Detecting Branching Structures using Local Gaussian Models

Li Wang, Abhir Bhalerao
Department of Computer Science
University of Warwick
Coventry CV4 7AL

November 26, 2001

Abstract

This report presents a method of detecting branching structure, such as blood vessels from retinal images, using a Gaussian Intensity model. Features are modelled with a Gaussian function parameterised by position, orientation and variance within some spatial window. Multiple features are modelled using a superposition of Gaussian models. A non-parametric classifier (k-means) is used to cluster components corresponding to each feature. Two different groups of images are used to test the methodology: artificial images and images of the human retina.

Contents

List of Figures

1 Introduction

Line, corner and branch detection in digital images is widely used in many computer vision applications such as image registration, object recognition and motion analysis. We are interested in using such methods for medical image segmentation (e.g. blood vessel detection in retinal images).

The detection and measurement of blood vessels can be used as part of the process of automated diagnosis of disease [1]. Thus a *reliable* method of detecting blood vessel structure in 2D or 3D tomographic images is needed. The intersections of the blood vessels create junctions or corners which are important dominant points for the whole structure since the information about a shape is concentrated at them.

Broadly speaking, corner detection techniques can be classified into two major categories. The first of these is boundary-based approaches that use pre-segmented contours (eg. [2]), while the second is based on the analysis of the raw gray-level data (eg. [3]).

In the case of boundary-based corner detection, the image is first pre-segmented. The Canny edge detector and zero-crossing methods are commonly used to extract the boundaries encountered [4]. Some methods then use a linking strategy to form chain codes. If a point at the object boundary makes discontinuous changes in direction or the curvature of the boundary is above some certain threshold, then that point is declared as a corner point [5]. Algorithms have been developed to detect corners along the boundary by measuring the eigenvalues of covariance metrics to locate the corner point [2]. Other researchers have extended the Hough Transform to find points of high local curvature from the edge pixels. This is done by accumulating positions of 'localisation' points in the Hough space, ie. for each edge pixel, a localisation point (similar to the centre of the radius of curvature) is computed by moving a certain distance away from the edge pixel orthogonal to the edge direction. Corners are then located by using the intersections of the localisation points [6]. The main weakness of all these approaches is that the performance of corner detection relies on the success or failure of the pre-segmentation step.

Gray-level corner detection methods can be divided into two groups: template based and geometry-based. A template based corner detector uses the similarity between a given template of a specific angle and the image data in a sub-window to find the corners [7]. Unfortunately, because of the complexity of all possible corner structures, it is impossible to design templates which can describe all situations and thus errors are inevitable. For geometry-based approaches, many of the algorithms use gradient information to locate corner positions. The product of gradient magnitude and the rate of change of gradient direction (curvature) with gradient magnitude are both used to measure the 'cornerness'. A corner is declared if the 'cornerness' is above certain threshold and the pixel is also nominally an edge point [8] [3]. Since these measures depend on second order differentials of the image, the algorithm is sensitive to noise. Furthermore, these approaches are only able to detect step-edge corners and do not address the problem of line junctions.

Some researchers have combined scale-space theory with the measuring of local curvature to detect junction points [1] [9], where the signal is smoothed by convolution with Gaussian kernels of different width, then the local curvatures are tracked through different scales to localise the corner point.

In this paper, we use a multiple resolution strategy differently exploiting from Wilson's work [10] and is specially aimed at finding branch structures of blood vessel in medical retinal images. A Gaussian intensity model is used to represent simple linear structures and the Multiresolution Fourier Transform (MFT) [10] [11] is also used to estimate parameters of the model.

The report is organised as follows: In section 2, the Gaussian model and the algorithm of feature estimation is presented. Section 3 is devoted to the feature reconstruction and hypothesis testing. Section 4 reviews the main steps in a general methodology for a scale space algorithm which is adapted to the junction detection problem. For a more detailed description about scale-space representation see [1] [9] [12] and [13]. Experimental results and discussion are presented in Section 5. Conclusions and ideas on how to extend the algorithm follow in Section 6.

2 Local Linear Feature Estimation

2.1 A Gaussian Intensity Feature Model

If an ideal linear feature is windowed by a smooth function $w()$, it can be regarded as a 2-dimensional Gaussian function [14], examples of which are shown in Figure 1.

The 2-dimensional Gaussian function can be written in the form:

$$G(\vec{x}) = (2\pi)^{-1/2}|C|^{-1/2}exp(-(\vec{x}-\vec{\mu})^T C^{-1}(\vec{x}-\vec{\mu})/2) \tag{1}$$

where $\vec{x}$ is the spatial co-ordinate, ie.

$$\vec{x} = (x, y)^T \tag{2}$$

and $\vec{\mu}$ is the mean vector and the covariance matrix $C = R^T V R$, where V is the diagonal matrix of variances, $V = \begin{pmatrix} \sigma_{x_1}^2 & 0 \\ 0 & \sigma_{y_1}^2 \end{pmatrix}$, R is the rotation matrix, $R = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$, θ is the angle to the x-axis. Figure 2 illustrates the meaning of the parameters used in the model.

Because of the complexity of real images, the model clearly can not be used to represent the whole image. However, it can be used on a small part of the image such as an image block of size $N \times N$. In another words, we can split the image into a set of blocks with different sizes, then try to fit the model in each region individually.

To estimate the parameters of our model, namely $[\vec{\mu}, \theta, {\sigma_{x_1}}^2, {\sigma_{y_1}}^2]$, a Windowed Fourier Transform is applied in each block before the feature extraction process:

$$Y(\vec{\omega}) = \sum w(\vec{x'} - \vec{x}) y(\vec{x}) e^{(-j\vec{x}'\vec{\omega})} \tag{3}$$

where $\vec{\omega}$ is the frequency co-ordinate,

$$\vec{\omega} = (u, v)^T \tag{4}$$

and $w(\vec{x'} - \vec{x})$ is a window function used to localise the signal. In this work, a cosine square function is used for $w()$.

Figure 3 shows the magnitude spectra of the windowed Fourier transform at different levels, ie. using a different scale window. For regions containing a single feature, the corresponding spectral energy lies orthogonal to the spatial orientation. For more complicated regions like branch points, there is a superposition of energies having a less clear DFT structure.

