different features of blood vessels in the retinal images and utilize the classical pattern classification methods to identify the vessel pixels. Staal *et al* presents an algorithm to detect vessels in retinal image based on ridges [32]. First, the ridges are detected by computing the eigenvector of the matrix of second order derivatives of the image corresponding to the largest absolute eigenvalue. Points belongs to the same ridges are grouped together. Grouping is done by a simple region growing algorithm. A set of features are extracted for all grouped ridges. With all these features, a kNN-classifier is used to classify the ridges into two categories – vessel or background. Two image databases are used to test the performance of the algorithm. One is Hoover's STARE, and the other is DRIVE. The proposed algorithm gains even higher true positive rate than Jiang's [30]. The biggest problem of this algorithm is that the computational cost is too heavy. Running on a Pentium III 1GHz, 1GB memory PC, the algorithm takes more than 15 minutes to process an image.

Zana et al proposed an algorithm combining morphological filters and cross-curvature evaluation to segment vessel-like patterns. First the linear-like objects, e.g., the vessels or part of them are highlighted with a morphological opening. A linear structure element is used. As long as the linear structure element cannot be included completely inside the vessel when it has orthogonal direction with the vessel, the opening operation can remove all of them. Conversely, when the structuring element and the vessel have parallel directions, the vessel will stay nearly unchanged. A bank of linear structuring elements with different directions is used and a sum of top-hats along each direction will brighten the vessels. Then the connectivity property is utilized. Noises are removed with a geodesic reconstruction of the opened images into the original image.

Detection of large vessels is usually easier than that of small vessels because of their high contrast. The challenge comes from the small vessels which are shallow and of smaller size. Most existing algorithms didn't perform well in terms of small vessel detection, even though their performance measured with Hoover's [13] method look good. As we will point out later, Hoover's measure strongly favors large vessels and loss of small vessels won't affect the measure much. In fact, in many applications, the accurate vessel width is far less useful than the complete vessel network structure, especially that of small vessels, e.g., neovascular detection, image registration, fractal dimension calculation, etc. Some of the algorithms such as Staal’s [32] have good performance on small vessel detection but it is very time consuming.

In this work, we proposed two novel algorithms to detect the blood vessels in retinal images. Gaussian tracing and Gabor-variance filter scheme contains steps of adaptive histogram equalization, initial point selection, forward predicting, backward verification and bifurcation point detection. A Gabor variance filter was developed to enhance the small vessels. Radial contrast based algorithm transforms the image from intensity domain to a high dimensional radial contrast domain. Different kinds of feature descriptors can be designed on that domain, and various pattern classification methods can be utilized to classify the blood vessel pixels from the background.

B. GAUSSIAN TRACING AND GABOR-VARIANCE FILTER

The solution includes two phases. In the first phase, large vessels are detected. Starting from automatically selected initial points on a contrast enhanced original image, a Gaussian tracing scheme traces the blood vessels using three major components: *forward prediction, backward verification* and *bifurcation point detection*. In the second phase, a Gabor-variance filter is applied on the enhanced image in the first phase and then the same tracing scheme is employed to detect small vessels. In this work, we still classify the vessels into two categories as stated in previous section: large vessels (with diameters larger than 3 pixels) and small vessels (all others).

1. Original image preprocessing

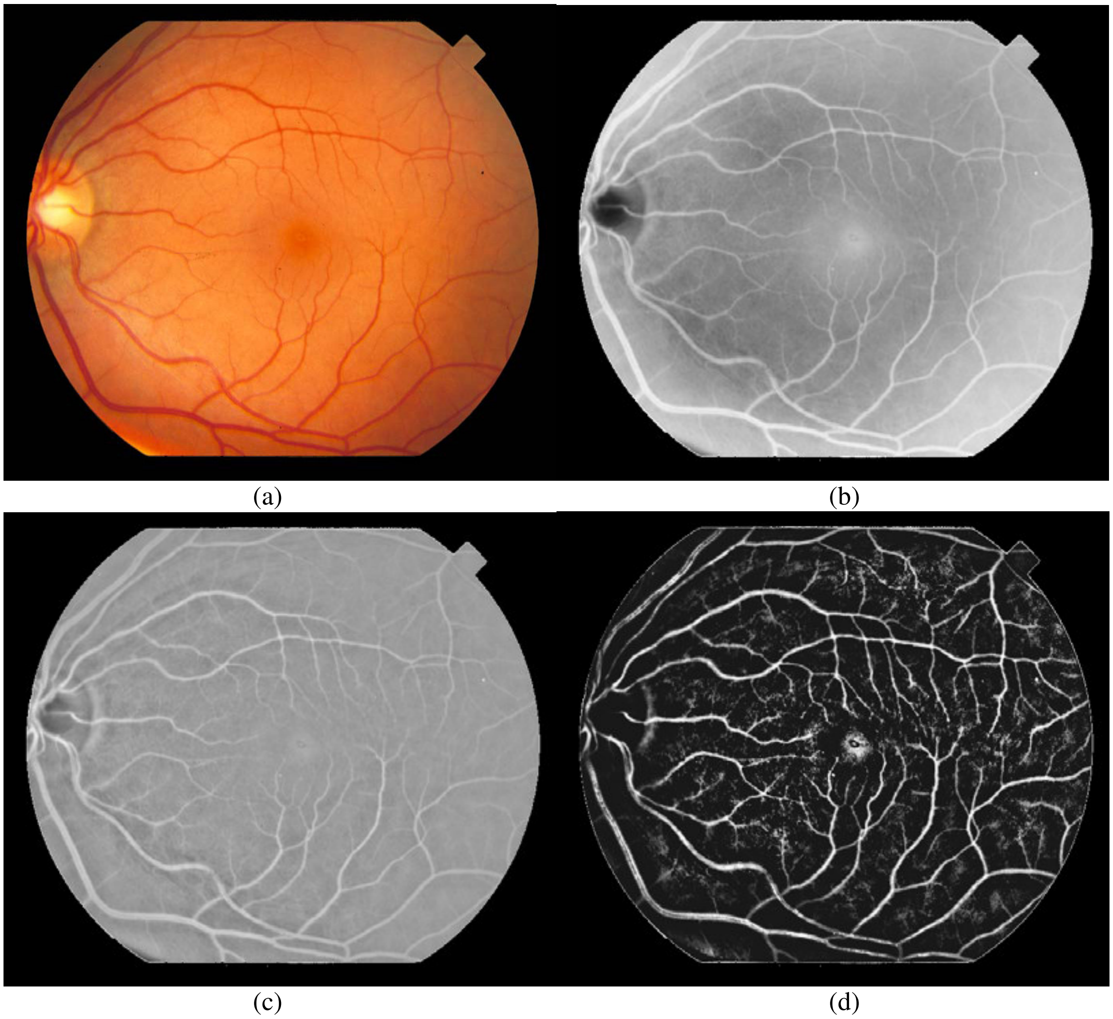
Unlike the blood vessels in the angiographic images, which are well highlighted with the help of a fluorescence agent, the blood vessels in color fundus retina images, especially the small vessels, have much lower contrast. It's necessary to do some preprocessing on the original images to enhance the vessels before the detection procedure starts. There's a common problem in most enhancement algorithms: over- and under-enhancement in an image with unbalanced intensity distribution. To reduce the effect from the bright and dim area in the retinal image, an image illumination equalization technique by Hoover in [33] is applied before enhancement. Let $I_p^{(g)}$ denote the green channel intensity value of a pixel p on the retinal image. The green channel is chosen because it contains most of the information regarding blood vessels in the retinal images [13]. The illumination equalization is done with the following equation:

$$I_p^{(ie)} = I_p^{(g)} - \overline{I_p^{(g)}} + 128$$

where $I_p^{(ie)}$ is the intensity value of pixel p after illumination equalization, $\overline{I_p^{(g)}}$ is the average intensity of p's neighborhood, which is a square area centered at p with width h=21. $I_p^{(ie)}$ is normalized to range [0, 255]. See Figure 4 for the green channel of an original retinal fundus image and the result after illumination equalization (Figure 4(b)). After this step, not only the overall intensity distribution is made even, but the bright area such as high-energy hard exudates and optical disc are also effectively suppressed. An adaptive histogram equalization (AHE) technique by Stark in [34] is employed to further enhance the blood vessels. While Stark's AHE is found effective in vessel enhancement, empirical study shows that its enhancement performance can be further improved by introducing a new parameter r as follow:

$$I(x,y)=\{\frac{1}{h^2}\sum_{q\in N_p}\delta[I_p^{(ie)}-I_q^{(ie)}]\}^r\cdot 255$$

where $\delta(x)$ is the delta function, i.e., $\delta(x)=1$ when x>0, otherwise, $\delta(x)=0$. The new enhancement method downgrade to Stark's AHE when r=1. When r increases, the contrast between vessels and the background becomes larger, but at the same time, some noises on the background are also enhanced, and the boundaries of some vessels are distorted. Also, when the window size h is small, the improved AHE tends to use the local information to do the enhancement, which may cause noisy results. On the other hand, if large value is chosen for h, the enhancement results become smoother, but detail information such as small vessels may be smeared and the computational burden increases dramatically. To determine the appropriate value for r, we did extensive experiments and the details can be seen in [35]. As a result, we found, based on the performance measure chosen in [35], the best enhancement performance is achieved when r=8 and h=81. These settings will be used in the rest of this section.



(a) (b) (c) (d)

Figure 4 (a) A color retinal fundus image that contains elements of diabetic retinopathy, (b) the inverted green channel of the color image(c), the result of illumination equalization (d) and the result of contrast enhancement with the improved AHE with h=81 and r=8.

2. V-tree structure

It's well known that the blood vessel system in human body has a hierarchical structure. Small vessels spin off from the large ones and they have similar structure but different sizes. To represent this structure, we adopted an n-nary tree data structure called *V-tree*. A V-tree can have multiple branches/flows under each node, which represents the bifurcation point of blood vessels. A flow between two nodes represents the blood vessel connecting two bifurcation points (or one bifurcation point and an end point). In retinal images, the root of a V-tree is the origin of the arteries/veins that enters the retina through optical disc. The links under the root are the arteries/veins and smaller vessels spin off from the larger ones at the bifurcation points. There are three cases of branching from a bifurcation point: 1) a larger vessel splits into two smaller ones; 2) two vessels cross over each other; 3) termination of a vessel (in this case, the node in V-tree is actually a leaf node). In a V-tree, circles are broken arbitrarily.

A vessel flow f can be modeled as connected short linear segments (f_i, i=0,1,2⋯) with approximately constant width and direction. Segments are separated by lines that are perpendicular to the vessel edges, called border lines. Connecting the central point of adjacent border lines generates the central line of the current segment, denoted by S_i, and all segment central lines forms the central line of the flow, denoted by S. The width of the current segment D_i is set to be the average length of its two border lines. The detection procedure is done segment by segment following a flow. The length of the current segment is set to be the width of last detected segment, and its direction is determined by the historical segments and a Gaussian filter outcome. At each segment, it has to be determined if it's near a bifurcation point or termination point.

3. Initial point selection

The whole detection procedure includes two stages to detect large and small vessels separately. In large the vessel detection stage, there are four major steps: initialization, forward prediction, backward verification and bifurcation point detection.

Initialization step uses a mesh grid to find which point to start with. At each point on the grid, a simple test with two decision rules is utilized to determine if it is on a vessel: 1) the point must be located between two adjacent edges detected by Sobel edge detector [36], and the distance between the two edges will be the initial prediction of the vessel width; 2) the output values of a 2D Gaussian filter along different directions have two local maxima along two directions that point at the forward and backward flow directions. An initial point will be removed if its hit in the detection procedure, since it's not necessary to start from that point again.

4. Forward prediction

Forward prediction is to estimate the width and direction of the current segment. Let (x_i, y_i) be the coordinate of the central point of the central line S_i, let $_i$ be the angle between flow direction and the horizontal line, and let w_i be the width of flow segment f_i. Given n consecutive detected segments $f_0 .. f_n$, we first estimate the central point of next segment by extrapolation with cubic spline on $(x_0, y_0) \cdots (x_n, y_n)$. Let P_x and P_y be two cubic spline functions to estimate x_{n+1} and y_{n+1} based on knots (i, x_i) and (i, y_i), $i=0..n$. The estimated central point $(\tilde{x}_{n+1}, \tilde{y}_{n+1})$ can then be calculated with $(P_x(n+1), P_y(n+1))$. The estimated flow direction $\tilde{\theta}^{(1)}_{n+1}$ is the tangential orientation at $(\tilde{x}_{n+1}, \tilde{y}_{n+1})$ (See Figure 5(a)), which can be calculated with:

$$\tilde{\theta}^{(1)}_{n+1} \approx \arctan(\frac{P_y(n+1) - P_y(n+0.5)}{P_x(n+1) - P_x(n+0.5)})$$

In a similar way, the width of next segment can be estimated with a cubic spline function P based on knots (*i, w*), *i=0..n*: $\tilde{w}_{i+1} = P_\theta(n+1)$.

The above estimation of $(\tilde{x}_{n+1}, \tilde{y}_{n+1})$ only considers the trend of flow. It is subject to the extrapolation error and sometimes suffers from over fitting. This will cause error in the flow direction estimation. To reduce the error and improve the accuracy of flow direction $\tilde{\theta}_{n+1}$, local homogeneity information is utilized. A two-dimensional Gaussian function is convoluted with the image along different directions. The convolution output for a particular direction θ centered at pixel (*x, y*), called *oriented feature response* (*OFR*) at (*x, y*) along direction θ, denoted by $\phi^{\theta}_{(x,y)}$, is expressed as:

$$\phi^{\theta}_{(x,y)} = I(x, y) \otimes g_\sigma$$

and g_σ is defined as:

$$g_\sigma(x', y', \theta) = \exp[-\frac{(x'\sin\theta + y'\cos\theta)^2}{2\sigma^2}]$$

where θ is the direction of the filter, $x' \in [0, l]$ and $y' \in [-w/2, w/2]$, with (x, y) as the origin point when the filter is applied to a pixel (*x, y*), and the filter is truncated to a rectangular shape with length *l* and width *w*. The standard deviation is set to be *w*/5. The direction with highest response is the second estimation of flow direction (See Figure 5(b)), denoted by $\tilde{\theta}'_{n+1}$:

$$\tilde{\theta}^{(2)}_{n+1} = \underset{\theta}{\arg\max}\, \phi^{\theta}_{(x_n, y_n)}, \text{ for } \theta \in \left(\theta_n - h \cdot \Delta\theta, \theta_n + h \cdot \Delta\theta\right)$$

subject to the constraint

$$\phi^{\tilde{\theta}'_{n+1}}_{(x_{n2}, y_{n2})} > \alpha \cdot \phi^{\theta_n}_{(x_n, y_n)}, \text{ where } 0.5 \le \alpha \le 1$$

This constraint is used to prevent over-detection at the termination points of the vessel flow. Combine the two estimations together we get the final estimation of current flow segment with:

$$\tilde{\theta}_{n+1} = c \cdot \tilde{\theta}^{(1)}_{n+1} + (1-c) \cdot \tilde{\theta}^{(2)}_{n+1}$$

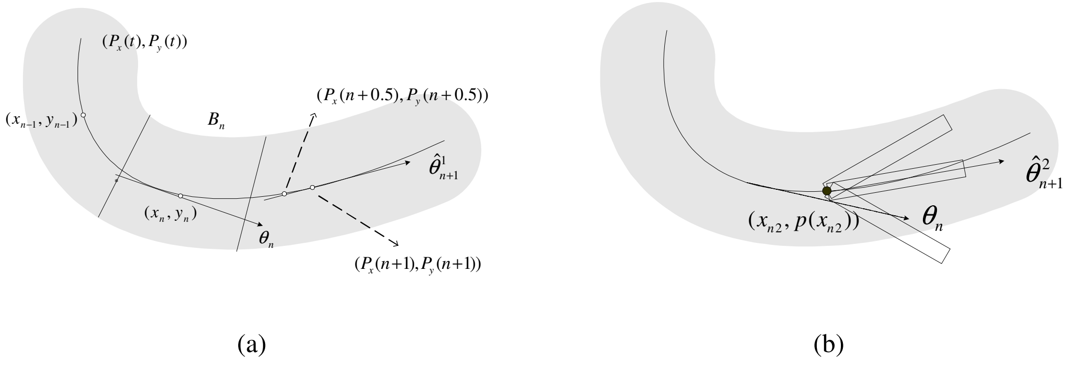


Figure 5 (a) Estimation of θ_{n+1} by extrapolation (b) and estimation of θ_{n+1} via blood vessel features.

On 83% of over 600 tested cases, the difference between $\tilde{\theta}^{(1)}_{n+1}$ and $\tilde{\theta}^{(2)}_{n+1}$ is less than 20° (See Figure 6(a)). Large discrepancies do happen, especially when other objects, typically another vessel, is at proximity (see Figure 6(c)), or the vessel flow has large curvature (see Figure 6(d)). When the curvature of the vessel flow is below a given threshold, the estimation from historical data $\tilde{\theta}_{n+1}^{(1)}$ should be given more weight (let $c \in [0.5,1]$). When the vessel flow curvature changes significantly, the estimation from OFR $\tilde{\theta}_{n+1}^{(2)}$ should be given more weight (let $c \in [0,0.5]$), since the extrapolation error in this case may be pretty large. The standard definition of curvature κ, is given by:

$$\kappa = d\theta / dl = (x' y'' - y' x'') / (x'^2 + y'^2)^{3/2}$$

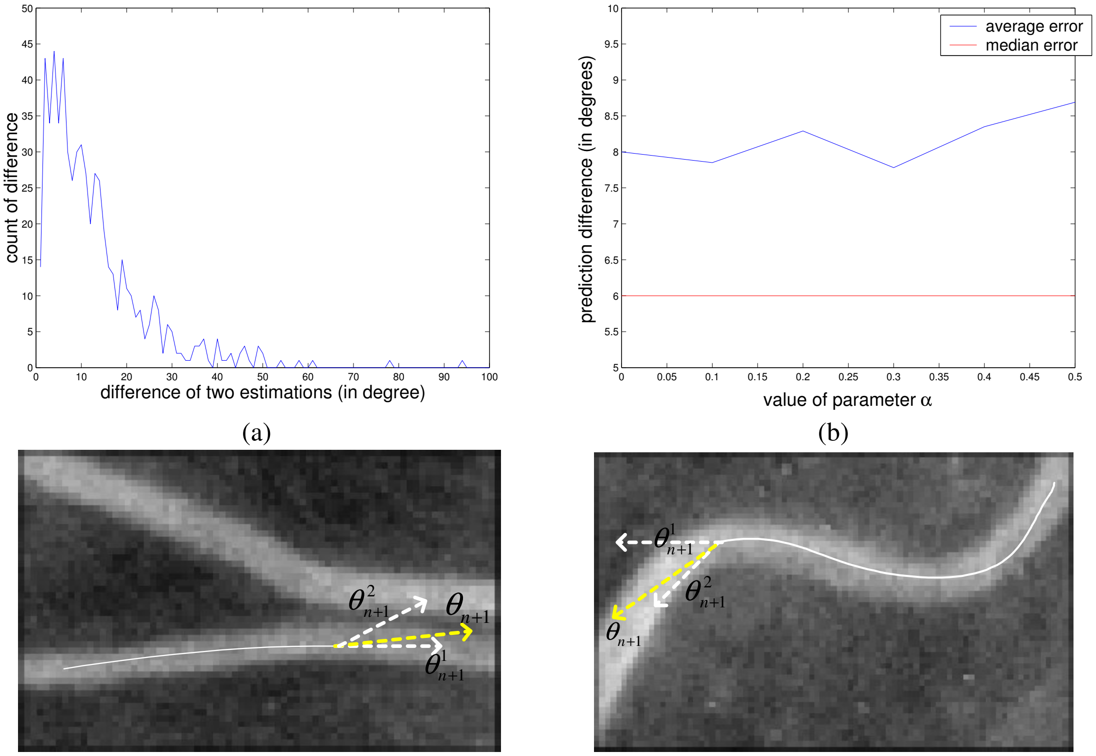
There are two ways to calculate κ: direction difference between two adjacent segments, and from the functions of the central curve approximated by $P_x(t)$ and $P_y(t)$ respectively. We set $\kappa = 0.1$ as the curvature threshold for the weight assignment.

The two curves in Figure 6 (b) represent the average and median values of the prediction difference between the estimated value of $\tilde{\theta}_{n+1}$ and the tangential direction of the approximated central curve of the flow through interpolation. The median and average differences are roughly 6° and 8° respectively. The error is minimized when $\alpha = 0.3$, which has been used for all the experiments reported in this work.

$\tilde{\theta}_{n+1}$ and $\tilde{w}_{n+1}$ are used to determine the most likely location of a new flow segment f_{n+1} based on extrapolation, and they need to be refined when we actually move to the f_{n+1}. The first step is to determine S_{n+1} and θ_{n+1}. The estimated central line $\tilde{S}_{n+1}$ is a straight line starting from the center of the previous border line along the orientation $\hat{\theta}_{n+1}$ with length l_{n+1}. When a flow has high curvature, it needs a smaller value of l_{n+1}. On each pixel (x, y)

of $\tilde{S}_{n+1}$, we find two edge points on the opposite sides of the cross section of the flow, by using the Sobel edge detector within the range of $\tilde{w}_{n+1}$. The middle point (x', y') of the two edge points is considered to be located on the real central line S_{n+1} of the segment, which is formed by connecting the set of the middle points in f_{n+1}. Obviously, S_{n+1} may or may not be the same as $\tilde{S}_{n+1}$. The actual direction θ_{n+1} of f_{n+1} can be determined by linear fitting. That is, θ_{n+1} is evaluated by the direction of the straight line that fits the set of the middle points in f_{n+1} with least square error.

Let $D(x, y)$ denote the distance between two edge points on the cross section of the pixel $(x, y) \in \tilde{S}_{n+1}$, the width of B_{n+1} can be calculated by taking an average on the local width of each point on the central line, i.e., $\tilde{w}_{n+1} = \sum_{(x,y)\in\tilde{S}_{n+1}} D(x, y) / l_{n+1}$. Let γ denote the angle deviation between $\tilde{S}_{n+1}$ and S_{n+1} , the width of segment f_{n+1} is adjusted by $w_{n+1} = \tilde{w}_{n+1} \cdot \cos\gamma$. Therefore after detection, f_{n+1} can be characterized by (S_{n+1}, w_{n+1}) , where S_{n+1} is characterized by $(\theta_{n+1}, (x_{n+1}, y_{n+1}))$.



(a) (b)

(c) (d)

Figure 6 Performance of direction prediction: (a) difference of two estimations, (b) prediction accuracy as function of α and two examples of prediction of θ_{n+1}: (c) $\alpha \geq 0.5$ and (d) $\alpha \leq 0.5$.

5. Backward verification

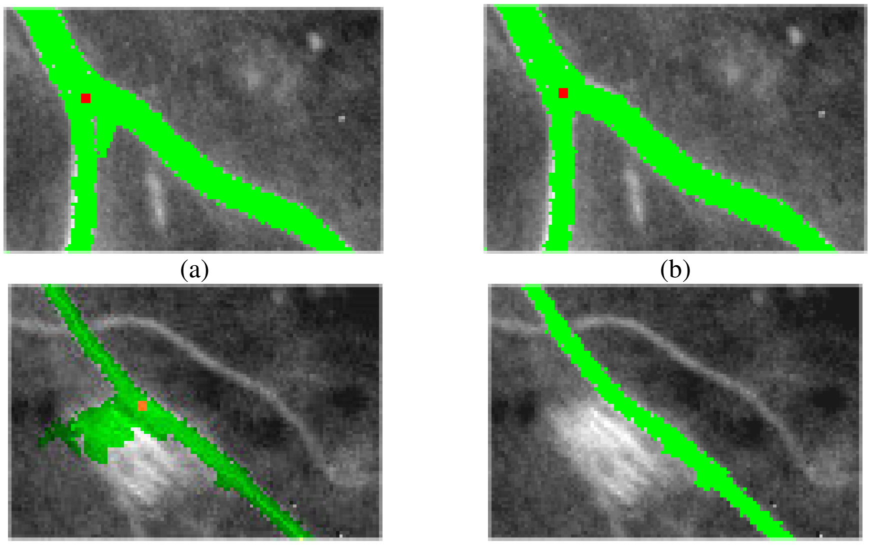
The purpose of backward verification is to remove the false detection from lesions near the blood vessels. It employs the width and OFR information of historical segments to find the false detections. A false detection is identified if its value of one of these features is deviated from those of its neighbors, i.e., an *outlier* [37-40].

We adopted the method in Grubbs' paper [40] to identify the outlier. Grubbs tabulated a series of *critical value* for different data size. Let and denote the mean and standard deviation calculated from the data points. If $z = |x - \mu| / \sigma$ exceeds the critical value for this data set size, where x is the data point to be tested, this data point is considered as an outlier. Our experiments show that Grubbs' method can effectively capture the false detections from lesions around blood vessels and falsely detected bifurcation points.

Figure 7 shows two examples of the power of backward verification. Figure 7(a) shows the segment adjacent to a bifurcation point which has an over estimated width. 10 historical segments are used in the verification procedure and cubic spline interpolation is utilized to correct the central point of the segment, the same procedure is also done on width. Figure 7(b) shows the result after correction. Figure 7(c) shows the situation when a blood vessel runs through a hemorrhage. A "branch" is falsely detected on the hemorrhage. The segment is identified as a false alarm because it's short and its width is too different from its parent flow. Figure 7(d) shows the result after backward verification.

6. Bifurcation point detection

Bifurcation points are where the vessels fork, or in the other view, where several vessels merge. The bifurcation points are searched on the central line *S*. On each point $p \in S$, the OFR along different directions are calculated. If p is a bifurcation point, its OFRs should have local maxima in the directions that the flow goes, and the number of this kind of directions should be larger than two, since every vessel segment will have two directions with local maxima in OFR, which indicates the forward and backward directions of that flow. That is, a direction θ_i is a flow direction on a bifurcation point, if



(a) (b)

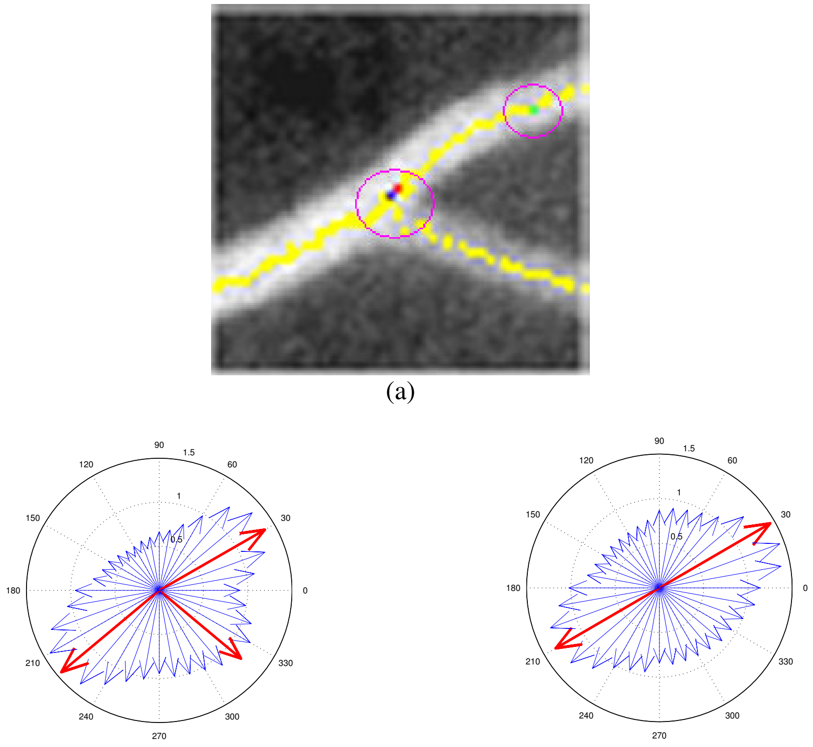
(c) (d)

Figure 7 False detection cases (a) and (c) and their correction (b) and (d) in the backward verification phases. The orange dots represent bifurcation points detected in the forward detection phase.

$$\theta_i = \arg\max_{\theta} \phi_p^{\theta}, \text{ for } \theta \in [\theta_i - h\cdot\Delta\theta, \theta_i + h\cdot\Delta\theta],\ i = 0,1,2\ldots, \text{ and}$$

$$\phi_p^{\theta_i} \geq m_p + c\cdot\sigma_p$$

where m_p and σ_p are the mean and standard deviation of ϕ_p along all directions, c>0, and h is a positive integer that determines the neighborhood size. In this work, $\Delta\theta = 10^{\circ}$, h=2, c=0.5. The OFR(s) at each point are normalized with the average. When approaching the real bifurcation point, following the flow there may be more than one point that satisfy the above criteria. The point chosen as the bifurcation point is the one with maximal summation of ϕ_p^{θ} along all directions. Figure 8(a) shows a vessel which forks into two smaller ones. The yellow line is the central line of the vessel. The blue point is the selected bifurcation point. Its OFR(s) plot in a polar coordinate is shown in Figure 8(b), with the local maxima shown in red. One can see that the three local maxima clearly indicate the three flow directions splitting at the bifurcation point. The red dot is another point that satisfies the bifurcation point criteria. The blue dot is chosen over the red one as the final bifurcation point because it has higher summation of all ϕ_p^θ. A OFR plot of a randomly chosen point on central line, marked as a green dot, is shown in Figure 8(c). One can see that there are only two local maxima existing along the direction in which the flow goes forward and backward.



(a)

(b)

(c)

Figure 8 (a) A bifurcation point connecting three flows, (b) the normalized OFR values along different orientations for a bifurcation point and (c) a non-bifurcation point.

7. Small vessel detection

The algorithm mentioned in the previous sections is for large vessels only. Small vessels have a much lower contrast against the background, which makes their OFR(s) quite close to each other. To deal with this problem, we proposed a new scheme to enhance the small vessels, the Gabor variance filter (GVF). GVF is convoluted with the original image at each point, and the variance of all the responses are used as the energy of that point. For a pixel (x, y) on the image, GVF is express with:

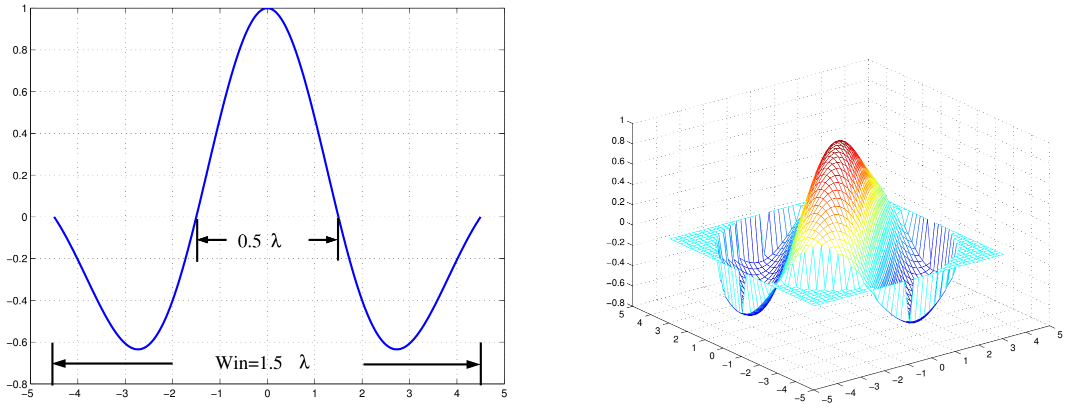
$$Gabor(u,v,\lambda,\theta,\varphi) = EXP(-\frac{u'^2}{2\sigma_x^2} - \frac{v'^2}{2\sigma_y^2})\cos(2\pi\frac{u'}{\lambda} + \varphi),$$

$$\begin{cases} u' = u\cos\theta + v\sin\theta \\ v' = -u\sin\theta + v\cos\theta \end{cases}$$

where, (u, v) is the relative coordinate with (x, y) as origin, λ, σ and φ are the spatial frequency, standard deviation (along x and y axes), and shift, respectively. In this case, we don't need shifting, so φ is set to 0. The GVF response of a pixel along direction θ is calculated as:

$$g_v(x,y,\theta) = I(x,y) \otimes Gabor(u,v,\lambda,\theta,\varphi),\ \theta = 0, \Delta\theta, ..2\pi$$

Figure 9 (a) shows the cross-section of Gabor filter by plane y=0. The filter is truncated and only the part in a square of width 1.5λ is kept (see Figure 9(b)). In the image after the first round of enhancement, blood vessels have higher intensity than the background (See Figure 9 (b)). If the filter is applied on a pixel on the vessel, when direction of filter is tuned to be aligned with the direction of the vessel flow, the Gabor filter response will be maximized since all high intensity pixels (vessel pixels) all fall on the peak of the filter and those falling in the valleys are low intensity background pixels; when the direction of filter is tuned to be perpendicular to that of the vessel flow, the filter response is minimized, and the value will be close to zero. If the filter is applied on the background pixels, since the background pixels have similar intensities along all directions, the Gabor filter responses along all directions will be similar to each other too. This inspires the idea of using the variance of the filter responses as the final energy. On vessel pixels, the variance of Gabor filter responses will be large since the response values are quite different to each other when the filter is tuned to different directions, while on background pixels, the variance will be small since the Gabor filter responses are close to each other.



