

行政院國家科學委員會專題研究計畫 成果報告

果蠅腦共軛焦顯微鏡影像分析及標準腦之研究(2/2)

計畫類別：個別型計畫

計畫編號：NSC93-2213-E-009-037-

執行期間：93年08月01日至94年09月30日

執行單位：國立交通大學資訊科學學系(所)

計畫主持人：荊宇泰

共同主持人：莊榮宏，江安世

報告類型：完整報告

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中 華 民 國 95 年 7 月 31 日

ABSTRACT

In this report, we present some results obtained during the execution of the project. The results include a software system, the IPToolBox, two examples that solved by using the software system, and the on going work. The IPToolBox consists of many image processing algorithms reported in the literatures. The two examples are respectively segmentation of the mushroom bodies using the GVF snake method and neuron tracing using the Hessian Matrix and graph search method. The on going work is to develop algorithms for fine structure segmentation including cell counting and improving the neuron tracing algorithm.

Keywords: active contour, snake, gradient vector flow, segmentation, confocal microscopic image, chain code encoding, Hessian Matrix, scale space, graph algorithm, shortest path.

1. FUNDAMENTAL IMAGE PROCESSING ALGORITHMS AND IPTOOLBOX

IPToolBox is a software system developed in my lab while the project was executed in the past years. IPToolBox, that has nice user interface, consists of most of the image processing algorithms reported in the literatures. We name some for examples.

File I/O: The system can read most of the image format. Not only the simple format such as BMP or JPEG can be recognized, the complex format such as DICOM (used in medical images) and some uncompressed TIFF format (used in confocal microscopy and multi-photon microscopy) can be read.

Filter: There are operations form image enhancement such as the mean filter and median filter that remove noises. The edge detection algorithms such as Sobel filtering and Canny edge detector are implemented. One of the important filter we implemented was the matched filter. The matched filter has been proved a useful tool in finding the vessel, neuron fiber, and neuron cells.

Binary Operation: This set of operations includes simple operation such as thresholding to the more complicated operations such as morphological operation, connectivity analysis, and water shed operations.

Geometric Operation: Some computational geometry algorithms such as convex hull, Delaunay Triangulation, and Voronoi Diagram are included.

FFT: Fourier Transform from 1D to 4D are implemented. Different filters, such as the low-pass filter, high-pass filter, and the band-pass filters are implemented.

Statistic: Simple texture analysis methods, such as variance, standard variation, and coefficient of variance, were implemented.

Edit: Very often editing images are needed. We provide user interface to edit image such as assigning a specification region a specific intensity.

Snake: A standard snake method and the GVF snake were implemented. We found this is very useful in finding the boundary of region of interest. We also found that this could be useful in finding the neuron fibers.

3D: Marching cube volume rendering algorithm is implemented. We deal with 3D volume data most of the time. This tool helps to present the result obtained from the other tools in the IPToolBox.

There are many other algorithms were developed and some of these are under development.

2. SEGMENTATION OF INSECT BRAIN NEUROPOLES IN THE CONFOCAL MICROSCOPIC IMAGES USING GVF SNAKE

In this section, we present a result obtained by using the IPToolBox. We use the Gradient Vector Flow (GVF) snake to segment the mushroom body in an insect brain of confocal microscopic images. The GVF snake has advantage of capturing local cavity. This also means that the GVF snake is sensitive to noises and needs good edge map. We employed the texture analysis and Canny edge detector to calculate a good edge map before the GVF was calculated. In this article, a new method based on the global shape of a snake to define a halting condition from moving a snake is presented.

2.1 INTRODUCTION

Confocal microscopy is widely used in neuroscience research because of its unique features in removing out-of-focus signals and volume visualization. This tool is particularly useful in studying brain neural networks projecting to a large space. Using the newly developed FocusClearTM reagent to clear biological tissues together with confocal microscopy, it is now possible to image the whole insect brain of 500 micron thick without physical sectioning [1]. Mushroom bodies are neuropil regions in the insect brain that have been implicated in associative learning and memory. Recently, a standard *Drosophila* brain averaging from several individuals has been established for comparing brain volumes between different wild types and mutants [2]. The so-called standard brain provides a common 3D framework as the first step for building a brain database allowing deposition of gene expression patterns from different laboratories. Consequently, averaged neuropils for detail analysis of brain's function are urgently needed. Automatic segmentation of neuropils is the one of the most fundamental task to establish averaged neuropils.