2.1.1 Feature Centroid estimation

If it is assumed that there is only a single feature in one block, the position of the feature, i.e. the distance of its centre from the origin with respect to the origin of the image plane, is linearly related to the phase spectrum [15]. Denoting the phase by $\phi(\vec{\omega})$, the DFT can be represented as

$$G(\vec{\omega}) = |G(\vec{\omega})| exp[-j\phi(\vec{\omega})] \tag{5}$$

where $G(\vec{\omega})$ is the Fourier transform of the spatial image. For a single linear feature, the phase spectrum, $\phi(\vec{\omega})$, can be modelled as

$$\phi(\vec{\omega}) = -\vec{\rho} \cdot \vec{\omega} \tag{6}$$

where $\vec{\rho}$ is the centroid vector and can be calculated by taking the partial derivatives of the phase in each of the directions. In the discrete case, by taking the difference in phase between neighbouring coefficients, the centroid vector of spatial position can be estimated as:

$$\rho_i = \frac{N}{2\pi} \sum_{\vec{\omega}} \hat{\phi}(\omega_i) \hat{\phi}(\omega_{i+1})^* \tag{7}$$

$$\rho_j = \frac{N}{2\pi} \sum_{\vec{\omega}} \hat{\phi}(\omega_j) \hat{\phi}(\omega_{j+1})^* \tag{8}$$

where $\frac{N}{2\pi}$ is the sampling interval.

2.1.2 Orientation

The MFT block which was modelled with Gaussian intensity profiles may be considered as having energy in an ellipse, centred on the origin. From Borisenko and Tarapovs' work [16], a moment of inertia tensor T can be calculated using the energy spectrum in place of mass,

$$T = \begin{pmatrix} T_{00} & T_{01} \\ T_{10} & T_{11} \end{pmatrix} \tag{9}$$

$$T = \sum \frac{\vec{\omega}\vec{\omega}^T \hat{E}(\vec{\omega})}{||\vec{\omega}||} \tag{10}$$

where N is the size of the block, the factor $||\vec{\omega}||$ is used to reduce the greater emphasis to energy further away from the origin. $\hat{E}(\vec{\omega})$ is the normalised energy at a given point (u, v) in the block, ie.

$$\hat{E}(\vec{\omega}) = \frac{|E(u, v)|}{E_{sum}} \tag{11}$$

where E_{sum} is the sum of the energy value in the block.

The major and minor axes of the ellipse are represented by the eigenvectors of the matrix. Since the orientation of maximum energy concentration, θ, is defined as the orientation of the major axis of the ellipse, it is indicated by the direction of the eigenvector, $\vec{e}_1$, which is associated with the largest eigenvalue, λ_1, i.e.

$$T\vec{e}_1 = \lambda_1 \vec{e}_1 \tag{12}$$

where $\vec{e}_1$ is defined as

$$\vec{e}_1 = (e_{10}, e_{11})^T \tag{13}$$

The orientation can then be obtained from

$$\hat{\theta} = arctan(\frac{e_{11}}{e_{10}}) \tag{14}$$

2.2 Multiple Linear Feature Estimation

If more than one linear feature is presented in a block, in order to perform the estimation, it is necessary to segment the spectrum into components corresponding to each feature. The complete spectrum of the region is modelled as the sum of the spectrum of each cluster:

$$G(\vec{\omega}) = |G(\vec{\omega})|exp[-j\phi(\vec{\omega})] = \sum_{l=1}^{K} |G_l(\vec{\omega})|exp[-j\phi_l(\vec{\omega})] \tag{15}$$

The use of the multiple linear feature model allows regions containing junction points or corners.

A partitioning method, K-means, is applied to separate the regions which are contributions from different features. K-means is an unsupervised, non-hierarchical clustering method, which is widely used in a number of image processing applications [17] [18]. It is an iterative scheme which attempts to both improve the estimation of the mean of each cluster, and re-classify each sample to the closest cluster. Firstly, it picks randomly selected initial seeds which are equal to the required number of clusters. Next, each component is examined and assigned to one of the clusters, depending on the minimum distance. The centroid's position of each cluster is recalculated at each iteration until no more components are changing class.

1. *Initialise $k = 2$ or $k = 3$ classes, choosing k pixels' coordinates as initial centroids at random from the image. Make sure that the pairwise distance between the k distance is large enough.*

2. *Using the phase gradient $\vec{\phi}_{i,j}$, convert each phase spectrum coefficient into a spatial vector $\vec{P}_{i,j}$. The sampling interval is $\frac{2\pi}{N}$ where N represents the size of the window. The spatial position is calculated by*

$$\vec{P}_{i,j} = \frac{N}{2\pi}\vec{\phi}_{i,j} \tag{16}$$

   Then, compare the distance between each pixel and each class centre and assign coefficient to the class to which it is closest.

3. *Recalculate the centroid for each class.*

4. *Repeat from step 2 until the movement of class centre is lower than a certain threshold t_m (we use $t_m = 2$ for 128×128 image).*

Different colours are used in Figure 4 to show the clustering approach of the K-means algorithm in given window which contain 2 and 3 features. After classifying the components belonging to each feature, the parameters of each feature can be individually estimated using equations (5)–(14).

After the parameters of Gaussian model corresponding to each feature have been estimated, the corner points, $q(\vec{x})$ can be localised as the intersection of each feature, denoted as $A_{l,m}$, i.e. $\forall \vec{x} \in A_l \cap A_m$ where $i \neq j$ and $A_{l,m} \in [A_1, A_2, A_3]$.

3 Reconstruction and Hypothesis Testing

3.1 Feature Reconstruction

If it is possible to synthesis the local spectrum using the estimated parameters, the feature model can be reconstructed. Some researchers [19] [20] have used the magnitude spectrum derived from the data and the estimated phase spectrum to generate the synthesised spectrum. In this paper, we use both the estimated phase and magnitude spectrum to reconstruct the model. Since the eigenvalues, denoted L_1, L_2 calculated previously, are inversely related to covariance matrix C', from

$$C' = R^T V' R \tag{17}$$

where

$$V' = \begin{pmatrix} \frac{2\pi}{N\sqrt{L_2}} & 0 \\ 0 & \frac{2\pi}{N\sqrt{L_1}} \end{pmatrix} \tag{18}$$

the synthesised spectrum, $\tilde{G}(\vec{\omega})$, can be generated using the model parameters,

$$\tilde{G}(\vec{\omega}) = |G'(\vec{\omega})| exp[-j(\phi'(\vec{\omega})] \tag{19}$$

where the estimated phase spectrum, $\phi'(\vec{\omega})$, is given by

$$\phi'(\vec{\omega}) = \phi'(\vec{\omega_i}) + \phi'(\vec{\omega_j}) \tag{20}$$

and

$$\phi'(\omega_i) = [\sum_{\vec{\omega}} \hat{\phi}(\omega_i)\hat{\phi}(\omega_{i+1})^*] \cdot \omega_i \tag{21}$$

$$\phi'(\omega_j) = [\sum_{\vec{\omega}} \hat{\phi}(\omega_j)\hat{\phi}(\omega_{j+1})^*] \cdot \omega_j \tag{22}$$

By taking an inverse DFT of $\tilde{G}(\vec{\omega}) \leftrightarrow Y'(\vec{x})$, the model reconstruction can be directly compared with the data, $Y(\vec{x})$ to test the goodness of fit.