(a) (b)

Figure 9 Determining the parameters of Gabor filter: (a) the truncated Gabor kernels for $\lambda = 6$, $\sigma_x = \sigma_y = 3$, $\phi = 0$, window size 9×9 on the x-z plane and (b) the 3-D view of the kernel at the orientation 45°.

Gabor variance filter response is orientation invariant. We use the following simulation approach to determine the parameters used in the filter. First, vessels are simulated by a Gaussian function

$$I(x,y)=I_f EXP[-\frac{(x-w/2)^2}{2\sigma^2}]+I_b\ ,\ (x,y)\in(0,w)\times(0,w)$$

where, w is the analysis window size. I_f and I_b are the highest intensity of vessel pixels (at the center of the vessel) and the intensity of the background, respectively. The standard deviation σ is set to be one-fifth of the blood vessel diameter d , i.e., $\sigma = d/5$, so that the filter coefficients beyond this range are negligible. We let d equal to 2 or 3 pixels, $I_b = 70$, and let I_f change from 90, 120 and to 150 in the simulation. To maximize the response when the filter is tuned to be in the same direction of the vessel, the peak part of the filter (between the first two zero-crossing points from origin, with the width $\lambda/2$) should be wide enough (controlled by the parameter λ) to cover the whole vessel, that is $d=\lambda/2$. The other parameters that affects the response value is the standard deviation σ_x . Let k denote the ratio between the vessel width d and σ_x , $k=\sigma_x/d$. Figure 10 shows the simulation result, which suggests that when the vessel diameter is 3, the GV response is maximized if σ_x is set to be equal to d and drops very quickly after that; when the vessel diameter is 2, GV response will keep going up as the standard deviation σ_x increases but slowly, after $\sigma_x = d$. As a conservative design, we set $\sigma_x = D_s$, where D_s denotes the maximal width of small vessels. As stated in previous sections, $D_s = 3$. So, the final parameter set is: $\sigma_x = \sigma_y = D_s = 3$, $\lambda = 2D_s = 6$.

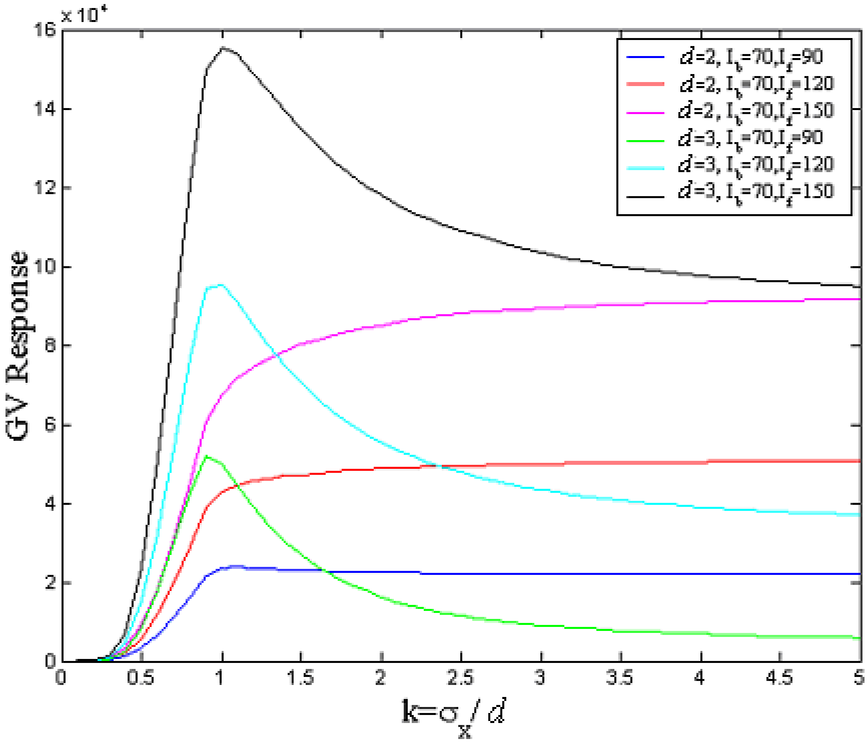


Figure 10 The GV responses for different foreground (I_f) and background (I_b) combinations as function of k values.

The result energy ε of a pixel (x,y) is calculated by taking the variance of g_v along all directions:

$$\varepsilon(x, y) = \frac{1}{n}\sum_{k=0}^{n-1}\left[g_v(k\Delta\theta) - \frac{1}{n}\sum_{k=0}^{n-1} g_v(k\Delta\theta) \right]^2, \text{ where } n = 2\pi / \Delta\theta$$

The result energy is stretched to range over the whole image in order to generate a new image. The result of a sample image is shown in Figure 11(b). GV filter picks out the small vessels and also inevitably some background noise at the same time. To remove these small noise segments, we apply the GV filter on the response map again and get the result like Figure 11(c), which is practically noise free, at the cost of minor small vessel signals. We call Figure 11(c) GV2 response map.

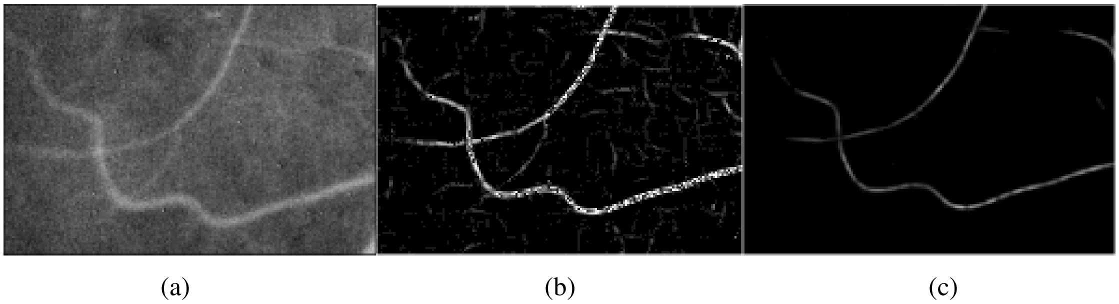


Figure 11 Extraction of small vessel feature map: (a) a sample image, (b) first phase GV response, (c) second phase GV response.

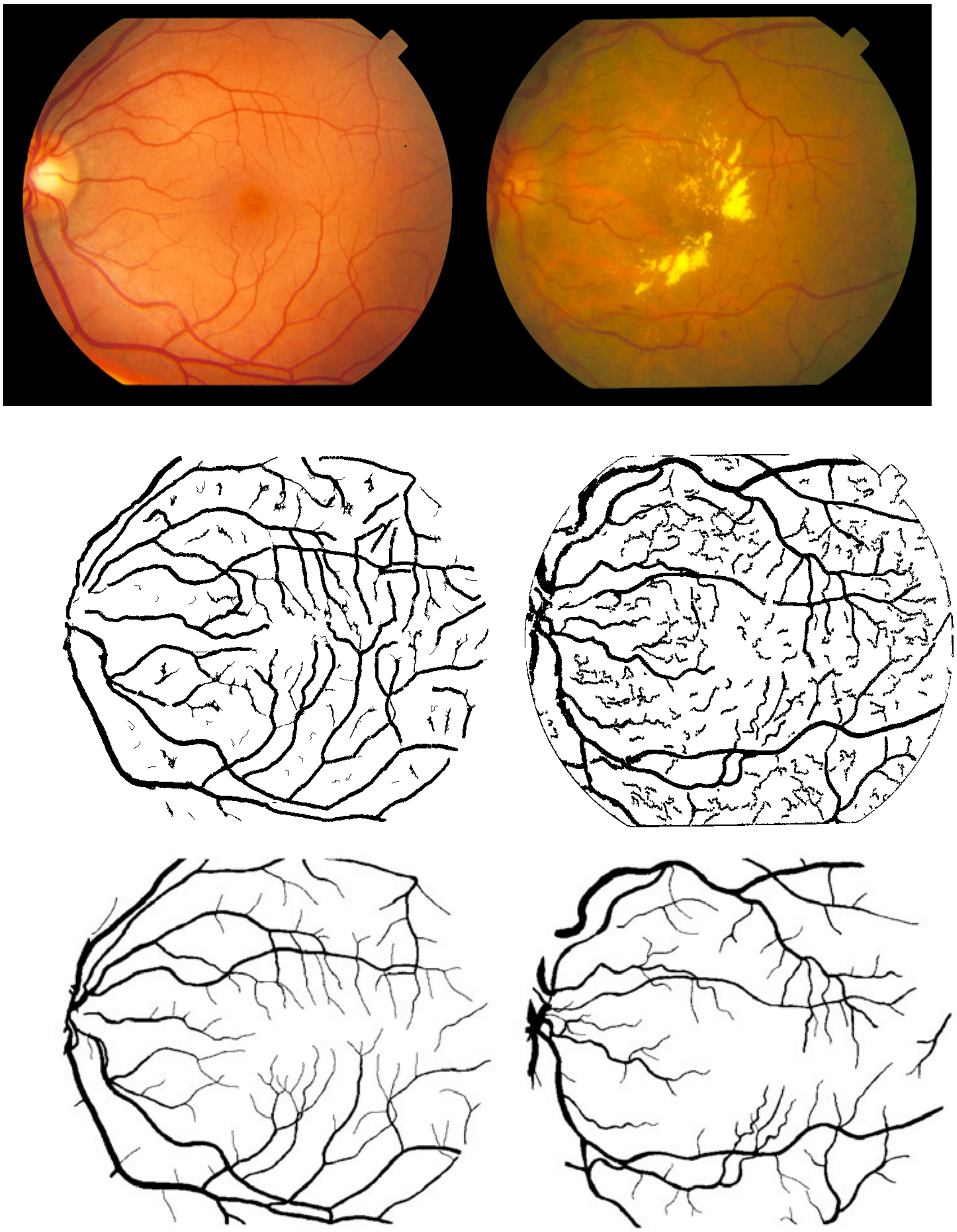


Figure 12 Two retinal fundus images and the detection results on them. First row, two original images in the STARE image database. Second row, the detection outcome. Third row, the first hand-labeled result.

The detection of small vessels is done on GV2 response map with the same scheme as that is applied to large vessels. Note that backward verification is not necessary since the estimated width is limited to a very small range ($[1..D_s]$).

The detection outcome of two images in STARE [13] image database are shown in Figure 12. STARE is a popular benchmark image database widely used by researchers in retinal image processing. It has 20 images. 10 of them are of normal people, and others are of patients with diabetic retinopathy. The performance of the algorithm on each of normal and abnormal images in STARE is shown in Table 2 and Table 3. At this time, we are using the performance measure defined by Hoover in [13]. Let M_t denote the ground truth, M denote the automatic detection outcome, and $\ell(\bullet)$ count the number of pixels in a pixel set. The true positive rate (TPR) and false positive rate (FPR) are defined as

$$TPR = \frac{\ell(\{p \mid M(p) \neq 0, M_t(p) \neq 0\})}{\ell(\{p \mid M_t(p) \neq 0\})}$$

$$FPR = \frac{\ell(\{p \mid M(p) \neq 0, M_t(p) = 0\})}{\ell(\{p \mid M_t(p) = 0\})}$$

STARE database provides two hand-labeled results — label-ah is made with less confidence than label-vk, and thus contains less vessel pixels. Taking label-ah as the “Golden standard”, the performance of the second human reader (label-vk) is found to be 89.5% TPR and 4.4% FPR. The overall performance of our method is 84.3% TPR and 3.9% FPR. Our TPR is a bit lower than second human reader but FPR is much lower. For the scheme proposed in [30] by Jiang, it must increase its FPR to 5% to achieve the same TPR as ours. Note that the basis of FPR calculation is the number of all non-vessel pixels which is much larger than the number of vessel pixels and thus small reduction in FPR is significant improvement. The scheme proposed in [13] also needs to increase its FPR significantly to match the same TPR as ours. Compared with these two schemes, our method needs less calibration. Better small vessel detection contributes to the higher performance of our algorithm. Statistics on ground truth (label-ah) of 20 images in STARE image database shows the pixel count of small vessels (with width less than or equal to 3 pixels) takes 42% of that of the whole vessel network, and 75% of which is captured by our algorithm.

Table 2 The performance on normal retinal images with Hoover's pixel-wised measure [13].

TPR(%)	91.0	88.0	89.9	82.7	90.6	85.1	82.7	86.2	86.2	80.0
FPR(%)	5.0	4.3	5.5	4.1	4.2	2.9	2.6	3.8	4.2	2.8

Table 3 The performance on retinal images with ophthalmological implications.

TPR(%)	84.5	82.6	84.9	79.1	73.8	84.4	83.9	86.5	78.1	85.2
FPR(%)	4.5	3.5	3.6	4.0	3.8	4.1	4.5	2.1	3.4	5.3

Hoover's measure strongly favors the large vessels. It counts the number of pixels that also appear on the ground truth. Since large vessels have many more pixels than small vessels, they take much more weight in the performance evaluation. For example, an algorithm A detects the large vessels well but misses most small vessels; and an algorithm B loses some boundary pixels on the large vessel edges but detects a lot of small vessels. A may have higher performance than B according to Hoover's measure. It must be pointed out that the boundary of vessels is pretty subjective to different image readers. There are some papers talking about precise boundary detection [41-43]. But there doesn't exist a way to define the vessel boundary numerically and precisely. It's doubtable whether the image readers of STARE database marked about the "real" vessel boundary. Also, the vessel diameter changes in the cardiac cycle [44], and it's related even to the velocity of the blood [45]. In a lot of medical applications, the vessel structure is more important, only width *change* or relative width *ratio* between different types of vessels are useful and accurate, vessel width is not necessary, e.g., vessel tortuosity analysis, retinal registration, fractal analysis, etc. A good vessel detection algorithm should be able to detect as many possible small vessels as well as large ones. To better evaluate an algorithms' performance on detecting vessel structure, we proposed a new measure called *central line based* performance evaluation:Let C_t and C denote the central line of M_t and M , respectively. The central line can be obtained by applying any thinning algorithm on the blood vessel map, e.g., [ref]. The true/false positive rates are defined as:

$$TPR = \frac{\ell(\{p \mid C(p) \neq 0, M_t(p) \neq 0\})}{\ell(\{p \mid C_t(p) \neq 0\})}$$

$$FPR = \frac{\ell(\{p \mid C(p) \neq 0, M_t(p) = 0\})}{\ell(\{p \mid C_t(p) = 0\})}$$

This performance measure only counts the number of pixels on the vessel skeleton, and is not affected by the subjective vessel width. Table 4 lists the performance comparison between Jiang's algorithms [30] and ours. One can see that Jiang's algorithm must use parameter set six (K6) to match the performance of our algorithm with central line based measure; but with this parameter set, Jiang's algorithm measured by Hoover's performance measure generates much more false detection (10.73%) than ours (3.9%) (See Figure 13 for the comparison between Jiang's and ours). Figure 14 shows two images in STARE and their detection result by Jiang’s and ours. Note that the central line based performance is evaluated based on the second hand labeled vessel pixels (label-vk) because it reveals more small vessel details. Jiang's algorithm takes 36 seconds with threshold step set to 1, and 8 seconds with threshold step set to 5 (coarser result). Our algorithm takes about 40 seconds to process one image.

Table 4 The performance comparison between Jiang's and ours, in Hoover’s measure and our central line based performance measure.

Detection performance Algorithm	Hoover measure (TPR/FPR)%	Central line measure (TPR/FPR)%
Our algorithm	84.30/3.9	60.56/0.39
Jiang's (K0)	44.13/1.15	31.46/0.004
Jiang's (K1)	60.08/1.97	40.19/0.015
Jiang's (K2)	70.72/2.79	45.34/0.032
Jiang's (K3)	80.69/5.19	54.09/0.13
Jiang's (K4)	83.84/6.37	56.31/0.18
Jiang's (K5)	86.04/8.13	58.73/0.26
Jiang's (K6)	87.81/10.73	61.53/0.39
Jiang's (K7)	89.63/14.82	64.83/0.61

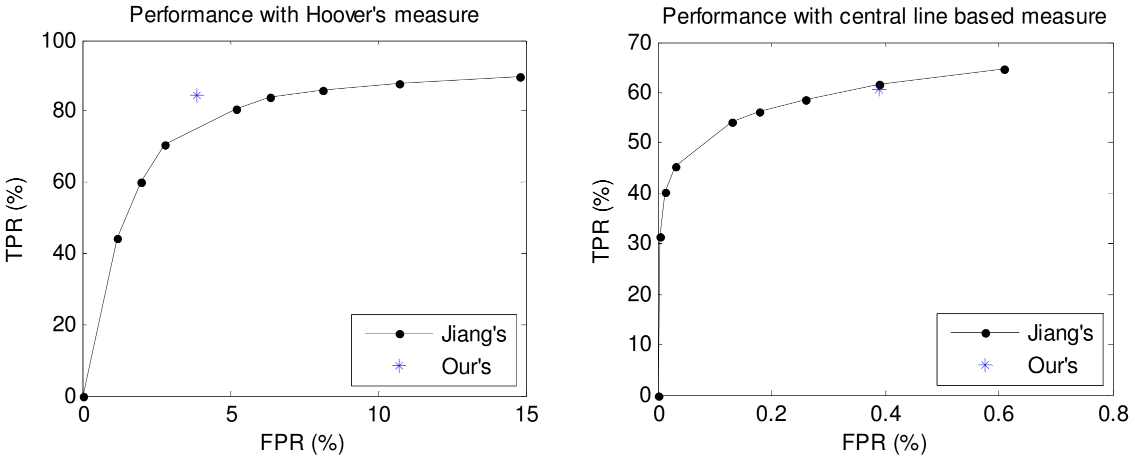


Figure 13 Performance comparison between Jiang’s and ours.

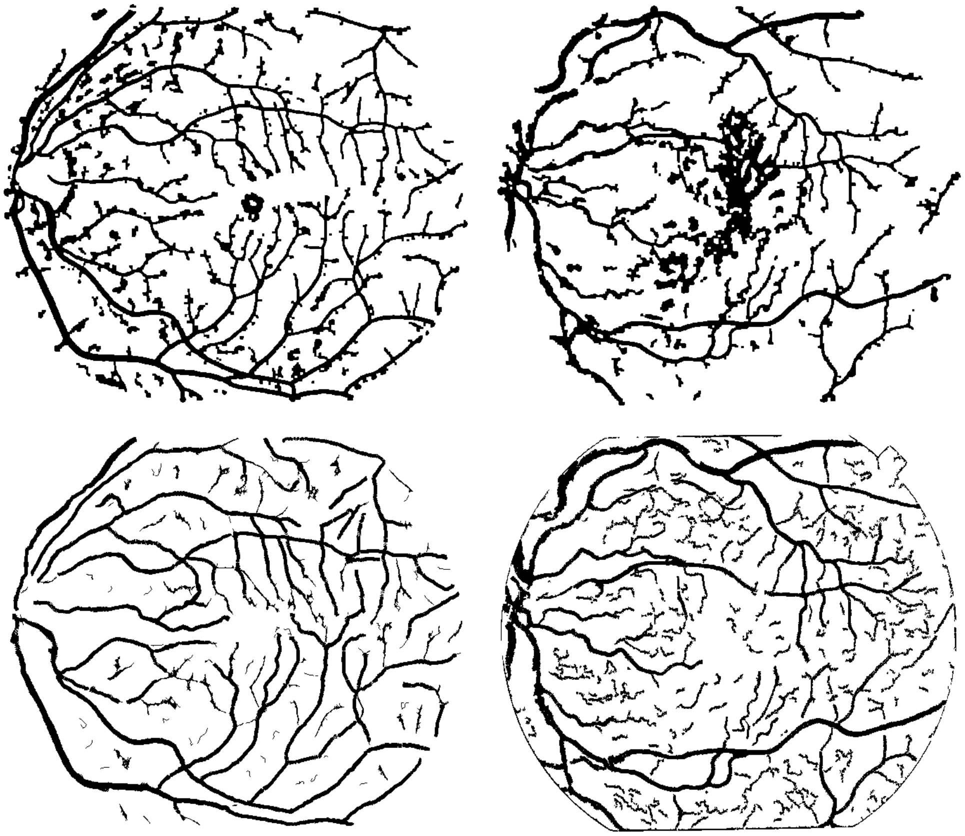


Figure 14 Comparison between Jiang's result and ours with the same central line based performance. First row, Jiang's result. Second row, our result.

One thing that must be mentioned is that the current algorithms, including this one, are tested on STARE, in which all images are of low resolution. As we mentioned in the previous sections, small vessels and micro-vascular are smeared or have very low contrast in the downsized images. But even in full resolution images (3000x2000 pixels, see a crop of image with small vessels in Figure 15), they are still very shallow. Our Gabor-variance filter can effectively enhance the shallow vessels and with the proposed tracing algorithm, a relatively complete vessel map can be extracted.

For small vessels, the precise vessel width is of less use and the structure of vessels contains more information, so the central line based performance measure is used here. The algorithm is tested on 11 images cropped from full resolution images that contain small vessels. The detection performance is show in Figure 16. The algorithm has 84.75% TPR and 0.15% FPR.

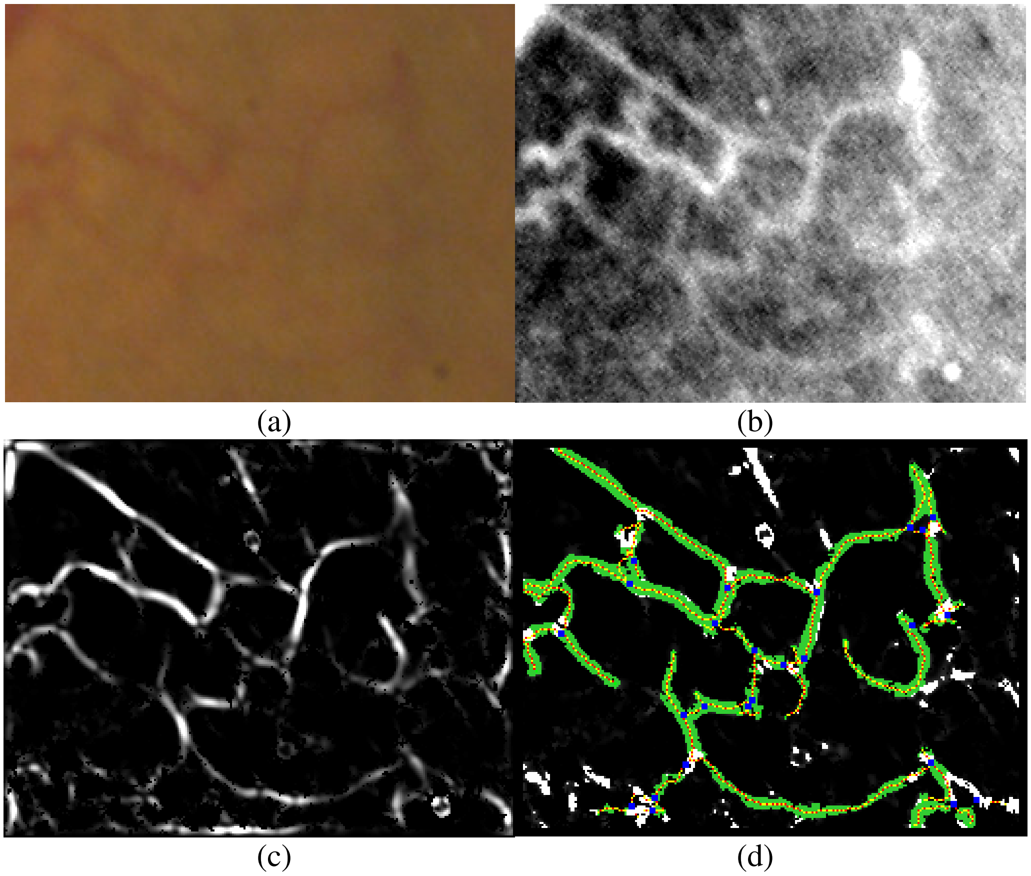


Figure 15 (a) A crop of image in full resolution that contains very shallow small vessels. (b) Simple enhancement by cutting off 1% of highest and lowest of the intensity histogram of the inverted green channel and stretching the rest. (c) Applying the Gabor-variance filter (d) The detection outcome by tracing. Blue dots are possible bifurcation points.

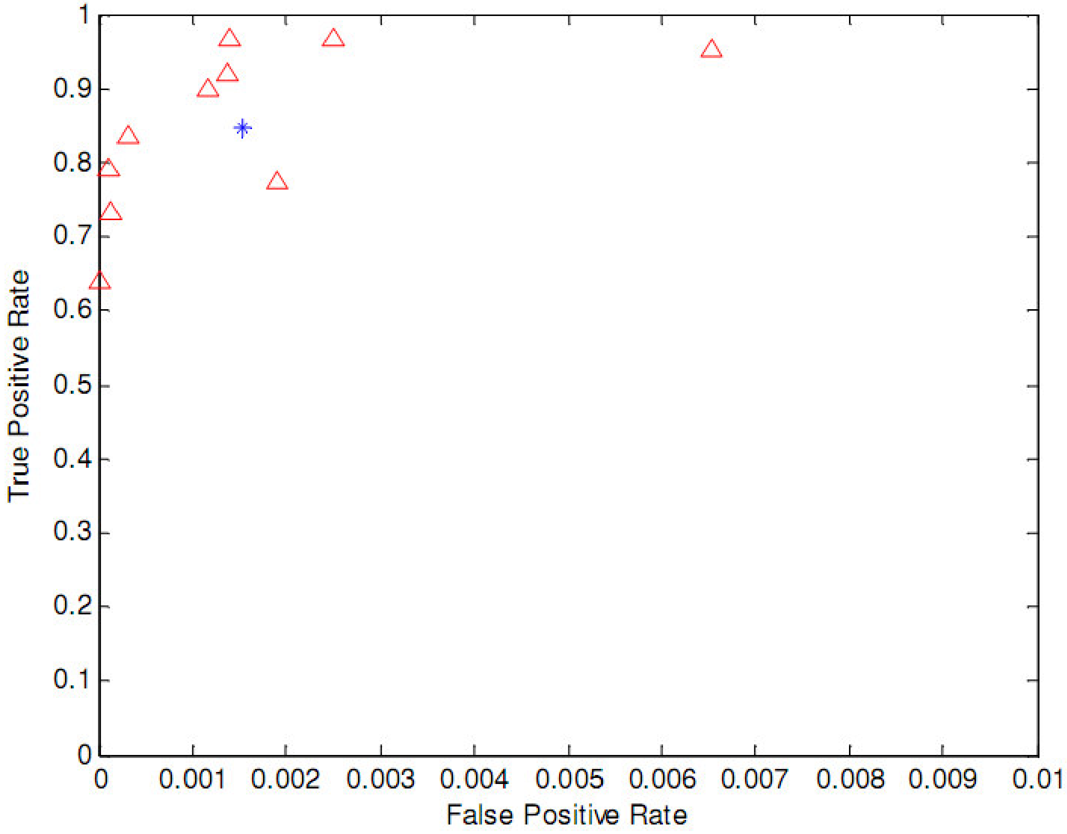


Figure 16 Performance of the tracing algorithm on high resolution images.

Because of the large size of the high resolution image, the processing time increases dramatically with the size being larger. Adaptive histogram equalization uses a big window with size 80 pixels and the forward prediction probes 32 directions, which involves 32 times convolutions with the original image. These are all time consuming. As stated above, to process one image with size 700x605, it takes 40 seconds. We will discuss another faster solution on high resolution images in the next section.

C. RADIAL CONTRAST BASED ALGORITHM

Most algorithms in the literature, including the last algorithm we proposed in the previous section, demonstrated different levels of success. But most work did not have a clear understanding of the intrinsic reasons why they perform in certain ways. There is a great need for the fundamental understanding of the blood vessel *detection features,* so that one can have a better understanding of the nature of the blood vessel pixels in the feature space, which would help tradeoff analysis of sensitivity, selectivity, and object characteristics, among other factors. The goal is to be able to push for the (highest) spatial resolution, (lowest) illumination levels, and (minimal and simple) parameter selections.

Anatomic study [14] suggests that BV branches can be characterized by *generation levels* based on their lengths, widths and locations. Up to nine generation levels can be identified from some BV networks. Most existing work considered the BV pixels have homogeneous properties and did not consider this issue explicitly. It might not be a major issue to disregard this property for low detection resolution investigation (large vessel detection), but our study suggests, as it will become clear later, one can improve the detection resolution, e.g., the smallest and shallowest BV segments which can be detected, by effective exploration of the similarity in pixels located at the same generation level. Illumination condition of image taking can also affected the detection performance significantly. The illumination can be significantly degraded by non-mydriatic clinical procedures, dark pigments of the patients, room lighting conditions, or inexperienced operators. Furthermore, the anatomical architecture of the retina suggests that blood vessels traverse among the seven thin retinal layers with the neural tissues at their proximity. As a result, illumination/intensity based detection methods may have difficult parameter adjustment problems. This issue could be remedied by equalization algorithms, or adaptive parameter adjustment, but the stability of these methods is not robust to the aforementioned factors in the extensive field experiments.

To address all these design issues, we developed a *radical contrast* based algorithm to detect different generation levels of blood vessels under various illumination conditions. Studies in human vision systems reveal that human eyes use contrast of image objects for inference of visual information. We developed a *radial contrast transform* (RCT) to map intensities of pixels to their *radial contrast vectors (RCV),* and then two functions measuring symmetry and energy are applied to the RCV(s) to form a two dimensional *feature space*. Different functions based on RCV can be designed to reduce the dimension of RCV and make pixels located at the BV of a generation level be clustered in close proximity in the feature space. This way, pattern recognition techniques can be applied to facilitate detection and classification of different generation levels of the BV network. In this work, it is found that simple thresholding technique can be used to do pattern separation to get the vessel pixel candidates. And then a procedure called shape denoising is utilized to remove pixels that are unlikely to be on the vessels. A post-processing step is then applied to generate the final binary vessel map.

In addition to design of pattern classification techniques for analysis of the feature space, the design of the transformation from illumination/intensity to contrast is just as critical. Major design factors of the transformation include the effects of *illumination*, object *shapes* (morphological structures), and *sampling* of (BV and background) pixels.

The transformation should be able to distinguish the shallowest vessels that are visible to human eyes. Blood vessels can be roughly characterized as relatively dark, long and thin objects that are visible to the naked eye even for very small vessels, when the intensity difference from the background is higher than the *just noticeable difference* (JND) level, i.e., 2% or higher of difference in the intensities of an object and its background. JND is a form of contrast measure. We adopt a contrast measure to screen for darker pixels (because BV pixels are darker than their background), and use the JND as the detection threshold of BV pixels, so that the limitation of computer based algorithms can be made consistent with that of the human vision system. The shape information is another important characteristic of an object. The transformation should allow people to construct, with the coefficients obtained through it, various feature space (e.g., the symmetry/energy space in this work and maybe some others) to cluster points that correspond to the BV pixels in feature space. Traditional contrast definitions are all isotropic and not able to describe non-isotropic shaped objects, e.g., blood vessels. Inspired by the idea in [16] and based on the contrast definition in [46], we developed the radial contrast transform that takes contrast information on different directions around the pixel being considered. The sampling problem is usually an issue for analog filters. Vessels, even in large size, have a limited pixel count through the cross-section. Those with a smaller size may only have 1-2 pixels in width. Analog filter based algorithms may fail because of insufficient sampling space. Since we are using the contrast, which is calculated with discrete information, we don’t have this problem.