In this section, we present an almost automatic method to segment the mushroom body in a stack of confocal microscopic volume images of cockroach brain. Difficulty in this task is mostly due to the vague boundary between the mushroom body and the background. The traditional low-level techniques such as intensity thresholding or gradient based edge detecting methods which only utilize local information could make mistakes during creating the boundary. To overcome this problem, the active contour model, known as "snake", which has been introduced by Kass, Witkin and Terzopoulos in 1987 [3], integrates the image feature extraction and representation phase into a single process. A snake is an energy-minimizing curve defined within an image domain that moves under the influences of the internal forces coming from within the curve itself and the external forces computed from image data. The internal and external forces are defined such that the snake will be attracted to an object boundary or other features within an image. Snakes are widely used in many applications, including edge detection [3], shape modeling [4], segmentation [5], and motion tracking [6]. In this study, we adopted the active contour model using the gradient vector flow (GVF) [7] as the external force to segment the mushroom body in the cockroach brain image.

As mentioned previously, the major difficulty to segment the mushroom body is due to the vague boundary between the mushroom body and the background. Vague boundary implies weak external force to attract the snake. To enhance the boundary of the mushroom body, we proposed to first apply the texture analysis to the image. We then applied the Canny edge detection method to calculate the boundary between the mushroom body and background. The gradient vector flow is finally calculated using the edge map obtained from the Canny edge detector. In the next section, we first state the proposed method. We then briefly describe the techniques involved in each step of the method. We also present a new halting condition to test if the snake converged. In Section 2.3, we demonstrate the result obtained using the proposed method. The conclusions are in Section 2.4.

2.2 METHOD

In this section, we first outline each step in the proposed method. We then describe the methods employed in each step.

The outline of the steps is stated in the following.

1. Texture analysis: Since the mushroom body has different texture from the background, image after texture analysis enhances the mushroom body as well as the boundary between the mushroom body and the background. The mushroom body in the image after texture analysis has low intensity in the image. Thus, the intensity thresholding can then be applied to segment the mushroom body.
2. Edge detection: Canny edge detection algorithm is then applied. The edges of the mushroom body have strong response. Applying the intensity thresholding can eliminate most of the noises. The resulted image is the edge map.
3. Calculate the GVF based on the edge map obtained.
4. Applied the active contour method to segment the mushroom body.

2.1 TEXTURE ANALYSIS

Texture analysis helps to divide an image into regions according to the textures of the regions. There are many ways to define the texture in a region. One of the ways is the “coefficient of variance”, CV , that is defined as the following equation.

$$CV = \frac{\sqrt{\sigma}}{\mu}. \quad (1)$$

In Eq. (1),

$$\mu = \frac{\sum_{k=1}^N f(k)}{N}, \quad (2)$$

and

$$\sigma = \frac{\sum_{k=1}^N f(k) - \mu}{N}. \quad (3)$$

Note that, a large CV means large deviation in the a region $f(k)$. On the other hand, a region has small CV implies that the region $f(x)$ has uniform distributed intensity. In our application, the mushroom body has small deviation of the intensity than the background, thus the coefficient of variance analysis produces small CV in the mushroom body.

2.2 CANNY EDGE DETECTOR

Canny edge algorithm was proposed by Canny in 1986 [8]. Three issues regarding an edge detector were addressed:

1. Correct Detection: The truth positive and the truth negative ratio should be high. And the false positive and the false negative ratio should also be high.
2. Good Localization: Points detected by the operator should be as close to the real edge as possible.
3. Good Response: Only one response to a single edge. If too many edges are detected, then obviously some of them must be false edges.

Based on the addressed issues, designing an optimal filter was modeled as an optimization problem. Since the optimal solution was difficult to solve, Canny used an approximation method to design a filter that is close to the optimal filter. The function is the derivative of a Gaussian as shown in Eq. (4).

$$g'(x) = -\frac{x}{\sigma^2} e^{-x^2/2\sigma^2} \quad (4)$$

The next step in the Canny edge detector is to find thin edges by using a “non-maximum suppression” technique. In this step, pixels that are not local maximal are removed. The last step is the hysteresis thresholding. Hysteresis thresholding sets a high threshold T_h and a low threshold T_l . Any pixel in the image that has a value greater than T_h is presumed to be an edge pixel. Then any pixel, that is connected to an edge pixel and has a value greater than T_l , is selected to be an edge pixel.