3.2 Hypothesis testing

Once the parameters have been estimated, the accuracy of the hypothesis is checked and the most fit model should be used to represent the corresponding data. In this work, we apply a probabilistic approach to test the model fit. The probability that a synthesised data $\vec{Y'}$ fits the original data $\vec{Y}$ is denoted as $P(G_K|\vec{Y})$, where $G_K, K = 1, 2, 3$ represents the hypothesis model, ie.

$$G_k = G_K(\mu_k, C_k, k) \tag{23}$$

As noted in [21], there are several kinds of algorithms which could be used for the feature matching. The most commonly-used is the inner product method. Given the model, a likelihood of the data can be approximated by

$$P(\vec{Y}|G_K) = \frac{Y \cdot Y'}{||Y||||Y'||} \tag{24}$$

which is simply a normalised inner product of the data with the estimated model. It is clear that when the synthesised spectrum is exactly the same as real spectrum, the value of P will be maximum and equal to 1. The more accurate the reconstruction $P \to 1$ measuring how well the feature model fits the actual data is used in a given region.

The above method can be applied for $K = 1, 2$ and 3 features hypothesis in each block of the image, then using the estimated parameters, the correlation result, P_k, can be obtained respectively. If none is bigger than a certain threshold, denoted as t_r, the block is considered as not contain any likely model, G_k. Otherwise, the hypothesis with the maximum correlation results, P_{max}, chosen from $P(\vec{Y}|G_K)$, gives G_{max} as the best feature model for the region. Figure 5 is an overview of the algorithm.

4 Scale-Space representation

The basic idea behind scale-space representation is to separate out information at different scales [12]. Any image can be embedded in a one-parameter family representation which derived by convolving the original image $F(\vec{x})$ with Gaussian kernels of increasing variance t.

$$S(\vec{x};t) = F(\vec{x}) * \mathcal{G}(\vec{x};t) \tag{25}$$

where $\mathcal{G}(\vec{x};t)$ denotes the Gaussian kernel which can be written as

$$\mathcal{G}(\vec{x};t) = \frac{1}{2\pi t} e^{-\frac{x_1^2+x_2^2}{2t}} \tag{26}$$

Under this representation, for a 2D image, the multi-scale spatial derivatives can be defined as

$$S_{\vec{x}^n}(\vec{x};t) = F(\vec{x}) * \mathcal{G}_{\vec{x}^n}(\vec{x};t) \tag{27}$$

where $\mathcal{G}_{\vec{x}^n}$ denotes a derivative of some order n.

After the whole stack of images is obtained, we can then extract corners at different scales. As stated in Kitchen's work, the corner can be detected by measuring the curvature of level curves, i.e. the change of gradient direction along an edge contour. One of the special choice is to multiply the curvature by the gradient magnitude raised to the power of three [9], which is:

$$k = S_{x_2}^2 S_{x_1^2} - 2S_{x_1} S_{x_2} S_{x_1 x_2} + S_{x_1}^2 S_{x_2^2} \tag{28}$$

One implementation result of this algorithm is shown in Figure 6. Figure 6(a) shows an original retina image as well as the images which have been smoothed by convolution with Gaussian kernels of different widths. The result of 50 strongest corner response k^2 after applying equation (28) is presented in Figure 6(b).

5 Experimental Results and Discussion

The reconstruction results and decision algorithm were tested using several synthetic and real images. Figure 7, 8, 9 show the reconstruction results and the correlation values P_k of artificial images for each hypothesis. In figure 7(a), there is only one feature in the block. We can see that the maximum correlation value P_1 is derived from one feature hypothesis, which is the best fitted model. Similarly, on two and three features respectively in figure 8(a), 9(a), it can be seen that maximum correlation values are both from the best fitted hypothesis.

Results for each hypothesis on a real retinal image are illustrated at different scales in figure 10, 11, 12. In figure 10(b), 11(b), 12(b), one feature hypothesis is used at different scales, (eg. 64×64, 32×32 and 16×16). Similarly, results from two features and three features hypothesises are shown in figure 10(c)(d); 11(c)(d); 12(c)(d). The last image of figure 10, 11, 12 show the results from choosing the best fit model based on the correlation testing in each block.

A synthetic image of the basic components of a blood vessel, shown in figure 13, was used to test the algorithm at different scales: (64×64 and 32×32). The regions which contain a corner are emphasised based on the decision model, i.e if two or three features hypothesis was used in the block, the geometric intersect of these features could then be found. The size of the circles in Figure 13 reflects the detection scale. It can be seen that the regions including junction points are labelled accurately.

In figure 14, the same test algorithm as used in figure 13, is applied to "pick up" the regions which may contain a corner or junction point in a real retinal image. Comparing the result which was given in figure 6, we can conclude that using the Gaussian model a greater number of the junction points or corners are detected than by the method of curvature measuring. The curvature method fails to find many of the branches at small scales, although this could perhaps be improved by parameter tuning. Our estimator, however, is still affected by the noise and the complexity in the real image so some failures or false-positives occur in the retinal image as it does not attempt to combine information across scales. Also, it does not distinguish between true junctions and points of high curvature.

6 Conclusions and Further work

This work uses and extends the ideas previously presents by Davies, Wilson and Calway [19] [22]. Its main contribution is that we apply a superposition of Gaussian models and use a synthesised magnitude to reconstruct the data. This allows us to derive a likelihood, $P(\vec{Y}|G_K)$, to select a model G_k, which models a junction with $K = 1, 2, 3$ branches. By using an explicit Gaussian intensity model to represent linear features with some width, it gives us a simple representation of linear and branching structures like blood vessels in medical images. The model and estimation readily extends to 3D [23].

The algorithm has been tested on both synthetic (clean) images and real (noisy) images. We compared our results against a scale space scheme [1] [9]. The results, shown in Figure 13 and 14, demonstrate that the junction points can be correctly detected in an artificial image. However, due to the influence of the noise, localisation errors still exist for real data.

This approach is still in its initial stages. The next step is to consider some ways of simultaneous fitting super-posed models to reduce the effects of noise to get better accuracy of the localisation [14]. Another development would be generalising the model including a classifier in order to explicitly label the junction. Furthermore, a neighbourhood linking strategy to track vessels between the branch point could be employed to extract the entire tree structure. We model the data over a range of window sizes so a scale-selection strategy could be usefully applied to confirm/select a hypothesis [17].

Acknowledgements

This work is supported by the UK EPSRC Grant GR/M82899.