1. RCT (Radial Contrast Transform) design

The goal of radial contrast transform is to construct a new contrast domain on which different kinds of analysis can be done. Analysis in intensity domains has some obstacles. Intensity based algorithms have a major drawback, different illumination condition or non-uniform intensity distribution make the parameter setting difficult for this kind of algorithm. This issue could be remedied by equalization algorithms, or adaptive parameter adjustment, but the stability of these methods is not robust to the aforementioned factors in our extensive field experiments. The shallow objects, such as small blood vessels, may have intensity values that are very close to their surrounding background. With intensity based solutions, it is hard to identify them from the background, which is the major reason why most intensity based algorithms failed to detect small vessels. By transforming to the contrast domain, the detection resolution can be dramatically increased. We will discuss this in detail later in this section.

A good transformation should be able to make it possible to distinguish the shallowest vessels that are visible to human eyes. It is well known that human vision systems use contrast of image objects for inference of visual information. Contrast has some outstanding merits for vision systems. First, it's intensity invariant. Contrast is a relative measure between an object pixel and its surrounding background. The overall intensity change in the retinal image won't change the contrast significantly. Second, the smallest contrast which is perceivable to human eyes has a psychological limit: *just-noticeable-difference* (JND). The notation JND is established by a 19th century psychologist Ernst Weber, after which Weber's measure is named. He also found that the ratio between

JND and initial stimulus is a constant 2% (or, 0.02), denoted by C_{min} in this work. Thus, JND can be used as a guideline to find proper parameter settings if necessary.

Contrast is defined as the ratio between the average intensities of an area and its surroundings [47], where the sizes of the analysis window needs to be decided based on the application needs. Michelson measure [48] and Weber measure [49] are two popular contrast measures. Michelson measure is defined as $C = \frac{f-b}{f+b}$, where f is the maximum luminance in the image and b is the minimum luminance. This measure is often used in images with simple periodical patterns. For images with complex patterns or texture, Weber's measure is more commonly used. Weber's measure is defined as $C = \frac{\Delta L}{L}$, where ΔL is the difference between the object's luminance and the background luminance L. This definition was improved with many other measures. Watson [50] gave the definition of the contrast at an image pixel as $I(x, y) = I_0[1 + C(x, y)]$, where $C(x, y)$ is the contrast at point (x, y) and I_0 is the uniform background luminance. Carbonaro and Zingaretti [46] proposed a contrast definition at pixel (x,y) as $C(x, y) = \frac{[I(x, y) - \overline{I}(x, y)]}{\overline{I}(x, y)}$, where $\overline{I}(x, y) = \sum_{i=x-p}^{x+p} \sum_{j=y-q}^{y+q} w_{ij} I(i, j)$ is the weighted average intensity of the neighborhood of the current pixel. In [49], Peli proposed a local band-limited contrast $c(x, y) = \frac{a(x, y)}{l(x, y)}$, where the band-limited image $a(x, y) = I(x, y) \otimes g(x, y)$ is derived by convolution of the image and a band-pass filter $g(x, y)$, and the local luminance mean image $l(x, y)$ is convolution of the original image and a low-pass filter whose frequency range is lower than the band-pass filter. Lin and Kao [51] defined a set of contrast in different frequency bands, $2^j f_{0,}$ *j= 1,2,* ⋯ The raw image is filtered by a set of band-pass filters $g_j(x,y)$ whose center frequencies are $2^j f_0$. Let $d_j(x,y)=r(x,y)\otimes g_j(x,y)$, the background luminance is $b_j(x,y)=\sum_{i=1}^{j-1} d_i(x,y)$, and the contrast at band j can be defined as: $c_j(x,y)=\frac{d_j(x,y)}{b_j(x,y)}$.

All these definitions are isotropic, i.e. they don't have directions. For objects like spots, it's not necessary to consider any direction information. On the other hand, it's not easy to describe the shape of vessels with these non-isotropic contrast definitions. To allow different types of objects to be distinguished, directional measure is necessary. Inspired by the idea in [16], we developed a *radial contrast transform* (RCT) to map the image pixel intensities into a high dimensional radial contrast domain. The number of dimensions is the number of directions along which the contrast values are calculated. Let p be a pixel with intensity I_p, and follow the contrast definition proposed in [46], the contrast of p along direction θ is defined as:

$$C_p^{(\theta)}=\frac{I_p-\overline{I}_p^{(\theta)}}{\overline{I}_p^{(\theta)}}$$

where,

$$\overline{I}_p^{(\theta)}=\frac{1}{r}\sum_{q\in N_p^{(\theta,r)}} I_q$$

$\overline{I}_p^{(\theta)}$ is the average intensity of p's neighbors along direction θ, r is the distance span of the neighborhood. The neighborhood of p along direction θ with size r is defined as:

$$N_p^{(\theta,r)} = \{(x_q, y_q) \mid x_q = \lceil x_p + k\cos\theta \rceil, y_q = \lceil y_p + k\sin\theta \rceil, k = 1..r\}$$

The neighborhood of p along direction θ is shown in Figure 17. All directional contrast values form the *radial contrast vector* (RCV) of p , denoted as $\mathbf{C}_p$.

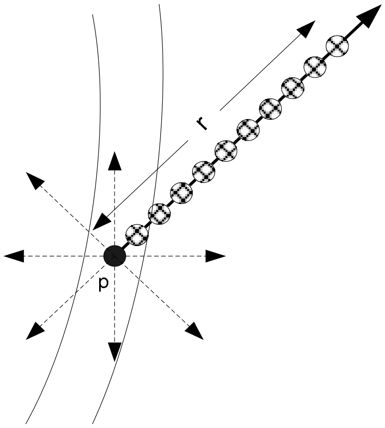


Figure 17 The linear analysis window along the direction θ**.**

The sampling problem is usually an issue for filter based solutions. Vessels, even in large size, have limited pixel count through the cross-section. Those with smaller size may only have 1-2 pixels in width. Filter based algorithms may fail because of an insufficient sampling space. RCT is based on the average intensity of p's surrounding background, thus the intensities of individual background pixels don't have much impact on the contrast value.

2. Feature descriptor

The radial contrast vector usually has a pretty high dimension (see parameter discussion in Section 3.3). It's possible that some analysis can be done on this high dimensional space directly, e.g., some training/classification algorithm can be done to identify the vessels from the result. But this classification procedure is coarse and has no intuitive guidance from the human. Also, the computational cost is very high on high dimensional space. With the RCV, different *feature descriptors* can be designed to identify the shape of particular objects, and thus add heuristics to the detection algorithm for those objects.

For blood vessels, we observed the following characteristics: 1) Blood vessels can be roughly characterized as relatively dark, long and narrow objects in retinal images; 2) the intensity of vessels change slowly along the vessels; 3) in short distance, the vessel width and direction don't change significantly, i.e., vessel pieces in short length can be modeled as a linear segment. Since intensity of vessel pixels doesn't change sharply along the vessel direction, vessel pixels should have fairly small contrast (either positive or negative) along the vessel direction. At the same time, since vessel pixels have lower intensity than the background, they have negative contrast along other directions along which their neighborhood doesn't fall completely inside the vessel. And, pixels around the center of the vessel, with high probability, will have similar intensity distributions across the vessel cross-section to the background along opposite directions. Thus, they will have similar contrast values along opposite directions. If the contrast values are drawn in a polar coordinate system (to distinguish positive and negative values, they are drawn in different colors, positive in red, negative in blue), one can see that the envelope of the values is of symmetric petal-like shape (See pixel 10-13 in Figure 18 and their contrast plots in Figure 19). We can design a descriptor called *symmetry descriptor*, denoted by D^S to describe this characteristic. The symmetry measure of a pixel (x, y) evaluated with D^S is defined as follow:

$$D^S_{x,y} = \frac{\sum_{m=0}^{N/2-1} \left| \mathrm{C}_p^{(\theta_m)} - \mathrm{C}_p^{(\theta_{m+N/2})} \right| / \overline{\left| \mathrm{C}_p \right|}}{\left| \mathrm{Max}(\mathrm{C}_p) - \mathrm{Min}(\mathrm{C}_p) \right|}$$

where N is the number of dimensions of RCV space (probed directions), $m \in [0..N-1]$ is index of probed contrast directions, $\theta_m = 2\pi \frac{m}{N}$, $\mathrm{Max}(\bullet)$ and $\mathrm{Min}(\bullet)$ take maximal and minimal values. In the numerator part, differences between contrast values of two opposite directions are summated and normalized by $\overline{\left| \mathrm{C}_p \right|}$, the mean of contrast magnitudes (the absolute values). For central pixels on the vessel, the contrast values have symmetric petal pattern, thus the summation is small. $\overline{\left| \mathrm{C}_p \right|}$ of pixels on vessels are much larger than those on the background, so after normalization the symmetry measure of vessel pixels becomes even smaller. The closer the pixels are to the edge, the less symmetric the two petals are, and the larger the summation is. For those pixels on background (See pixel 1-9, 14-16 in Figure 18 and their contrast plots in Figure 19), the contrast values are random, and contrast values on opposite directions usually will not cancel out each other, thus the summation will be large after normalization. The absolute value of the difference between maximal and minimal contrast value is a detector of the peaks along vessel directions.

As we stated above, the vessel pixels have lower intensity than the background, thus their contrast values along the directions other than vessel directions should be some negative values. Therefore, these values should be smaller than the just-noticeable-difference (We say that these contrasts are *negatively sharper* than C_{min}), so they can be perceived by the human eyes. Along the vessel directions, since the pixel intensities don't change sharply, the magnitude of the contrast along these directions are small. Furthermore, the number of directions that fall inside the vessel is small too. Thus, we can expect that for most vessel pixels the summation of their contrast values is lower than that of most background pixels. So, another feature descriptor D^{E}, the energy of the contrast can be used as another heuristic to distinguish vessel pixels from background ones. The contrast energy of a pixel (x, y) is defined as follow:

$$D_{x,y}^{E} = \sum_{m=0}^{N-1} \mathrm{C}_{p}^{(\theta_m)}$$

For blood vessel pixels, they have symmetric negatively sharper contrasts than C_{min} along opposite directions (except those along vessel directions). And background pixels have random contrast values along different directions, so they don't have this characteristic. Therefore, vessel pixels should have more number of contrast pairs that are both negatively sharper than C_{min}. A new feature descriptor D^{NP} applied on a pixel (x, y) is defined as follows:

$$D_{x,y}^{NP} = \mathrm{count}(\theta_m | \mathrm{C}_{p}^{(\theta_m)} \leq -C_{\min}, \mathrm{C}_{p}^{(\theta_{m+N/2})} \leq -C_{\min}, m = 0..N/2-1)$$

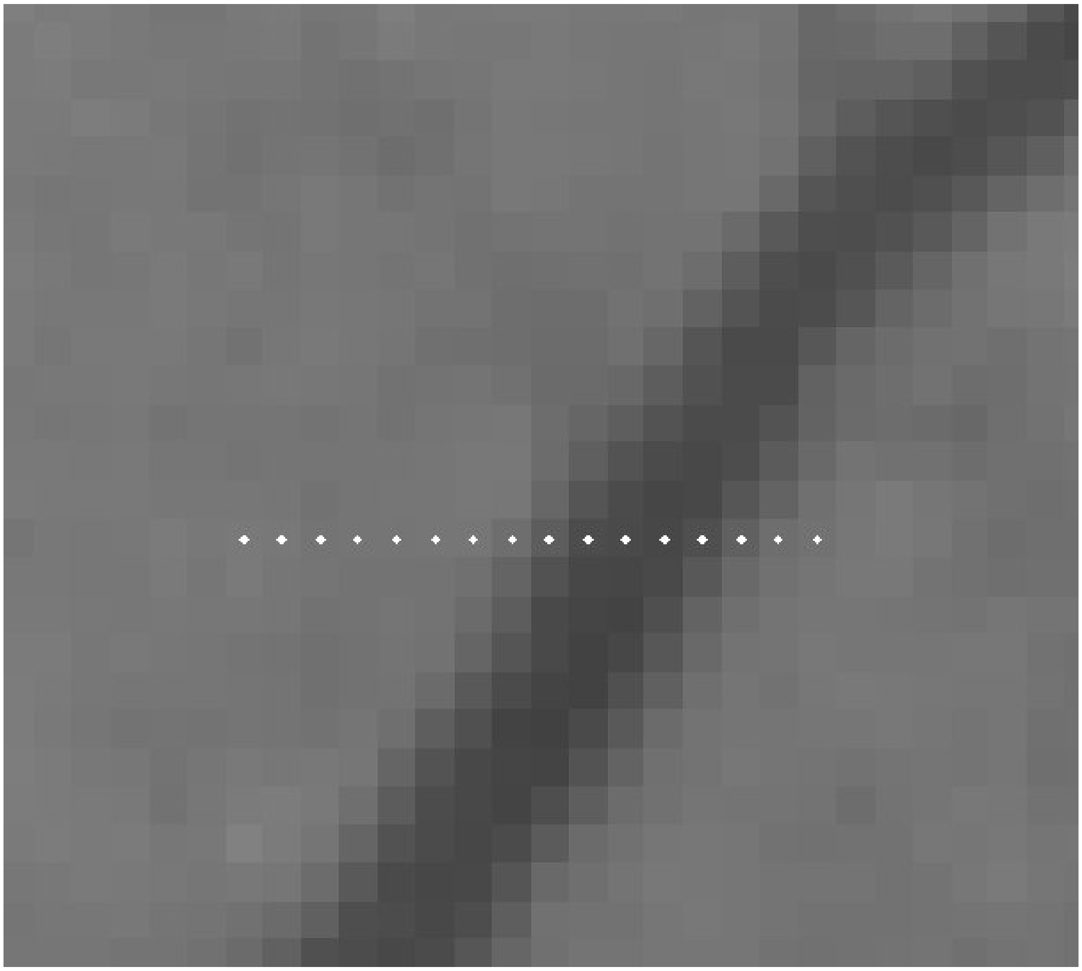
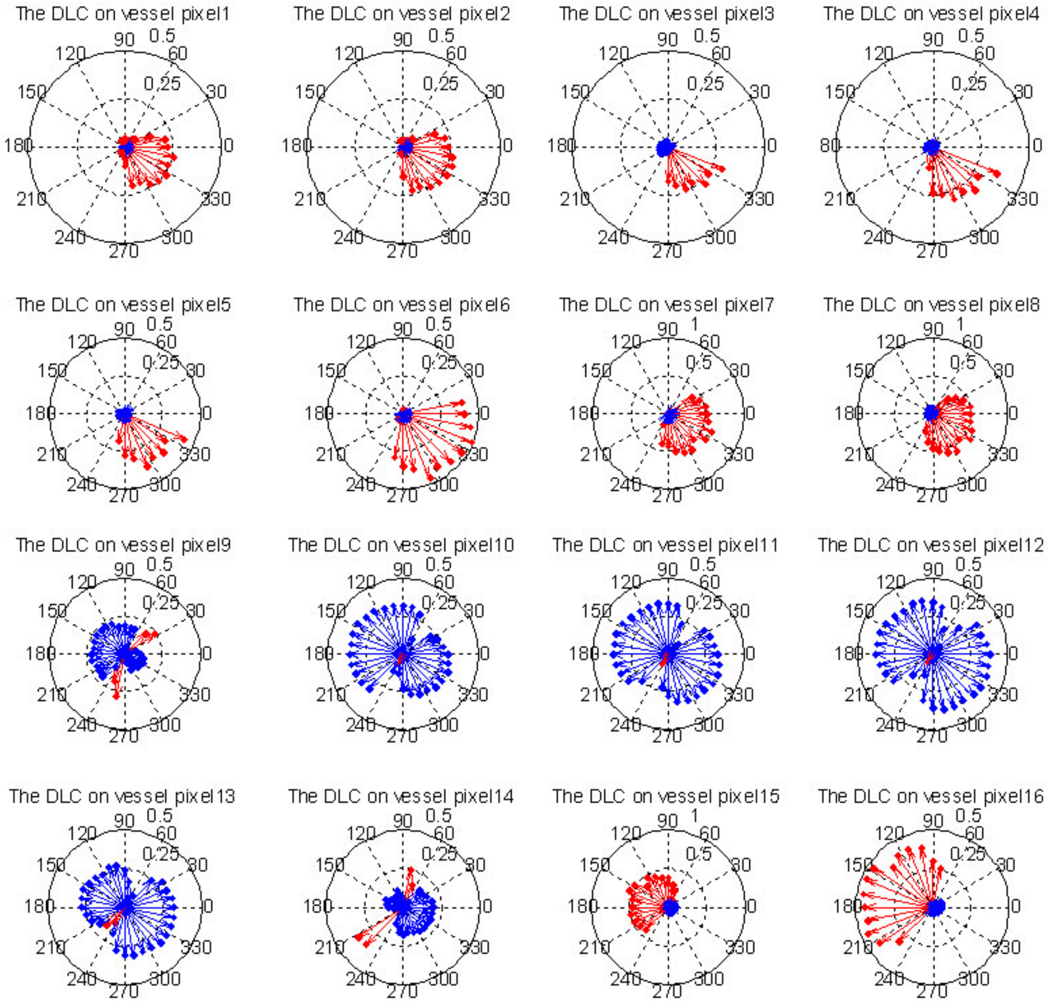


Figure 18 16 pixels that run across a vessel (im077 in STARE, (425, 442) – (440, 442)).



The DLC on vessel pixel1

The DLC on vessel pixel2

The DLC on vessel pixel3

The DLC on vessel pixel4

The DLC on vessel pixel5

The DLC on vessel pixel6

The DLC on vessel pixel7

The DLC on vessel pixel8

The DLC on vessel pixel9

The DLC on vessel pixel10

The DLC on vessel pixel11

The DLC on vessel pixel12

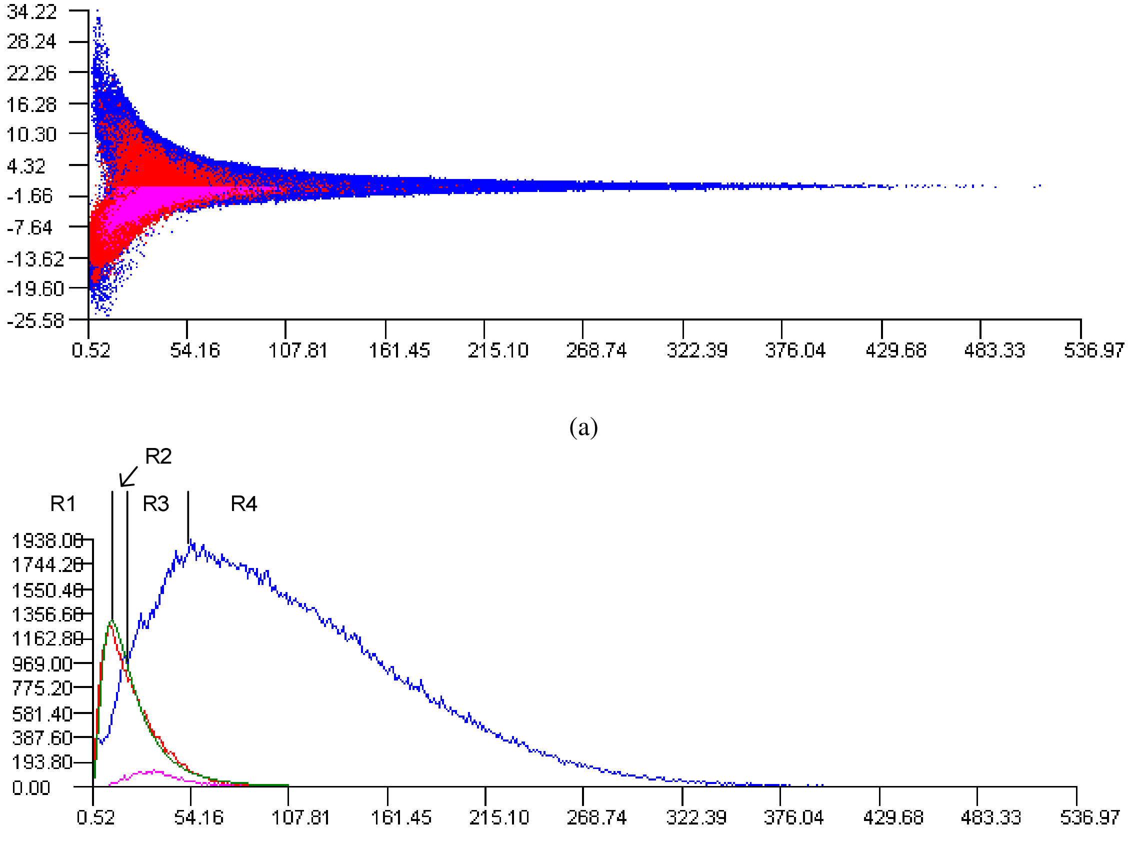
The DLC on vessel pixel13

The DLC on vessel pixel14

The DLC on vessel pixel15

The DLC on vessel pixel16

Figure 19 Contrast plots of the 16 pixels in Error! Reference source not found. **in polar coordinate, where negative values are show in blue, positive values in red. Neighborhood size r=15. Contrast values along 32 directions are calculated.**



(a)

(b)

Figure 20 (a) The symmetry/energy feature space (b) The histogram plot of symmetry measure. Curve in red is for vessel pixels, blue for non-vessel pixels, purple for edge pixels. Green curve is the fitted lognormal curve. R1-R4 are marks of different regions on the plot.

2.1 Distribution of vessel pixels in feature spaces

Using the feature descriptors, the dimension of the problem space can be reduced dramatically while heuristics is achieved to guide the detection. In this section, we use D^S and D^E to build a two dimensional feature space, and do some analysis on it. We use the STARE, a very popular benchmark retinal image database, to evaluate the effectiveness of the feature descriptors. STARE provides two sets of hand labeled results, label-ah and label-vk. In our experiment, we use label-vk, which has much more small vessel details than label-ah, as the ground truth. Distributions of vessel/non-vessel pixels in feature space are analyzed, and their relation is studied. And at last, we found that a simple threshold based classification can be done to effectively separate the two categories.

2.2 Histogram plots in feature spaces

Histogram plots in feature space of both vessel (C1, in red) and non-vessel pixels (C2, in blue) are shown in Figure 20:

- Marginal distribution of $D_{i,j}^{S}$

It is found that vessel pixels exhibit a single-tailed distribution, and that non-vessel pixels exhibit sigmoid curve like distribution, the slope of which differs from the one image to another due to the different non-vessel pixel properties in each image. C1 decays while C2 increases. And it's also observed that when C1 approaches zero, C2 is about to reach its peak value.

The pixels with symmetry measures close to zero are those on the central line of the vessels or those pretty close to them. The number of these pixels is small. As the symmetry measure grows, more and more vessel pixels farther from the central lines are included on both large and small vessels. This creates a sharp increase in the number of vessel pixels. When the symmetry measure approaches the value of the edge pixels, pixels on small vessels drop out first and then those on larger vessels do. This procedure is characterized on the vessel pixel histogram plot (C1) as a gradually decayed long tail. On the background pixel histogram plot (C2), pixels with small symmetry values are mostly those on the edge of the field of view (FOV) or those which fall right between two vessels, in which area the contrast values are almost all positive and symmetric. But these pixels will have high energy measure, which will be discussed later. When symmetry value becomes larger, background pixels near the vessel boundary are included. Because our symmetry measure is always positive, it is not able distinguish the pixels inside/outside the vessels near the vessel boundaries. The pixels inside/outside the boundary in similar distance to the boundary will have similar symmetry measure. We will use the energy measure to classify them later. The number of background pixels keeps increasing as the symmetry value increases. And the corresponding background pixels are farther and farther away from the vessels. When the vessel is out of the neighborhood of the background pixels, the background pixels are of uniform intensity plus some random noise inside their neighborhood, which leads to a high symmetry measure.

The image pixels can be divided into four groups based on their symmetry measure (See Figure 20(b)), by the symmetry measure values that correspond to the peak count value of vessel pixels, background pixels and the histogram value that the two curves meet. Pixels in R1 corresponds to the strong vessel candidates and number of non-vessel pixels in this region is rather small and most are at the rim of the edges of blood vessels. It’s pretty subjective for different image readers to include them in the vessel pixel set or not. Pixels in R2 and R3 are the main overlapping area of vessel and non-vessel pixels. This can be foreseen by the properties of the “fuzziness” of $D_{i,j}^{S}$ term where vessel edges and furcation pixels are all included. In R2, the count of vessel pixels dominates non-vessel pixels. The non-vessel pixels in this area usually indicate the edges between bright spots where pixels have the similar properties as that of vessel pixels (e.g., the dark area between two hard exudates). In R3, non-vessel pixels count dominates and these non-vessel pixels consist mainly of the pixels very close to the edges of the vessels. Threshold is made in R3 using the adaptive thresholding method that will be illustrated in Section C.3.1. Most of the vessel pixels are retained at the cost of the misclassification of some points close to the edges of the vessel as well as the edges between bright spots as vessel pixels. Further steps will be done in removing these misclassified points by *shape denoising* illustrated in Section C.3.2. Pixels in R4 are all classified without doubt as non-vessel pixels.

- Marginal distribution of $D_{i,j}^{E}$

It is found that nearly all the vessel pixels present values below zero while a large portion of non-vessel pixels have positive value. For vessel pixels that have positive $D_{i,j}^{E}$ values, further experiment shows that these are pixels very close to the edge of the big vessels. It's highly subjective to include these pixels as the vessel pixels or not. For example, we noticed that in STARE, even the two image readers have different decision on the vessel boundaries. And these pixels do not affect the extraction of blood vessel skeleton. In the applications such as vessel tortuosity analysis and image registrations, these pixels don't matter at all. In these applications, simple hard thresholding with threshold 0 can be used to choose the candidates of vessel pixels at the cost of missing only a few edge pixels. We will have more discussion on this later.

- How vessel and edge pixels are related in feature space

We further conduct the experiment to study the relationship between distributions of edge and vessel pixel histogram along $D_{i,j}^{S}$ axis. The plot (C3) shown in Figure 20 in pink color represents the pixel count of edge pixels versus the symmetry values. It is found that C3 is enclosed in C1 since edge pixels are a portion of the whole vessel pixel set. Also, there is a peak shift between the two plots with the peak of C3 lagged behind that of C1. This is because of the fact that the vessel pixels with small symmetry value are those around the central line of blood vessels while most edge pixels are of smaller symmetry measure.

2.3 Mathematic modeling of distribution of “pixel count vs. $D_{i,j}^{S}$“ of vessel pixels

The distribution of “pixel count vs. $D_{i,j}^{S}$“ of vessel pixels can be modeled as the log-normal distribution with the probability density function of the form:

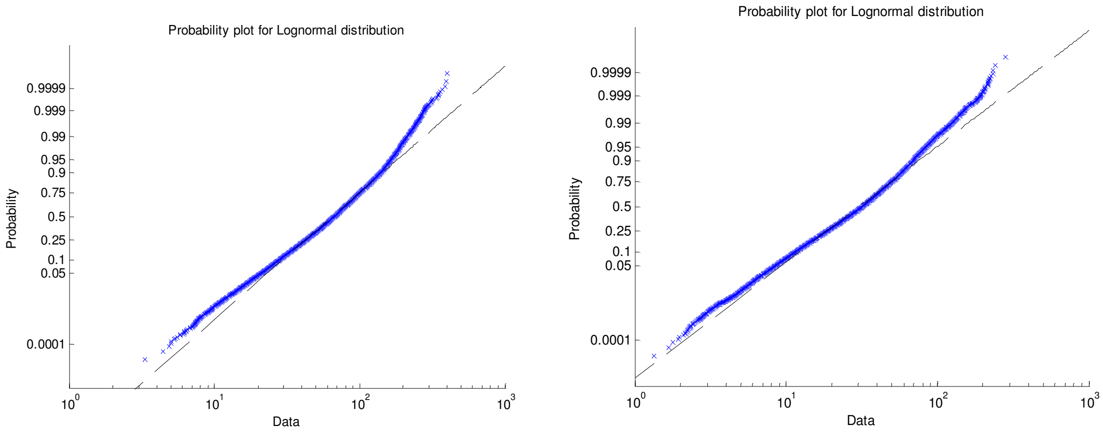
$$f(x;\mu,\sigma)=\frac{1}{x\sqrt{2\pi\sigma^2}}\operatorname{Exp}(-\frac{(\ln x-\mu)^2}{2\sigma^2})$$

for $x > 0$, where μ and σ are the mean and standard deviation of the variable's logarithm (by definition, the variable's logarithm is normally distributed).

Log-normal distribution parameters μ and σ can be determined using the maximum likelihood estimation. The estimators for μ and σ can thus be deduced as follows:

$$\widehat{\mu} = \frac{1}{n}\sum_{k} \ln x_k \text{ , } \widehat{\sigma}^2 = \frac{1}{n}\sum_{k} (\ln x_k - \widehat{\mu})^2$$

where, x_k is the $D_{i,j}^S$ value for each vessel pixel and n is the total number of hand-labeled vessel pixel counts. The green curve in Figure 20 is the lognormal distribution curve with parameters estimated from the vessel pixels and normalized by the histogram bin size. It can be seen that the theoretical curve matches the actual vessel curve well. The matching of the distribution and the actual vessel pixel count can also be shown with a probability plot [52]. Figure 21 shows the probability plot of two images (one abnormal - im0001, one normal - im0077) in STARE. It is demonstrated that lognormal distribution applies well to the modeling of the vessel distribution in $D_{i,j}^S$ descriptor space, except in the area when the symmetry measure is high. The number of vessel pixels in this area is small, and most of these vessel pixels are on the boundary of the large vessels or very fine small vessels. At these pixels, the radial contrast values are small and of more randomness. Thus their distribution may deviate from the lognormal distribution. At the lower end of the plot in im0001, there is also a bit deviation, which is in fact pretty small (check the ticks on y-axis), and they may be caused by the bright hard exudates in the image. Taking the exponential of the normalized pixel count, if the converted data samples match the normal distribution, the original data set matches lognormal distribution. Thus, by testing the normality of the converted data set, we can verify if the original data set matches the lognormal distribution. Shapiro-Wilk test [53] is the most well known method to test the normality of given data set. The test result shows that with the significant level set to 0.05, the null hypotheses are rejected, for both vessel and edge pixel count. Thus the vessel and edge pixel histograms both match the lognormal distribution.



(a) (b)

Figure 21 (a) Probability plot of an abnormal image im0001 (b) plot of a normal image im0077.

3. Adaptive candidate vessel pixels detection

The previous section discussed the mathematical modeling of the distribution of the vessel pixels and that of edge pixels against $D_{i,j}^{S}$ measure. Similar discussion is done for the $D_{i,j}^{E}$ measure. It is also discussed how the two measures are related to each other. In this section, we take advantage of these properties to identify the vessel candidate pixels from the background pixels in the 2D feature descriptor space. It is conducted in three steps.

3.1 Adaptive threshold in $D_{i,j}^{S}$ and $D_{i,j}^{E}$ descriptor spaces

As we discussed before, edge pixels histograms fall inside that of the vessel pixels. C3 decays with C1, and about the same time, they approach zero. This is because the vessel pixels, with larger symmetry measures, are those close to vessel boundaries, i.e., the vessel edge. We have no idea what the vessel pixel histogram is since it needs the knowledge of the vessel pixel locations which is exactly what we are looking for. But we can get the histogram of edge pixels with edge detectors. A simplified Canny edge detector is used to detect the vessel edges in a retina image. In this simplified version, after the non-maximal suppression step, the procedure of connecting the "broken" edges is omitted since it's not necessary that the edge map is complete and contains all the vessel edges. As far as most large vessel edges are captured, the edge pixel histogram will be a good proximate to that of the full edge map. The values of symmetry measure of these edge pixels are calculated and a lognormal curve is fitted. The geometric mean μ and the geometric standard deviation σ are used to estimate confidence intervals for the vessel pixel histogram. 2σ upper confidence interval bound is taken, i.e., the symmetry value of $\mu + 2\sigma$ is used as the adaptive threshold in $D_{i,j}^{S}$ descriptor space. As illustrated in the previous section, this adaptive threshold introduces a small portion of non-vessels pixels that are those close to vessel edges and the edges between bright spots (e.g., the dark area between two hard exudates that are close to each other). Further refinement of candidate vessel pixels is done in the next section.

3.2 Refinement of candidate vessel pixels

The candidate vessel pixels contain some background pixels that happen to match the thresholds of the two measures. Further refinement is necessary to remove these "noises" in the detection result. In this step, several methods are proposed.