2.3 ACTIVE CONTOUR METHOD AND GRADIENT VECTOR FLOW

An active contour model is a mapping: $\Omega = [0, 1] \rightarrow R^2$, and $S \rightarrow v(s) = (x(s), y(s))$. We define the model as a space of admissible deformation A and a function E to be minimized. This function E is written in the form

$$E = \int_0^1 E_{int}(v(s)) + E_{ext}(v(s)) ds, \quad (5)$$

where E_{int} represents the internal energy of a snake due to bending, and E_{ext} the external force that is derived from image features. E_{int} serves to impose smoothness constraint on the snake. E_{ext} pushes or pulls the snake toward desired features such as edges. As internal and external energy are formulated, we deform the snake by minimizing Eq. (5). The internal energy can be written as:

$$E_{int} = (\alpha(s)|v'(s)|^2 + \beta(s)|v''(s)|^2) / 2 \quad (6)$$

where $\alpha(s)$ and $\beta(s)$ are weighting parameters that control the snake's elasticity and rigidity, $v'(s)$ and $v''(s)$ denote the first and second derivatives of the curve v with respect to s . The energy function that attracts snake to salient features in images is the external energy. External energy is derived from the image.

Traditional active contour model suffers from the following three key difficulties: First, the active contour model has narrow capture range. Second, active contour model lacks the ability to handle boundary concavities. Third, snake has the tendency to become stagnant at a nearby local energy minimum. For the third problem, the main difficulty is due to noise. Preprocessing of the image is necessary. Texture analysis mentioned previously is employed to reduce noise. However, if the noise is too strong, it still requires user assists to manually adjust the snake to escape from the local minimum. For the first and second problem, Xu and Prince proposed a method called the Gradient Vector Flow (GVF) in 1998 [7] to solve the problems.

The GVF field is defined to be a vector field $p(x,y) = [g(x,y), h(x,y)]$ that minimize the energy

$$\xi = \iint \mu(g_x^2 + g_y^2 + h_x^2 + h_y^2) + |\nabla f|^2 |p - \nabla f|^2 dx dy. \quad (7)$$

When $|\nabla f|$ is small, the energy is dominated by the sum of the squares of the partial derivatives of the vector field, yielding a slowly varying field. On the other hand, when $|\nabla f|$ is large, the second term dominates the integrand, and is minimized by setting $p = \nabla f$. This produces the desired effect of keeping p nearly equal to the gradient of the edge map when it is large, but forcing the field to be slowly-varying in homogeneous regions. Using calculus of variations [9], it can be shown that the GVF field can be found by solving the following Euler equations:

$$\mu \nabla^2 g - (g - f_x)(f_x^2 + f_y^2) = 0, \quad (8)$$

and

$$\mu \nabla^2 h - (h - f_x)(f_x^2 + f_y^2) = 0, \quad (9)$$

where ∇^2 is the Laplacian operator.

Using GVF to server as external forces, we can move the snake to minimize the energy of the contour shown in Eq.(5). The solution can be solved by using the greedy approach [10] or the finite difference method [3]. We used the finite difference method in our implementation.

2.4 IMPLEMENTATION DETAILS

Two issues are discussed in this section, the reparameterization of a contour and the halting condition of a moving snake.

When a snake deforms, the distances between consecutive vertices on the snake change. It is required to resample the snake along its path [11]. We resample the snake according to a user pre-defined parameter l_{des} . The distance between the neighboring vertices is kept between $0.5l_{des}$ and $1.5l_{des}$. When the distance is less than $0.5l_{des}$, we merge two adjacent vertices into a single vertex. If the distance is larger than $1.5l_{des}$, we insert a vertex between the vertices.

The next issue is the halting condition for a snake. One possible condition is to set a limit for the number of iterations. Another approach is to monitor the improvement (change of energy) between two consecutive iterations. The snake stops moving if the improvement is small enough. Due to the following reason, we present a new method to determine the halting condition.

Since the external force is provided by the GVF, the vertices on the snake move toward “valley” of the external force field when a snake converges. Since the bottom of the valley is not at the same height, some points in the valley still move toward nearby local minimal even when the snake converge to the boundary. In such cases, in order to reduce the energy of the snake, vertices of a snake will have a strong tendency to pile up in the “deepest pocket” in the valley. To solve this problem, we define the stopping criterion as the similarity of the shapes of contours. If the similarity between two snakes is within a threshold, we consider these two snakes to be the same and confirm that the snake is converged. To evaluate the similarity of the shapes of contours, we adopt a modified chain code curve encoding method [12] and the longest common subsequence techniques[13].