References

[1] M. E. Martinez-Perez, A. D. Hughes, A. V. Stanton, S. A. Thom, A. A. Bharath, and K. H. Parka, “Retinal blood vessel segmentation by means of scale-space analysis and region growing,” in *Proceedings of the International Conference on Image Processing*, 1999, vol. 2, pp. 173–176.

[2] D. M. Tsai, H. T. Hou, and H. J. Su, “Boundary-based corner detection using eigenvalues of covariance matrices,” *Pattern Recognition Letters*, vol. 20, pp. 31–40, 1999.

[3] Z. Zheng, H. Wang, and E. K. Teoh, “Analysis of gray level corner detection,” *Pattern Recognition Letters*, vol. 20, pp. 149–162, 1999.

[4] F. Mokhtarian and R. Sulomela, “Robust image corner detection through curvature scale space,” *IEEE Trans. on PAMI*, vol. 20, no. 12, pp. 1376–1381, 1998.

[5] H. C. Liu and M. D. Srinath, “Corner detection from chain-code,” *Pattern Recognition*, vol. 23, pp. 51–68, 1990.

[6] E. R. Davies, “Application of the generalised hough transform to corner detection,” *IEE Proceedings*, vol. 135, no. 1, pp. 49–54, 1988.

[7] R. Mehrotr, S. Nichani, and N. Ranganathan, “Corner detection,” *Pattern Recognition*, vol. 23(11), pp. 1223–1233, 1990.

[8] J. A. Noble, “Finding corners,” *Image and Vision Computing*, vol. 6(2), pp. 121–128, 1988.

[9] T. Lindeberg, “Junction detection with automatic selection of detection scales and localization scales,” in *Proc. 1st International Conference on Image Processing*, Nov. 1994, vol. 1, pp. 924–928.

[10] R. Wilson, A. D. Calway, E.R.S. Pearson, and A. Davies, “An introduction to the multiresolution fourier transform and its applications,” Tech. Rep. RR170, University of Warwick, UK, January 1992.

[11] A. H. Bhalerao, *Multiresolution Image Segmentation*, Ph.D. thesis, University of Warwick, U.K., 1991.

[12] T. Lindeberg, “Scale-space theory: A basic tool for analysing structures at different scales,” *Journal of Applied Statistics*, vol. 21, no. 2, pp. 225–270, 1994.

[13] T. Lindeberg, "Feature detection with automatic scale selection," *Int.J. of Computer Vision*, vol. 30, no. 2, 1998.

[14] A. Bhalerao and R. Wilson, "Estimating local and global image structure using a gaussian intensity model," *Medical Image Understanding and Analysis*, 2001.

[15] A. Papoulis, *Signal Analysis*, McGraw-Hill, New York, 1977.

[16] A. I. Borisenko and I. E. Tarapov, *Vector and Tensor Analysis with Applications*, Dover Publications, New York, 1979.

[17] A. Davies and R. Wilson, "Curve and corner extraction using the multiresolution fourier transform," Tech. Rep. RR 202, University of Warwick, UK, November 1991.

[18] B. Kovesi, J. M. Boucher, and S. Saoudi, "Stochastic k-means algorithm for vector quantization," *Pattern Recognition Letters*, vol. 22, pp. 603–610, 2001.

[19] A. Davies, *Image Feature Analysis using the Multiresolution Fourier Transform*, Ph.D. thesis, University of Warwick, UK, August 1993.

[20] C. T. Li, *Unsupervised Texture Segmentation Using Multiresolution Markov Random Fields*, Ph.D. thesis, University of Warwick, U.K, 1998.

[21] P. Smith, D. Sinclair, R. Cipolla, and K. Wood, "Effective corner matching," in *British Machine Vision Conference*, U.K., 1998.

[22] R. Wilson, A. D. Calway, and E. R. S. Pearson, "A generalized wavelet transform for fourier analysis: the multiresolution fourier transform and its application to image and audio signal analysis," *IEEE Tran. IT, Special Issue on Wavelet Representations*, 1992.

[23] A. Bhalerao and R. Wilson, "A fourier approach to 3d local feature estimation from volume data," in *British Machine Vision Conference(BMVC 2001)*, 2001, vol. 2, pp. 461–470.

