

Blood Vessel Segmentation in Retinal Fundus Images

Sahinaz Safari Sanjani
PhD Candidate
Electrical Engineering
Stanford University

Jean-Baptiste Boin
PhD Candidate
Electrical Engineering
Stanford University

Karianne Bergen
PhD Candidate
Institute for Computational and
Mathematical Engineering
Stanford University

Abstract—The segmentation of retinal blood vessels in the retina is a critical step in diagnosis of diabetic retinopathy. In this paper, we present a new method for automatically segmenting blood vessels in retinal images. Five basic algorithms for segmenting retinal blood vessels, based on different image processing techniques, are described and their strengths and weaknesses are compared. A hybrid algorithm that combines the confidence outputs of each basic algorithm is described. The performance of each algorithm was tested on the DRIVE dataset. Our hybrid algorithm performs segmentation with similar precision and sensitivity as a trained human observer.

I. INTRODUCTION

Diabetic retinopathy is the leading cause of blindness among adults aged 20-74 years in the United States [1]. According to the World Health Organization (WHO), screening retina for diabetic retinopathy is essential for diabetic patients and will reduce the burden of disease [3]. However, retinal images can be difficult to interpret, and computational image analysis offers the potential to increase efficiency and diagnostic accuracy of the screening process.

Automatic blood vessel segmentation in the images can help speed diagnosis and improve the diagnostic performance of less specialized physicians. An essential step in feature extraction is blood vessel segmentation of the original image. Many algorithms have been developed to accurately segment blood vessels from images with a variety of underlying pathologies and across a variety of ophthalmic imaging systems [9].

This work focuses on developing existing retinal blood vessel segmentation algorithms, comparing their performances, and combining them to achieve superior performance. For this project, the Digital Retinal Images for Vessel Extraction (DRIVE) database of retinal images was used [6], [7]. This database contains 40 images, 20 for training and 20 for testing. These images were manually segmented by two trained researchers. The algorithms were implemented on the original images and the hand segmentations were used to evaluate the performance of the developed algorithms.

The next section of this report explains five distinct vessel segmentation algorithms developed and applied to the DRIVE database. This section is followed by the pipeline developed for combining these algorithms for superior performance. The performance results of all these algorithms are then presented and compared.

II. BASIC VESSEL SEGMENTATION ALGORITHMS

Retinal vessel segmentation algorithms have been heavily researched. There are several approaches to the segmentation. Among these approaches, five of them were chosen for implementation in this project. These methods utilize different image processing techniques and each offer different advantages and disadvantages in vessel segmentation [9]. These properties are then exploited to develop a hybrid algorithm.

A. Matched Filtering

Matched filtering for blood vessel segmentation has first been developed in 1989 [11]. Since then, several different algorithms were developed based on this approach [12] [13] [14] [15] [16] [17]. All of these algorithms are based on the following observations from the retinal blood vessels [11]:

- 1) Blood vessels usually have limited curvature. Therefore, the anti-parallel pairs can be approximated by piecewise linear segments.
- 2) It is observed that the vessel diameters (observed in 2D retinal images as widths) decrease as they move radially outward from the optic disk and range from 2 to 10 pixels in the resulting images from DRIVE database.
- 3) The cross section gray level pixel intensity of blood vessels has a Gaussian profile. Their profile can be approximated by a Gaussian curve:

$f(x, y) = A(1 - k \cdot \exp(\frac{-d^2}{2\sigma^2}))$, where d is the perpendicular distance between the point (x, y) and the straight line passing through the center of the blood vessel in a direction along its length, σ is the spread of the intensity profile, A is the gray-level intensity of the local background and k is a measure of reflectance of the blood vessel relative to its neighborhood.

For the implementation of this algorithm, a 2D matched filter of Gaussian profile is used. 12 different kernel filters are implemented in 15° increments to cover all directions. The kernels have a σ of 2, and are truncated at a neighborhood of $N = \{(u, v) \mid |u| \leq 3\sigma, |v| \leq \frac{L}{2}\}$, where $L = 9$. The mean value of each kernel is then subtracted from it. These kernels are then used as convolution masks across the image. All 12 kernels are convolved with the image and at each neighborhood, the filter that generates the maximum result is considered the correct vessel orientation.

It is worthy to note that the matched filtering algorithm was implemented on the green channel of the fundus images. The reason for this decision is that this channel offers the highest intensity contrast between the vessels and the background as experimentally determined and also confirmed by literature [9].

After running the matched filtering algorithm, the image is further edited based on area and proximity of detected regions. Fig. 1 displays the results of this algorithm.

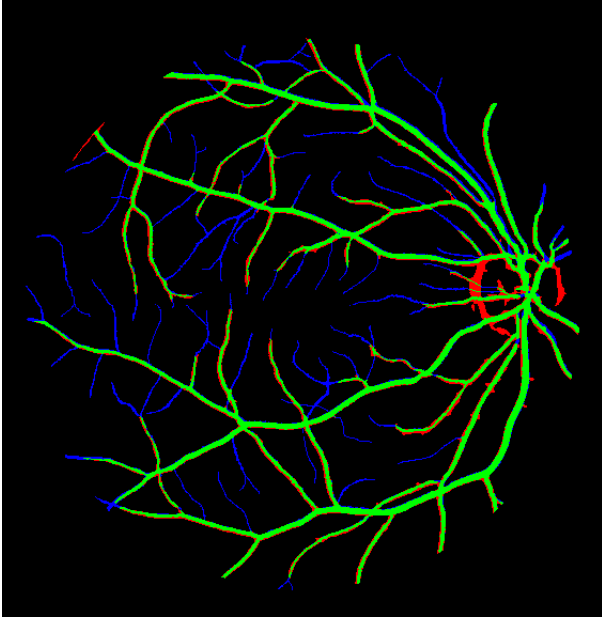


Fig. 1. Sample result for matched filtering algorithm on Image 19 of DRIVE database. True positives (green), false positives (red), false negatives (blue), true negatives (black).

TABLE I
MATCHED FILTER RESULTS

Image	Accuracy	Sensitivity	Precision	Specificity
01	92.39	73.24	71.56	95.41
02	92.48	74.41	76.28	95.78
03	91.84	66.69	76.78	96.37
04	92.80	62.09	81.99	97.79
05	92.62	65.58	79.10	97.12
06	92.05	67.16	75.99	96.34
07	91.72	60.72	74.39	96.67
08	90.47	63.77	63.30	94.47
09	92.46	68.28	69.95	95.87
10	92.58	68.07	71.61	96.11
11	92.12	62.10	74.60	96.74
12	92.15	71.29	69.54	95.29
13	91.91	63.27	77.76	96.87
14	92.84	71.53	70.37	95.81
15	92.33	74.28	62.03	94.51
16	91.63	63.01	72.34	96.17
17	91.53	64.30	68.16	95.56
18	92.59	72.52	67.98	95.33
19	93.09	76.59	71.11	95.48
20	92.17	76.70	62.32	94.13
Mean	92.19	68.28	71.86	95.89

B. Supervised Pattern Recognition

Supervised methods have been shown to perform well on the problem of blood vessel segmentation [18], [19], [20], [21], [22]. These methods vary widely in their choice of features and type of classifier used, but all perform pixel-based classification.