Chain code encoding method is first proposed by Freeman [12]. The concept is to divide a curve into a fixed length segment and encode the curve according to the direction of segments along the curve. In general, encoding method has 4-connectivity (Fig. 1) and 8-connectivity (Fig. 2). We subdivide the plane into 4 (8 respectively) regions in 4-connectivity (8-connectivity respectively) for encoding. Given a starting vertex on a curve, we compute the vector of adjacent vertices along the curve. Then we encode the curve into a bit string according to the directions of the vectors. We call the bit in a bit-string as *direction bit*. Fig. 3 shows an example in a 8-connectivity encoding of a curve.

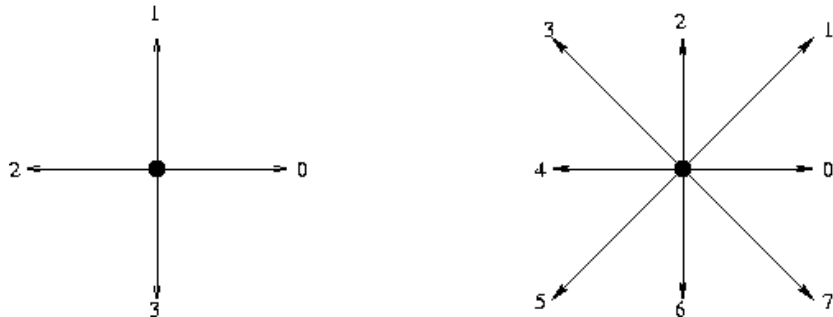


Fig. 1 4-connectivity and 8-connectivity for chain code encoding

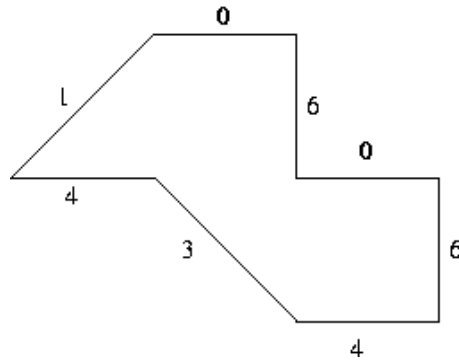


Fig. 2 An example of 8-connectivity chain code encoding in clockwise order

When two curves are encoded into two bit-strings, we find the longest common subsequence (LCS) of these strings by the dynamic programming technique [13]. We define a similarity function as follows:

$$S(v1, v2) = (Len(v1) + Len(v2) - 2 \times LCS(v1, v2)) / (Len(v1) + Len(v2)). \quad (10)$$

In Eq. (10), S denotes the similarity function. $v1$ and $v2$ are curves to be matched. $Len(v)$ denotes the length of the curve v , and $LCS(v1, v2)$ denotes the length of the longest common subsequence of $v1$ and $v2$. Note that $0 \leq S(v1, v2) \leq 1$. If two curves are identical, $S(v1, v2) = 0$. A small S implies two curves are identical. We set a threshold T_s that if S is smaller than T_s , we consider that the snake has been converged.

A standard chain code has a problem. In figure 4, vector **A** and vector **B** are in different encoding regions. Thus vector **A** and **B** is considered pointing in different direction and are encoded into distinct direction bits. But if the angle in vector **A** and **B** is small, they should be considered pointing in the same direction. Thus, we made a modification in defining the cost function in the longest common subsequence algorithm to solve this problem. In the original longest common subsequence algorithm, a string bit represents vector's direction. Now we let each string bit to be the vector itself. The comparison of string bits is replaced by the inner product of the vectors. If the value of the inner product is larger than a given threshold value θ , we consider these two string bits are identical. According to our experiment, the modified method provides a reliable stopping criterion.

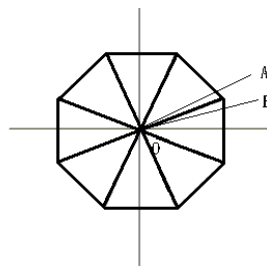


Fig. 3 Vector **A** and **B** are similar. But they are considered very different in chain code encoding.

2.5 USER INTERFACE

Snakes can be trapped into local minimal or wrongly converged to a boundary due to noises or ill initialization. When this happens, user assist is used to help the snake to escape from the trapped configuration. The other situation that needs a user assist occurs when the topology of contour

changes. When a contour splits into two contours or two contours merge into one contour, we need user assists to manually modify the snakes.