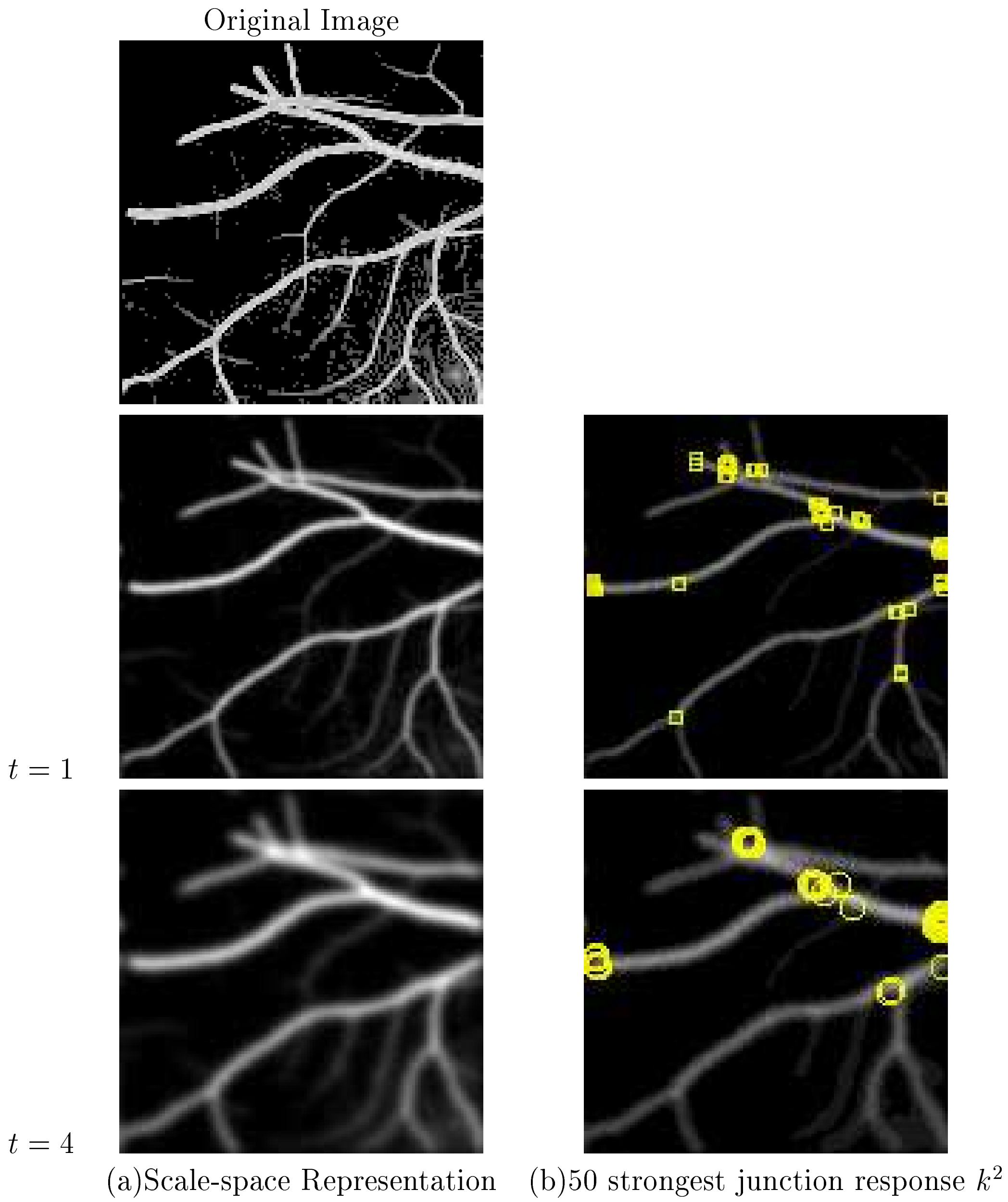


Figure 1: Scale space representation and Junction response at different scale(The size of the circles reflect the detection scales)

(a) Original Image

(b) Windowed Model

(c) Original Image

(d) Windowed Model

Figure 2: Examples of a Gaussian Intensity Model for linear features

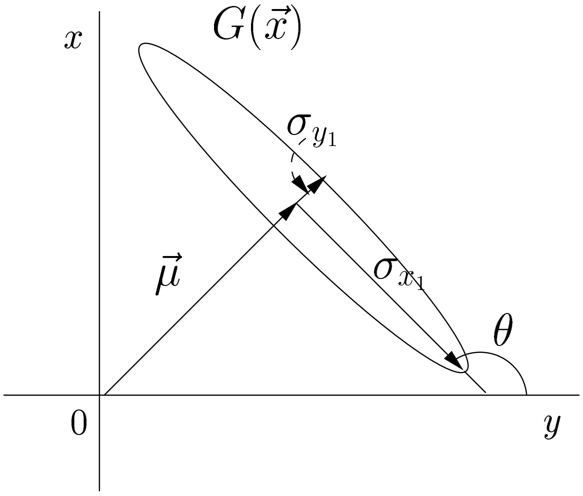
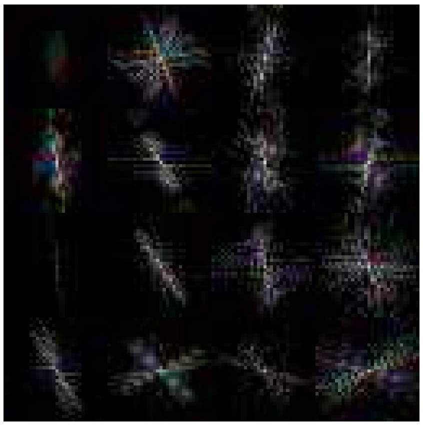


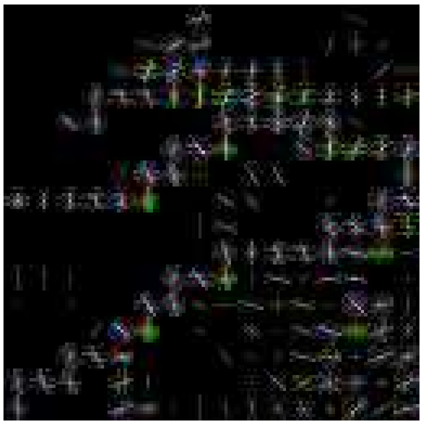
Figure 3: Parameters of Gaussian Model $G(\vec{x})$

(a) Original Image



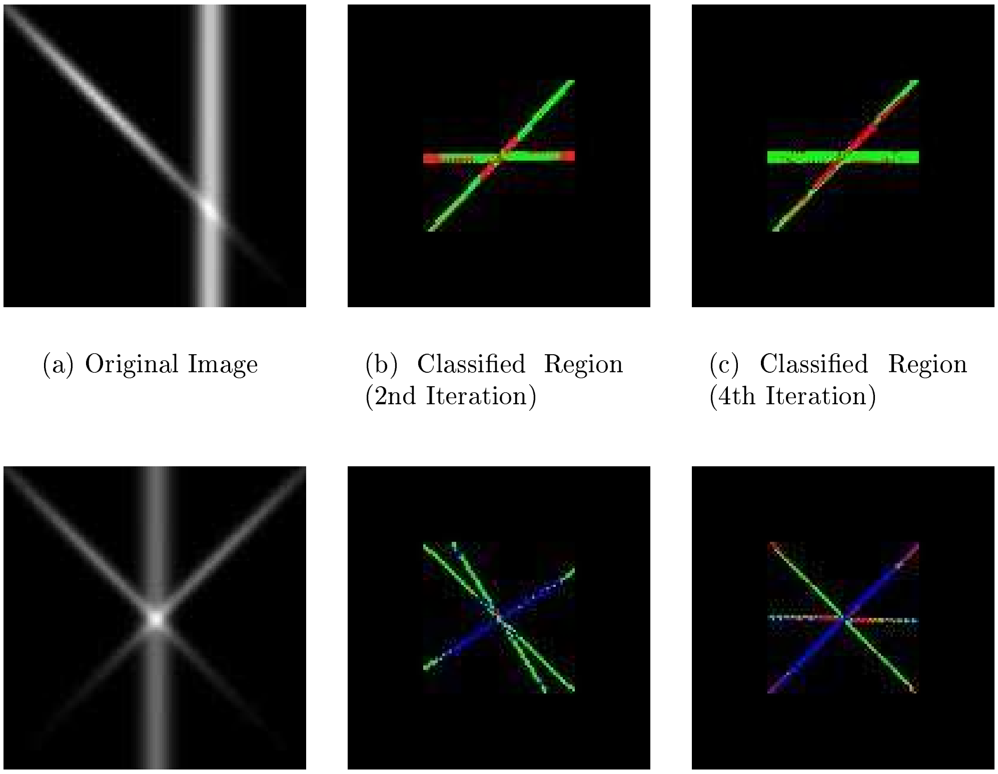
(b) Windowed Fourier transform for $N = 64$ block sizes

(c) Windowed Fourier transform for $N = 32$ block sizes



(d) Windowed Fourier transform for $N = 16$ block sizes

Figure 4: Windowed Fourier transform of a example retinal image (a) showing DFT Magnitude Spectra at different scales $(b), (c), (d)$.