The disadvantage of any supervised method that ground-truth classes from a training set are required. Though these may not always be available or convenient to obtain in practice, for our application this data is available to researchers in the DRIVE [6], [7].

The method described below is adapted from the work of Marín et. al. [22].

1) *Preprocessing*: In following with [22], three preprocessing steps are applied to the images before the features are extracted. The algorithm uses only the green color channel in the RGB colorspace. The first preprocessing step is morphological opening with a three-pixel diameter disk structuring element to reduce the effect of the *central vessel light reflex*, a brighter section along the vessel ridges.

The second preprocessing step, called *background homogenization*, produces uniform background gray levels across the entire set of images. The local background gray level is computed by applying a 69×69 mean filter to the image. The background is then subtracted and the resulting gray levels are scaled from 0 to 1. Finally, a constant is added to the image gray levels so the mode gray level value in image is set to 0.5.

The final preprocessing step is a top-hat transformation on the complement of the image using an eight-pixel radius disk as the structuring element. This final preprocessing step enhances the dark regions in the original image, including the blood vessels, while removing brighter regions such as the optic disk.

2) *Neural Network classifier*: A neural network is used to classify each pixel in the test images as *vessel* or non-vessel. The feature vector associated with each pixel includes seven features, five based on local gray-level information and two based on Hu moment invariants. Hu moment invariants were selected for their scale and rotational invariance.

The gray-level features are computed for a pixel, (x, y) , using the pixel gray-level value, and the gray-level statistics in a 9×9 window, $W_9(x, y)$ centered at (x, y) . The five features include the center pixel gray-level value, the gray-level standard deviation within the window, and the absolute differences between the center pixel gray-level and the minimum, maximum and mean gray-level values in the window. Additionally, for each pixel, the 1st and 2nd Hu moments, I_1 and I_2 are computed for a 17×17 neighborhood window multiplied point-wise by a zero-mean Gaussian of the same size. The absolute value of the logarithm of the Hu moments ($|\log(I_1)|$ and $|\log(I_2)|$) are used as the final two features associated with the pixel. The features are scaled so that each has zero mean and unit variance.

The training set included 27503 pixels (8096 vessel, 19407 non-vessel), representing a relatively small percentage (0.61%) of pixels in the training images. The pixels in the training

set were selected by the authors of [22], available at <http://www.uhu.es/retinopathy/eng/bd.php>.

The structure of the neural network used is a multi-layer feed-forward back propagation neural network, with seven input nodes, three hidden layers with 15 nodes each and one output node. The transfer functions for the hidden layers are linear, and the transfer function for the output layer is the log-sigmoid function, $\text{logsig}(x) = \frac{1}{1+\exp\{-x\}}$. 70% of the training set was used for training and the other 30% for cross-validation to prevent over-fitting of the classifier.

No post-processing was applied to the results of the neural network classifier besides binarization. The output of the classifier was nearly binary (the exception being a small number of pixels along the edges of vessels with values very close to 1), so a threshold of $\tau = 0.75$ was used for all images.

A disadvantage of this method is that because the classification is pixel-by-pixel, the result often has many smaller disconnected segments. Therefore, post-processing methods designed to reduce noise by removing small connected components will also remove these disconnected segments.

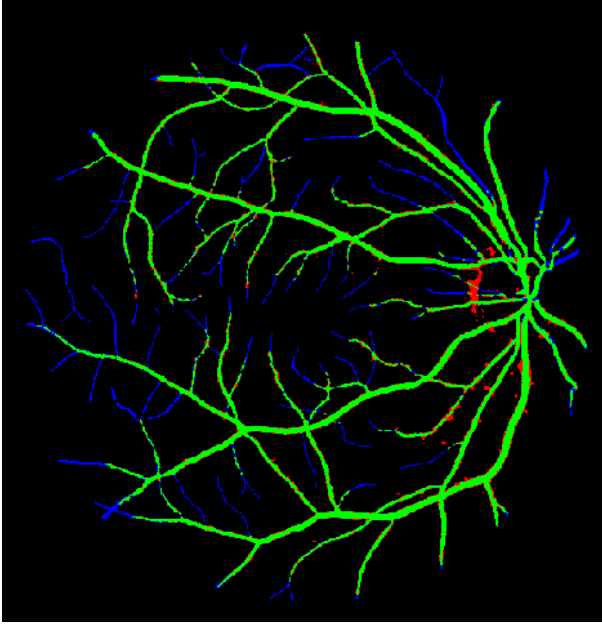


Fig. 2. Sample result for neural network algorithm on Image 19. True positives (green), false positives (red), false negatives (blue), true negatives (black).

C. Multi-scale Line-detection

This method is based on the work of Nguyen et. al. [23]. The idea behind this approach is that the blood vessel structures can be approximated as piecewise linear, so line detection on multiple scales can be used to separate the blood vessel structure from the background. By using lines of multiple lengths, vessels of different sizes and scales can be detected; problematic features, such as the small-scale vessel central light reflex (described above) have limited impact on the result at larger scales.

TABLE II
NEURAL NETWORK PERFORMANCE RESULTS

Image	Accuracy	Sensitivity	Precision	Specificity
01	94.40	76.16	80.11	97.15
02	94.58	73.46	88.44	98.31
03	93.34	69.20	82.26	97.45
04	94.20	63.49	90.08	98.92
05	93.99	67.73	84.97	98.12
06	93.71	67.14	85.11	98.07
07	93.35	65.73	80.48	97.57
08	92.93	63.30	76.41	97.19
09	94.19	68.86	79.01	97.57
10	94.36	71.13	79.47	97.51
11	93.76	65.50	82.78	97.97
12	94.17	72.43	79.22	97.28
13	93.63	66.70	85.15	98.08
14	94.21	75.80	75.26	96.67
15	94.21	74.03	71.33	96.55
16	93.97	68.44	82.45	97.81
17	93.80	70.15	77.44	97.13
18	94.02	72.56	74.68	96.81
19	95.51	80.70	81.74	97.53
20	94.49	76.97	72.92	96.59
Mean	94.04	70.47	80.47	97.51

1) *Preprocessing*: Background homogenization (described in Neural Network preprocessing) without denoising was applied to the inverted green channel of each RGB image. To limit the impact of the optical disk, bright regions (gray-level values exceeding a fixed threshold) are replaced with a local average gray-level calculated with a 69×69 mean filter.

2) *Line Detection*: A total of seven scales are used for line detection, with the line detectors of lengths 3, 5, ..., 15. For each scale, the following procedure was carried out. For each pixel, the mean gray-level in a local 15×15 window, $I^{avg}(x, y)$, is computed. For scale s , line detection is performed by computing the weighted average of gray-level values along lines of length S for each of 18 different angles, $0^\circ, 10^\circ, \dots, 170^\circ$. The largest response, $I^s(x, y)$ over all directions is calculated for each pixel. The line response for scale s is the difference between the maximum line detection response and the average gray-level, $R^s = I^s - I^{avg}$. The line response is rescaled, \tilde{R}^s to have zero mean and unit variance.