3. EXPERIMENTAL RESULT

We present the experimental results by using two images. The first image is shown in Fig. 4. In this figure, the original image is in (a). In (b), the image after texture analysis is shown. (c) shows the image after intensity thresholding of the image (b). Images (d) and (e) are obtained by first applying the Canny edge detector and the intensity thresholding of the image after Canny edge detection. The initial contour is shown in (f). (g) is the image consists of the initial contour and the calculated GVF. Images in (h) and (i) are the converged snakes for both the left and right mushroom bodies. The converged snakes for the mushroom bodies superimposed in the original images are shown in (j) and (k). The second image has worse contrast than the previous one. We show that the proposed method can still segment the mushroom body well. In the first row in Fig. 5, there are the original image, the image after texture analysis, and the image after intensity thresholding of the resulted image after texture analysis. The second row consists of the images that are obtained by the Canny edge detection, intensity thresholding of the resulted image after Canny edge detection, and the calculated GVF as well as the initial contour. In the third row, there are two images showing the converged snakes. Even the contrast is much worse than the previous image, the proposed method can still well segment the mushroom body.

4. DISCUSSION

We presented a method to segment the mushroom body in the confocal microscopic image. The proposed method employed the texture analysis and the Canny edge detector method to compute the edge map. The active contour method using GVF as the external force was then used to segment the mushroom body. The proposed method worked well even when the boundary of the mushroom body is not very sharp.

When we apply the method to segment the 3D mushroom body in a stack of volume images, the final converged contour in one slice serves as the initial contour in the next slice. The user assists are needed when the topology of the contour changes or there are unexpected noises. This method helps to reduce the required of human intervene to process a whole stack of volume data.

The disadvantage of this approach is the computing time required in the preprocess step. To compute the GVF of a 256 by 256 image needs 5 seconds in an AMD 1.5G XP CPU PC. However, since it generally takes less than an hour to process a set of volume data, and it does not need any human intervene in the preprocessing step, the proposed method can save many man power in the task of segmentation of the mushroom body.

(a)

(b)

(c)

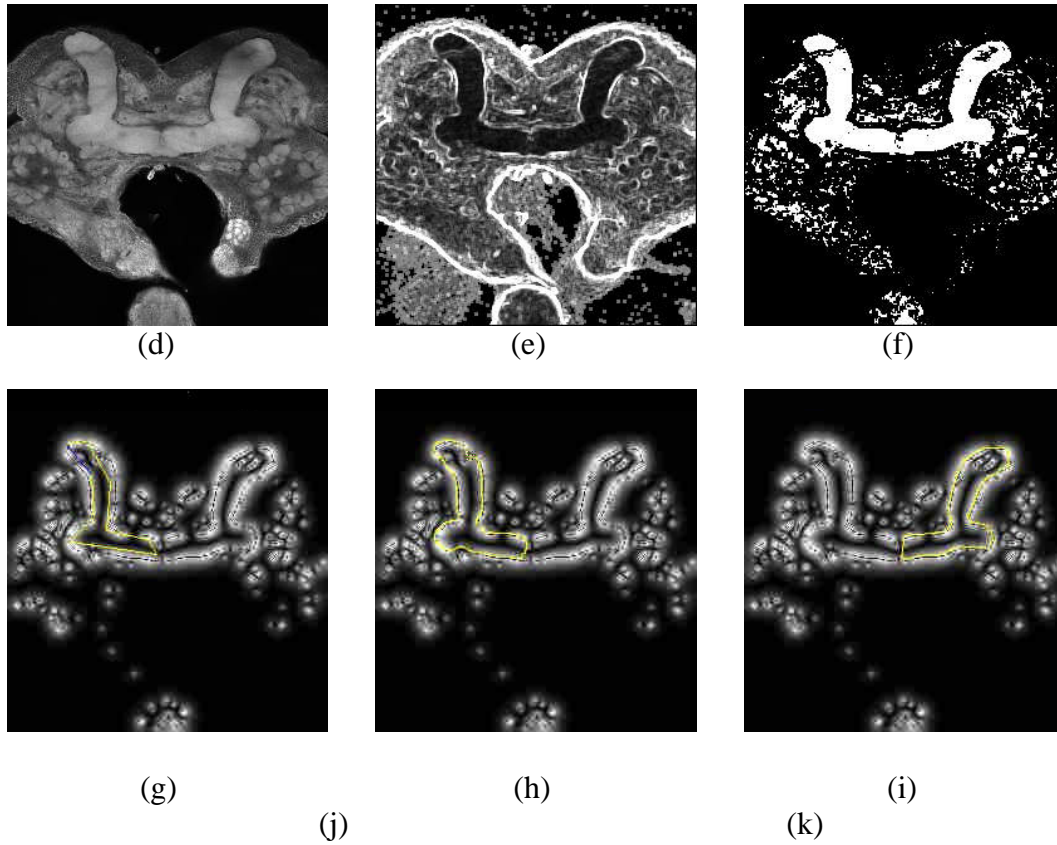
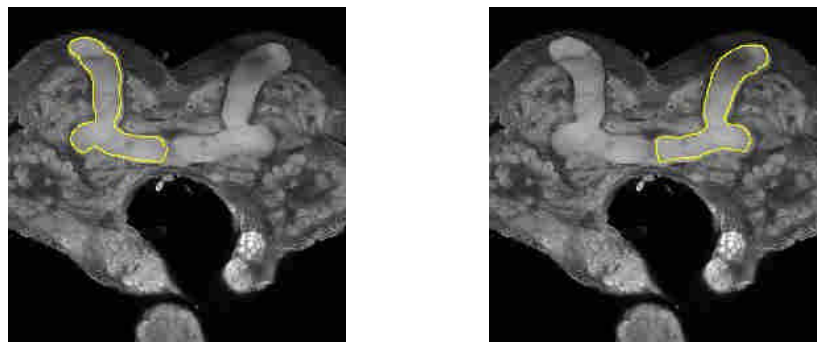