1) K-Mean algorithm

The two thresholds found in $D_{i,j}^{S}$ and $D_{i,j}^{E}$ descriptor spaces specify a rectangular region in the two dimensional feature space of $D_{i,j}^{S}$ and $D_{i,j}^{E}$. By taking into account the joint probability distribution of pixels in this region in the feature space. Vessel pixels present quasi-linear correlation pattern in the feature space. Thus, the most of the vessel pixels are located around the point with the means of the two lognormal fitting curves of the symmetry and energy measure histograms. At the same time, the misclassified non-vessel pixels are of higher symmetry measures. Therefore, two clusters are expected in the feature space. K-means [54] algorithm with cluster number set to two (vessel and background pixels) is used to do the refinement of candidate vessel pixels. The result of vessel pixel candidates found with K-Mean is shown in Figure 22.

Other pattern classification methods such as SVM, neuron-networks, can also be applied to make the refinement. Note that an important property of blood vessels has not been utilized thus far – the continuity of the vessel, the following scheme is proposed to identify the vessel pixels.

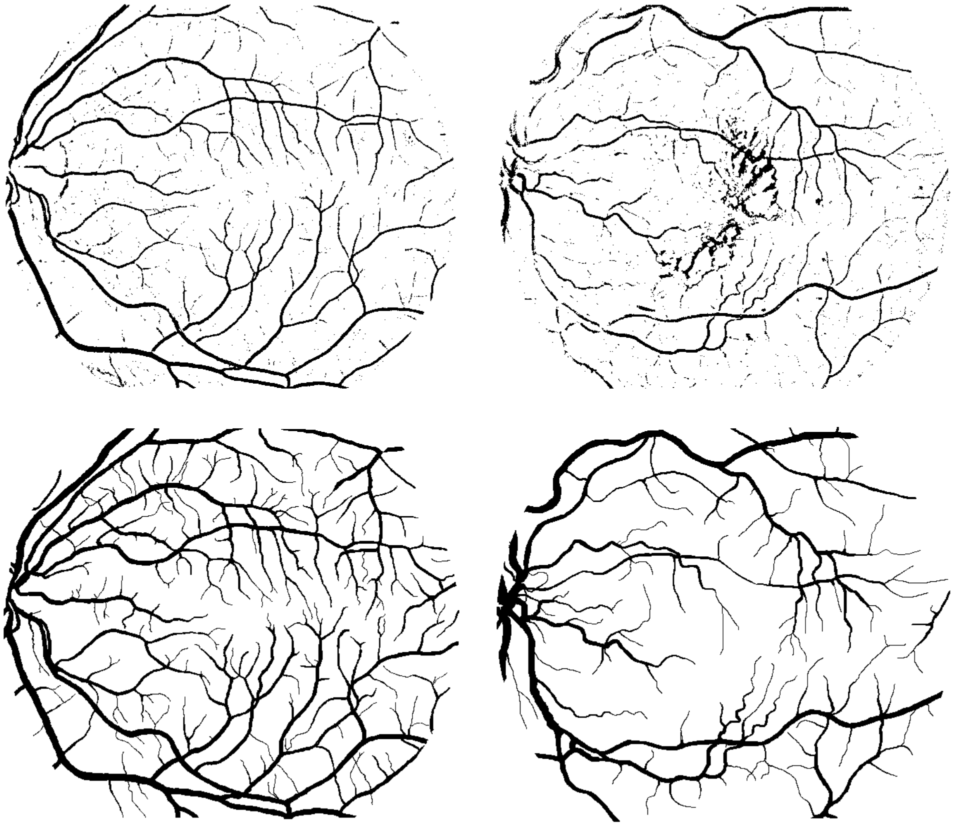


Figure 22 First row, the vessel candidate pixels found with K-Means algorithm. Second row, the second hand labeled vessel map.

2) Shape denoising

The vessel pixel candidates obtained with the two adaptive thresholds still contain non-vessel pixels. Up till now, we haven't used an important property of the vessel: continuity. As we assumed at the very beginning, blood vessels can be modeled as connected linear segments. And the intensity change of vessel pixels along the vessel direction is gradual. This gives us a hint on how to exclude non-vessel pixels, since non-vessel pixels don't have this characteristic. The noise removal process includes the following steps:

- Vessel direction calculation

We know that if a pixel p is on the vessel, along the vessel direction, its neighbors should have similar DCV(s) to p's, since the pixel intensities along vessel and their background usually don't change sharply. We use the following equation to find ϕ_p, the direction of vessel on a vessel pixel p:

$$\phi_p = \min_{\theta \in \Theta} \arg \sum_{q \in N_p^{(\theta,r)}} \left| \mathrm{C}_p - \mathrm{C}_q \right|, \text{ where } \Theta = \{\theta \mid \mathrm{C}_p^{(\theta-\pi/2)} \le -C_{\min}, \mathrm{C}_p^{(\theta+\pi/2)} \le -C_{\min})\}$$

This equation picks the direction along which the summation of the distance between the RCV(s) of current pixel p and its neighbors is minimal. On most vessel pixels, there are two vessel directions, and this equation only picks the one with the minimal response. The other direction can be acquired by simply adding π to ϕ_p.

- Remove the noise

Based on the continuity characteristic, a vessel pixel and its neighbors along vessel directions should have similar ϕ. We do the check over every vessel pixel candidate and mark those satisfying the following inequation as vessel pixels:

$$\mathrm{Count}(\forall q \in \{N_p^{(\phi_p,L/2)} \cup N_p^{(\phi_p+\pi,L/2)}\} \mid \phi_q = \phi_p \pm \varepsilon \text{ or } \phi_q = \phi_p + \pi \pm \varepsilon) \ge \beta L$$

where ε is an error tolerance parameter that allows certain error in the direction fitting; L is the length of minimal length of the linear segment that can be detected and $\beta \le 1$ is another error tolerant parameter . This means, on each pixel p, we check along two possible vessel directions starting from neighborhood size L/2, if the number of neighbor pixels with the same vessel direction (ϕ_p or $\phi_p + \pi$) is larger than βL, we mark p as vessel pixel. The result of the vessel map after shape denoising is show in Figure 23.

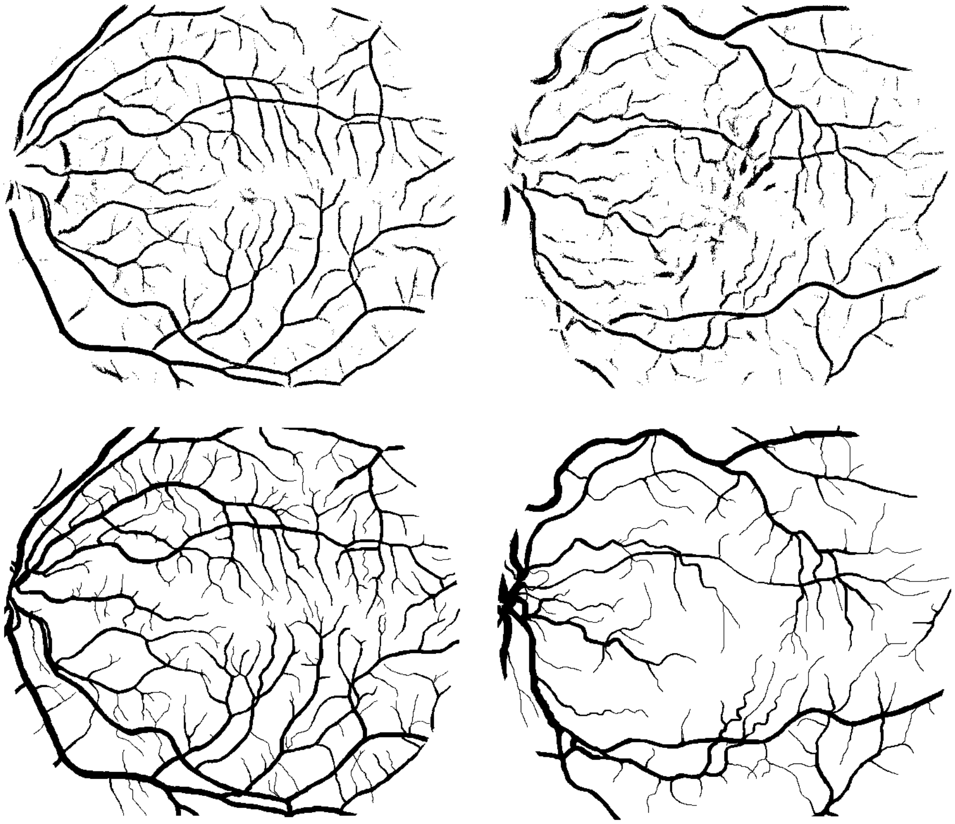


Figure 23 First row, the result of vessel detection after shape denoising; second row, second hand labeled vessel map.

This method has some common points with the line detection algorithm by Duba and Hart [55] with Hough transform. Hough transform can detect parametric curves on a binary image. Usually the binary image is obtained with an edge detector. The location information of the edge pixels are transformed in to a parameter space. Duba and Hart used a normal-parameterization to express the lines, i.e., a line is specified by its distance from the origin ρ and the angle θ between its normal and the positive x-axis: $x\cos\theta + y\sin\theta = \rho$. Fixing x and y, one can get a sinusoidal curve in the parameter space. By restricting the value of θ to be $[0,\pi]$, the sinusoidal curves corresponding to all pixels on the same straight line pass through the same point (θ,ρ) in the parameter space. The detection is done with several steps: 1) Map the pixel on spatial space to curves on the parameter space; 2) the parameter space is quantized with grid of size $d_\theta \times d_\rho$. An counter array of size $d_\theta \times d_\rho$ is created and its elements are assigned to the cells in the grid. Any curve passing through a cell increases the counter of that cell with one. The larger the counter is, the more curves pass through the cell, and the more pixels fall on the straight line in the spatial space corresponding to the cell. By finding the local maxima(s) in the counter array for parameter space, one can detect the lines in the spatial space.

From the pixel level filtering viewpoint, RCT has its root in the Hough transform on the contrast space. First, in our method, the transform is done in a small local range around the pixel being checked (denoted by *p*), i.e., the neighborhood of *p*. Second, there are infinite straight lines that go through *p*, only one of them is checked – the direction θ_p calculated as the vessel direction at *p*. Thus, *p* has a unique mapping point (θ_p, ρ_p) in the parameter space, where ρ_p is the distance of the straight line passing through *p* and has direction θ_p. The detection is done by picking out the point in parameter space whose value in the counter array exceeds a given threshold αL. The small neighborhood size ensures that the distance won't be too large between the detected points on the same line, in order to satisfy the assumption that the vessel can be modeled as linear segments in small neighborhood.

3.3 Parameter setting

In the previous sections, the adaptive thresholding technique was introduced to set the thresholds for symmetry and energy descriptors. There are several other basic parameters that should be set too. First, the neighborhood size *r*. For the neighborhood size should be larger than the width of the largest vessel to be detected. As mentioned in the introduction, the width of arteries in retina is about 0.3mm, which is about 10 pixels in images in STARE database and the images obtained in the EDRSS project. *r* is set to be 15 to provide enough space for the neighborhood to expand. Second, the minimal linear segment length is *L*. This value decides the resolution of the vessel length that can be detected. In this work, it is set to 5 to capture finer vessels. Third, the number of directions is *N*. The ration angle $\Delta\theta = \frac{360^\circ}{N}$ decides the resolution in search of blood vessels. To detect all vessels, this value is set based on the small vessels. Since the smallest vessel width is 1, and L=5, if $L \sin\Delta\theta = 1$, all of the pixels on and between two adjacent directions can be covered. Therefore, $\Delta\theta = 11.5^\circ$, $N = \frac{360^\circ}{\Delta\theta} = 32$. At last, the two error toleration parameters are β and ε. β should be less than 1 but close enough to 1 in order to avoid too many false alarms. In this work, it's set to 0.8. ε decides the tolerance of vessel direction continuity. This value should not be too large, or the assumption that vessels can be modeled as connected short linear segments will be broken. In this work, we use $\varepsilon = 2 \bullet \Delta\theta$.

4. 2-pass fast vessel detection

The previous algorithm performs well on large vessels and parts of small vessels. For those extremely shallow vessels, it doesn't work well. Note that in the shape denoising step in the previous section, the vessel direction is determined by the minimal sum of differences between a pixel and its neighbors along all directions. This holds well in most cases. But for those shallow small vessels, because the contrast values along different directions are pretty small and close to each other, the vessel direction found with this method may not be reliable.

As we assumed in the section of feature descriptor, in short length, vessels can be modeled as a linear segment. Let p be a pixel on the vessel with direction . Since along the direction $\theta \pm \pi / 2$, the most number of of p's neighbors fall in the background, p should have the sharpest negative radial contrast along these two directions. If these two contrasts are negatively sharper than $C_{\min}$, p is visible to human eyes. And, along the vessel direction (θ and $\theta + \pi$) p's neighbors with the size L should have the same property, i.e., having contrast negatively sharper than $C_{\min}$ along the direction $\theta \pm \pi / 2$. L is the length of the minimal segment that can be detected (See Figure 24). So, a simple algorithm can be proposed to detect the finer vessels. The steps are listed as follows:

1) For every pixel p on the image, check if it has contrasts along two opposite directions with values negatively sharper than a threshold T_{BV} , i.e., $C_p^{(\theta \pm \pi/2)} \le T_{BV}$

2) If that direction exists, check if $C_q^{(\theta \pm \pi/2)} \le T_{BV}$, for L of all q in p ‘s neighborhood of size L along direction θ, where is a error tolerance parameter.

3) If both 1) and 2) are satisfied, mark p as vessel pixel.

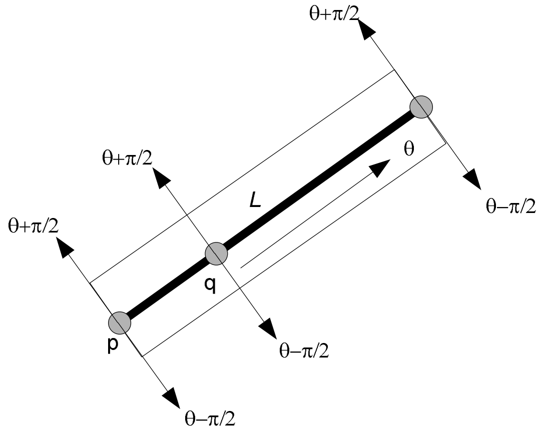


Figure 24 Illustration of fast vessel pixel detection.

With different diameter ranges, large and small vessels should be detected with different parameters in this method. For large vessels, the neighborhood size is the same as in previous section: r=15. Minimal linear segment length L is set to 15 to avoid to many false detections. Error tolerant parameter is set to 0.8. The number of probed directions N is set to 32. And the contrast threshold T_{BV} is set to -0.05, a value with higher contrast to avoid false detections. For small vessels, the neighborhood size is smaller to take more local information for more accurate contrast: r=6. L is set to 5, N=32 and $T_{BV} = -0.02$ to detect more small vessel details. The algorithm is run twice with different parameters and the results are combined to get the final detection outcome.

The detection results of two sample images (im0077 and im0001 in STARE) are shown in Figure 25. The performance of the algorithm is summarized in the performance comparison section. This algorithm is fast, an image of 700x605 pixels (STARE) can be processed in 10 seconds, and it can be further reduced to 1.2 second on a 2x quad-core CPU server with the multi-core code library introduced in Section III. From the example images, one can see that a lot of small vessels have been extracted, but at the same time, some shallow large vessels are missing (see Figure 25, the detection outcome of im0001, second row, right side). This is because of the parameter setting of large vessels doesn't fit the shallow objects. By adjusting the contrast threshold T_{BV} , the missed large vessels can be partly detected, but there would be some over detections. To solve this problem, the algorithm that combines K-mean vessel detection in symmetry-energy feature space and this fast vessel detection is introduced in the next section.

5. Combination of the previous two algorithms

The previous two algorithms both have their own merit. Feature space based algorithm can detect the large vessels more accurately, while fast detection algorithm is more sensitive to small vessels. By combining them together, it is expected that a better detection performance can be achieved. The combined algorithm first use the feature space based algorithm to get most large vessels and then employ the fast detection algorithm with the parameter set of small vessel. The results are combined together to get the final detection outcome. See Figure 26 for the detection outcome of two sample images. The result of fast detection algorithm is drawn in red and superposed on the result of feature space based detection algorithm (in white). The performance of this combined algorithm is compared with other algorithms in the next section.

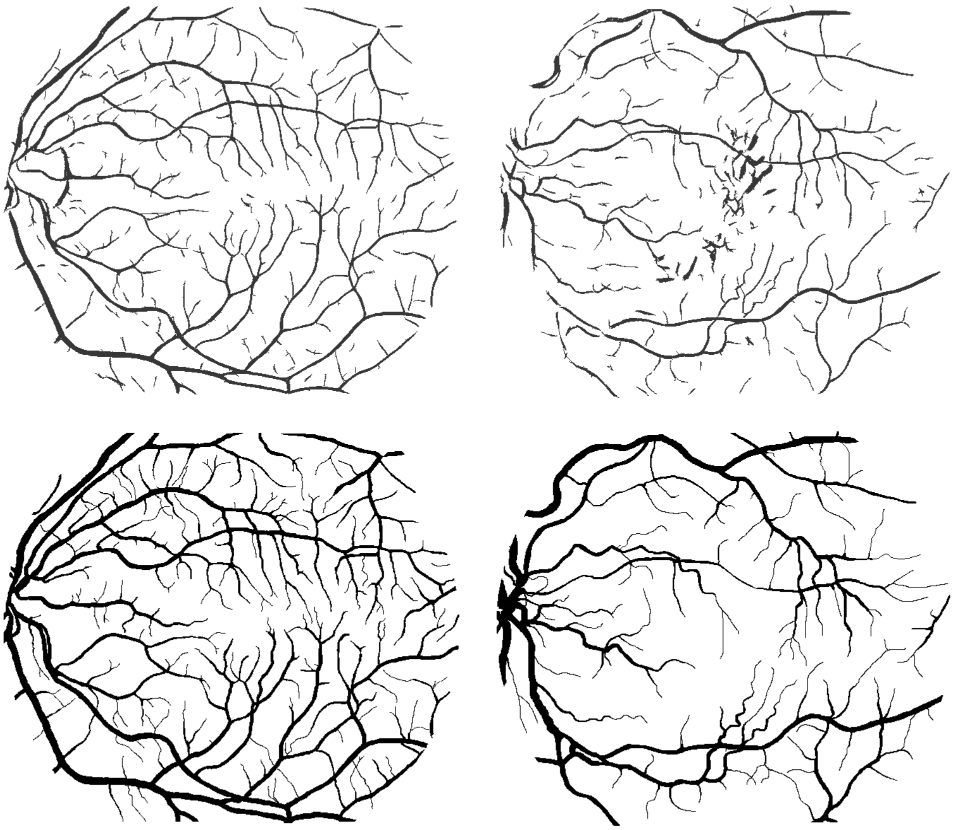


Figure 25 Result of two pass fast vessel detection. First row, the detection outcome of 2-pass fast detection algorithm. Second row, the second hand labeled vessel map.

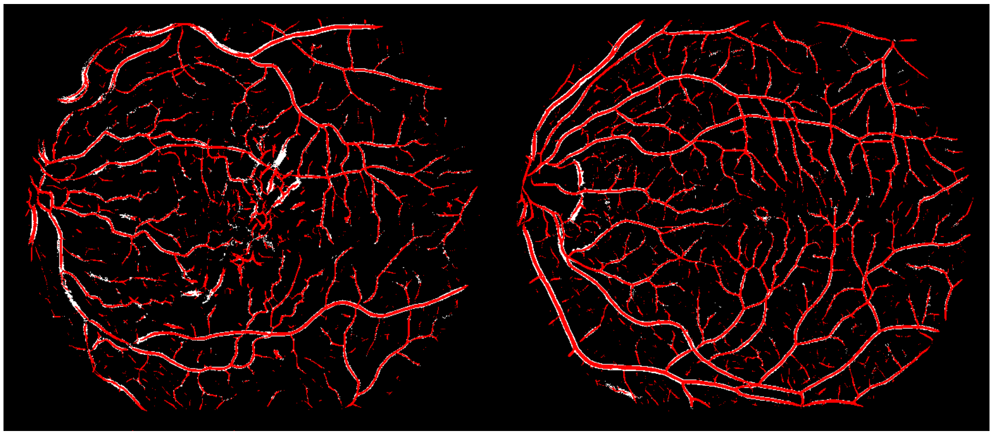


Figure 26 The detection outcome of the combined algorithm. The result of fast detection algorithm is superposed on that of shape denoising algorithm.

6. Performance evaluation

The performance of the proposed feature spaced based algorithm, 2-pass fast detection algorithm and combined algorithm are compared with two recently published paper (Jiang's [30] in STARE and Staal's [32] in DRIVE). In Jiang's algorithm, 8 sets of parameters are used to generate results with different tradeoff between sensitivity and selectivity. From parameter set K0 to K7, the sensitivity of the algorithm increases and the selectivity drops. See Table 5, The K-Mean classification algorithm achieves TPR/FPR of 59.31%/3.42% based on Hoover's measure, and 62.75%/0.37% with central line based measure. The shape denoising algorithm gets 74.48%/3.58% with Hoover's measure, and 57.24%/0.45% with central line based measure. The 2-pass fast algorithms gains about the same performance as Jiang's with parameter set K1 or K2 (65.3%/2.4%), based on Hoover's performance measure. But with central line based measure, Jiang's algorithm only achieved detection performance of 40.19%/0.015 or 45.34%/0.032, which is 30.6%~38.5% lower than our algorithm (67.5%/0.2%). To match the performance of our algorithm in central based measure, Jiang's algorithm must use the most sensitive parameter set K7, which is still lower than ours with a higher false positive rate. In that case, with Hoover's measure, Jiang's TPR is 27% higher than ours, but its false positive rate is 6.2 times as ours. The combined algorithm achieves higher TPR than 2-pass fast algorithm with both Hoover's (19.85% higher) and central line based performance measure (8.3% higher), and at the same time, a bit higher FPR (4.4% versus 2.4%). Figure 27 shows the performance curve of Jiang's algorithm and the performance of our algorithms. One can see clearly that our algorithms have much better performance with central line measure and similar performance as Jiang’s in Hoover’s measure.

Table 5 The performance comparison between Jiang's and ours in STARE.

Algorithm	Detection performance	Hoover measure (TPR/FPR)%	Central line measure (TPR/FPR)%	Time Cost (sec)
Ours	K-Mean	59.31/3.42	62.75/0.37	20
	Shape denosing	74.48/3.58	57.24/0.45	35
	2-pass fast algorithm	65.3/2.4	67.5/0.2	15
	Combined algorithm	78.19/4.4	73.10/0.38	37
Jiang's	(K0)	44.13/1.15	31.46/0.004	8~36
	(K1)	60.08/1.97	40.19/0.015	
	(K2)	70.72/2.79	45.34/0.032	
	(K3)	80.69/5.19	54.09/0.13	
	(K4)	83.84/6.37	56.31/0.18	
	(K5)	86.04/8.13	58.73/0.26	
	(K6)	87.81/10.73	61.53/0.39	
	(K7)	89.63/14.82	64.83/0.61	

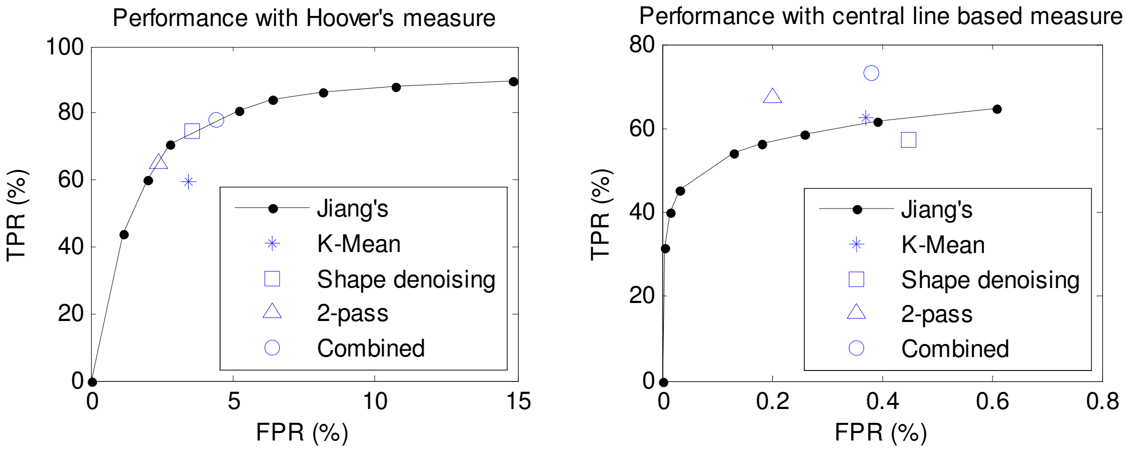


Figure 27 The performance comparison between Jiang’s and our algorithms.

DRIVE is another widely used benchmark image database in retinal image processing research. It contains 20 images. For each image, Staal’s algorithm generates a probability map, in which, a pixel with higher intensity has higher probability to be on the vessel. To measure its performance, several thresholds are applied on the probability map (21-121, with interval 20). See Table 6, The K-Mean classification algorithm achieves TPR/FPR of 59.41%/2.67% based on Hoover's measure, and 76.61%/0.83% with central line based measure. Shape denoising algorithm gets 64.26%/2.06% and 72.58%/0.23%. 2-pass fast algorithm gains about the same performance (69.37%/5.05% and 82.35%/0.50%) as combine algorithm (71.24%/3.13% and 82.50%/0.40%) on DRIVE. See Figure 28, with Hoover’s measure, our algorithms has a lower performance, but with central line based measure, the 2-pass and combined algorithm both gain higher performance. To achieve the same performance with central line based measure as ours, the threshold on Staal’s result map should be set to be 41, at which level; our algorithms gain a lower TPR than Staal’s in Hoover’s measure. The difference is the boundary pixels on the vessel edges. Staal's results have a much thicker/over-detected large vessel boundaries, this makes the TPR higher but also causes a higher FPR. This fact shows that Hoover's measure can't fully evaluate the algorithm performance on vessel structure. On the other hand, Staal's algorithm is time-consuming. It needs more than 15 min to process one image, while the combined algorithm can process one image in 30 seconds and the 2-pass fast algorithm uses only less than 10 seconds to process one image. By utilizing the parallel code library introduced in next section, the time cost of our algorithms can be further reduced linearly with the increase of the number of processors. For example, in our test bed with 2x quad-core system, the two algorithms can process one image in 3.9 second and 1.1 second.

There's tradeoff between speed and accuracy when determining which algorithm to use in practice. For applications that need fast vessel structure detection, e.g., registration, fractal analysis, etc., 2-pass fast algorithm can generate result with relatively high accuracy with low time cost; and for those needs more accurate vessel width detection, such as vessel dilation alert, the combined algorithm can generate result with higher accuracy but the time cost is higher too.

Table 6 The performance comparison between Staal's and ours in DRIVE.

Detection performance Algorithm		Hoover measure (TPR/FPR)%	Central line measure (TPR/FPR)%	Time Cost (sec)
Ours	K-Mean	59.41/2.67	76.61/0.83	16.5
	Shape denoising	64.26/2.06	72.58/0.23	27
	2-pass fast algorithm	69.37/5.05	82.35/0.50	13
	Combined algorithm	71.24/3.13	82.50/0.58	28
Staal's (Different Thresholds)	121	69.65/1.94	71.84/0.13	15 min
	101	73.50/2.57	74.31/0.18	
	81	77.53/3.49	76.82/0.27	
	61	81.42/4.80	79.56/0.41	
	41	85.40/6.82	82.99/0.62	
	21	90.06/11.08	89.19/1.11	

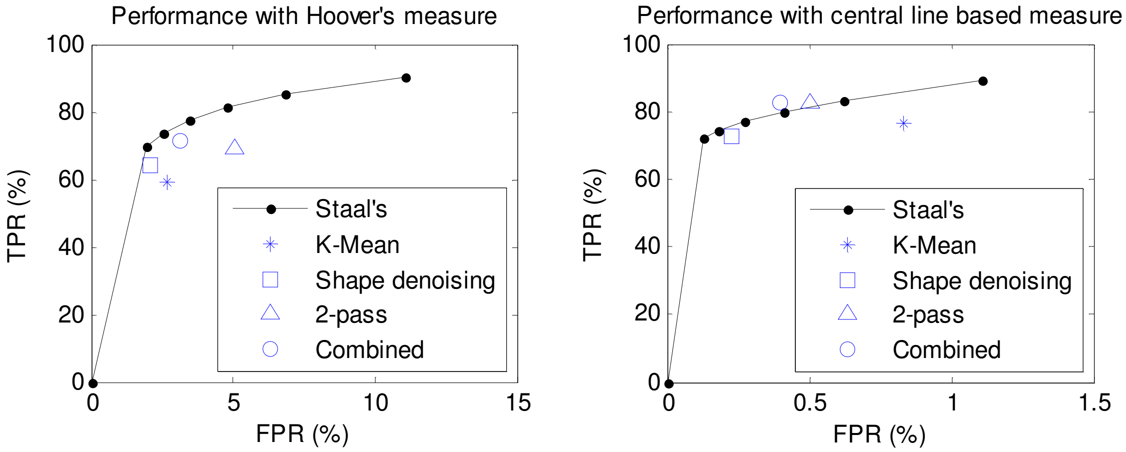
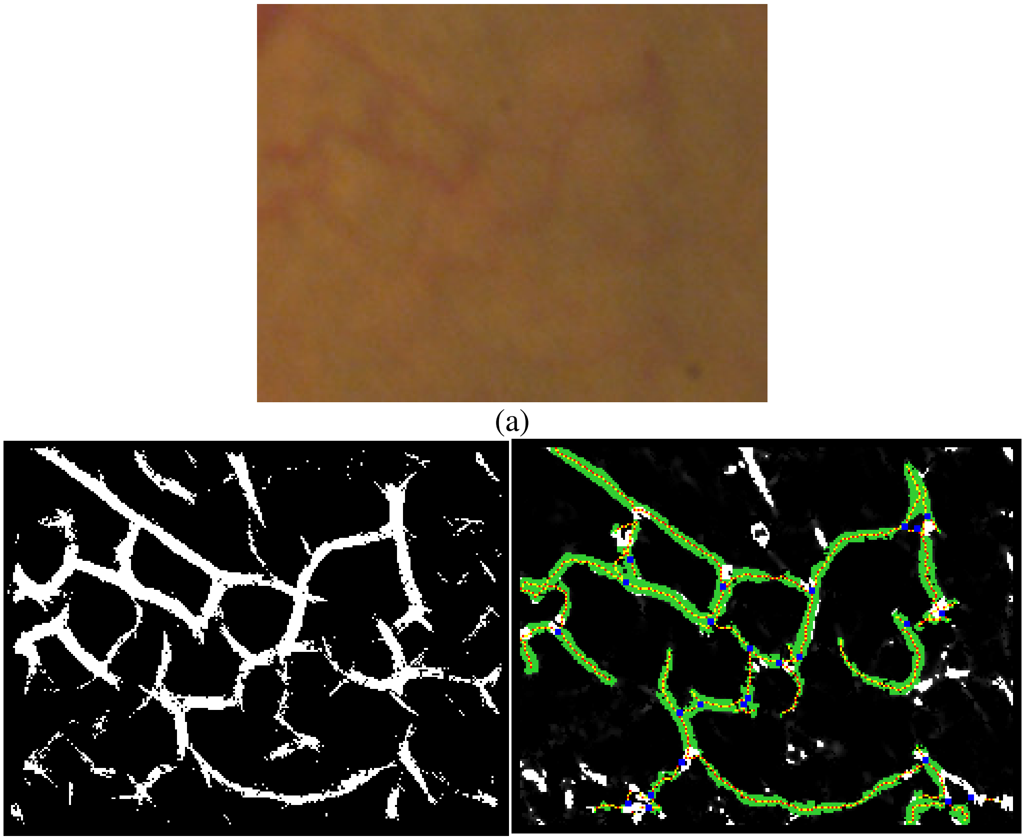


Figure 28 The performance comparison between Staal's and our algorithms.