The multi-scale line response is obtained by computing a linear combination of the line responses for each scale and the original gray values in the image, I . The weighting used for each line response is proportional to the scale of the response:

$$R = \frac{1}{64} \left(\sum_s s \tilde{R}^s + I \right).$$

The final output is scaled so that the values range from a 0 to 1.

3) *Post-processing*: The result above is used as an input to the hybrid algorithm we describe in the next section. For the multi-scale line detection to produce a binary classification as a stand-alone algorithm, the following binarization and post-processing method is used.

Otsu's method was used to obtain a threshold to binarize the multi-scale line response. The threshold returned by Otsu's method was reduced by 25% to retain finer features in the

image. Finally, all connected components with an area below a fixed threshold of 100 pixels were eliminated.

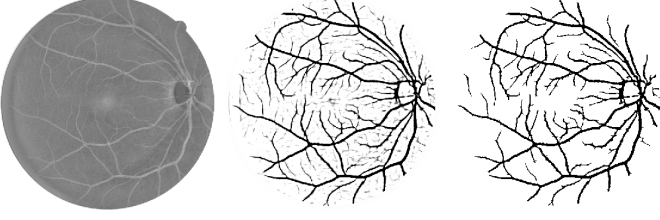


Fig. 3. Stages of Multi-line detection method. Preprocessed image (left), multi-scale line detection result (center), and post-processed binary classification (right).

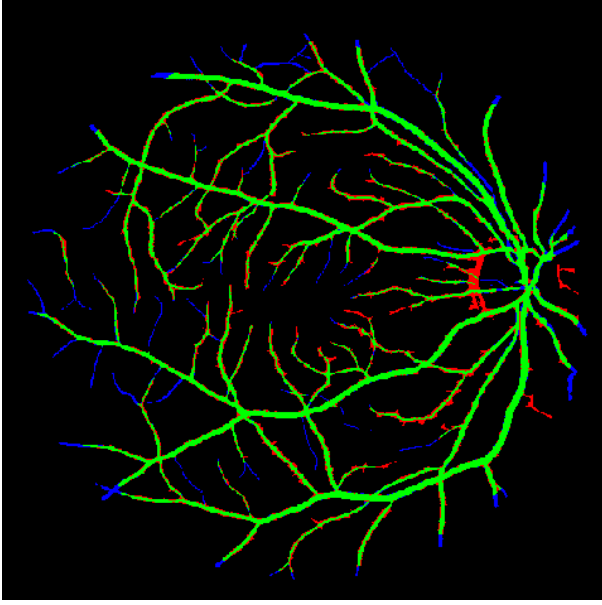


Fig. 4. Sample result for multi-scale line detection on Image 19. True positives (green), false positives (red), false negatives (blue), true negatives (black).

D. Scale-space analysis

A scale-dependent approach is attractive for vessel segmentation because the vessels vary in width. We implemented the algorithm by Martinez-Perez et al. [24], which separates the information between different scales. We can then compute the gradient and the hessian at each pixel for each scale in order obtain a classification that combines all scales.

The luminance channel of the RGB image was used to obtain maximum contrast. We can represent an image $I(x, y)$ at each scale s by convolving with a gaussian kernel of standard deviation s , $G(x, y; s)$:

$$I(x, y; s) = I(x, y) * G(x, y; s)$$

The partial derivatives, necessary for computing the gradient and the hessian can then be computed as follows :

$$I_x = I(x, y) * sG_x$$

TABLE III
MULTI-SCALE LINE DETECTION PERFORMANCE RESULTS

Image	Accuracy	Sensitivity	Precision	Specificity
01	93.36	81.49	71.72	95.15
02	93.72	77.67	79.88	96.55
03	92.35	74.39	73.43	95.41
04	94.09	73.05	80.79	97.33
05	93.73	75.30	77.80	96.62
06	92.67	72.58	74.73	95.97
07	92.90	73.11	73.24	95.92
08	91.36	73.74	63.43	93.89
09	92.95	76.43	67.70	95.15
10	93.29	77.08	69.84	95.49
11	92.45	72.78	70.13	95.38
12	92.83	79.26	68.47	94.78
13	92.72	73.84	74.56	95.84
14	92.51	82.12	64.26	93.89
15	92.34	79.82	59.83	93.79
16	92.89	77.00	76.60	96.46
17	93.63	78.57	68.44	94.90
18	94.59	81.69	68.76	95.18
19	94.09	85.57	73.73	95.83
20	93.50	85.12	64.86	94.50
Mean	93.09	77.53	71.11	95.41

$$I_y = I(x, y) * sG_y$$

$$I_{xx} = I(x, y) * s^2G_{xx}$$

$$I_{xy} = I(x, y) * s^2G_{xy}$$

$$I_{yy} = I(x, y) * s^2G_{yy}$$

The strength of an edge is characterized by the norm of the gradient. We expect the gradient magnitude to be large at vessel edges only, but low on the background and vessels centers. The gradient values are also used in the thresholding method applied later in the pipeline.

The strength of the ridge (vessel) will be evaluated by the largest eigenvalue of the Hessian. Only positive values are considered since we have dark vessels on a bright background. The eigenvalues can be easily and efficiently computed by standard methods.

To compare values across scales, larger weights must be applied to the lower scales, so that they span similar values. A weight of $1/s$ is optimal. Finally, we combine multiple scales by taking the maximum response at each pixel over all scales, and then normalizing the values to the interval $[0, 1]$.

As a compromise between the computation time and the performance, we selected scales from $s_{min} = 1.5$ to $s_{max} = 10$, incremented by 0.5. This choice is dependent on the size of the image, and must be adapted if the image resolution changes.

The figure 5 gives the output for three scales. For the low value of s , the curvature map (column (b)) keeps the very small vessels but the large ones are not completely reported, while for the large value of s , the large vessels are correctly reported but not the small ones. Finally, the last column shows the combined result, which gives a better estimate for likelihood of a vessel.

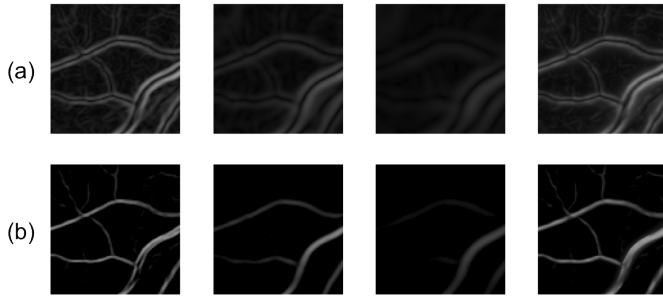


Fig. 5. Scale-space analysis for a 100×100 portion of a retinal image. From left to right : $s = 1.5, 3, 5$, weighted inversely to the scale, and for the last column : final output by putting all the scales together. (a) Gradient. (b) Curvature (from hessian).