Fig. 4 The first image. (a) is the original image. (b) and (c) The image after texture analysis and intensity thresholding. (d) and (e) show the images after Canny edge detector and intensity thresholding to obtain the edge map. (f) and (g) shows the initial contour. (h) and (i) are the converged snakes for the left and right mushroom body. In (j) and (k), we draw the contours on the original image.



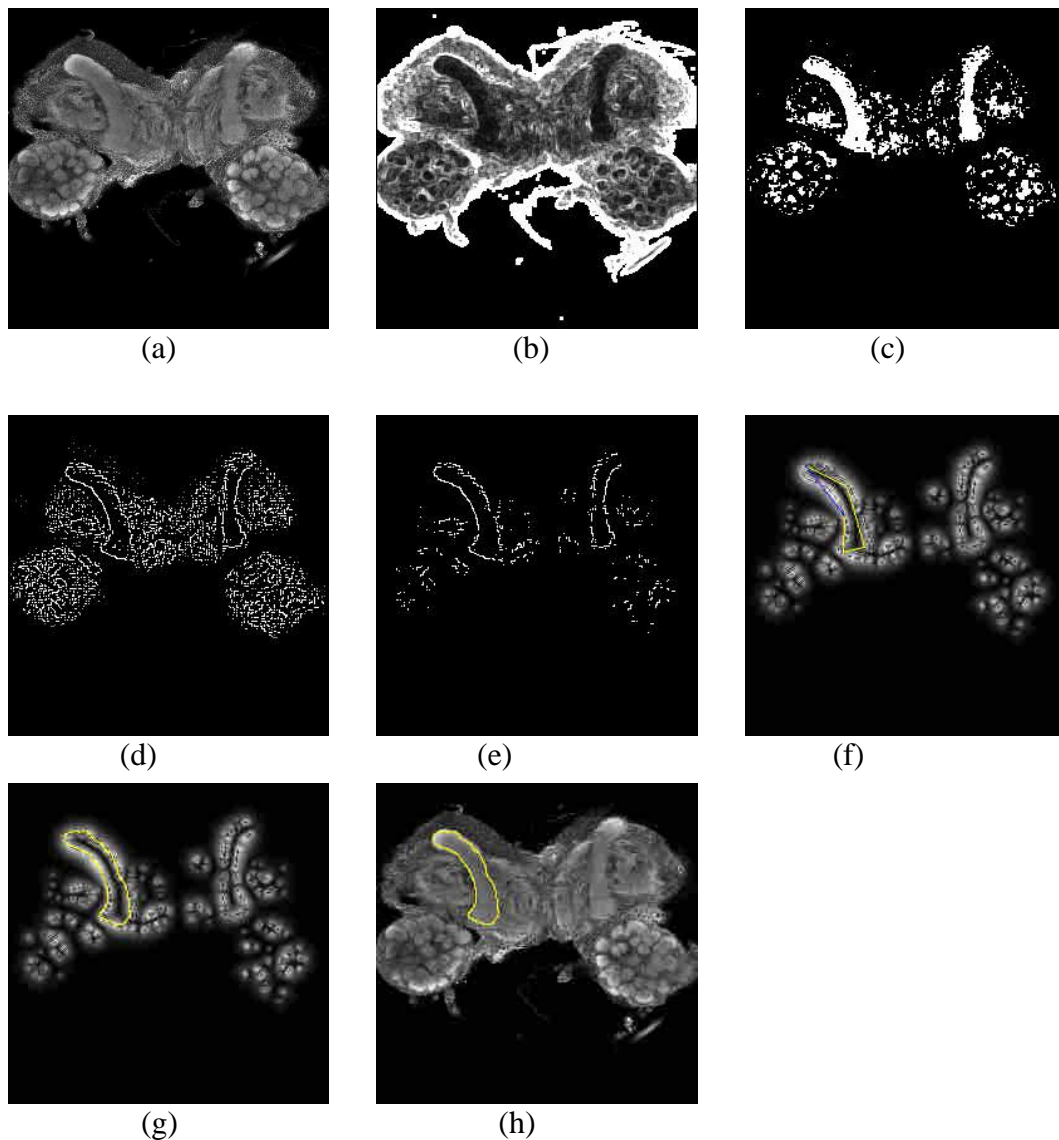


Fig. 5 The contrast of this image is worst than the one shown in Fig. 5. (a) The original image. (b) The image after texture analysis. (c) Intensity thresholding of the image after texture analysis. (d) and (e) are the images after Canny edge detector and intensity thresholding. (f) The initial contour. (g) The converge snake for one of the mushroom body. (h) is the converged contours drawn over the original image.

3 NEURON TRACING

Finding the centerline of the tubular structure helps to segment or analyze the organs such as the blood vessels or neuron fibers. This section described a semi-automatic method using the minimum cost path finding and Hessian matrix analysis in scale space to calculate the centerline of tubular structure organs. Unlike previous approaches, exhaustive search for line-like shapes in every scale is prevent. Centerline pixels candidates and the width of the vessel are extracted by

analyzing the intensity profile along the gradient vectors in the image. A verification procedure using Hessian matrix analysis with the scale obtained from the gradient analysis is applied to those candidates. Results obtained from the Hessian matrix analysis are used to construct a weighted graph. Finding the minimum cost path in the graph gives the centerline of the tubular structure. The method is applied to find the centerline of the vessels in the 2D angiogram and the neuron fibers in the 3D confocal microscopic images.

Many organs, such as blood vessel and neuron, inside human body are tubular shaped. knowing the centerline of these organs help to obtain quantitative information, such as shape and size of the organ, for further analysis. In this paper, we present a semi-automatic method to extract the centerline of the tubular structure in 2D and 3D images. The method was applied to segment the 2D vessel in angiogram and the 3D neuron in the confocal microscopic images.