7. Blood vessel detection in high resolution images

The fast detection algorithm can also be used in vessel detection, especially the small/micro-vessel detection, in high resolution images. The small/micro-vessels are very shallow in the high resolution images. After downsizing, a lot of details will be smeared. With high resolution images, more detail can be revealed, and the screening system can have more information/evidence to make decision on referral of the patient to doctors for further examination. In the previous section, we introduced a Gaussian tracing and Gabor-variance based algorithm to detect small vessels in high resolution images. The result is promising but the time cost is high. We applied a 1 pass fast algorithm to detect small vessels in high resolution images. The neighborhood size r is set to 15 to capture most arterioles and venules. And the sensitivity controlling parameter C_{min} is set to be -0.01 to capture more details since in the high resolution images, the small vessels are pretty shallow; minimal segment length L is set large to 20 to avoid too much false detections.

The outcome of the fast detection algorithm is shown in Figure 29. Compared with the result of tracing algorithm, this algorithm captures more detail and more importantly, it runs much faster. It can process an image in about 15 second (this number is a bit larger than that in 2-pass fast algorithm because of the larger minimal segment to be probed), while the tracing algorithm needs about 40.



(a)

(c)

(d)

Figure 29 (a) A sample cropped from full resolution image (b) The detection outcome of fast detection algorithm. (c) The result of the previous tracing algorithm.

The detection performance is show in Figure 30. The algorithm gains 87.27% TPR and 0.20% FPR, which detects more vessels with slightly FRP increase, compared with the tracing algorithm in the previous section.

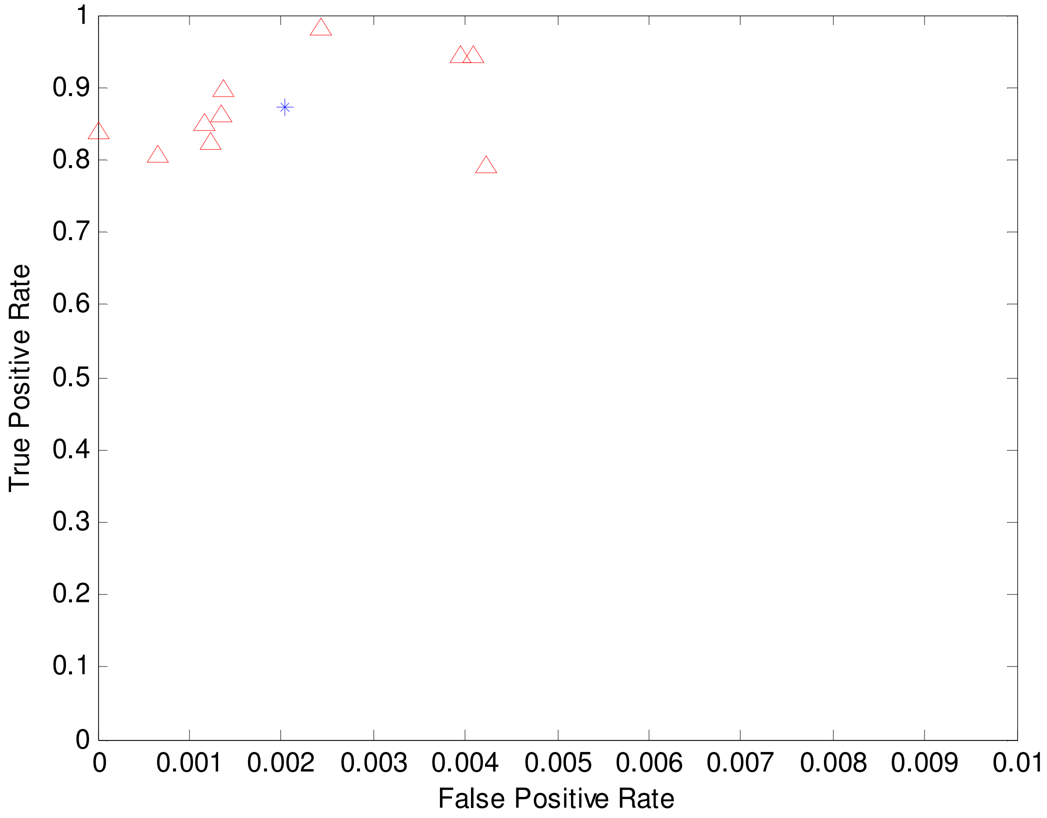


Figure 30 Performance of 1-pass fast detection algorithm on high resolution images.

8. Detection of other objects in RC domain

In real clinical practice, such as the project of *Early Diabetic Retinopathy Screening System* (EDRSS) we developed in the past two years, not only blood vessel detection is necessary, existence and counts of some lesions are important indicators of disease. In this section, the detection of some common lesions that serves for large scale screening in EDRSS is introduced.

8.1 Microaneurysm detection

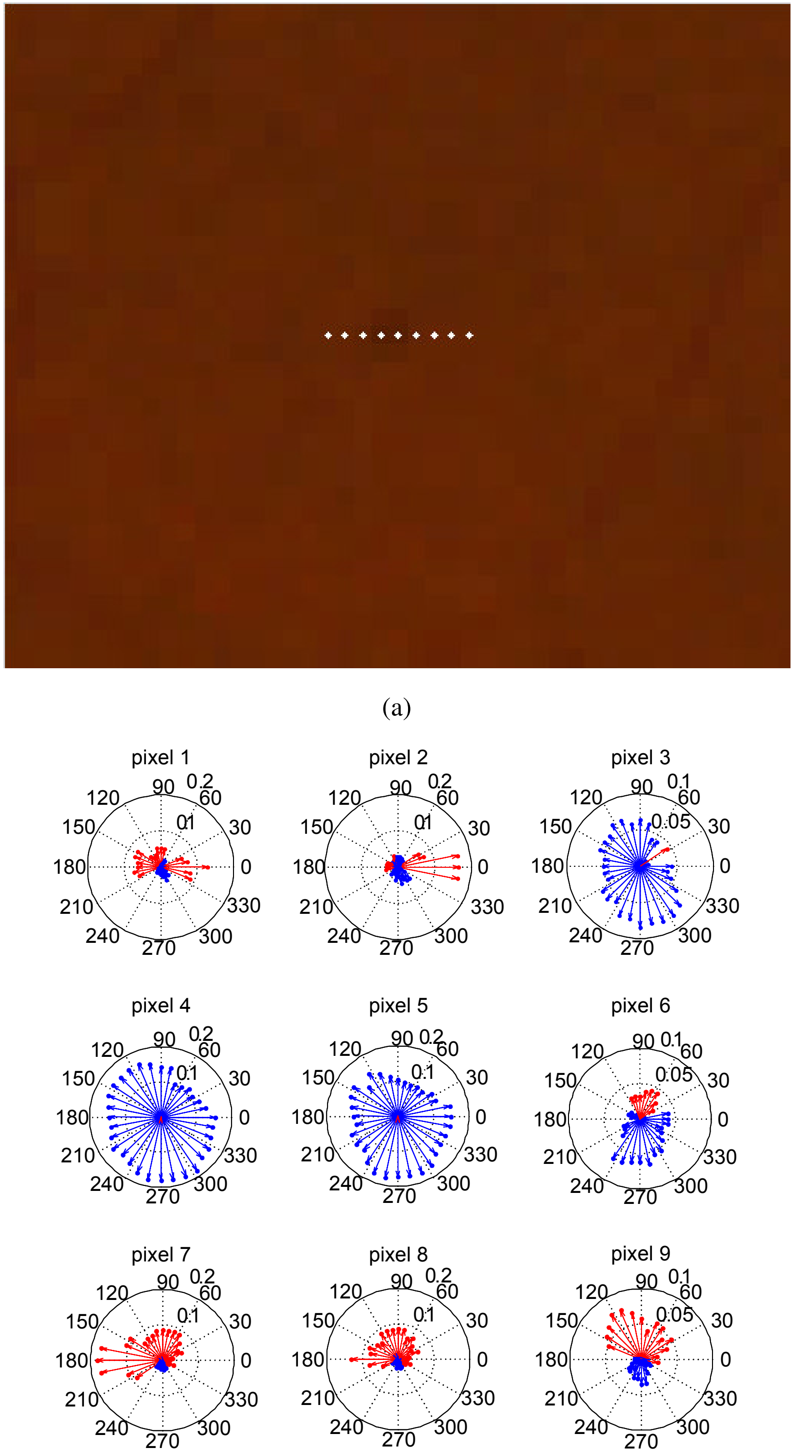
Microaneurysm (MA), known as an abnormal blood-filled dilatation of a retina blood vessel, is one of the earliest visible signs in diabetic retinopathy. MA is caused by the fluid accumulation and pressure in the capillaries [56]. They are characterized as dark red circular spots with diameter of 10-100 m, which are 1-3 pixels in STARE and EDRSS. Blood leakage may occur because of the weakness of capillary wall. Largest microaneurysms are found to be no larger than 1/12 of the optic disc diameter. Microaneurysms are dynamic. They appear and disappear throughout the disease and are usually located around the disc and macular areas of the retina [56].

The total number of MAs has been shown to be closely correlated with the severity of a variety of diseases [57]. Identifying and counting of MAs in retina fundus images manually is very time consuming and even well trained operators will make considerable errors. Computer-aided solutions are faster, more objective and indefatigable. These excellent properties make computer-aided solution greatly helpful to (early stage) disease diagnostic and treatment.

Cree *et al* developed a retina image MA detection solution in 1997 [58]. His approach is specifically tuned for fluorescent angiograms, which is an invasive technique and not ideal for mass screening. Hipwell et al further developed Cree’s methods to detect

MAs in digital red-free fundus photographs [59]. They used bilinear top-hat transform and the process includes shade correction, vessel and hemorrhage removal, and candidate MA detection. Hafez and Azeem presented a MA detection scheme based on a circular Hough transform and an adaptive edge technique without vessel removal in fluorescein angiograms [60]. However, it's known that geometric transform based approaches are mostly used to recognize lines or curves (circles for example) in binary images that have been edge detected, with a Robert, Sobel or Canny edge detector, for instance.

Microaneurysms are characterized as dark, round shape spots in the green channel retinal images. In a small neighborhood size on which the radial contrast values are calculated, the pixels on MA should have negatively sharper contrast than $C_{\min}$. See Figure 31 for nine pixels that run across a microaneurysm and the contrast radial plot for these nine pixels. One can see clearly that microaneurysm pixels (pixel 4 and 5) have negatively sharper contrast than $C_{\min}$ along all directions. Thus, we can create a feature descriptor $D = \text{Count}(\theta \mid C_p^\theta \le T_{MA})$, where T_{MA} is set to $-C_{\min}$ to gain the highest sensitivity. A simple threshold $k = N$, where N is the number of probed directions, can be set to identify all the MA(s). Consider that the relatively small size of microaneurysms (1-3 pixels), the neighborhood size *r* is set to five to improve the computational efficiency. A sample image that contains two microaneurysms and the detection result is shown in Figure 32. The image is cropped for better visualization of microaneurysm.



(a)

(b)

Figure 31 (a) Nine pixels that run across a microaneurysm. (b) The plot of radial contrasts for these nine pixels.

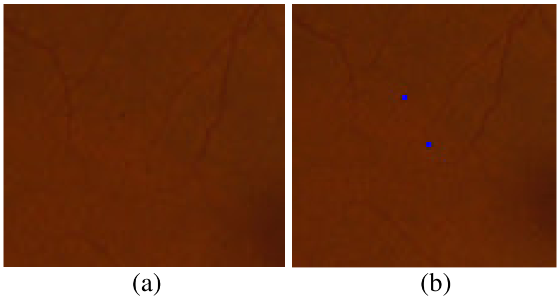


Figure 32 (a) An image containing 2 microaneurysm. (b) The detection outcome of microaneurysms on the image.

There may be some false alarm on the blood vessels, with the BV map obtained in previous sections, they can be easily removed.

8.2 Hard exudates detection

Hard exudates are the deposits of plasma protein left behind as the rest of edema fluid is absorbed. They will be finally removed by macrophage cells. The exudates have a bright yellow and waxy appearance and sharp edges against the background. And they may form circular shape around the vessels [56, 61, 62]. The relationship between exudates and edema can be used to distinguish them from drusen, a symptom of age-related macular degeneration, which is not diabetic retinopathy. Researcher has found that exudates cannot be identified efficiently based only on shape, texture and size [63], and domain knowledge must be incorporated.

Wang *et al* [64] proposed a scheme that extract color features from the RGB space and use statistics classifier derived from Bayes' rule to detect lesions in the retina image, which is only effective to when the whole image is under the evenly distributed illumination. Luo *et al* [65] introduced a blockwise object detection algorithm based on watershed technique.

False alarms are removed by comparing the average intensity inside and outside the contour. In [61] Hsu *et al* incorporate the domain knowledge to identify the hard exudates. A dynamic clustering algorithm is used to detect hard exudates on the difference map created by subtracting a smoothed image from the original image. Finally, domain knowledge is employed to differentiate hard exudates from drusen and cotton wool. Walter *et al* [66] proposed a method of optic disc and exudates detection. They used morphologic operator to find the position of optic disc and then determine the boundary with watershed technology.

Hard exudates are bright yellow color spots in the retinal images. Because of its irregular shape, it's hard to define some specific shape descriptors to identify them effectively. In another point of view, within large enough neighborhood size, they can be considered as big bright spot, thus their radial contrast along all probed directions should be *positively* sharper than Cmin. $D = \text{Count}(\theta \mid C_\theta \geq T_{HE})$, where $T_{HE} = C_{\min}$, with threshold $k = \alpha N$. The error tolerance parameter α is used to relax the threshold. Because hard exudates have an irregular shape, if along some direction most of a HE pixel's neighborhood falls into the HE itself or along some directions, the neighborhood pixels have higher intensity; the radial contrast of that HE pixel would be smaller than the given threshold or even negative. See Figure 33 for 25 pixels that run through a block of hard exudates, and Figure 34 for their contrast plots. Pixels 6-18 are those on the exudates. One can see that these pixels have positive contrasts along almost all directions. Most of them have one negative contrast pointing to the center of exudates where the intensity values are the highest. α is set to be 0.8 in this work. The sensitivity of the detector can be adjusted by changing the value of T_{HE}. Note that HE(s) are usually much brighter than the background. T_{HE} can be set to a larger value to remove possible false alarms, in this work, $T_{HE} = 0.05$. Image im0001 in STARE database and the detection outcome of hard exudates on it are shown in Figure 35. The neighborhood size r for HE detection should be large enough to cover most HE(s). In this work, *r*=60 for HE detection.

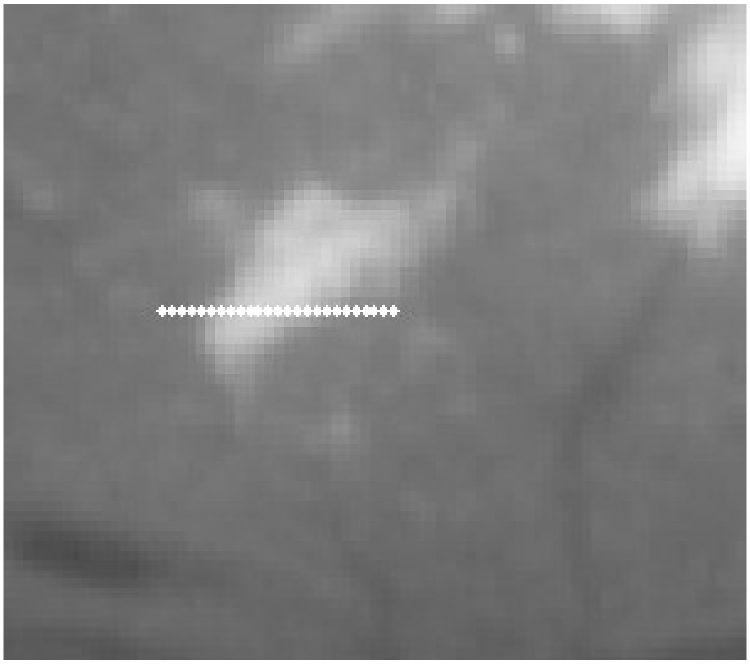


Figure 33 25 pixels running across hard exudates.

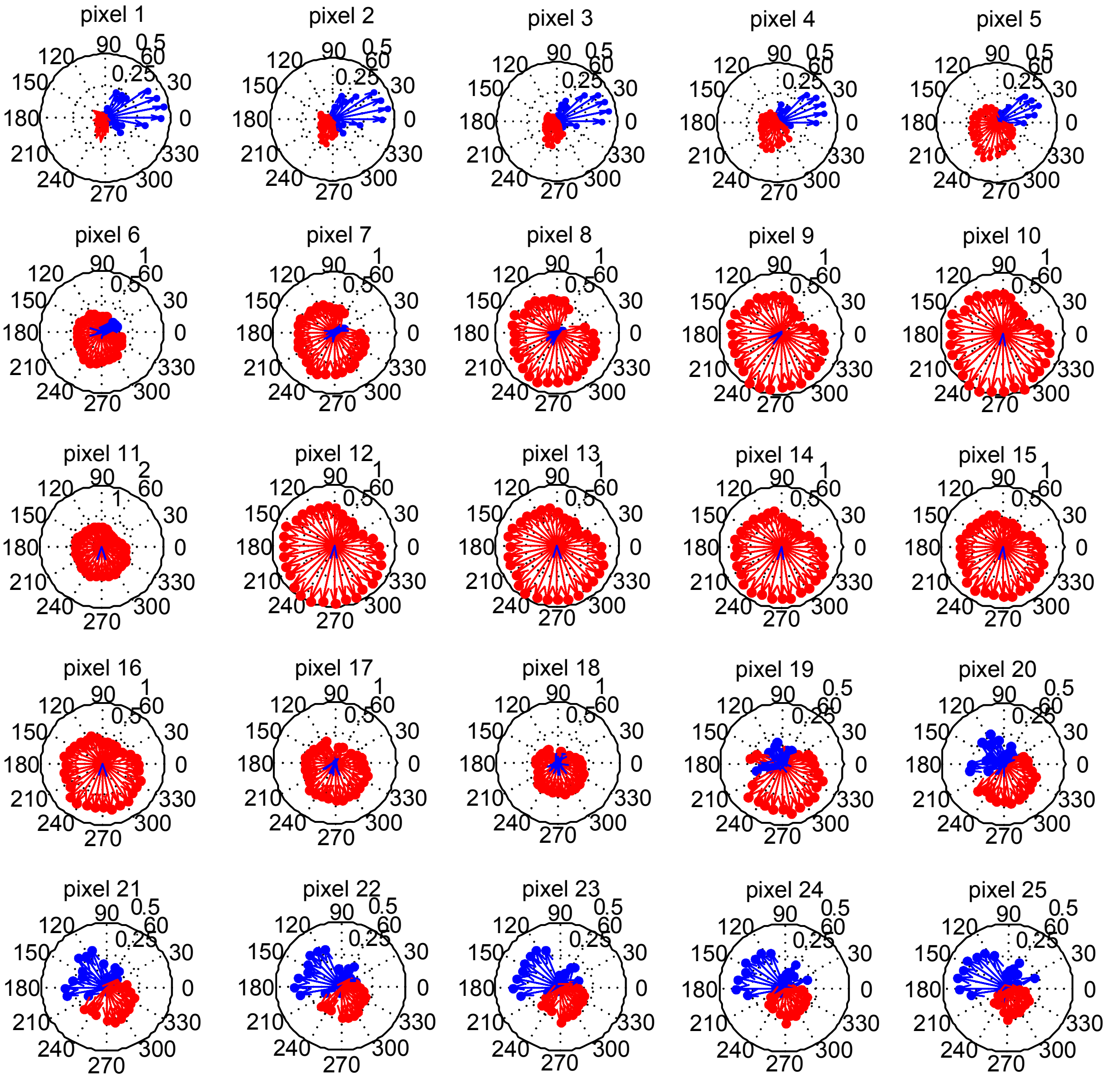


Figure 34 Plots of radial contrast for the 25 pixels.

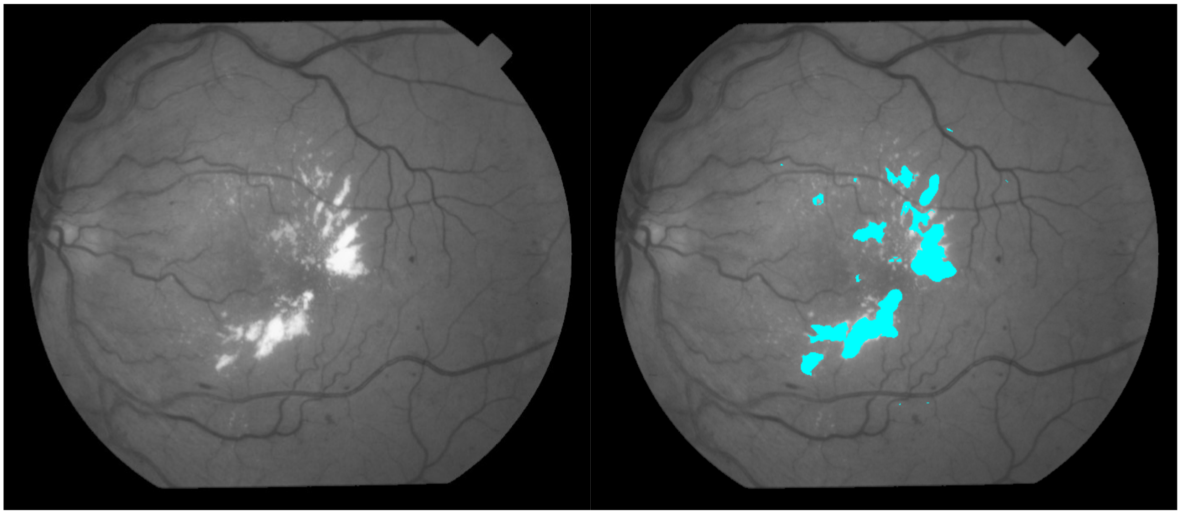


Figure 35 (a) im0001 in STARE (b) Detected hard exudates.

8.3 Hemorrhage

Hemorrhages are the result of blood leaking from the damaged capillaries. There are three categories of hemorrhages: dot, flame or blot. Dot hemorrhages are dark red spots and usually not distinguishable from microaneurysms, but usually with a larger size. Flame hemorrhages are distinguished by their flame like shape due to the blood leakage in alignment with the nerve fibers. Blot hemorrhages usually have irregular shape and larger than blot hemorrhages (See Figure 36 for an illustration).

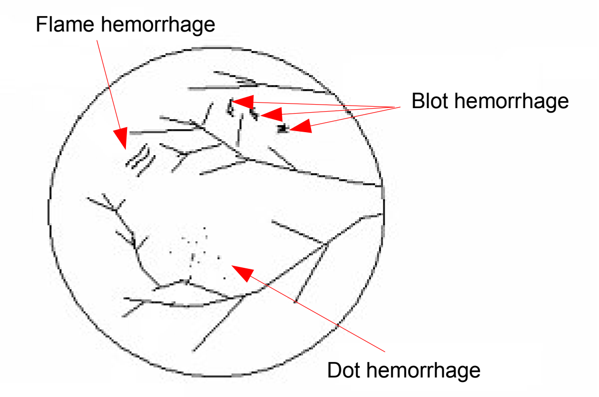


Figure 36 An illustration for different types of hemorrhages.

There is little research work specifically on hemorrhage detection. Most of the research work on microaneurysm detection can be applied directly on hemorrhage due to their overlapped features. Flame and blot hemorrhages are hard to detect due to their irregular shape. Larsen *et al* [67] define the hemorrhage as “round or irregularly shaped, sharply or diffusely outlined deep red areas of the color of venous blood or darker”, and microaneurysm as “small round lesions with a well defined edge and brighter rim”. In their paper, the description of detection algorithm is very rough and contains no detail. The algorithm picks up seed points of candidate with local image properties and then grows them. Optic disc and vessels are removed prior to the processing. They don’t distinguish hemorrhages from microaneurysms and the performance is measured based on both lesions.

The feature descriptor used for large vessels can be used directly on the flame/blot hemorrhage detection, but with a larger neighborhood size. A line of pixels that run across a blot hemorrhage is show in Figure 37, and their radial contrast plot is drawn in Figure 38. One can see hemorrhage pixels have radial contrasts that are negatively sharper than C_{min}, along almost all directions. It can be seen in Figure 39 that some vessels are also detected, because they have the same characteristics as hemorrhages. It's hard to distinguish large blood vessels from the flame or blot hemorrhages. In current protocol in EDRSS project, existence of dot hemorrhage is used as an indicator of diabetic retinopathy. Blot/flame hemorrhage are not used.

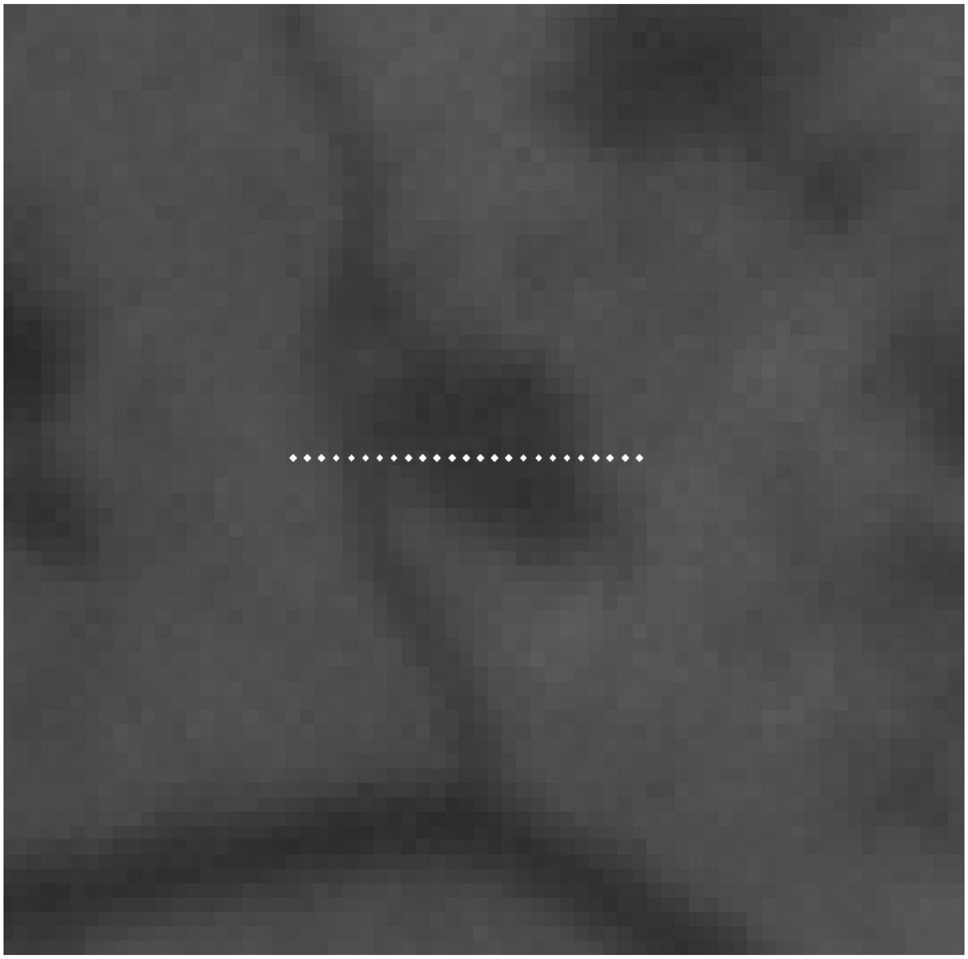


Figure 37 25 pixels running across a blot hemorrhage.

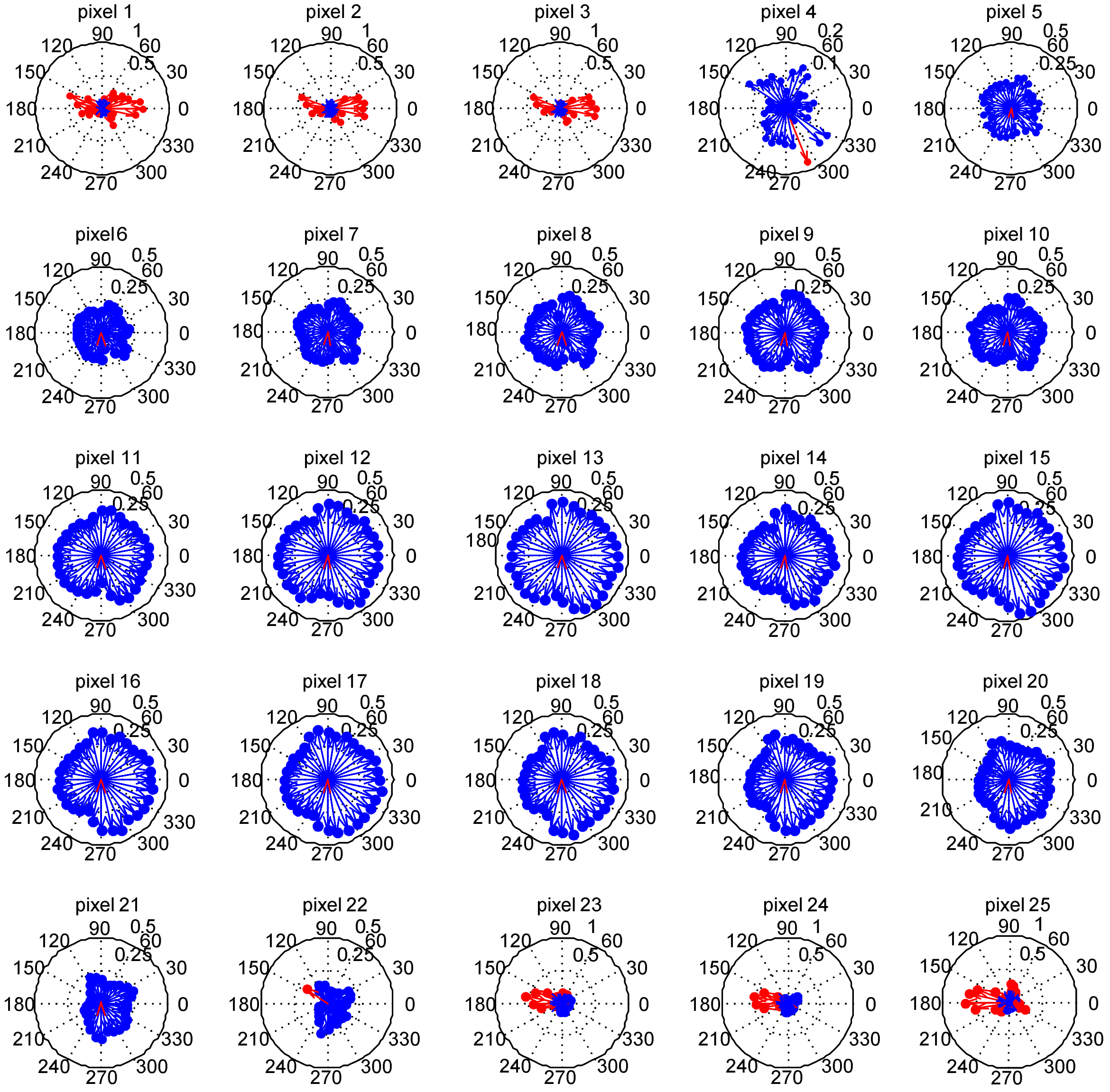


Figure 38 Plots of radial contrast for the 25 pixels.

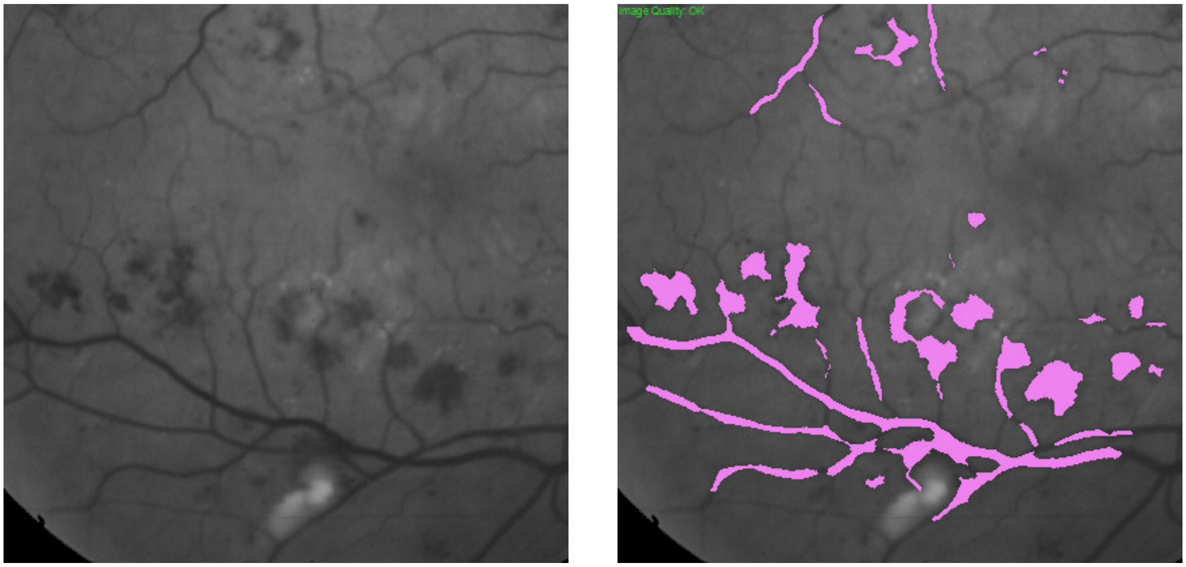


Figure 39 A sample image of hemorrhage and the detection outcome.