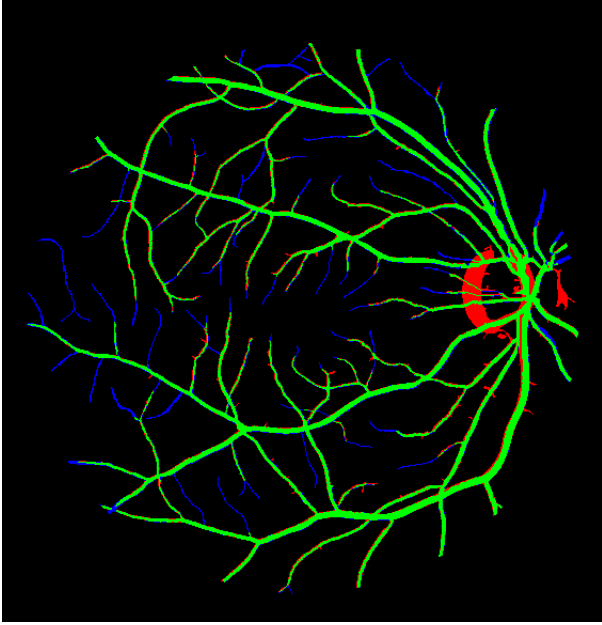


Fig. 6. Sample result for scale-space analysis algorithm on Image 19. True positives (green), false positives (red), false negatives (blue), true negatives (black).

E. Morphological processing

Another approach we can use is based on morphological operators. The underlying idea behind this technique can be illustrated with a simple example.

The function given in figure 7(a) has clear “bumps” that represent two vessels, which are brighter than the surrounding background (as in the inverted green image channel). We demonstrate the results of two top-hat operators with structuring elements of different size, the first size being the width of the largest vessel (figure 7(b)), and the second one being the width of the smallest vessel (figure 7(c)). As we can see, the top-hat operator will get rid of the low variation background. And, we observe that the result is optimal when the size of the structuring element is approximately the same as the size of the vessel. This is the motivation for applying a series of top-hat operators with 8 different scales, spanning the whole range of possible widths of the vessels.

TABLE IV
SCALE-SPACE RESULTS

Image	Accuracy	Sensitivity	Precision	Specificity
01	93.88	74.11	79.54	97.00
02	93.79	66.84	90.39	98.70
03	92.56	66.06	81.69	97.33
04	93.99	63.83	90.27	98.88
05	93.44	64.35	86.20	98.29
06	93.06	63.89	85.28	98.10
07	92.98	59.11	85.28	98.37
08	90.97	48.83	72.81	97.27
09	93.38	58.64	82.77	98.28
10	93.64	66.97	79.28	97.48
11	93.70	64.88	84.30	98.14
12	93.81	68.65	81.19	97.60
13	92.54	62.06	83.10	97.82
14	94.36	71.71	80.07	97.52
15	94.88	74.31	77.28	97.36
16	93.97	66.77	86.16	98.30
17	93.13	62.12	80.12	97.72
18	94.39	70.84	80.20	97.61
19	94.88	80.43	79.45	96.98
20	94.69	72.94	78.34	97.45
Mean	93.60	66.37	82.19	97.81

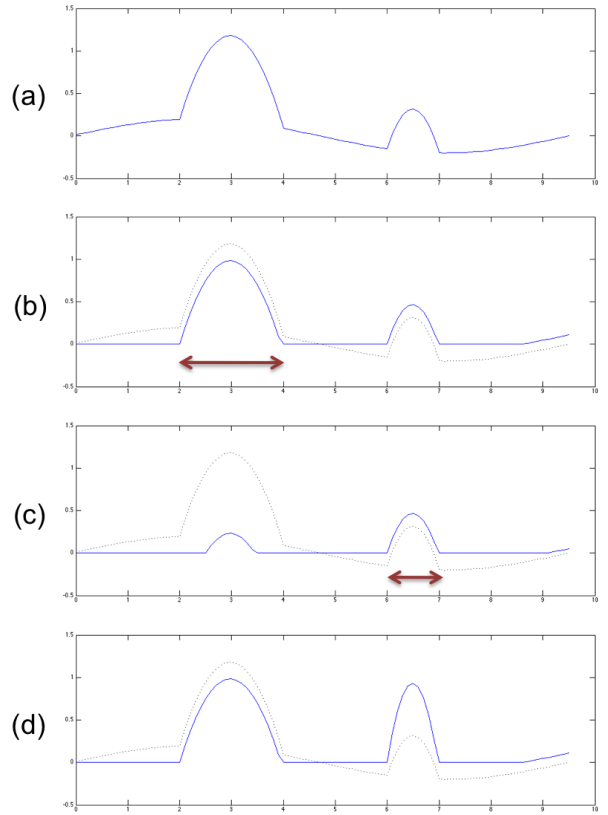


Fig. 7. Simplified example of the application of a top-hat operator in the case of two vessels of different sizes on a noisy background. (a) Original function. It is reproduced as a dotted line for the other figures. (b) Application of a top-hat operator with a structuring element whose size is represented by the red arrow, and which is similar to the size of the largest vessel. (c) Application of a top-hat operator with a structuring element with a size similar to the size of the smallest vessel. (d) Maximum of the functions obtained in (b) and (c) and with a weight 2 for the second one.

In 2D, the principle remains the same, and we selected a circular structuring elements, because the vessels have no privileged direction.

Because the regular top-hat operator is very sensitive to noise, we use a modified top-hat operator, as used by Mendonca et al. [25], and defined as follows :

$$tophat(img) = img - \min(open(close(img, W_c), W), img)$$

The small noisy fluctuations will not be retained thanks to the closing operator that we apply first. The structuring element associated to this closing operator W_c , is small but, larger than the typical size of the noise we want to attenuate. We used a disk of radius 2 pixels for our application.

The main structuring element W is chosen as a disk with a radius varying from 1 to 8 pixels. This gives 8 results. We average consecutive paired consecutive images to obtain 4 results, which helps reduce the noise.

The last operation is the equivalent of the one we apply in figure 7(c); for a given pixel, the response will be given by a maximum on the values for each scale, weighted with larger weights for lower scales, to compensate for the weaker response. In the end, we normalize the output so that our values range between 0 and 1.