Centerline of a tubular structure can be obtained by first segment the region of interest to establish a binary image of the ROI. A further procedure can then be applied to calculate the centerline. A problem with this approach is that the ROI could be difficult to segment. For example, segmentation of the coronary arteries in the angiogram is a difficult problem.

Coronary arteries angiography is still the most common modality for physicians to assess the severity of coronary narrowing or stenosis during percutaneous coronary intervention procedure. Accurate analysis of coronary arteries in digital angiographic images is valuable and important to clinical needs. The major difficulty in automatic extraction of coronary arterial structures in angiogram lies in (1) low signal-to-noise ratio due to poor X-ray penetration, (2) vessel overlaps, and (3) superimposition of other tissues such as ribs, spine, or cardiac chambers. Traditional signal-based edge detection algorithms [錯誤! 找不到參照來源。6,17,18,19,20,21,22] were unable to effectively or accurately detect the desired structures. The existing methods specific to vessel extraction can be categorized into (i) model-based [23,24,25] (ii) tracking-based [26,27,28], (iii) classifier-based [29], and (iv) filter-based [30, 31,32] techniques. In model-based methods, the coronary arterial tree is produced based on a pre-defined coronary artery model in the form of a “graph” structure. In tracking-based methods, the process proceeds with an initial start-of-search location followed by an automatic tracking process by exploiting the spatial continuity of the vessel’s centerline, orientation, diameter, and density. In classifier-based methods, a clustering algorithm is employed with properly preprocessed data to differentiate vessel or non-vessel regions. In filter-based methods, the coronary arteries are enhanced and located so that they can be subsequently detected in the image. However, any of the above approaches cannot guarantee to work well in all of the cases.

Another case studied in this paper is tracing neuron fibers in the confocal microscopic images. A neuron consists of nucleus, dendrite, axon, and terminal button. Knowing the centerline of the neuron helps to analyze the shape of the neuron. Due to the low contrast of the image and the shape of the neuron, segmentation of neuron fibers in confocal microscopic images is a much difficult task than segmentation of the vessels in angiogram. To calculate the centerline by first segmentation of the neuron fibers might not be an appropriate approach.