(a) Original Image

(b) Classified Region (2nd Iteration)

(c) Classified Region (4th Iteration)

(d) Original Image

(e) Classified Region (2nd Iteration)

(f) Classified Region (4th Iteration)

Figure 5: Clustering Approach using K-means (Different colour represents the components belong to each feature in frequency domain.)

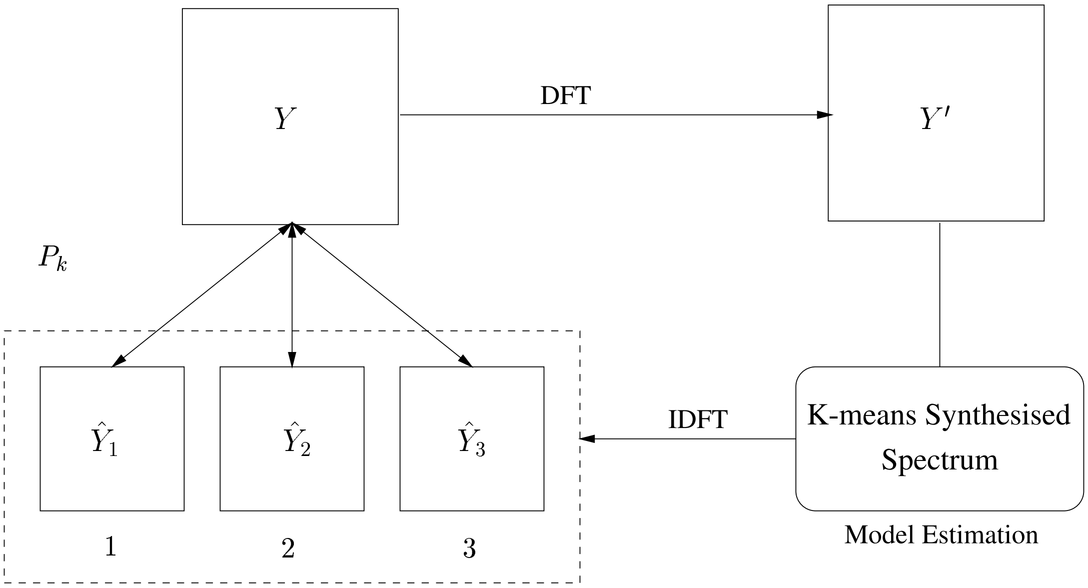
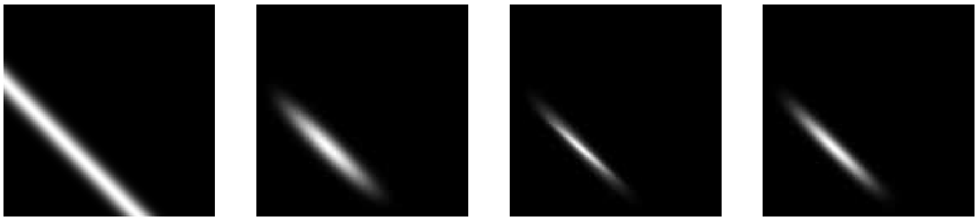


Figure 6: Hypothesis testing algorithm



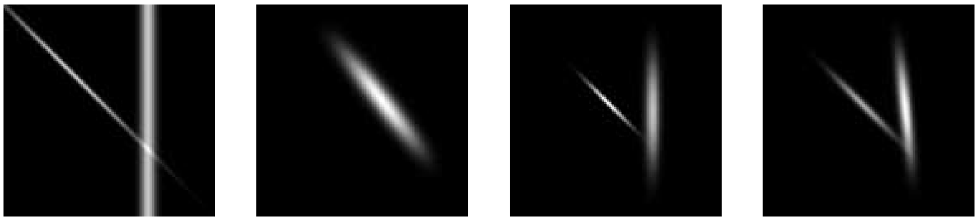
(a) Original Image

(b) 1 feature hypothesis

(c) 2 feature hypothesis

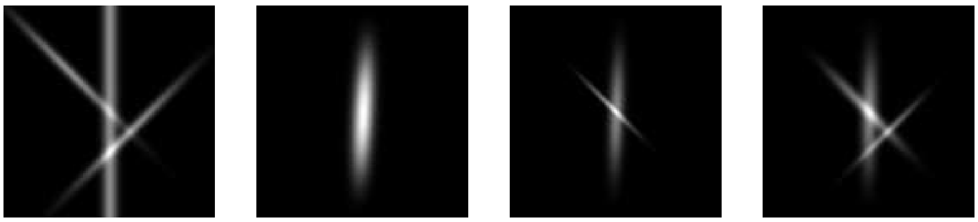
(d) 3 feature hypothesis

Figure 7: Estimation result for each hypothesis $P_1 = 0.97, P_2 = 0.89, P_3 = 0.95$



(a) Original Image (b) 1 feature hypothesis (c) 2 feature hypothesis (d) 3 feature hypothesis

Figure 8: Estimation result for each hypothesis $P_1 = 0.22, P_2 = 0.97, P_3 = 0.90$

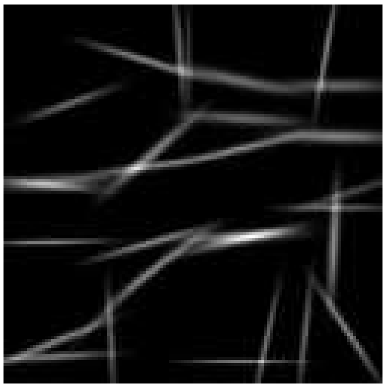


(a) Original Image (b) 1 feature hypothesis (c) 2 feature hypothesis (d) 3 feature hypothesis

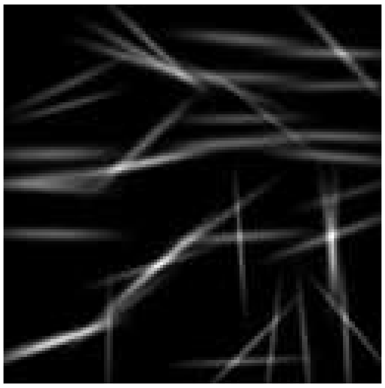
Figure 9: Estimation result for each hypothesis $P_1 = 0.70, P_2 = 0.90, P_3 = 0.97$