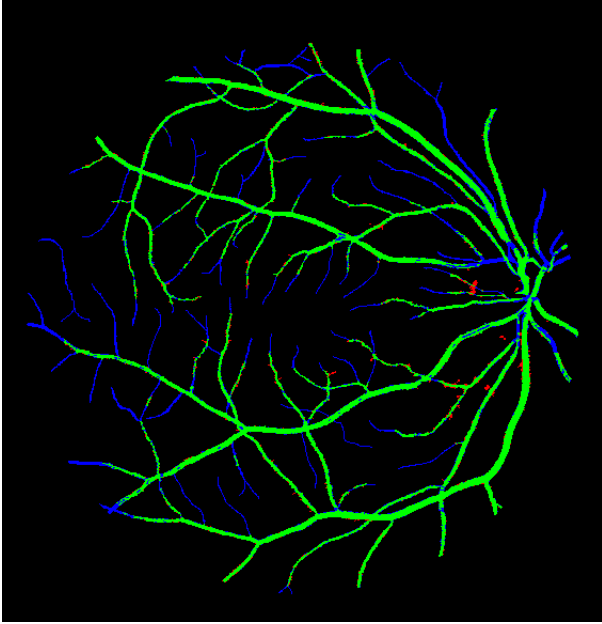


Fig. 8. Sample result for morphological processing algorithm on Image 19. True positives (green), false positives (red), false negatives (blue), true negatives (black).

TABLE V
MORPHOLOGICAL PROCESSING RESULTS

Image	Accuracy	Sensitivity	Precision	Specificity
01	94.11	66.98	86.71	98.38
02	94.29	67.98	93.18	99.09
03	92.28	59.40	85.59	98.20
04	94.16	62.31	93.78	99.33
05	93.20	55.34	94.85	99.50
06	93.41	62.53	89.57	98.74
07	93.45	56.50	93.22	99.35
08	92.80	48.66	92.46	99.41
09	94.23	60.46	89.37	98.99
10	93.84	57.30	90.24	99.11
11	93.94	61.50	89.95	98.94
12	94.09	67.01	84.69	98.17
13	93.09	58.56	91.58	99.07
14	94.58	66.53	85.96	98.49
15	95.47	66.90	88.23	98.92
16	94.14	62.21	92.61	99.21
17	93.71	62.03	85.09	98.39
18	94.33	63.21	85.91	98.58
19	96.00	72.95	94.21	99.35
20	95.05	71.89	81.81	97.98
Mean	94.01	62.51	89.45	98.86

III. HYBRID ALGORITHM

A. Overview

In order to take advantage of the strengths of each of these methods, we have developed a hybrid algorithm for retinal blood vessel segmentation. Our hybrid algorithm combines the result the five methods above to achieve an improved segmentation.

Four of the five methods above (with the exception of the neural network-based method which was essentially binary in practice), return a continuous value between 0 and 1 for each pixel. While thresholding or other post-processing can be used to obtain a binary output for each method, these values on $[0, 1]$ can also be thought of as a confidence or probability that a given pixel is part of the blood vessel. Therefore, our set of five basic segmentation methods gives us five different (and possibly conflicting) confidence estimates for each pixel.

Our hybrid algorithm is a method to map the confidence estimates returned by each basic algorithm into a single binary classification. This is done in two steps. The first is to combine the five confidence estimates at each pixel into a single value between 0 and 1. The second step is to apply a sophisticated thresholding technique, based on a priori knowledge of blood vessel characteristics, to obtain a final binary classification.

B. Weighting and Modification

The five confidence estimates derived from our five basic algorithms are combined using a simple, unweighted, average of the confidences returned by each individual algorithm.

The idea behind this approach is that we expect that averaging the five results will reduce noise. Since each method relies on different image processing techniques, we expect the noisy regions will differ across methods while local vessels segments will be detected by multiple methods.

A modified version of the hybrid algorithm was also implemented based on the following observation: four out five

of the implemented algorithms perform poorly around bright exudate areas and the optic disk, as observed in Fig. 1, Fig. 2, Fig. 6 and Fig. 4. Only the morphological vessel segmentation approach performs well in these areas, as observed in Fig. 8. However, this method misses many details away from the optic disk.

In order to combine the advantages of all the implemented algorithms, a mask was developed to separate out areas of exudate and optic disk (bright areas) from the rest of the retinal image. This mask was developed based on the green channel of the original retinal image, which offers the highest vessel intensity contrast as mentioned in the matched filtering section. Adaptive histogram equalization is the next step in order to compensate for the varying illumination of the image taken by the fundus camera. A simple hard threshold is then used to create the mask. The white areas of the mask are dilated to avoid edge effects from the bright areas.

For this version of the hybrid algorithm, the white areas of the mask are replaced by the vessel segmentation result from the morphological algorithm. The rest of the image uses the original hybrid algorithm with the averaging scheme indicated above. Figure 9 displays this modification to the hybrid algorithm. The resulting confidence estimates, shown in the figure, are passed to the thresholding algorithm described in the next section.

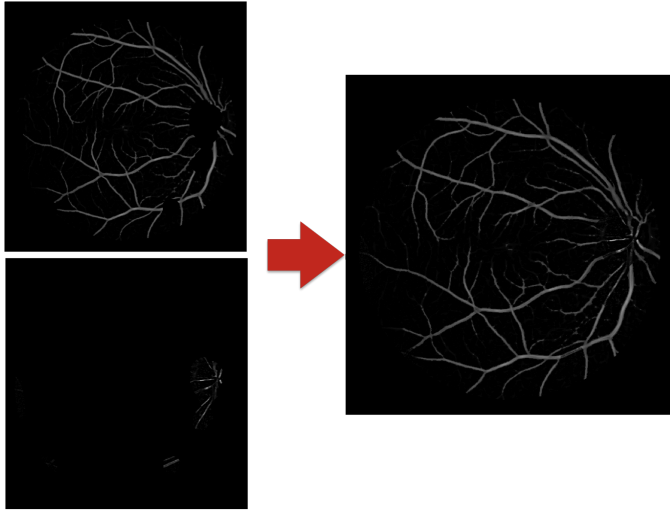


Fig. 9. Stages of modified hybrid algorithm. original hybrid image minus the optic disk and other unusually bright areas (top left), morphological vessel segmented image in the areas of interest (bottom left), and combined image for the modified hybrid algorithm (right).

C. Thresholding: from a soft to a hard classification

The next step of the pipeline is to turn our soft classification into a hard one. Until now, we only produced a map where each pixel has a value between 0 and 1, this number expressing the confidence or likelihood of the presence of a vessel. We would like to map this into a binary classification, *vessel* or *background*.

The first approach one might use for binarization is Otsu's method : the vessel pixels should have a high value while the

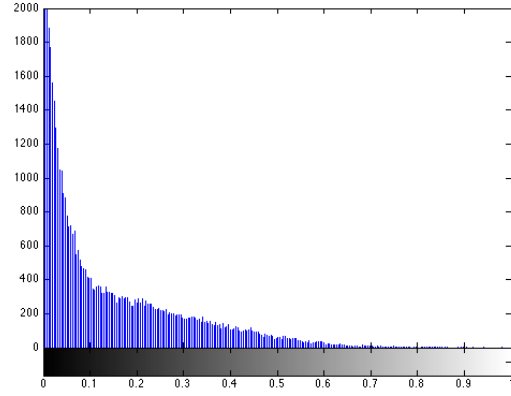


Fig. 10. Histogram of the output of the scale-space algorithm. For a better visibility, the low values are clipped.

background values should be close to zero. But because of the imperfections of our methods, a vessel pixel can also have a low value. Checking the histogram of the output of one of our algorithms, Figure 10, helps us understand why this algorithm cannot be properly applied here. The distribution of the pixel values between vessel and background is not well separated. This can lead to pixels of the vessel that will turn out to be below the threshold, and thus incorrectly labelled as background. As a consequence, the vessel will appear as disconnected segments (cf. Figure 11(b)).