III. EARLY DIABETIC RETINOPATHY DETECTION FIELD TEST

A. INTRODUCTION

Diabetes can cause implications in the retinal system, such as microaneurysm, hemorrhage, edema, hard exudates, macular degeneration, glaucoma, etc. These diabetes related eye implications are called diabetic retinopathy (DR). It has become one of the major reasons of blindness in developed countries such as United States and Canada. Early detection and treatment can dramatically reduce the risk of blindness. Images are enhanced, denoised, and different objects (e.g., blood vessels, different kinds of lesions) are extracted from the image. Based on some predefined criteria on the detected features, a screening result is given to decide whether the patient needs to be referred to an eye specialist for further exam. In Appendix B, the system design of EDRSS is introduced, and different parts of the system are described in detail. In EDRSS, the most important part is the batch processing servers. Its performance, either on accuracy or time efficiency, would affect that of the whole system significantly. A code library, which can take advantage of multiple core architecture to parallel the image processing tasks and improve the computational efficiency, is introduced in Appendix C. The radial directional contrast based algorithms are used to detect vessel map and other lesions that are used to make the screening decision.

B. SCREENING STRATEGY AND PERFORMANCE EVALUATION

The screening system has a set of predefined rules to screen out the patients that needs further exam by the doctor. These rules are generated from the doctors' reading of the images taken from the field clinics. Currently, the major indicators defined in the protocol of abnormal eyes are the existence of hard exudates and dot hemorrhage, and high number of microaneurysms. In this project, to gain high throughput of the system, we choose to use the 2-pass fast algorithm to detect blood vessels.

Before the system runs any tests on the images, the quality of the images is checked. There are three common problems on the low quality images: underexposure, overexposure and unbalanced-exposure, see Figure 40 for examples. In the three cases, underexposure is the worst one, since it provides least information. Overexposure is a better case, some information may be saturated. Unbalanced exposure doesn't affect the system performance much since the radial contrast based algorithm is not sensitive to the gradual intensity change. But in this case, if the dark area has too low intensity, it will be treated as underexposure. A simple algorithm is designed to identify the three kinds of images. First, intensity histogram in green channel is drawn, if the portion of pixels with intensity lower than 15 (left tail) or larger than 235 (right tail) is larger than 2% of the total number of pixels in the field of view, the image is then marked as low quality. The field of view is fixed for specific camera so the non-FOV area can be removed with a pre-constructed mask. The result of the list images in Figure 40 are shown in Table 7.

Table 7 The image quality decision procedure. Red numbers are those exceed the threshold (2%).

Case	Normal	Underexposure	Overexposure	Unbalanced-exposure
Left tail (%)	0.96	30.82	0.3	5.75
Right tail (%)	0	0	4.18	0.08
Decision	Good	Low quality	Low quality	Low quality

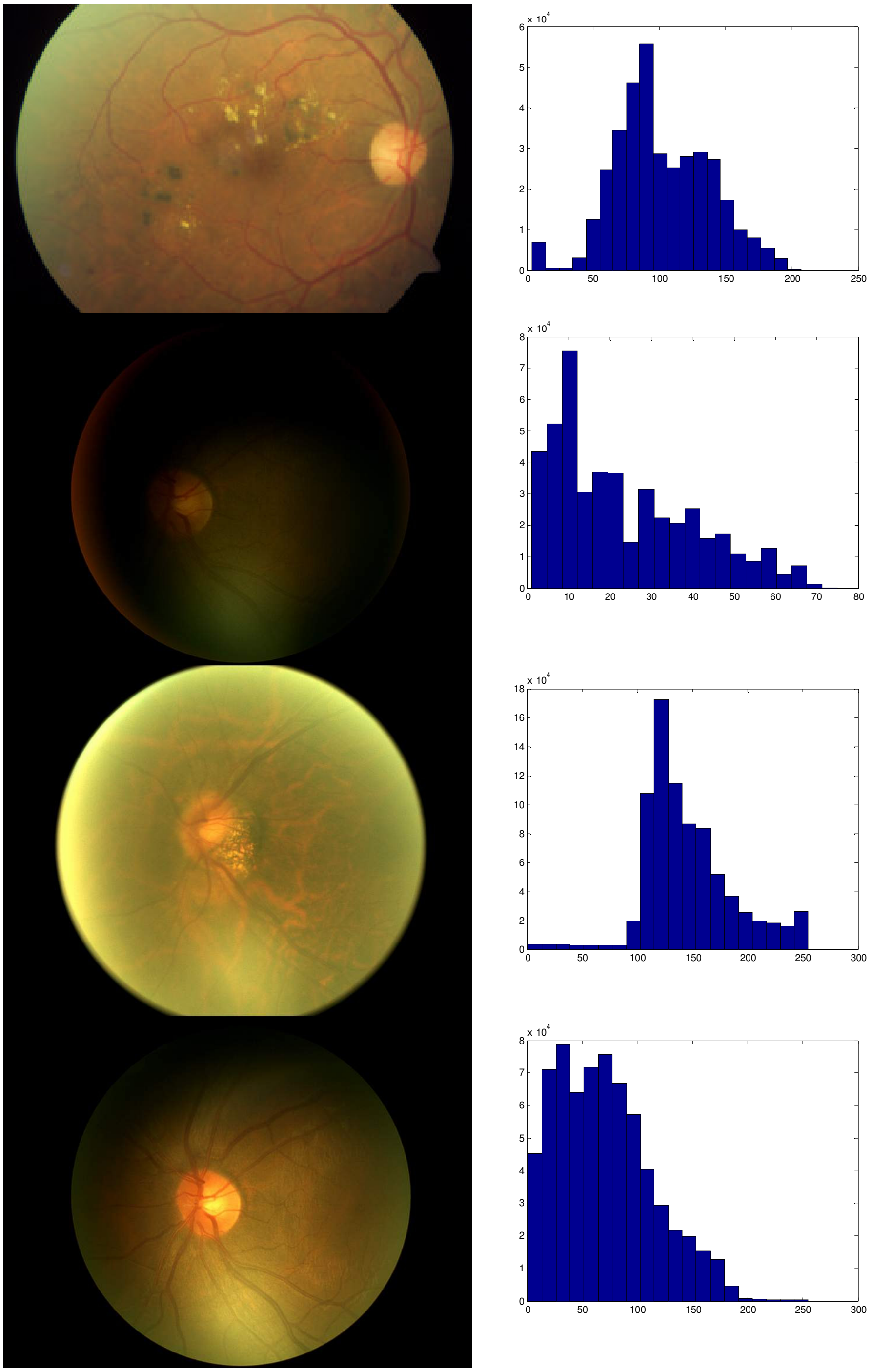


Figure 40 Common low quality images. Left column, from top to bottom: underexposure, overexposure, unbalanced exposure; right column, from top to bottom: the histogram of green channel for the images.

Some times the underexposed images may have information that can be used for screening purposes. A simple curve adjust on the intensity histogram can improve the image quality significantly. The curve adjust actually stretches the histogram and change most of the low intensity pixels to a higher level. In this project, the curve is interpolated with three points (0, 0), (I_{in}, I_{out}), (255, 255), where I_{in} is the intensity value under which intensity of 98% of all pixels are, and I_{out} is a fixed number 192, which is three quantiles of the intensity range (255*3/4 = 192). An example can be found in Figure 41. Note that the contrast enhancement procedure not only enhanced the vessels and lesions to be detected, but also the background noise, see Figure 41 for a microaneurysm that is invisible in the original image.

The 2-pass fast blood vessel detection algorithm is very sensitive to detect vessels. It can actually detect the vessel map without the enhancement procedure described above. Figure 42 (second row) shows the detection outcome of the 2-pass BV detection algorithm on the original and the enhanced underexposed image. One can see that the results are almost the same, and the outcome from enhanced image has a bit less false detections. The large amount of false detections is caused by the low signal/noise ratio in the underexposed images. After the enhancement, the noise is also enhanced. In both cases, the noise interferes with the contrast calculation and causes the false alarm. To solve this problem, we utilize the 1-pass detection algorithm that was used in previous section to detect blood vessels in high resolution images. It uses a larger neighborhood size to capture major vessels with the loss of some fine small vessels. These small vessels are smeared by the noise. The parameters are set as follow: r=15, L=20, =0.8, N=32, and $T_{BV} = -0.02$. The detection result is shown in **Error! Reference source not found.** (third row). One can see that the false detection has been dramatically reduced. This is because the larger neighborhood size acts as a low pass filter in the contrast calculation, which reduced the effect from the noise. The enhancement again doesn’t help on improving the detection performance. The performance comparison of the radial contrast based algorithm between original and enhanced low quality images shows that our algorithm can work well without the help of image enhancement, which is usually time consuming and with the risk of over enhancement.

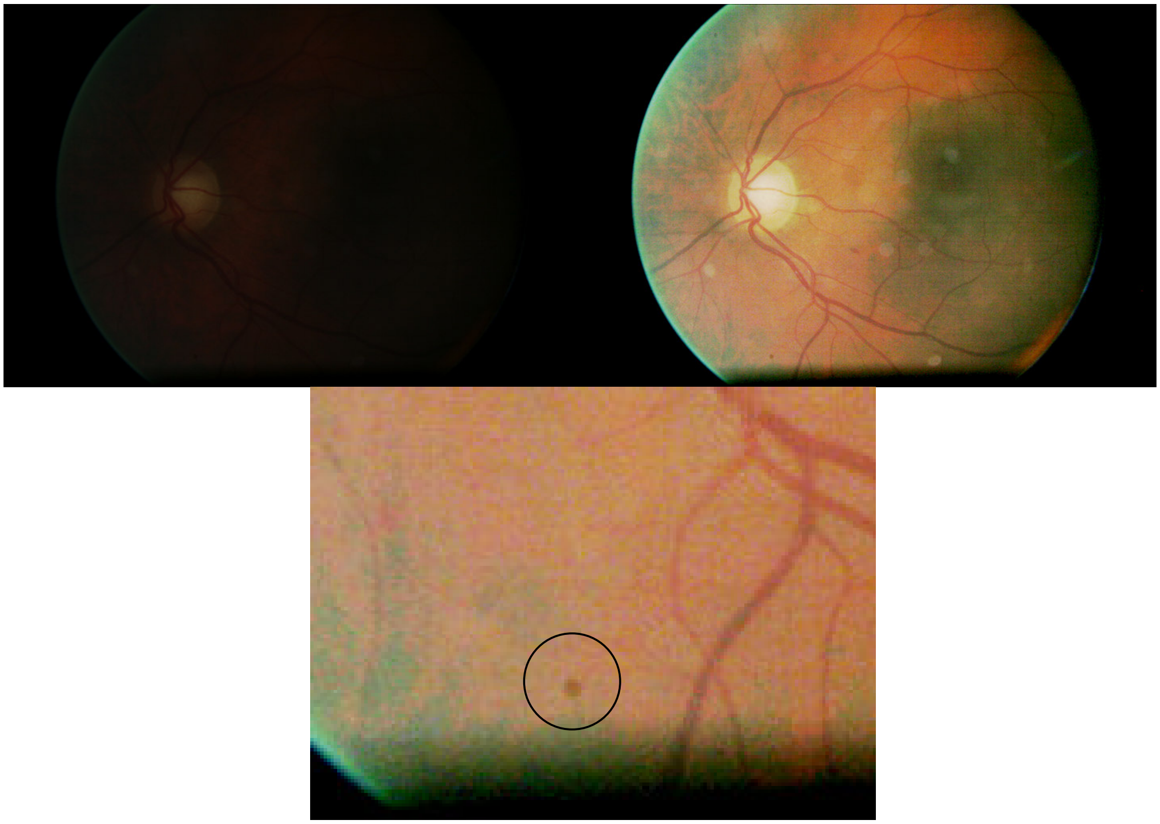


Figure 41 First row, an underexposed image (left) and the one after simple curve adjustment (right); second row, a crop of enhanced image, showing the enhanced noise.

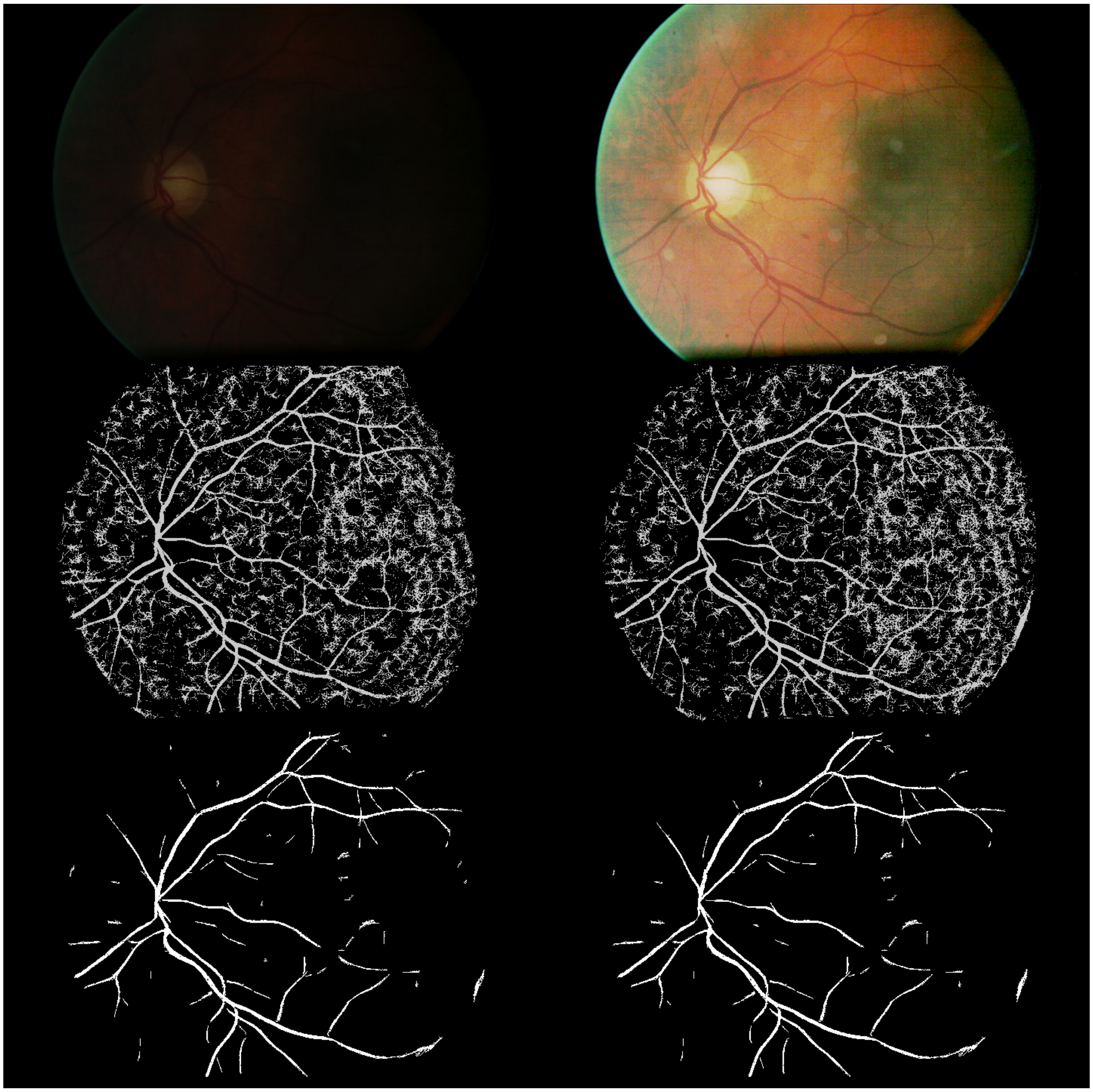


Figure 42 First row, an underexposed image and the enhanced image. Second row, the detection outcome with 2-pass fast algorithm. Third row, the detection outcome of 1-pass detection algorithm.

The image enhancement doesn’t help on the computer side, but it’s very important for the human reader. Human readers don’t have the ability to enhance the images with their eyes. The enhanced images can help them read the image more effectively and accurately.

It must be mentioned that in the high noise environment, the detection of microaneurysm by computer or even human reader may become unreliable because of their small size. There's a high possibility that some noise on the background has the similar characteristics as microaneurysm. The detection outcome of low quality images are for reference purpose only. In current stage of the project, low quality images that are unreadable to human readers will be excluded in performance test.

Once the image passed the quality check, the system first run the detection on hard exudates, which are bright spots in the image, with the algorithm described in the previous section. If any hard exudates are found, the patient is put in the list of "*Highly abnormal*". If the first test is passed, the system runs a test of dot hemorrhages, which have the similar characteristics as microaneurysms, but with larger size. If any dot hemorrhage is found, the patient is marked as "Highly abnormal". Otherwise, the system runs the detection algorithm of microaneurysm, find all possible candidates. The blood vessel map of the whole image is extracted. Microaneurysm candidates that fall on the vessels will be removed and the rest candidates are considered as the "real" ones. The number of them is counted. In the field test, it is noticed that a batch of images may contain some microaneurysm like objects that are located at exactly the same position in most of the images. These are actually the dust on the lens. Figure 43 shows two such images. To removes these false detections, the detection outcome of all images are put together to make a comparison. If a detected microaneurysm candidate appears more than 1/10 of all the images (or at least 3 different images) at the exactly the same location, it is considered as a dust and removed from all the images.

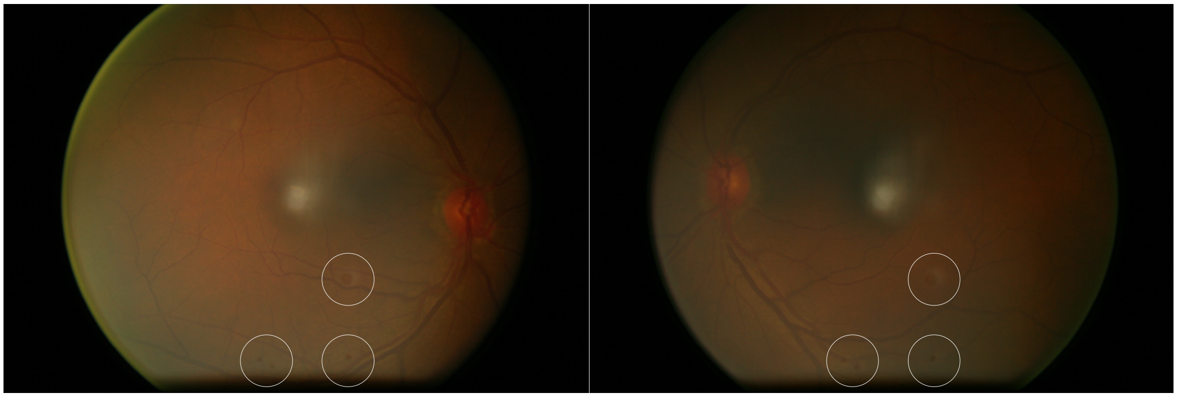


Figure 43 Two images that have microaneurysms/dot hemorrhages at exactly the same locations.

Different sensitivity of the system on microaneurysm counting is used to give more information that may help the system make the decision for the patient. The sensitivity is controlled by the parameter T_{MA}. In current protocol, if in a low sensitivity test ($T_{MA} = 0.08$), there is more than one microaneurysms found in the image, the patient is put in list of "Highly abnormal". In a medium sensitivity test ($T_{MA} = 0.05$), if the number of microaneurysms is larger than five, the patient is marked as "Warning". And if the number exceeds 10 in a high sensitivity test, the patient is marked as "Warning". Otherwise, the patient is marked as "OK". Usually, every patient will have more than one image taken. If any image is marked as warning or dangerous, the patient is marked as "DR" (Diabetic Retinopathy).

The images taken from the field were read by two doctors, and the computer also gives its recommendation for the patients. The results are compared. It's noticed that the even the two doctors (A and B) may have different opinions on the same image. For one image, the two doctors' reading may be:

Table 8 Different cases of two human readers' reading on the same image. (DR – diabetic retinopathy)

A's reading	B's reading	Category
DR	DR	Strong case
Non-DR	Non-DR	
DR	Unreadable	Weak case
Non-DR	Unreadable	
Unreadable	DR	
Unreadable	Non-DR	
Non-DR	DR	Disagreement case
DR	Non-DR	

The cases are divided into three cases as what Table 8 describes. The test data contains 385 images for 192 patients. Doctor A marks 34 of them as DR and 158 as Non-DR and Doctor B marks 37 of them as DR and 149 as Non-DR. The difference of the total number of patients is because of some unreadable images to Doctor B, which are excluded. Taking Doctor A's reading as the ground truth, EDRSS recognized 76% DR patients and screens out 77% Non-DR patients. Taking Doctor B's reading as ground truth, only 57% DR patients are recognized, and 74% non-DR patients are screened out. In all the DR patients, only 12 patients are marked by both doctors. EDRSS recognize 10 out of the 12 patients. In cases that the two doctors have different opinions (weak cases), 80% patients with DR are recognized and 67% patients without DR are screened out by EDRSS.

The low rate to screen out non-DR patients is due to several reasons. In some images, the dust and figure prints on the lens may cause bright artifacts, which look pretty similar to hard exudates. See Figure 44 for an example. In this case, these images will be marked as "DR" immediately. A method similar to dust removal in microaneurysm detection is applied to remove the artifacts. Detection outcome of images are compared, and those bright objects appeared at the same location for more than 1/10 of all images (or at least 3 images) are considered as artifacts and are removed and excluded in decision making procedure. This procedure helps remove some of the artifacts, but some artifacts may still remain in the result images and cause the computer make wrong decisions.

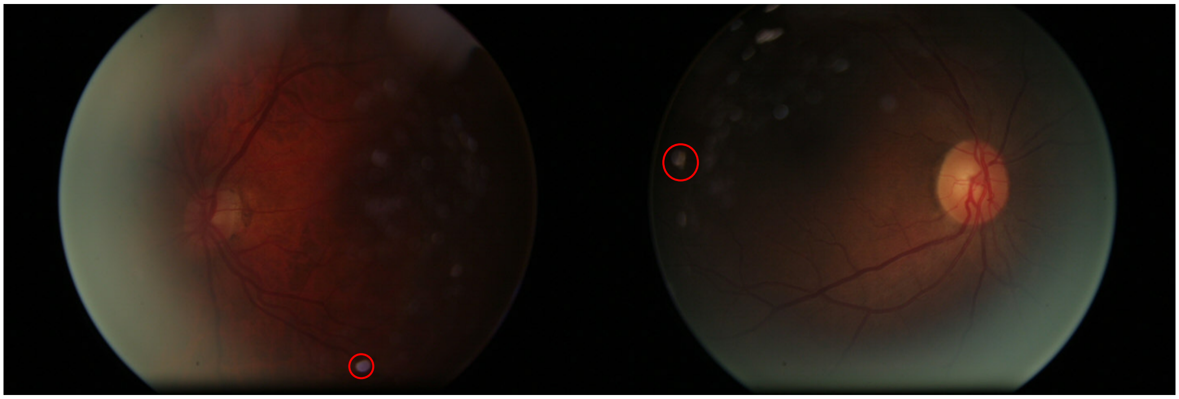


Figure 44 Artifacts caused by dust or finger prints.

See Figure 45, the circled dot hemorrhage in the first image and the spot hard exudates in the second one are is missed by both doctors but found by EDRSS. See Figure 46, the circled microaneurysm-like object near the bottom is in fact a dust on the camera lens, while the one on the upper part is a real microaneurysm.

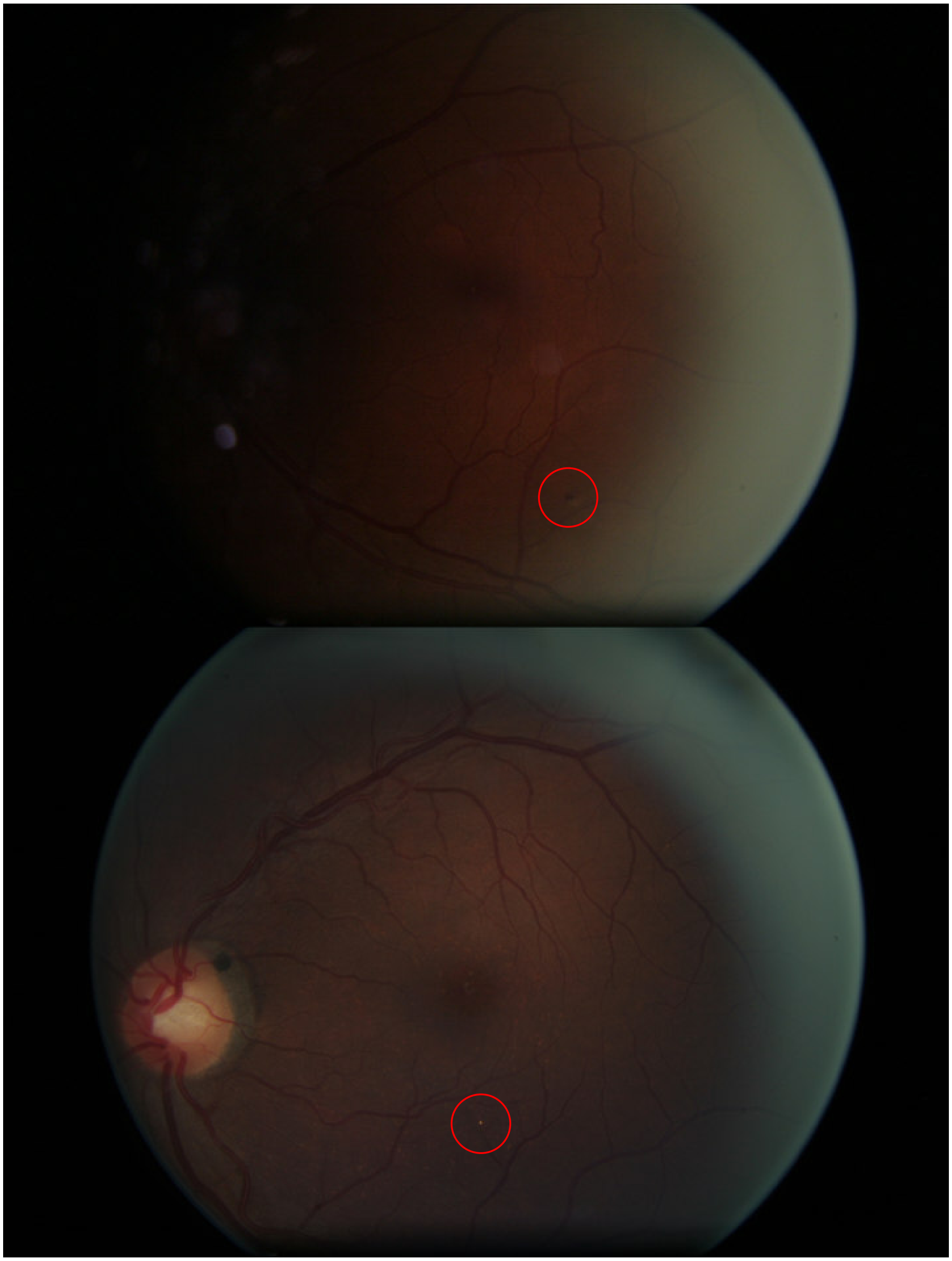


Figure 45 Upper image, a dot hemorrhage found by computer, but missed by both doctors; lower image, a spot hard exudates missed by two doctors.

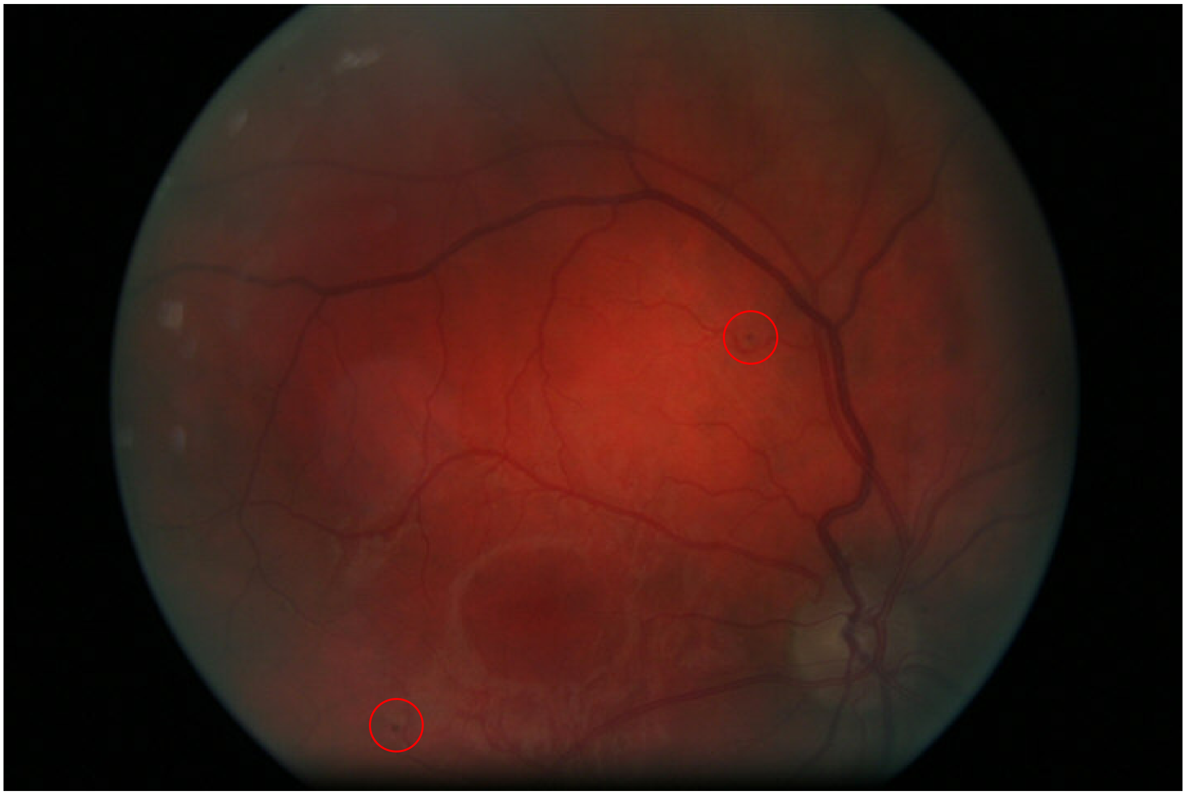


Figure 46 A microaneurysm-like object that is in fact a dust on the camera lens.

Some times the images are marked as "DR" because the doctors found some lesions that are beyond the ability of the system. Current system uses only three types of lesions to make the decision: hard exudates, dot hemorrhages and microaneurysms. Other types of lesion or anomalies are beyond the ability of the system. See Figure 47, the left image has some unknown object in the image. This image was marked by Doctor A as “non-DR” but “DR” by Doctor B. Our system has no knowledge of this object and marked it as “non-DR”. The right image is from a patient with age related macular degeneration, on which our system has no knowledge. This case is marked by both doctors as “DR”, but our system marked it as “Non-DR”.