(a) Original Image

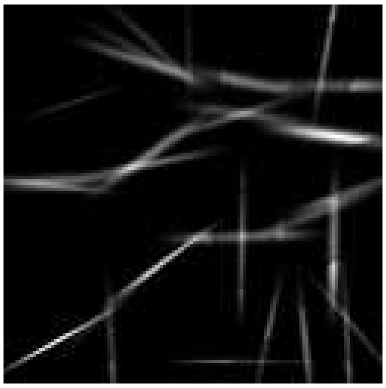
(b) 1 feature hypothesis



(c) 2 feature hypothesis

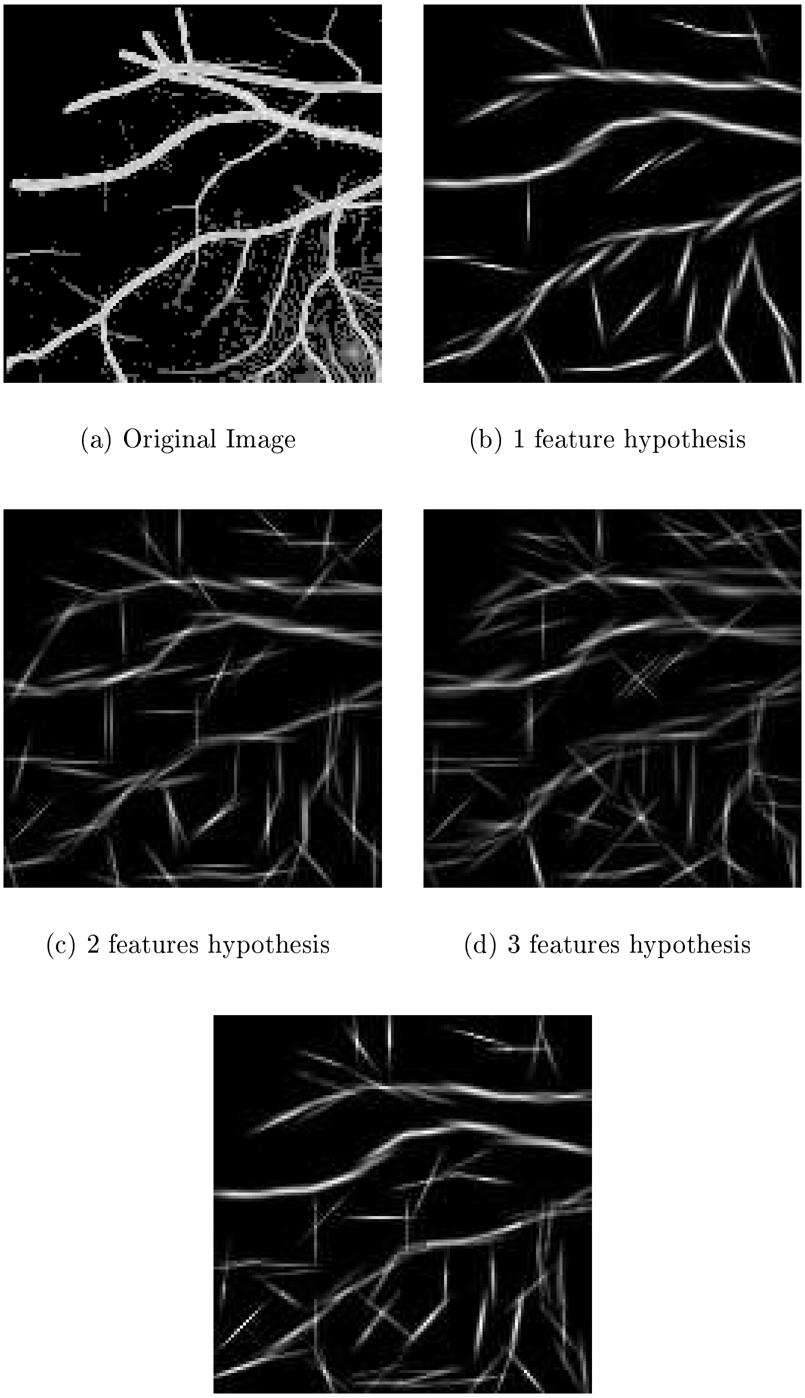


(d) 3 feature hypothesis



(e) best fitted hypothesis

Figure 10: Hypothesis testing in a real retinal image for $N = 64$ block sizes



(a) Original Image

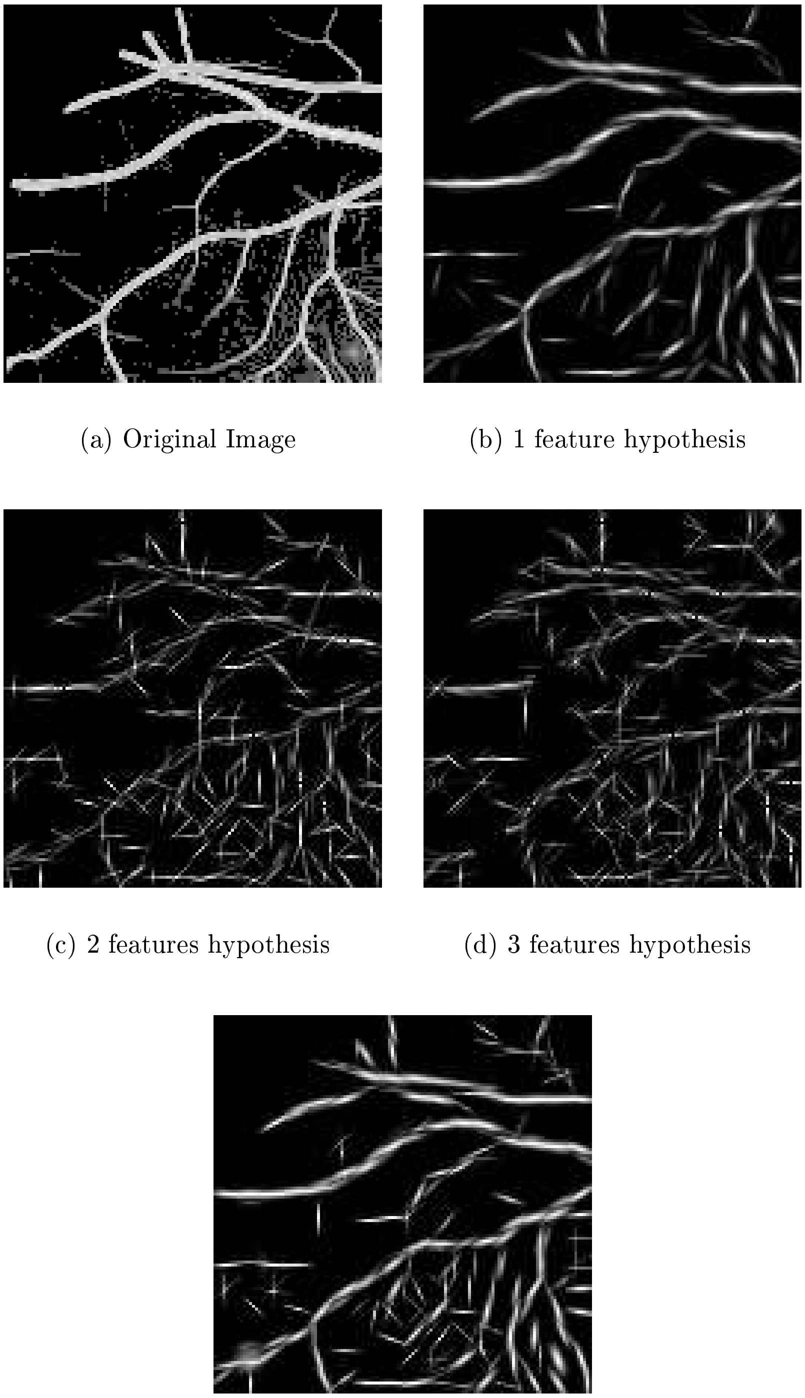
(b) 1 feature hypothesis

(c) 2 features hypothesis

(d) 3 features hypothesis

(e) best fitted hypothesis

Figure 11: Hypothesis testing in a real retinal image for $N = 32$ block sizes



(a) Original Image

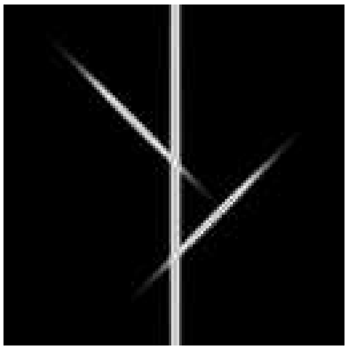
(b) 1 feature hypothesis

(c) 2 features hypothesis

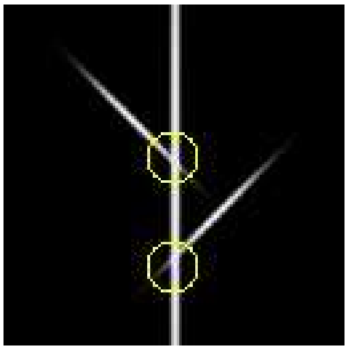
(d) 3 features hypothesis

(e) best fitted hypothesis

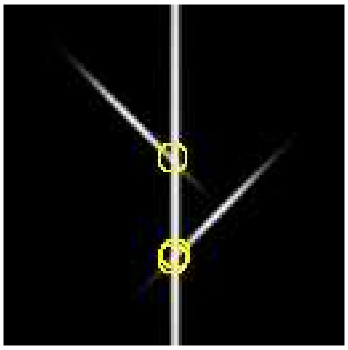