To improve this method, we need to take into account the properties of the retinal blood vessels. We know that the vessels form a vascular tree of connected vessels. This insight allows us to create a new algorithm that will iteratively grow the tree, using the connectivity, as proposed by Martinez-Perez et al. [24]

The first step of the algorithm is to establish pixels that will be considered with certainty as background or vessel pixels. To do that, we divide the pixels in two classes using the Otsu threshold t in the global image. The pixels below t will form the initial background class, and the ones above t will form the initial vessel class. We characterize these two classes with their mean μ_b and μ_v (respectfully, for the background class and the vessel class) and their standard deviation σ_b and σ_v . We consider that the pixels that are above μ_v are always vessel pixels, while the pixels below μ_b are always background pixels.

We use those labelled pixels as seed pixels for our growing algorithm. We will alternatively grow the background and the vessel regions by looking at the pixels connected to these seeds. We will also use the gradient information; the pixels with low gradients are either in the middle of a vessel or in the background, so they are more easy to classify. We characterize the histogram for the gradient values with the mean μ_g and the standard deviation σ_g .

At first, we classify only the pixels that have a clear value for the likelihood (either very high or very low), and a low

value for the gradient. For the class vessel :

$$\mu_v - a\sigma_v \leq p \text{ and } \gamma \leq \mu_g + a\sigma_g$$

and for the class background :

$$p \leq \mu_b + a\sigma_b \text{ and } \gamma \leq \mu_g$$

We take the parameter $a = 1$ for the first step, and we increment it by 0.5 for each run, until there are no pixels left to classify. This means that we become progressively more permissive in our labelling. The edge pixels outside of the vessels usually have a higher value for the gradient, and the fixed condition $\gamma \leq \mu_g$ for the background vessels will prevent these problematic edge pixels to be labelled at this stage.

In the second stage, the same iterations are performed but this time without the condition on the gradients. We iterate until all pixels are classified.

An example of the results achieved by this algorithm is given in Figure 11. We can see that even though it does not perform as well as the manual classification, it takes advantage of the tree structure of the vessels and performs much better than the basic thresholding algorithm.

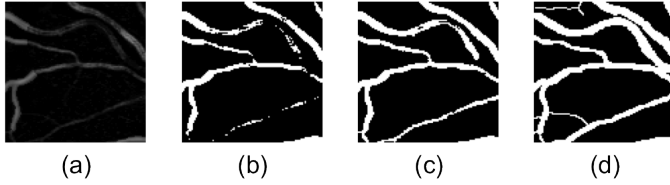


Fig. 11. (a) Detail of an original image obtained after application of our 5 vessel detection algorithms. (b) Result of Otsu's method. (c) Result of the tree growing method. (d) Manually labelled reference image

TABLE VI
HYBRID (ORIGINAL) RESULTS

Image	Accuracy	Sensitivity	Precision	Specificity
01	94.37	81.66	77.99	96.37
02	94.74	79.46	85.42	97.53
03	93.43	74.27	81.08	96.88
04	94.44	69.41	88.28	98.50
05	94.20	73.08	84.22	97.72
06	93.95	72.55	84.15	97.64
07	93.54	72.85	78.61	96.84
08	92.78	67.21	74.78	96.61
09	94.28	74.39	78.21	97.08
10	94.13	76.33	76.87	96.69
11	94.06	72.24	81.17	97.42
12	94.26	76.68	78.91	96.91
13	93.68	72.28	82.69	97.38
14	94.38	80.30	75.35	96.34
15	94.37	80.06	71.25	96.10
16	94.07	76.24	79.63	96.90
17	93.78	74.99	76.33	96.56
18	94.06	79.02	73.52	96.11
19	95.37	85.89	79.32	96.75
20	94.41	81.71	72.18	96.01
Mean	94.11	76.03	79.00	96.92

TABLE VII
HYBRID (MODIFIED) RESULTS

Image	Accuracy	Sensitivity	Precision	Specificity
01	94.46	73.43	83.85	97.77
02	94.21	67.95	92.56	99.00
03	93.49	71.38	83.59	97.48
04	93.52	56.29	95.45	99.56
05	94.23	68.62	88.36	98.50
06	93.96	68.44	87.80	98.36
07	94.06	62.55	91.61	99.09
08	93.69	63.29	84.32	98.24
09	94.59	68.38	84.92	98.29
10	94.58	72.34	82.46	97.78
11	93.97	60.56	91.34	99.12
12	94.43	71.27	83.84	97.93
13	93.39	61.57	90.62	98.90
14	94.80	75.37	80.79	97.51
15	94.90	74.78	77.15	97.33
16	94.45	69.18	87.76	98.47
17	93.99	64.15	85.63	98.41
18	94.61	71.83	81.19	97.73
19	95.62	81.98	83.21	97.60
20	95.01	77.91	77.74	97.18
Mean	94.30	69.06	85.71	98.21

IV. RESULTS

TABLE VIII
SECOND OBSERVER

Image	Accuracy	Sensitivity	Precision	Specificity
01	94.93	79.62	81.22	97.23
02	94.94	82.29	83.58	97.16
03	94.06	74.26	83.17	97.44
04	94.82	78.43	82.22	97.37
05	94.63	73.82	84.99	97.93
06	93.61	75.45	78.72	96.62
07	94.50	68.61	87.44	98.48
08	94.28	66.00	85.05	98.33
09	94.57	76.91	77.12	96.94
10	94.73	71.77	82.25	97.88
11	94.64	75.99	81.70	97.44
12	95.09	77.08	82.86	97.70
13	93.89	80.46	77.60	96.13
14	95.49	77.08	83.34	97.94
15	95.37	80.11	76.74	97.16
16	94.92	77.89	82.61	97.51
17	94.95	73.30	83.69	97.99
18	94.86	85.83	74.00	96.05
19	95.34	90.70	75.75	95.98
20	94.41	87.00	69.07	95.30
Mean	94.70	77.63	80.66	97.23

A. Comparison of Algorithm Performance

To measure the performance of our algorithms, we look at the precision and sensitivity (recall) for each of 20 test images for each basic algorithm and the two hybrid algorithms (see Appendix VI-A).