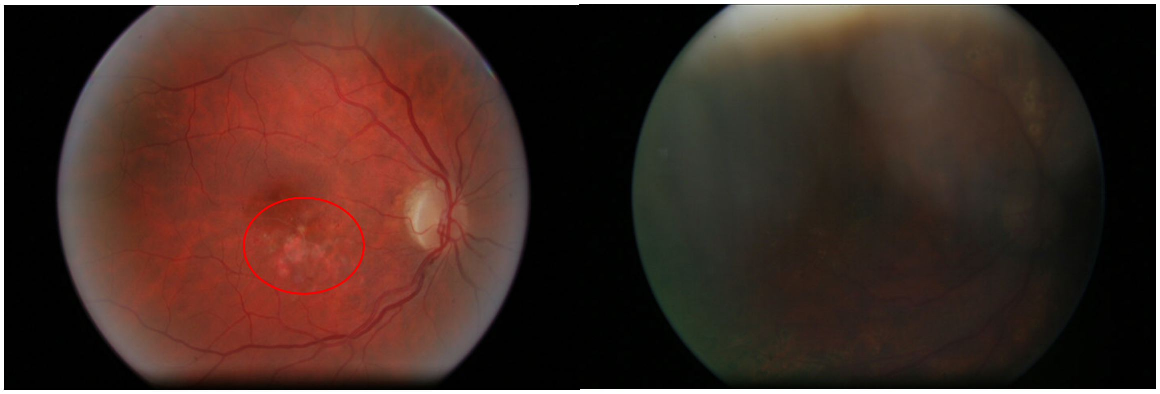


Figure 47 Left, an image with unknown object; right, an image with age related macular degeneration.

IV. CONCLUSION

In this dissertation, I have investigated the problem of retinal blood vessel detection. Automated blood vessel detection is a challenge problem because of the variety of illumination conditions, patient races, low contrast small vessel detection, and the tradeoff between accuracy and computation efficiency. The detection result of retinal blood vessel is the basis of many important medical applications. By analyzing the distribution of vessel location, width and tortuosity, some severe diseases can be predicted in early stage, which dramatically reduce the risk of serious impact to the patient's life quality. Also, blood vessel network is the only obvious structure that exists in all retinal images. It can be used as the landmark for retinal image registration and biometric authentication.

Two automatic vessel detection algorithms are proposed in this work. One of them is Gaussian tracing and Gabor-variance filter. In this algorithm, large and small vessels are traced with the same scheme but on different maps. The original image is pre-processed with illumination equalization and improved adaptive histogram equalization. Starting from some automatically selected initialization points, the algorithm trace along the vessels with forward prediction and backward verification. After large vessels are detected, the enhanced image is further processed with a novel Gabor-variance filter to highlight the small vessels. The same tracing algorithm is run on the result image with different set of parameters to detect small vessels. Evaluated with Hoover's performance measure, the overall true positive rate (TPR) of our algorithm on 20 test images (STARE image database) is 84.3% and the false positive rate (FPR) is 3.9%, with Hoover's performance measure. With the proposed central line based measure, the TPR and FPR are 60.56% and 0.39%, respectively. In both case, our algorithm outperforms Jiang's algorithm in comparable TPR and FPR level.

The other algorithm is a radial contrast based. It transforms the intensity information of an image into a high dimensional radial contrast domain. Different feature descriptors are designed to introduce the human intuitions to do more accurate detection of both large and small vessels. In this work, we introduced two major feature descriptors: symmetry and energy descriptors. Symmetry descriptor represents the symmetry of radial contrasts of vessel pixels along different directions. It describes the pedal like shape of radial contrast of vessel pixels in the polar-like coordination. And the energy descriptor is based on the fact that vessel pixels should always have more contrasts negatively sharper than the just noticeable difference, which makes the vessel pixels visible to the human eyes. The histograms of vessel/background pixels in the two descriptor space are analyzed. And it's found that the histogram of vessel pixels in symmetry descriptor space has a lognormal distribution. The thresholds for the two measures of symmetry and energy are calculated adaptively with the information achieved from edge pixels.

Different classification methods are utilized to classify the vessel pixel from the background in the feature space. K-Mean classifier is tested and the TPR/FPR with Hoover's and central line based measure are 59.31%/3.42% and 62.75%/0.37%, respectively. A shape denoising procedure is employed to remove false detections. Evaluated with Hoover's performance measure, the overall performances are 74.48%/3.58% and 57.24%/0.45%. A 2-pass fast detection algorithm is also introduced which can detect more small vessel details with a much higher speed, with the cost of some shallow large vessels. A combination of the shape denoising algorithm and 2-pass algorithm is proposed to get higher detection accuracy. The TPR/FPR for 2-pass fast detection algorithm is 65.3%/2.4% in Hoover's measure, and 67.5%/0.2% in central line based measure. That for combined algorithm is 78.19%/4.4% and 73.10%/0.38% respectively. The two algorithms outperform Jiang's algorithm with comparable time cost. With DRIVE image database, the K-Mean and shape denoising algorithm gain lower performance than Staal's while the 2-pass and combined algorithm get a bit higher performance than Staal's. But our algorithms are much more efficient than Staal's. Our algorithms can be done in less than 10 seconds (2-pass fast algorithm) or 30 seconds (combined algorithm) on a PC with 2GHz CPU while Staal's needs 15 minutes to process one image on a PC with 1GHz CPU.

In this work, I also introduced a large scale early diabetic retinopathy screening system (EDRSS). The system contains three parts. The client tool collects retinal images and demographic data of patients in the clinics. Images and data are transferred to a daemon program for verification before sending to the server. The daemon program does the authentication and controls the data flow to the server. It coordinates the data and images to be transferred to different servers. It also validates the consistency between data and images to make sure that one image belongs to one patient and one patient has at least one image. Server side includes database server, FTP server, file server, web server and a batch processing server. The system provides rich information and functionality for different users. Doctors can view and annotate the images, gives the diagnosis information to help improve the system performance. They can also view the detection outcome from the batch processing server and compare the results with theirs.

A code framework for image processing is designed to take advantage of multi-core computer architecture to distribute image processing task to multiple CPU(s) to improve the computational efficiency and research productivity. This code library also makes creating new filters very easy. It contains three parts: operation target, which includes ColorBgra class and a Canvas class and its derived class RectangularCanvas; operation, including IOperation interface, PixelOperation class, WindowBasedOperation, OperationSet and OperationPipeline; utility class, including some utility functions such as read/write image. These classes make the coding of new image processing algorithms much easier. And the computational time is linearly reduced as the number of processors increases.

The performance of EDRSS is evaluated based on the image readings of two doctors. It should be mentioned that there are different opinions on the same images even between the two readers. For the cases that both doctors diagnosed as diabetic retinopathy (DR), the system recognized 83% patients with DR. Taking doctor A’s reading, which is more consistent with computer’s reading, as ground truth, the system recognize 76% patients with DR and screens out 77% patients without DR. And for the cases that the two doctors have different opinions, the system recognizes 80% patients with DR and screens out 67% patients without DR.

The image quality affects the system performance dramatically. In fact, a lot of patients who actually don’t have DR are misclassified because the images contain bright artifacts that are recognized as hard exudates by the system. Inexperienced operators often take images that are not well focused, or under bad lighting environment. The pigment of retina of people with different races can also cause the performance drop. Our algorithm performs well in both normal and underexposed images without image enhancement.

The study in this work can not only be applied to retinal images, but also detection problems of any linear shape objects such as roads on the satellite map. The concept of the code framework can also be applied to other projects. This work is expected to build a solid foundation for further study on medical related application with image processing methods.

REFERENCES

[1] H. Friedman, "Blood circulation of the eye.," in *Blood Vessels and Lymphatics in Organ Systems*: Orlando: Academic Press Inc., 1984, pp. 247-258.

[2] A. Bill, "Blood circulation and fluid dynamics in the eye," *Physiological Reviews,* vol. 55, pp. 383-417, 1975.

[3] H. Buettner, R. Machemer, S. Charles, and D. R. Anderson, "Experimental derivation of choroidal blow flow. Retinal morphology, early receptor potential, and electroretinography.," *American Journal of Ophthalmology,* vol. 75, pp. 943-952, 1973.

[4] B. W. Zweifach, "Basic mechanisms in peripheral vascular homeostasis," in *Transactions of the Third Conference on Factors Regulating Blood Pressure*, B. W. Zweifach and E. Schorr, Eds. New York: Josiah Macy, Jr. Found., 1949, pp. 13-52.

[5] J. V. Forrester, A. D. Dick, P. G. McMenamin, and W. R. Lee, *The Eye: Basic Sciences in Practice*. London, UK: Elsevier Health Sciences, 2001.

[6] A. Alm, "Ocular circulation.," in *Adler's Physiology of the Eye: Clinical Application*, W. M. Hart, Jr., Ed. St. Louis, Mo: Mosby-Year Book Inc., 1992, pp. 198-227.

[7] J. P. Gieser, M. Mori, N. P. Blair, and M. Shahidi, "Findings on retinal topography and thickness mapping in age-related macular degeneration," *Retina, The Journal of Retinal And Vitreous Diseases,* vol. 21, pp. 352-360, 2001.

[8] J. B. Hickam and R. Frayser, "Studies of the retinal circulation in man: observations on vessel diameter, arteriovenous oxygen difference, and mean circulation time.," *Circulation,* vol. 23, pp. 302-316, 1966.

[9] J. T. Ernest, "Retinal circulation in diabetes mellitus," *Archives of Ophthalmology,* vol. 104, pp. 986-988, 1986.

[10] A. Harris, T. A. Ciulla, H. S. Chung, and B. Martin, "Regulation of retinal and optic nerve blood flow," *Archives of Ophthalmology,* vol. 116, pp. 1491-1495, 1998.

[11] J. E. Grunwald, "Laser doppler velocimetry studies of retinal hemodynamics in diabetes mellitus," in *Biology of the Ocular Microcirculation*, R. N. Weinreb, W. L. Joyner, and L. A. Wheeler, Eds. Amsterdam, the Netherlands: Elsevier Science Publishers BV, 1991, pp. 149-159.

[12] P. J. Saine and M. E. Tyler, *Ophthalmic Photography: Retinal Photography, Angiography, and Electronic Imaging*, 2nd ed. Boston, MA: Butterworth-Heinemann, 2002.

[13] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Transactions on Medical Imaging,* vol. 19, pp. 203-210, 2000.

[14] T. L. McKay, D. J. Gedeon, M. B. Vickerman, A. G. Hylton, D. Ribita, H. H. Olar, P. K. Kaiser, and P. Parsons-Wingerter, "Selective inhibition of angiogenesis in small blood vessels and decrease in vessel diameter throughout the vascular tree by triamcinolone acetonide," *Investigative Ophthalmology and Visual Science,* vol. 49, pp. 1184-1190, 2008.

[15] R. Zwiggelaar, T. C. Parr, and C. J. Taylor, "Comparing line detection methods for medical images," in *18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* Amsterdam, 1996.

[16] R. Kutka and S. Stier, "Extraction of line properties based on direction fields," *IEEE Transactions on Medical Imaging,* vol. 15, pp. 51-58, 1996.

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

[18] H. Li and O. Chutatape, "Fundus image features extraction," in *Proceedings of the 22nd Annual International Conference of the IEEE*, 2000, pp. 3071-3073.

[19] Y. Wang and S. C. Lee, "A fast method for automated detection of blood vessels in retinal images," in *The 31st Asilomar Conference on Signals, Systems & Computers*, 1997, pp. 1700-1704 vol.2.

[20] I. C. Liu and Y. Sun, "Recursive tracking of vascular networks in angiograms based on the detection deletion scheme," *IEEE Transactions on Medical Imaging,* vol. 12, pp. 334-341, 1993.

[21] O. Chutatape, L. Zheng, and S. M. Krishman, "Retinal blood vessel detection and tracking by matched Gaussian and Kalman filters," in *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1998.

[22] G. Tascini, G. Passerini, P. Puliti, and P. Zingaretti, "Retina vascular network recognition," in *SPIE Conference on Image Processing*, Newport Beach, CA, 1993, pp. 322-329.

[23] L. A. Zhou, M. S. Rzeszotarski, L. J. Singerman, and J. M. Chokreff, "The Detection and quantification of retinopathy using digital angiograms," *IEEE Transactions on Medical Imaging,* vol. 13, pp. 619-626, 1994.

[24] H. Shen, B. Roysam, C. V. Stewart, J. N. Turner, and H. L. Tanenbaum, "Optimal scheduling of tracing computations for real-time vascular landmark extraction from retinal fundus images," *IEEE Transactions on Information Technology in Biomedicine,* vol. 5, pp. 77-91, 2001.

[25] S. Ying, "Automated identification of vessel contours in coronary arteriograms by an adaptive tracking algorithm," *IEEE Transactions on Medical Imaging,* vol. 8, pp. 78-88, 1989.

[26] A. Can, H. Shen, J. N. Turner, H. L. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," *IEEE Transactions on Information Technology in Biomedicine,* vol. 3, pp. 125-138, 1999.

[27] X. Gao, A. Bharath, A. Stanton, A. Hughes, N. Chapman, and S. A. Thom, "A method of vessel tracking for vessel diameter measurement on retinal images," in *International Conference on Image Processing*, Thessaloniki, Greece, 2001, pp. 881-884.

[28] L. Gang, O. Chutatape, and S. M. Krishnan, "Detection and measurement of retinal vessels, in fundus images using amplitude modified second-order Gaussian filter," *IEEE Transactions on Biomedical Engineering,* vol. 49, pp. 168-172, 2002.

[29] M. E. Martinez-Perez, A. D. Hughes, A. V. Stanton, S. A. Thom, A. A. Bharath, and K. H. Parker, "Retinal blood vessel segmentation by means of scale-space analysis and region growing," in *Lectures Notes in Computer Science*. vol. 1679, C. Taylor and A. Colchester, Eds. Berlin, Germany: Springer-Verlag, 1999, pp. 90-97.

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

[31] A. Sim and E. d. Ves, "Segmentation of macular fluorescein angiographies. A statistical approach," *Pattern Recognition,* vol. 34, pp. 795-809, 2001.

[32] 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,* vol. 23, pp. 501-509, 2004.

[33] A. Hoover and M. Goldbaum, "Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels," *IEEE Transactions on Medical Imaging,* vol. 22, pp. 951-958, 2003.

[34] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *Image Processing, IEEE Transactions on,* vol. 9, pp. 889-896, 2000.

[35] D. Wu, "Segmentation, registration, and selective watermarking of retinal images," in *Ph.D Disseration, Computer Science* College Station: Texas A&M University, 2005, p. 120.

[36] W. K. Pratt, *Digital Image Processing*. New York: Wiley, 1978.

[37] W. Baarda, "A testing procedure for use in geodetic networks," in *Publications on Geodesy*, vol. 2. Amsterdam, Netherlands: Geodetic Commission, 1968.

[38] V. Barnett and T. Lewis, *Outliers in Statistical Data*, 3rd ed. New York: John Wiley, 1994.

[39] R. H. Ronald, *Methods and Applications of Linear Models: Regression and the Analysis of Variance*. New York: John Wiley, 1996.

[40] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics,* vol. 11, pp. 1-&, 1969.

[41] X. Gao, A. Bharath, A. Stanton, A. Hughes, N. Chapman, and S. A. Thom, "Measurement of Vessel Diameters on Retinal Images for Cardiovascular Studies," in *Medical Image Understanding and Analysis*, 2001.

[42] F. P. Miles and A. L. Nuttall, "Microvessel diameter estimation - error bias correction of serial measurements," *Biorheology,* vol. 28, pp. 315-332, 1991.

[43] F. P. Miles and A. L. Nuttall, "Matched-filter estimation of serial blood-vessel diameters from video images," *IEEE Transactions on Medical Imaging,* vol. 12, pp. 147-152, 1993.

[44] H. C. Chen, V. Patel, J. Wiek, S. M. Rassam, and E. M. Kohner, "Vessel diameter changes during the cardiac cycle.," *Eye,* vol. 8, pp. 97-103, 1994.

[45] B. Khoobehi, G. A. Peyman, and K. D. Vo, "Relationship between blood velocity and retinal vessel diameter," *Investigative Ophthalmology & Visual Science,* vol. 33, pp. 810-810, 1992.

[46] P. Zingaretti and A. Carbonaro, "Route following based on adaptive visual landmark matching," *Robotics and Autonomous Systems,* vol. 25, pp. 177-184, 1998.

[47] A. P. Dhawan, G. Buelloni, and R. Gordon, "Enhancement of mammographic features by optimal adaptive neighborhood image-processing," *IEEE Transactions on Medical Imaging,* vol. 5, pp. 8-15, 1986.

[48] R. Zwiggelaar, S. M. Astley, C. R. M. Boggis, and C. J. Taylor, "Linear structures in mammographic images: Detection and classification," *IEEE Transactions on Medical Imaging,* vol. 23, pp. 1077-1086, 2004.

[49] E. Peli, "Contrast in complex images," *Journal of the Optical Society of America a-Optics Image Science and Vision,* vol. 7, pp. 2032-2040, 1990.

[50] A. B. Watson, H. B. Barlow, and J. G. Robson, "What does the eye see best," *Nature,* vol. 302, pp. 419-422, 1983.

[51] T. H. Lin and T. Kao, "Adaptive local contrast enhancement method for medical images displayed on a video monitor," *Medical Engineering & Physics,* vol. 22, pp. 79-87, 2000.

[52] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey, *Graphical Methods for Data Analysis*. Boston: Duxbury Press, 1983.

[53] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika,* vol. 52, pp. 591-611, 1965.

[54] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, 1967, pp. 281-297.

[55] R. O. Duba and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM,* vol. 15, pp. 11-15, 1972.

[56] G. C. Vafidis, "Features of Diabetic Eye Disease.," in *Diabetic Eye Disease, Identification and Co-management.*, A. R. Rudnicka and J. Birch, Eds. London, UK: Reed Educational and Professional Publishing Ltd., 2000, pp. 52-67.

[57] E. M. Kohner and M. Sleightholm, "Microaneurysms in the development of diabetic retinopathy," *Diabetologia,* vol. 42, pp. 1107-1112, 1999.

[58] M. J. Cree, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester, "A fully automated comparative microaneurysm digital detection system," *Eye,* vol. 11, pp. 622-628, 1997.

[59] J. H. Hipwell, F. Strachan, J. A. Olson, K. C. McHardy, P. F. Sharp, and J. V. Forrester, "Automated detection of microaneurysms in digital red-free photographs: A diabetic retinopathy screening tool," *Diabetic Medicine,* vol. 17, pp. 588-594, 2000.

[60] A. S. Abdel, M. Hafez, and G. Auda, "Using circular Hough transform for detecting microaneurysms in fluorescein angiograms of the ocular fundus," in *International Conference on Industrial Electronics* Cairo, 2001.

[61] W. Hsu, P. M. D. S. Pallawala, M. L. Lee, and K.-G. A. Eong, "The role of domain knowledge in the detection of retinal hard exudates," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, pp. II-246-II-251 vol.2.

[62] J. G. Lawrenson, "Histopathology and Pathogenesis of Diabetic Retinopathy.," in *Diabetic Eye Disease, Identification and Co-management.*, A. R. Rudnicka and J. Birch, Eds.: Reed Educational and Professional Publishing Ltd., 2000, pp. 32-43.

[63] H. M. Leibowitz, D. E. Krueger, and L. R. Maunder, "The Framingham Eye Study Monograph. An opthalmological and epidemiological study of cateract, glaucoma, diabetic retinopathy, macular degeneration, and visual acuity in a general population of 2631 adults, 1973-1975," *Surv Opthalmol.,* vol. 24, pp. 335-610, 1980.

[64] H. Wang, W. Hsu, K. G. Goh, and M. L. Lee, "An effective approach to detect lesions in color retinal images," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, 2000, pp. 181-186 vol.2.

[65] G. Luo, O. Chutatape, H. Li, and S. M. Krishnan, "Abnormality detection in automated mass screening system of diabetic retinopathy," in *Computer-Based Medical Systems, 2001. CBMS 2001. Proceedings. 14th IEEE Symposium on*, 2001, pp. 132-137.

[66] T. Walter, J. C. Klein, P. Massin, and A. Erginay, "A contribution of image processing to the diagnosis of diabetic retinopathy - Detection of exudates in color

fundus images of the human retina," *IEEE Transactions on Medical Imaging,* vol. 21, pp. 1236-1243, 2002.

[67] M. Larsen, J. Godt, N. Larsen, H. Lund-Andersen, A. K. Sjolie, E. Agardb, H. Kalm, M. Grunkin, and D. R. Owens, "Automated detection of fundus photographic red lesions in diabetic retinopathy," *Investigative Ophthalmology and Visual Science,* vol. 44, pp. 761-766, 2003.

[68] G. R. Cherrick, S. W. Stein, C. M. Leevy, and C. S. Davidson, "Indocyanine green: Observations on its physical properties, plasma decay, and hepatic extraction," *Journal of Clinical Investigation,* vol. 39, pp. 592-600, 1960.

[69] A. Dreher, P. Tso, and R. Weinreb, "Reproducibility of topographic measurements of the normal and glaucomatous optic nerve head with the laser tomographic scanner.," *American Journal of Ophthalmology,* vol. 111, pp. 221-229, 1991.

[70] P. Vieira, A. Manivannan, P. F. Sharp, and J. V. Forrester, "True colour imaging of the fundus using a scanning laser ophthalmoscope.," *Physiological Measurement,* vol. 23, pp. 1-10, 2002.

[71] R. Brewster, *http://www.getpaint.net/*. Source code retrieved on September, 2007.

APPENDIX A: OTHER RETINAL IMAGING MODALITY

A. FLUORESCEIN ANGIOGRAPHY

Fluorescein angiogram is obtained using UV lights with the help of a fluorescein. Majority contents in sodium fluorescein injected into the patients binds to the serum protein albumin [62], which in a healthy individual won’t cross the capillary walls. The dye circulates though the patient’s circulatory system, and finally goes into the retinal vasculature. A specialized camera flashes a blue light into the eye and captures several images of the retina. High contrast images are produced with the help of special filters inside the camera that only allow the fluorescent light to be photographed.

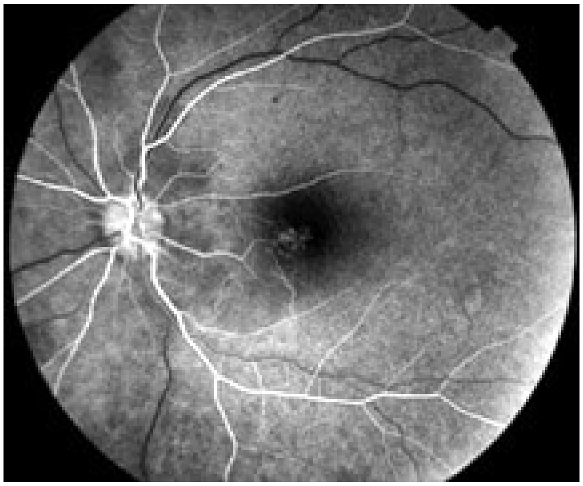


Figure 48 An example of fluorescein angiogram. (Source: The New York Eye and Ear Infirmary/Ocular Imaging Center)

However, fluorescein angiography is not ideal for large scale screening of diseases because of the invasive injection of the fluorescein into the patient and possible adverse health effects including nausea or allergic reactions. An example of a fluorescein angiograph is shown in Figure 48.

B. INDOCYANINE GREEN ANGIOGRAPHY

The indocyanine green dye test has long been used in cardiac blood flow studies, but has limited use in retinal imaging due to its infrared fluorescence spectrum. Improvement in digital camera sensitivity now allows the real-time visualization and recording of the transition of dye through the retinal vasculature. Indocyanine green angiography is used mainly for choroid imaging. The dye will not leak extensively through the capillaries of the choroid and is highly protein-bound, enabling the photographer to obtain a clearer image of the choroid compared to fluorescein angiography [68]. It is often used in combination with fluorescein angiography to aid in the detection and treatment of choroidal neovascularization.

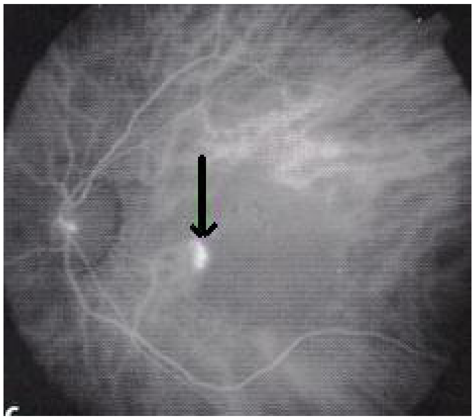


Figure 49 An example of indocyanine green angiogram. (Source: The New York Eye and Ear Infirmary/Ocular Imaging Center)

With the same reasons described in the previous imaging modality, it's not ideal for large scale screening, either. Figure 49 displays an indocyanine green angiograph of a patient with retinal pigment epithelial detachment.

C. SCANNING LASER OPHTHALMOSCOPE

A scanning laser ophthalmoscope (SLO) can obtain higher contrast than normal fundus imaging, and also has the merits of uniform illumination and optical sectioning. No dilation of the pupil is necessary. Using optical coherence tomography with a dual scanning laser ophthalmoscope, one can obtain three-dimensional imaging of the retina and detect pathological changes at the tissue level. One of the most widely used applications of the SLO is the Heidelberg retina tomograph (HRT). This system attempts to quantify the optic nerve heads through combining a series of optical sections of the area [69]. The major drawbacks of this imaging modality are: requirement of computer stereo pairs, smaller field of view, high cost. A SLO image of a patient with characteristic symptoms of diabetic retinopathy is given in Figure 50. The image is taken from [70].

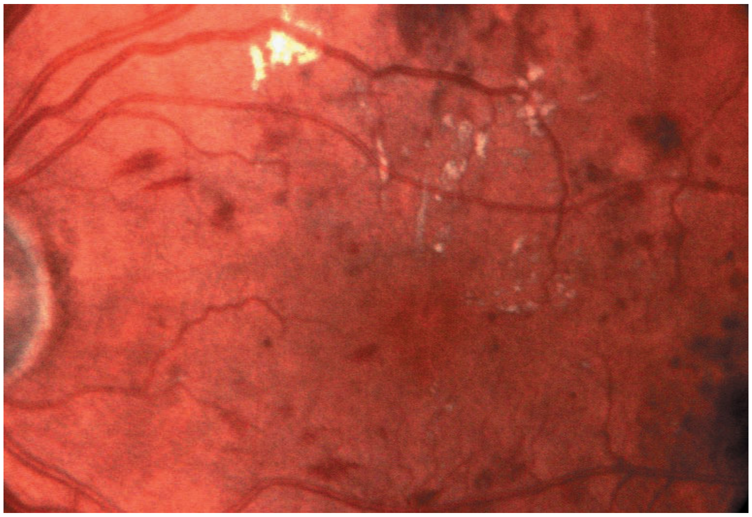


Figure 50 An example of a SLO image. (Soure: [70])

APPENDIX B: EARLY DIABETIC RETINOPATHY SCREEN SYSTEM

A. SYSTEM ARCHITECTURE

The whole system consists two major conceptual parts: client is used by the technicians or doctors to acquire images and input patient data; servers store all the patient information and images, process the images in batch and present the results on web pages. Images are taken with a Canon CR-DGi retina fundus camera, transferred to the computer via USB connection, named with certain naming convention and stored in local hard drive. Demographic data of every patient are filled and stored as well. After a batch of patients has been finished taking pictures, the images and demographic data are transferred to the server via VPN (Virtual Private Network) over internet. Before they are transferred, demographic data and images are validated so that every image corresponds to *exactly* one demographic datum, and one demographic datum corresponds to *at least* one image. Once the transferring finishes, the batch is marked as *final*. A finalized batch can’t be modified any more on server side. There’s a daemon program in server side that accepts all requests of data transferring to the server from client side. The daemon coordinates the demographic data and image transferring to different servers and validates the data/image consistency again. Once the validation succeeds, images transferred to FTP server will be moved to the file server for permanent storage and detection batch processing will be started by the daemon. Final detection results can be viewed via web site exposed over internet. Authentication is required for viewing the web pages. User can download the final report with simple mouse clicks. See Figure 51 for an overview of the system architecture.

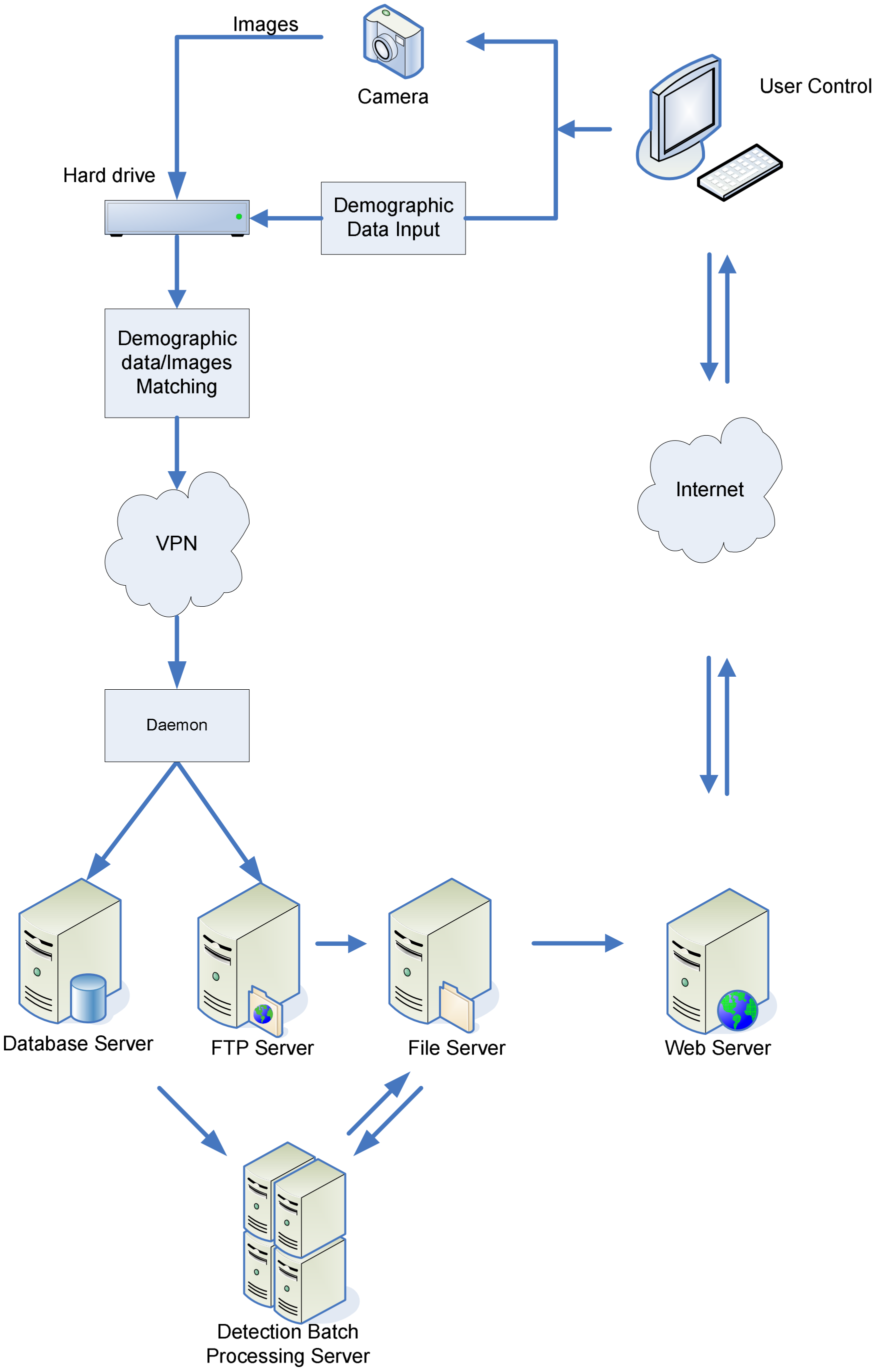


Figure 51 The architecture of early DR screening system.

B. INSTRUMENT CONTROL AND DATA ACQUISITION TOOL

The instrument control and data acquisition (ICDA) tool is deployed remotely in the clinic where the retinal images are taken. It receives images taken from a special retinal fundus camera and accepts input of patient information (demographic data) from the operator, stores them, does data validation and cross-checking, and transfers them to Daemon (see next section).

Retinal camera is specialized low-power microscope with an attached camera. A magnified view of the visual condition of the retina can be photographed and documented directly with it and then used by the ophthalmologists or trained medical professionals to detect, monitor or diagnose the progression of the eye diseases. Canon CR-DGi, a non-mydriatic retinal camera with advanced digital imaging technique, is used in our system. Canon provides rich SDK (Software Development Kit) functions to let computer communicate with the camera via USB connection and download image immediately after it's taken. These images are then stored in local hard drive. Basic image quality control is designed to evaluate the image quality and warn the operator if the image quality is really poor. Currently our QC system recognizes three types of quality problems: overexposure, underexposure and unbalanced exposure. The images are divided into batches. A batch of images is taken in a certain period of time in one specific clinic.