Figure 12: Hypothesis testing in a real retinal image for $N = 16$ block sizes



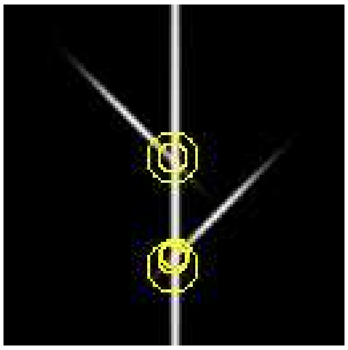
(a) Original Image



(b) Label the branch point for $N = 64$ block sizes

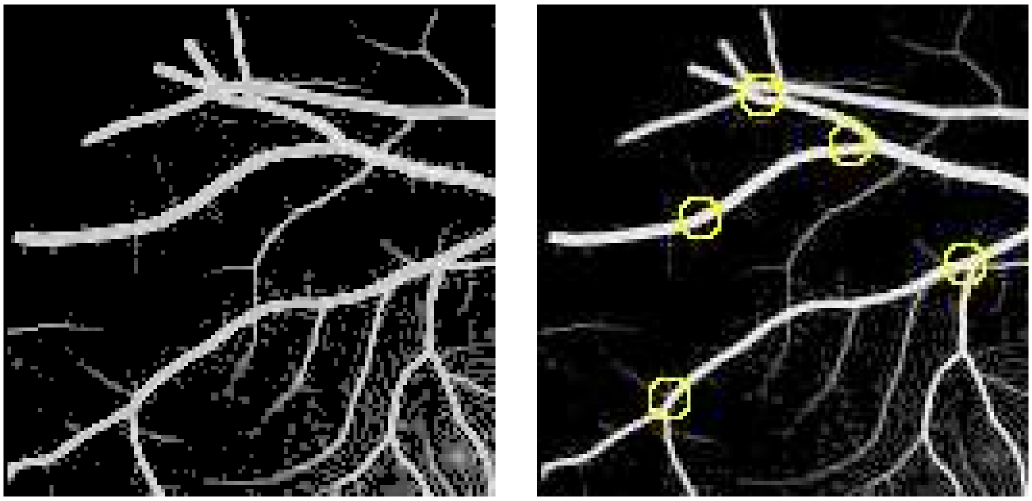


(c) Label the branch point for $N = 32$ block sizes



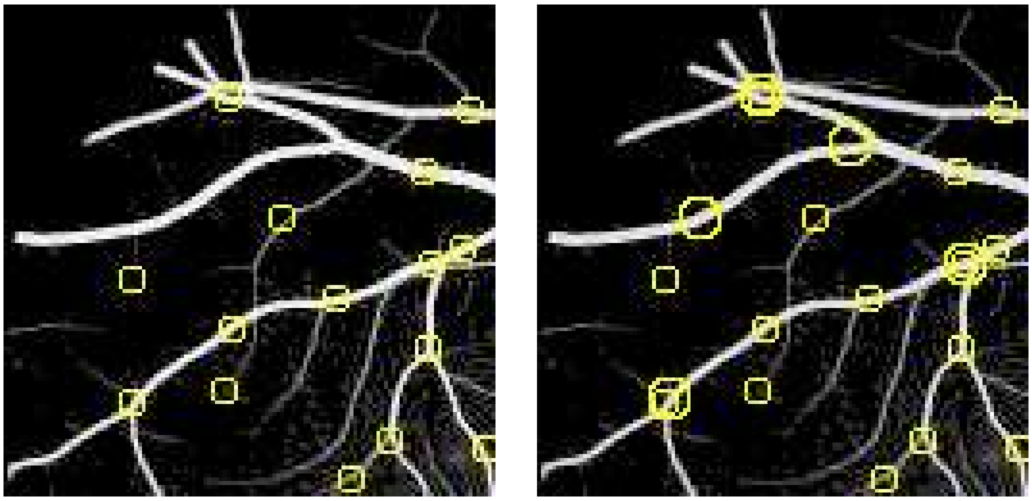
(d) Branch point detect through the different scale

Figure 13: Label the branch point in different scale of an synthetic image (the size of the circles reflect the detection scales)



(a) Original Image

(b) Label the branch point for $N = 64$ block sizes



(c) Label the branch point for $N = 32$ block sizes

(d) Branch point detect through the different scale

Figure 14: Label the branch point in different scale of retina image (the size of the circles reflect the detection scales)