We use as a reference the images manually labelled by a second observer to estimate the consistency of two different trained human observers. We realize that the human is far from perfect in this task; compared to the first human observer, the second one only achieves a mean sensitivity of 77.63% and a precision of 80.66%, which average to 79.14%. An algorithm

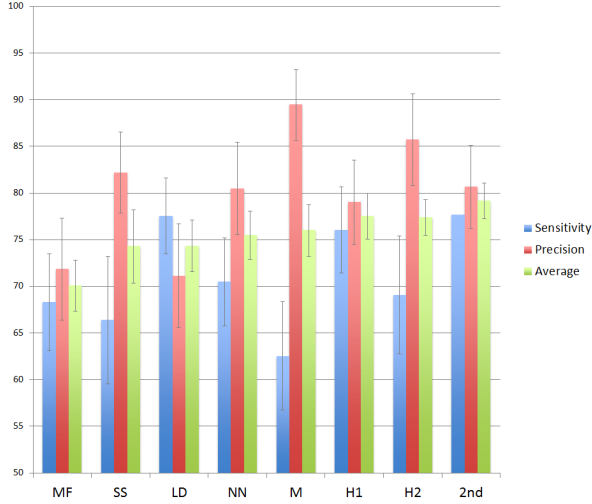


Fig. 12. Sensitivity, precision and average of those two values for each of the algorithms. From left to right : matched filtering, scale-space, line detection, neural network, morphological, hybrid algorithm, modified hybrid algorithm, and hand-labeled vessel segmentation by a second observer

that achieves similar performance to these values can be said to perform similarly to a human, which is our ambition. The fact that the segmentation results hand-labelled by two humans are so different also proves the difficulty of our task; it can be difficult to judge if a pixel belongs to a very small vessel or to a noisy background and which pixels form the discrete edge of the vessel.

If we compare our five algorithms, we see that they lie in different zones of the trade-off curve between sensitivity and precision. The line-detection algorithm has higher sensitivity (it misses fewer small vessels) at the cost of a lower precision (more false positives), while the other algorithms have a higher precision. It is important to note a particular strength of the morphological algorithm; as we noticed before, compared to the other algorithms this one obtains fewer false positives around the optical disk, resulting in the highest precision among the methods. However, its sensitivity is low, with weaker performance on small vessels.

The original hybrid algorithm has a higher average sensitivity/precision than all the other algorithms. Its value is 77.51%, only 1.63% lower than the second method. The sensitivity and precision are also quite similar to the values for a human.

Our second algorithm, which included the masking around the optical disk, which was designed to obtain a higher precision (we get fewer of the false positives around the optical disk) meets this performance goal, but at the cost of a lower sensitivity. On the whole, the average is 77.39%, which is very close to that of the original hybrid algorithm. Depending on the relative importance of small vessel or suppression of false positives in a given application, we may prefer one algorithm over the other.

V. CONCLUSION

This project focuses on blood vessel segmentation in retinal fundus images for the potential application of automatic diabetic retinopathy diagnosis. In this project, five algorithms were implemented based on methods from relevant literature. These five algorithms were then combined using two different approaches in order to take advantage of their individual advantages.

Several performance measures were computed in order to evaluate the performance of each of the developed methods. These measures include accuracy, precision, sensitivity, and specificity. Each of the developed algorithms offer trade-offs between these performance metrics. However, the modified hybrid algorithm results tend to have superior performance when averaging all the performance metrics.

ACKNOWLEDGMENT

The authors would like to thank David Chen for his continuous guidance throughout the project.

REFERENCES

- [1] Klein R, Klein B. "Vision disorders in diabetes." In: National Diabetes Data Group, ed. Diabetes in America. 2nd ed. Bethesda, MD: National Institutes of Health, National Institute of Diabetes and Digestive and Kidney Diseases; 1995, pp. 293-337.
- [2] Zhang X et al. "Prevalence of Diabetic Retinopathy in the United States," 2005-2008 JAMA. 2010; 304(6):649-656. <http://jama.jamanetwork.com/article.aspx?articleid=186384#ref-joc05096-1>
- [3] World Health Organization. Prevention of Blindness and Visual Impairment. <http://www.who.int/blindness/causes/priority/en/index6.html>
- [4] Selvathi et al. "Automated Detection of Diabetic Retinopathy for Early Diagnosis using Feature Extraction and Support Vector Machine." *International Journal of Emerging Technology and Advanced Engineering*. 2012; 2(11):103-108.
- [5] G. S. Ramlugun, V. K. Nagarajan, C. Chakraborty, "Small retinal vessels extraction towards proliferative diabetic retinopathy screening," *Expert Systems With Applications*, 2012, vol. 39, pp. 1141-1146.
- [6] J.J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, B. van Ginneken, "Ridge based vessel segmentation in color images of the retina", *IEEE Transactions on Medical Imaging*, 2004, vol. 23, pp. 501-509.
- [7] M. Niemeijer, J.J. Staal, B. van Ginneken, M. Loog, M.D. Abramoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database", *SPIE Medical Imaging*, Editor(s): J. Michael Fitzpatrick, M. Sonka, SPIE, 2004, vol. 5370, pp. 648-656.
- [8] A.D. Hoover, V. Kouznetsova, M. Goldbaum, Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response, *IEEE Transactions on Medical Imaging* 19 (2000) 203-210.
- [9] M.M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvarab, A.R. Rudnickac, C.G. Owenc, S.A. Barmana, "Blood vessel segmentation methodologies in retinal images A survey", *Computer Methods and Programs in Biomedicine* Volume 108, Issue 1, October 2012, Pages 407-433.
- [10] U.M. Akram and A.S. Khan, "Automated Detection of Dark and Bright Lesions in Retinal Images for Early Detection of Diabetic Retinopathy." *Journal of Medical Systems*, Volume 36, Issue 5, November 2012.
- [11] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, 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.
- [12] B. Zhang, L. Zhang, L. Zhang, F. Karray, "Retinal vessel extraction by matched filter with first-order derivative of Gaussian," *Computers in Biology and Medicine*, vol. 40, pp. 438-445, 2010.
- [13] M.G. Cinsdikici, D. Aydin, "Detection of blood vessels in ophthalmoscope images using MF/ant(matched filter/ant colony) algorithm," *Computer Methods and Programs in Biomedicine*, vol. 96, pp. 85-95, 2009.
- [14] M. Amin, H. Yan, "High speed detection of retinal blood vessels in fundus image using phase congruency", *Soft Computing: A Fusion of Foundations, Methodologies and Applications* pp. 1-14, 2010.
- [15] M. Al-Rawi, M. Qutaishat, M. Arrar, "An improved matched filter for blood vessel detection of digital retinal images," *Computers in Biology and Medicine*, vol. 37, pp. 262-267, 2007.
- [16] A.D. Hoover, V. Kouznetsova, 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.
- [17] J. Xiaoyi, 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.
- [18] C. Sinthanayothin, J.F. Boyce, H.L. Cook, and T.H. Williamson, "Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images," *Br. J. Ophthalmol.*, vol. 83, pp. 902-910, 1999.
- [19] J.V.B Soares, J.J.G. Leandro, R.M. Cesar, Jr., H.F. Jelinek, and M.J. Cree, "Retinal vessel segmentation using 2D Gabor wavelet and supervised classification," *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1214-1222, Sept. 2006.
- [20] C.A. Lapascu, D. Tegolo, E. Trucco, "FABC: retinal vessel segmentation using AdaBoost," *IEEE Trans. Med. Imag.*, vol. 14, no. pp. 1267-1274, Sept. 2010.
- [21] E. Ricci and R. Perfetti, "Retinal blood vessel segmentation using line operators and support vector classification," *IEEE Trans. Med. Imag.*, vol. 26, no. 10, pp. 1357-1365, Oct. 2007.
- [22] D. Marín, A. Aquino, M.E. Gegúndez-Arias, and J.M. Bravo, "A New Supervised Method for Blood Vessel Segmentation in Retinal Images by Using Gray-Level and Moment Invariants-Based Features," *IEEE Trans. Med. Imag.*, vol. 30, no. 1, Jan. 2011.
- [23] U.T.V. Nguyen, A. Bhuiyan, L.A.F. Park, and K. Ramamohanarao, "An effective retinal blood vessel segmentation method using multi-scale line detection," *Pattern Recognition*, vol. 46, no. 3, pp. 703-715, March 2013.
- [24] M.E. Martinez-Perez, A.D. Hughes, A.V. Stanton, S.A. Thom, A.A. Bharath, K.H. Parker, "Retinal blood vessel segmentation by means of scale-space analysis and region growing," *Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer-Verlag, London, UK, pp. 90-97, 1999.
- [25] A.M. Mendonca, A. Campilho, "Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction," *IEEE Trans. Med. Imag.*, vol. 25, pp. 1200-1213, 2006.