Before the images are taken, the demographic data of patients must be inputted by the camera operator. This step is to ensure that every image belongs to a specific patient. In the field test of the project, it is not rare that some images in the batch can't be matched to any patient. This enforcement effectively avoids this kind of error. The demographic data reflects the patient's medical status *in the batch*. It includes: personal information (Medicaid ID, gender, race, birthday, etc), some basic measurement (such as weight, blood sugar, blood pressure, etc.), diabetic medicine currently taken by the patient and the medical problem the patient is currently having. Note that the same patient may have different demographic data in different batches, e.g., patient's weight or blood pressure may change with time and be different in different batches.

Data consistency is a critical problem in a lot of applications. In the system, two types of consistency must be guaranteed. First, every patient should have *at least* one picture taken, or that patient should not appear in this batch. Second, every picture should be related to *exactly* one patient. If a picture is not related to any patient, it should not appear in this batch; if a picture is related to more than one patient, there must be something wrong. In the client tool, demographic data inputting is enforced to be done before pictures are taken. Operator can select a patient profile that has already been previously filled up to take pictures. Until another profile is selected or created, the images taken are associated with the current one by the patient's medicaid ID and the batch ID. The image file name is generated by combining the medicaid ID and the date and time when the image is taken. The second consistency is guaranteed by this enforcement. Before the demographic data and images are transferred to the server, patients list are scanned to check if any patient has no images associated with them. If such patients are found, client tool will refuse to transfer the data and images to the daemon (We will discuss the daemon in detail in Section C). This additional check guarantees that all data transferred to server satisfy the first consistency. Keeping the data consistency saves a lot of time correcting the miscommunication between the clinic and the data processing center and reduce the error in screening introduced by the images that are incorrectly labeled.

C. DAEMON

Daemon program is located between client tool and all servers. It acts as a gateway to control and record all data flows. All data transferred to the server side must go through the Daemon. This design may cause a performance bottleneck at the Daemon, but considering that the amount of data exchanged between the client tool/daemon and daemon/server is not that significant, and it makes controlling and logging easy, we finally used this design.

When images and demographic data are ready in the client side, client tool connects to Daemon, provides the credential assigned by the system administrator to identify itself. After the information is verified, Daemon will assign a verification code for this connection (i.e., session code) to the client side. Before this connection is terminated, every action taken by the client tool must be associated with the verification code. The verification code is used to distinguish connections from different client tools in different clinics. After the connection is established, the client side sends demographic data of the current batch in XML format to Daemon. Daemon interprets the data and gets ready to verify the image files that will be transferred by the client tool to the FTP server. Client tool then requests a one time use username and password from Daemon to upload all image files to the FTP server. After each image file is transferred, Daemon is acknowledged. After all images are transferred, Daemon validates the data and checks their consistency with the received images again. If everything looks fine, it stores demographic data to the database server, it will move all images to file server for permanent storage, and assign the processing job to processing servers (see next section). All actions (including connecting, login, demographic data transferring and image uploading, etc.) will be logged permanently to the database. The working flow is shown in Figure 52.

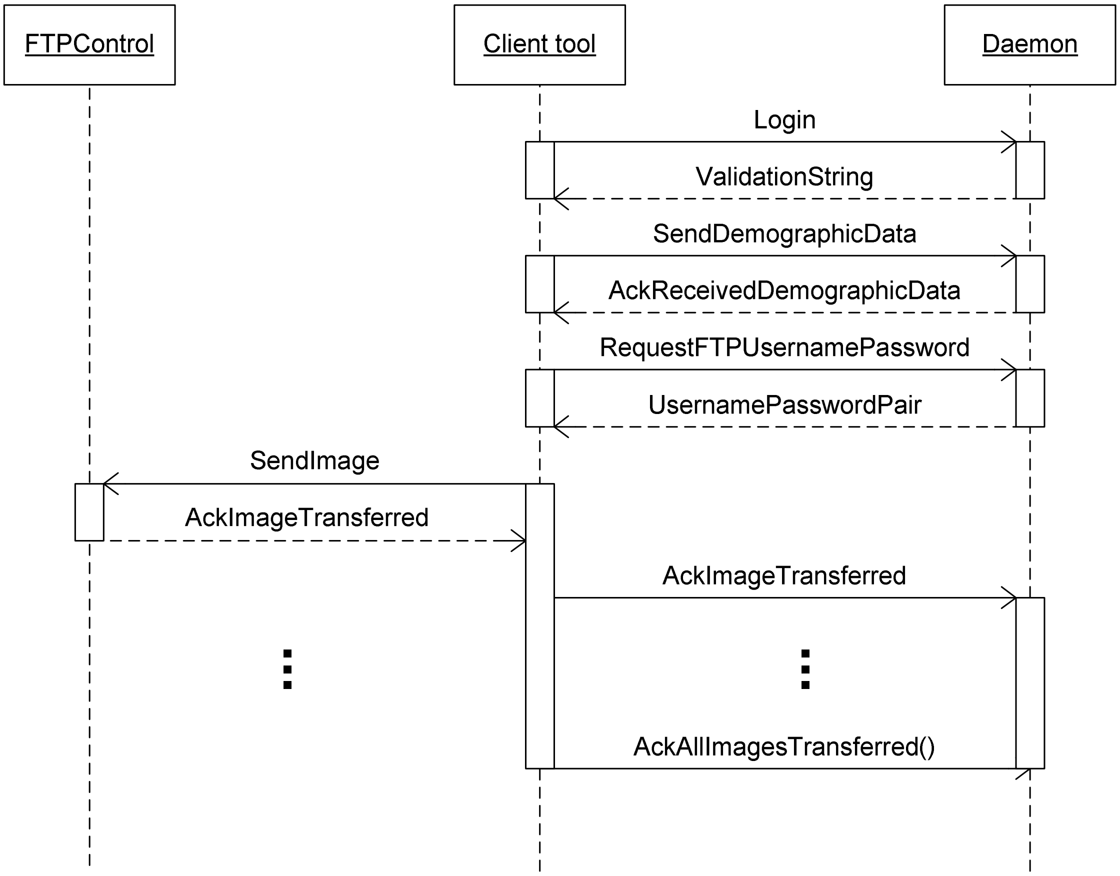


Figure 52 The information flow among FTPControl client tool and Daemon.

D. SERVER SIDE

Server side conceptually includes database server, FTP (File Transfer Protocol) server, file server, web server and batch processing system. Database servers act as the information center to store all the information including authentication information, patients' demographic data, batch information and logs. Usually there are at least two database servers that are hot backup for each other. If one of them failed, the system can keep running. In this project Microsoft SQL server 2005 is selected for its tight integration with the C# language, which is the primary language used to implement the whole project. FTP server accepts images transferred from client tool and temporally stores them. The access to FTP server is controlled by the Daemon. Every time the client tries to transfer the image to FTP server, it must request a one time use username/password pair from Daemon, which is valid only in that connection. The FTP server included in Microsoft Windows IIS server is chosen for the easy control of it via C# program. All uploaded image files are finally moved to file server for permanent storage. In file server, images are organized in batches. Each batch of images is stored in a folder named with the batch ID. Once a batch is moved from FTP server to the file server, this batch of images are made permanent, i.e., unchangeable. If a later adding/deleting/changing of images or demographic data is necessary, a folder with a new version number for that batch will be created under the batch folder and the images will be uploaded in the folder of latest version. This design is for liability purpose. Only the images inside the folder with latest version number will be processed by the batch processing system. Batch processing system is the most important part in the whole project. It analyzes the retinal images, extracts useful objects and features such as blood vessels and some lesions (microaneurysms, hemorrhages and hard exudates, etc), and based on these information and some predefined criteria (e.g., the number of microaneurysm, existence of some significant lesions such as hemorrhages and exudates), it gives a suggestion whether the patient should go to see a doctor for further exam. The batch processing system may consist of multiple servers and each of them may have more than one CPU(s). Once a new batch is available for processing, Daemon assigns it to one of the servers. This server picks up images in this batch one by one and processing it, stores the detection results in an XML file on the file server, which will later be used by report generation system to create the final report.

APPENDIX C: PARALLEL IMAGE PROCESSING LIBRARY

In this appendix, we introduce an image processing code library that can take advantage of the popular multi-core computer architecture to distribute the image processing task to multiple cores and execute them in parallel. This code library was used extensively in EDRSS project and improved the system performance significantly, in time cost. This code library is inspired by an open source project [71] but the implementation is different and some new concepts have been added and implemented.

In all the servers, batch processing system is the most important one. Its performance has direct impact on the whole system. Images may come from various clinics located all over the Texas State. The system should be able to process them quickly and correctly. And in later plan of the EDRSS project, the processing subsystem will be integrated with the client tool in one box so that the patients can get the final suggestion from the computer system in real-time whether they should see a doctor for further examination.

In nowadays, multi-core system has become the mainstream of the computer architecture, even in personal computers. The computational power of the computers has become stronger and stronger. Microsoft .NET framework provides the ability for threads to run on different CPUs without digging into the detail of the low-level thread management. In .NET `Thread` namespace, there's a class called ThreadPool. It holds given number of threads in it and a queue of tasks can be assigned to the thread pool. When a thread in the pool is available, the first task in the queue is assigned to that thread to execute. It is determined by the .NET framework on which CPU the thread will be executed.

It provides great convenience to distribute the task to multiple CPU(s) to execute at the same time, which boosts the system performance significantly.

In most image processing algorithms, certain operation is applied pixel by pixel on the image and the result of one operation usually doesn't affect those on other pixels, i.e., the operations are pixel-wise. This inspires the idea to distribute the operations to multiple cores and do them in parallel manner. In the development of the system, I found that Paint.NET [71] used basically the same thought as mine. My solution has a different focus to make the implementation of new filters/algorithms easier and faster. Also, different kinds of events are added to help make the flow control easier for the asynchronous operations.

Having such a code library in hand improves the productivity of research work, in to aspects: first, the coding efficiency will be much higher. With several lines of code one can easily invoke an existing algorithm and get the result image with a single call. And writing new filters will be easy work (as shown later in this section). It can be done by creating a new class and writing code to apply the operation on a single pixel. All tricky thread management and result rendering will be done by the code library. Second, the computational cost will be significantly cut if the code is run on multi-core machine. The experiment time will be reduced and researchers can get the result faster.

The whole framework contains three parts: operation targets, operations and utility functions. Operations are applied on the target and utility functions are for utility work like loading/writing images, combining images, etc. Each of the parts will be described in detail in the following sections.

A. OPERATION TARGET – COLORBGRA AND CANVAS

1. ColorBgra

Operation target physically is the image on which the operations are applied. In this code library, two classes are used to represent the image – ColorBgra and Canvas. An image is actually a two dimensional array of pixels. A pixel can be abstracted to a point of color with four channels: blue, green, red and alpha, where the first three channels describe the color and the alpha channel describes the transparency of that point. The coordinate of that point in the array corresponds to that of the pixel on the image. A meta class can be created to describe the necessary property of a pixel's color information. To save time and utilize the existing resources, I modified the ColorBgra class in Paint.NET directly with proper copyright permission. The ColorBgra class contains the following most frequently used properties:

- R, G, B, A – the four channels of the pixel's color
- Intensity – The intensity value calculated from R/G/B channels.

2. Canvas

Usually, algorithms/filters are applied on a region of interest (ROI), which is part of (or the whole) image. In digital image processing, images are usually of rectangular shape and can be described with a two dimensional array of ColorBgra objects. Region of interest can have any shape, square, rectangle, circle, etc. A Canvas class is created to describe the ROI on an image. Canvas class is an abstract class that cannot be instantiated. All the abstract method inside it must be implemented in the derived classes. There are several important concepts and properties/methods in this class:

- BaseMap

  BaseMap is a two dimensional array of ColorBgra that holds all the information of the image. Canvas is part of BaseMap and cannot go beyond the range of BaseMap. The part outside BaseMap will be trimmed.

- SetCanvas(params object[])

  This method is an abstract method. It has no implementation in this class and must be implemented by every derived classes of Canvas. Different canvases may have different implementation on this. For example, a rectangular canvas class may provide the implementations with two parameters of Point type that indicate the coordinates of the upper-left and bottom-right corners; and a circle canvas class may provide the implementation with two parameters of which one is of Point type indicating the coordinate of the center and the other is a float number giving the radius. This method works with the following method to uniquely define the canvas shape.

- IsVisible(int x, int y)

  This method returns a boolean value indicating if the given coordinate is in the region of interest (the canvas). This is also an abstract class. Different derived classes have different implementations. In rectangular canvas, a point is visible if its coordinate is in between the two corners, while in a circle canvas, its distance to the center must be no larger than the radius to make it visible. This method is different from OutOfBaseMap(int x, int y), which indicates if the given coordinate is out of the BaseMap.

- Bound

Bound is the smallest rectangle that contains the entire canvas. In a very large image, it's not necessary to scan the whole image and test on every pixel with IsVisible(int x, int y) to find those in the region of interest, which is a waste of time. Bound provides a way to save the scanning time. One only needs to scan the range in the bound, which is usually much smaller than the whole BaseMap. Also, it's useful to divide the whole image into several parts and process them one by one. One just needs to set the canvas to be the smaller sized crop and apply the operations on it, save the result and proceed to the next crop. This is critical when the system resource is limited and can't handle the big image in one pass.

- Other methods

Other methods in Canvas class includes: CloneCanvasInBound(), it returns a new canvas object with the smaller BaseMap only inside the bound. This saves a lot of storage spaces when processing the pixels on the canvas. Another useful function is CopyBaseMapInBoundFrom(Canvas). It copies the content of the BaseMap inside the bound from the given canvas to the current canvas object. It's useful to save the processing result, especially when the image is too big, the processing is applied inline and the original image memory is used to save the result to save memory space. HasSameCanvasShape(Canvas) is used to check if two canvas has the same shape. These methods are all abstract methods and must be implemented in the derived classes.

3. RectangularCanvas

In most image processing algorithms, the region of interest is just a rectangle. RectangularCanvas is a derived class of Canvas to present the canvas with rectangular shape. This class can be used to divide a large image into several small images, process them separately and combine the final result together.

The class diagram of ColorBgra/Canvas/RectangularCanvas and their relations are shown in Figure 53.

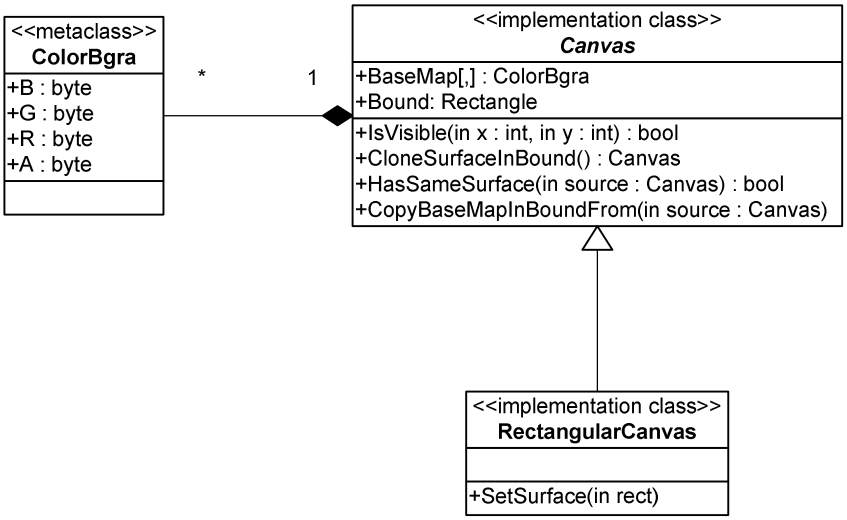


Figure 53 The class diagram of the image processing framework.

B. OPERATION

Operation is the algorithm that runs on the image pixels that produce a value for that pixel on the result image. In this project, several levels of abstraction for operations are created to make a hierarchical structure that provides rich functionality, flexibility and parallelization (See Figure 54).

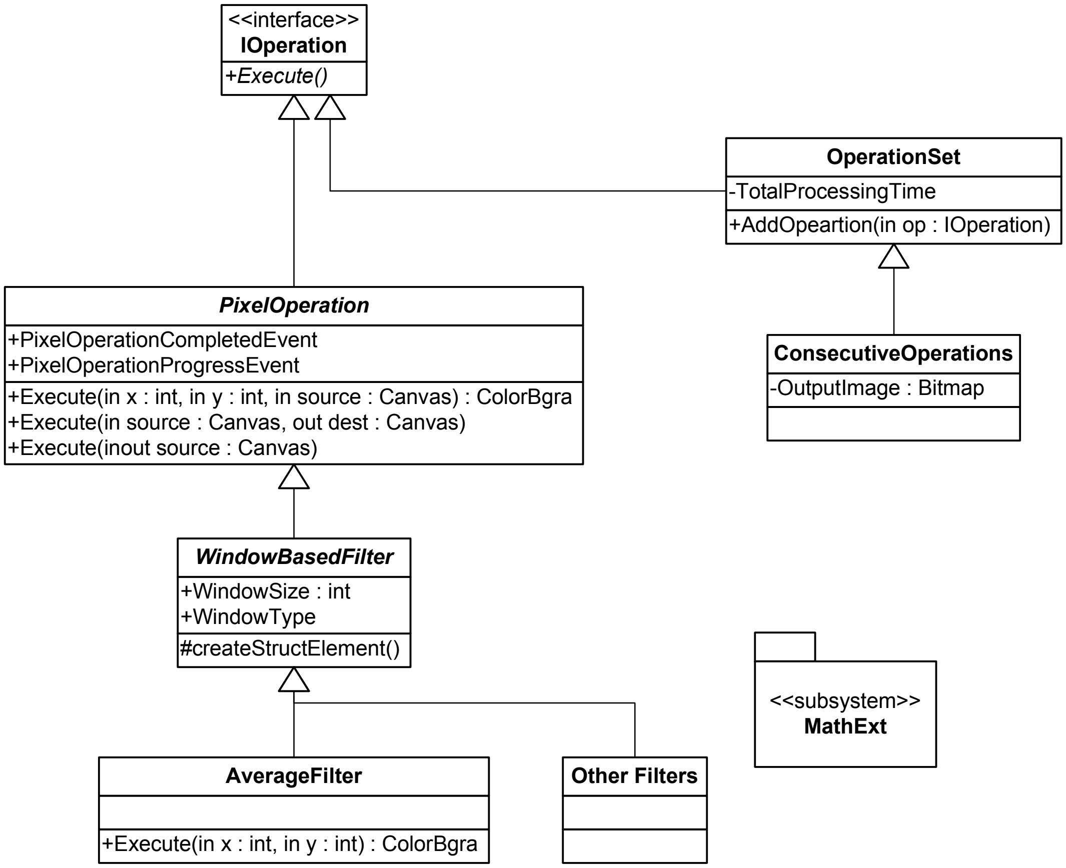

<<interface>>
IOperation
+Execute()
OperationSet
-TotalProcessingTime
+AddOpeartion(in op : IOperation)
PixelOperation
+PixelOperationCompletedEvent
+PixelOperationProgressEvent
+Execute(in x : int, in y : int, in source : Canvas) : ColorBgra
+Execute(in source : Canvas, out dest : Canvas)
+Execute(inout source : Canvas)
ConsecutiveOperations
-OutputImage : Bitmap
WindowBasedFilter
+WindowSize : int
+WindowType
#createStructElement()
<<subsystem>>
MathExt
AverageFilter
+Execute(in x : int, in y : int) : ColorBgra
Other Filters

Figure 54 The class diagram of operations.

1. IOperation

On the top of the hierarchy is a highly generalized interface *IOperation* that only has one abstract method and three events. The abstract method `Execute()` simply states that every operation should have a way to get executed. In the three events, OnBeforeStart is fired before the operation is applied on the image; OnCompleted is fired after the the whole image has been processed; and OnProgress is fired as the image is processed and certain progress has been made (e.g., every 1%). The interface defines the highest abstraction of all operations. It also has another important use to provide a general type that can be used anywhere an operation parameter should be provided.

2. PixelOperation

PixelOperation class is an abstract class that includes all methods and properties needed for an operation that is applied on a single pixel. It has a link to the source canvas that is generated from the original image, and a link to a destination canvas that saves the result. Also,

This class does the job to divide the task of executing operations on all pixels on the image into several subtasks, assigned them to different threads and let them execute on multiple CPU(s). We call this operation parallelization. In this project, there are two levels of parallelization: pixel level and row (column) level. In pixel level parallelization, the operation on each pixel is assigned to a single thread to execute, while in row (column) level, operations on a row (column) of pixels are assigned to a thread to be executed one by one in a blocking way inside the thread. Usually the number of CPU(s) is far less than the number of pixels on the image, so pixel level parallelization will bring too much overhead of context switching. So in this project, I choose row (column) level parallelization. The operation parallelization is done with overloaded Execute method with different signatures. They are described in detail as follow:

- protected abstract ColorBgra Execute(int x, int y)

This is an abstract class, which has no implementation in PixelOperation class. It just provides a signature of the method that applies the operation on a single pixel whose coordination is given by the parameters of the method. It returns a result of ColorBgra type, which represents the result on that pixel. All derived (real) classes of operations should implement this method and fill in what specifically the algorithm does. This method is critical in simplifying the implementation of new filters/algorithms. When adding a new filter, one just need to implement the operation on a single pixel and doesn't have to take care of anything else as long as the new class is derived from PixelOperation class. An example is shown in Figure 55.

- public void Execute(int row)

This method utilizes the previous one to execute the operation on a row (or column) of pixels that are on the canvas one by one and stores the results in destination canvas. This method is executed by a thread, and inside the thread, the execution is done in a blocking way, i.e., the row of pixels are processed one by one. This method will be called by the next method.

- private void Execute(object obj)

This method is the working method that a thread actually executes. In this method, the previous method is called and executed in a blocking manner. After it's done, this row is marked "completed" in the task list, and OnProgress event will be fired to update the progress of the whole operation. If all rows have been completed, OnCompleted event will be fired to notify the caller of the thread that the whole operation has been completed.

```
using System;
using System.Collections.Generic;
using System.Text;
using CodeLib;

namespace CodeLib.Filter {
  /// <summary>
  /// Invert an image.
  /// </summary>
  public class Invert : PixelOperation {
    /// <summary>
    /// Constructor.
    /// </summary>
    public Invert() {
      this.name = "Invert";
    }

    /// <summary>
    /// Execute
    /// </summary>
    /// <param name="x"></param>
    /// <param name="y"></param>
    /// <returns></returns>
    protected override ColorBgra Execute(int x, int y) {
      ColorBgra c= this.workingSurface.BaseMap[x,y];
      return ColorBgra.FromBgr((byte)(255 - c.B), (byte)(255 - c.G), (byte)(255 - c.R));
    }
  }
}
```

Figure 55 A code snippets that show the easiness of creating new filter.

- public void Execute(Canvas source, Canvas dest)

This method executes the current operation on the source canvas and saves the result to the destination canvas. It makes a task list that processing job of each row (column) in the image is assigned to a thread in the thread pool and marked as "incomplete". The status of the processing job will be managed by the threads themselves.

- public Canvas Execute(Canvas source)

This method execute the current operation on the given source canvas, and returns the result canvas.

Figure 56 shows how the overloaded Execute methods work together and implement the operation parallelization. The pink box indicates a *critical section*, which means, when one thread is running this part, other threads must wait it to finish before they enter this critical sections, i.e., critical section is locked when running, and at any time, there's only one critical section running.

By distributing the computational burden to multiple CPU(s), the time cost can be dramatically reduced. If the algorithm can be done in only one CPU with time *t*, the processing time will be *t/n* if it is run in this parallelized framework on *n* CPU(s). There will be some overhead of context switching. That cost can be neglected.

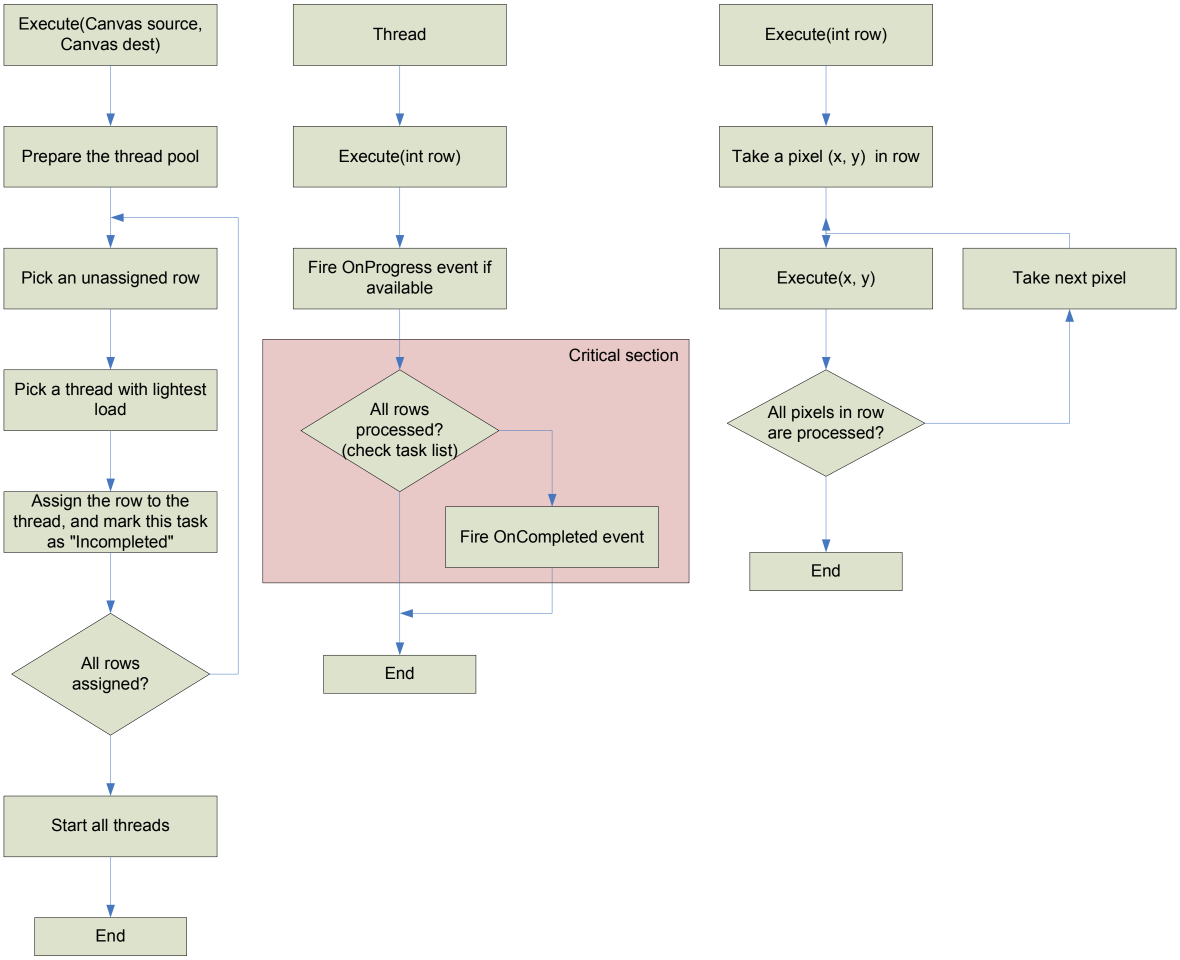

Execute(Canvas source, Canvas dest)
Prepare the thread pool
Pick an unassigned row
Pick a thread with lightest load
Assign the row to the thread, and mark this task as "Incompleted"
All rows assigned?
Start all threads
End
Thread
Execute(int row)
Fire OnProgress event if available
Critical section
All rows processed? (check task list)
Fire OnCompleted event
End
Execute(int row)
Take a pixel (x, y) in row
Execute(x, y)
Take next pixel
All pixels in row are processed?
End

Figure 56 The operation parallelization.

3. WindowBasedOperations

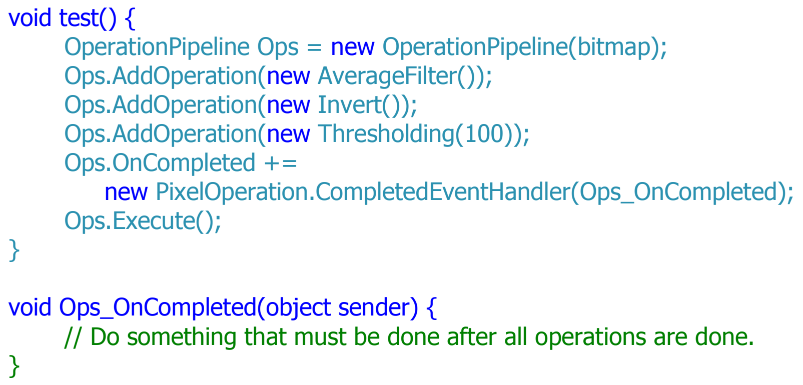
Most of image processing algorithms only need the information in a local range of the pixel being processed. This class is created to provide the abstraction on this level. It provides different types of window shape (square, rectangle, circle, diamond, etc) and the ability to mask out the pixels outside the window. Note the concept of window is different from Canvas. Window describes a local area around a pixel, and Canvas is an area on the base map that will be processed pixel by pixel by the algorithm.

4. OperationSet

There is often needs to apply the same operation on a set of images, e.g., in a batch processing system, and sometimes people just want to set up a batch of image processing jobs that would take long time and let the computer run over night. OperationSet provides this ability. One can add image processing tasks to the set with AddOperation(IOperation) method, and execute them as a batch. Note that every operation in the set must have its own input image/canvas. The execution order of the operations is not defined, which means, even though operation A is added to the set before Operation B, A may be executed after B.

5. OperationPipeline

In image processing jobs, the original image may go through several steps before the result image can be achieved, the result of previous step is the input of next step. For example, a Sobel edge detector contains several steps, first the image is applied a Gaussian filter to reduce the noise; second, two kernels are convoluted to the image and the response map is calculated; then the result image is applied a threshold; finally the skeleton of the result binary map after thresholding is obtained with a thinning algorithm. Since the operations are all asynchronous, to organize them together and execute them one by one in order, an event handler of OnCompleted event must be created for every operation to begin the execution of next operation, which generates a lot of redundant code and affects the readability of the code. OperationPipeline is a class that helps solving the problem. It automatically manages the event handler and executes the operations in the order they were added to the pipeline. Figure 57 shows a code snippet that shows the easy use of OperationPipeline.



```
void test() {
      OperationPipeline Ops = new OperationPipeline(bitmap);
      Ops.AddOperation(new AverageFilter());
      Ops.AddOperation(new Invert());
      Ops.AddOperation(new Thresholding(100));
      Ops.OnCompleted +=
          new PixelOperation.CompletedEventHandler(Ops_OnCompleted);
      Ops.Execute();
}

void Ops_OnCompleted(object sender) {
      // Do something that must be done after all operations are done.
}
```

Figure 57 The code snippets for OpeartionPipeline.

C. UTILITY CLASSES

In image processing system, besides the most important algorithm part, there are a lot of other utility jobs to be done to help the system running. In this project a Utility class is included. It contains:

- ReadBgraData(Bitmap)

  This method reads the color information from the bitmap and returns a two dimensional array of ColorBgra objects, the length of the two dimensions are the width and height of the image, respectively.

- WriteBgraData(ColorBgra[,])

  This method creates a blank image with the width and height to be the length of the dimension of the given ColorBgra array, and writes the color information to the image. It is usually used to save the processing result to the image for visual display and permanent storage (save as image file).

- CombineGrayScaleImages(Bitmap[])

This method combines several gray scale images together by taking the highest intensity value of all the images for each pixel. All images must have the same size.

VITA

Ming Zhang received his B.S degree in Electrical Engineering in 1999, from Southeast University in Nanjing, China. From 1999-2002, he worked as a research assistant in Southeast University doing research work and lab management.

Mr. Zhang started working toward his Ph.D degree in Texas A&M University in August 2002, under the supervision of Dr. Jyh-Charn (Steve) Liu, and graduated with his Ph.D in August 2008. His research areas of interest include medical image processing and pattern recognition. His permanent address is:

Department of Computer Science, Texas A&M University,
College Station, TX, 77843-3112