VI. APPENDIX

A. Performance Measures (continued)

To measure the performance of our algorithms, we computed four statistics for each of the 20 test images. These measures are accuracy, precision, sensitivity (recall), and specificity, and are consistent with the performance metrics used in the literature. Let, TP denote true positives (vessels correctly identified as vessel), TN denote true negatives (non-vessel pixels correctly identified as background), FP denote false positives (background pixels incorrectly identified as vessel), and FN denote false negatives (vessel pixels incorrectly identified as background). The performance measures are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

For the rest of this part, we will only consider the precision and the sensitivity. It is interesting to study this particular couple because there is usually a trade-off between these values. An algorithm with a high precision (few false positives) will also tend to have a low precision (small vessels not detected very well). On the other hand, an algorithm that detects more of the small vessels will tend to have also more false positives.

B. Weight selection

The way we chose the weights when we put the results of our 5 algorithms together can seem a bit arbitrary : we gave the same weight for the 5 likelihood maps. But as we saw earlier that the algorithms do not achieve the same performances, some being better for sensitivity, and others for precision so a different weighting may be better

In order to see what could be the set of weights that would give the highest average of precision and sensitivity, an option is to try many of them. Since these vectors lie in a subspace of dimension 4 (5 weights of sum 1), it means we have a very large space to explore. This is even more time consuming as we need to compute the average for all the 20 images of the training set for each set of parameters. In order to save some time, we do not need to compute the 20 outputs of the 5 algorithms and instead we can store these 100 images after computing them once. But we still need to run the thresholding algorithm (as well as the additional optional masking for the optical disk if needed) each time. This is why we only ran it for values of weights that were multiple of 0.1, and we did not include the values that would not give good results anyway, based on other results.

We did this search for both versions of our algorithm : the original hybrid one, giving higher sensitivity, and the modified hybrid, giving higher precision. The figure 13 gives the results for one of the searches. Each point corresponds to one set of weights and it is plotted according to the mean precision and sensitivity that were obtained on the 20 test images. The red circle shows the performance achieved for equal weights, and which corresponds to the specifications that were reported in the previous sections. The corresponding set of weights maximizing the average precision/sensitivity is plotted with a star. We can see that we get a better sensitivity and a better precision compared to the previous choice of equal weights.

The weights that we find are the following. For the original hybrid algorithm (resp. modified algorithm), we get :

- 0 (0) for the matched filtering algorithm
- 0.2 (0) for the neural network algorithm
- 0.5 (0.5) for the multi-scale line detector
- 0.1 (0.4) for the scale-space analysis
- 0.2 (0.1) for the morphological processing

For the original hybrid, this gives a mean sensitivity of 77.66% (compared to the previous 76.03%), a mean precision of 78.57% (compared to the previous 79.00%) and an average of 78.12% (compared to the previous 77.51%). For the modified

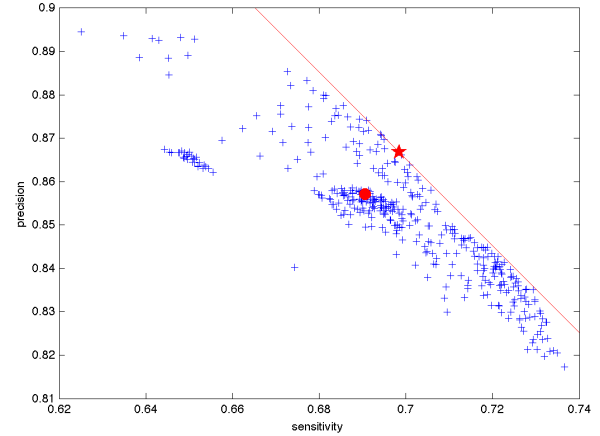


Fig. 13. Cloud of points corresponding to different sets of weights tested for the modified hybrid algorithm. 429 different sets of weights were tested for this algorithm.

hybrid, this gives a mean sensitivity of 69.84% (compared to the previous 69.06%), a mean precision of 86.69% (compared to the previous 85.71%) and an average of 78.26% (compared to the previous 77.39%).

First, we see that this improved set of weight performs strictly better than the set of equal weights that we used before, and the average of the sensitivity and the precision becomes even closer to the average for a human (79.14%).

Then, what is striking is that some of the weights are 0. This means that an algorithm can strictly decrease the performance if it is used, and instead it is better to use the other algorithms. This is the case of the matched filtering algorithm in both cases, and of the neural network algorithm in the second case.

Finally, this last result should be taken with a grain of salt. Indeed, since we already used our training set to train our supervised algorithms, we had to do this search of optimal parameters on the testing set. But it is also the set that we used to assess the performance of our algorithm. To assess the real performance, we should have used another testing set. The problem is that we had only two sets, the other images being from other databases so a bit different to treat (different resolution, etc., so we would have to adjust our algorithm for them).

This problem could be avoided with a larger database, and we could fine-tune the weights to get optimal results.

C. Member contributions

1) Karianne:

- Implementation of Neural Network-based method and associated codes
- Implementation of Multi-scale Line detection method
- Write-ups of sections of report on Neural network based method, multi-scale line detection method, abstract, and overview of the hybrid algorithm

2) *Sahinaz:*

- Implementation of matched filtering method and associated codes
- Implementation of mask for the modified hybrid algorithm
- Write-ups of sections of report on Introduction, Conclusion, Matched Filtering, and Modified Hybrid Algorithm.

3) *Jean-Baptiste:*

- Implementation of scale-space analysis method
- Implementation of morphological processing method
- Implementation of thresholding algorithm
- Implementation of the hybrid algorithms
- Write-ups of sections of report on Scale-space analysis method, Morphological processing method, Thresholding algorithm and most of the Results section.