In this paper, we developed a method to calculate the centerline of tubular structure directly from the image. The technique is developed based on the Hessian matrix analysis in scale space. The Hessian matrix analysis evaluates the local curvatures to explore the second order structure surrounding at a point such that results of analysis is independent of image properties, and is only dependent to the local shape of the point. Hence, they generally are suitable for different types of images. Combined with the Hessian matrix analysis, multi-scale technique is used to detect every

scale of feature and eliminates the effect of noise to construct a robust and feasible extraction algorithm. By finding the centerline directly from the image, we could develop an easier method to segment the tubular structure.

For completeness, we briefly state the background of the multi-scale technique. Lindeberg [33,34,35] revealed the way to detect ridge lines in an 2D image for computer vision using analysis of local directional derivatives to find the area at which the shape conform the ridge definition in differential geometry. His algorithm calculates directional derivatives of every pixel in scale-space, which represents different level of detail, from the coarse to the fine, of the image. The directional derivatives of a pixel at some scale can be viewed as a kind of description of the local shape at that scale. Evaluation and selection those descriptions in scale-space results the possible ridge, or say, centerline of line structures. Sato et al. [36] present a generalized measure of similarity of line in 3D space using calculation of eigenvalues of the Hessian matrix, which is equivalent to the directional derivatives. They designed a delicate formula to estimate the likelihood of being line-shape from eigenvalues of the Hessian matrix, and build a 3D line filter by it for vessel enhancement. Frangi et al. [37] also propose another line filter based on analysis of the Hessian using their own formula. It is shown by the experimental result that good noise and background suppression of this method. A line extraction algorithm using the response of line filter mentioned above and the minimum cost path finding is proposed by Wink et al. [38]. The response from the line filter is regarded as weights of edges in a graph where image pixels are viewed as nodes and there are edges connected between any two adjacent pixels in it. Hence, given two points as the beginning and ending of a vessel or neuron, finding a minimum cost path (with highest weight) extracts the most possible centerline of the vessel or neuron. Wink et al. improved their tracking algorithm by involving scale parameters into minimum cost path finding [39]. In previous methods using minimum cost finding combined with the line filter to obtain centerlines, the response using as weights of graph is generally taken from the maximum response over the scales in scale-space. In this method, however, response in every scale is reserved as a part of the weights in path finding, rather than discarding all except the max one. This strategy improves the accuracy of the algorithm but need huge of memory. This restricts the application of the algorithm to the case of handling 3D volume data.

In this section, we present a semi-automatic method to segment tubular structure in 2D image or 3D volume data. The method reduces the memory required so processing large volume in a personal computer is possible. In the next section, we describe some preliminaries. In Section 3.2, we present our method. The results are shown in Section 3.3. We have conclusion and discussion in Section 3.4.

3.1 PRELIMINARY

We briefly describe the 3D Hessian Matrix analysis. The 2D Hessian Matrix is the same but ignore the third dimension. Hessian matrix for a point p under scale σ in 3D space can be written as the following equation

$$H_{\sigma}(p) = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (11)$$

where I_{ab} is the partial derivative of I in a, b direction $\partial^2 I / \partial a \partial b$. If I is continuous, $I_{ab} = I_{ba}$, H can be written as a symmetric matrix

$$H_{\sigma}(p) = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (12)$$

Let λ_1 , λ_2 , and λ_3 be the eigen values of $H_{\sigma}(p)$, and $\lambda_1 \leq \lambda_2 \leq \lambda_3$. If (x,y,z) is on the centerline of a neuron fiber, we have $|\lambda_1| \approx 0$ and $|\lambda_2|$ and $|\lambda_3|$ are large. Furthermore, let \mathbf{v}_1 , \mathbf{v}_2 , and \mathbf{v}_3 are respectively the eigen vectors corresponding to the eigen values λ_1 , λ_2 , and λ_3 . We have \mathbf{v}_1 points along the direction of the medial axis, and \mathbf{v}_2 and \mathbf{v}_3 are normal to the medial axis.

Computing the second partial derivative of the volume data with scale $\sigma = \omega$ is implemented in the frequency domain. Given a point p , the scale $\sigma = \omega$ defines a Gaussian function $g(p, \sigma)$ that the mean is at point p with variance σ . To compute the second partial derivative of the volume data is equivalent to

1. Fourier transform the volume data into frequency domain.
2. Compute the second partial derivative of the Gaussian Function and transform it into the frequency domain.
3. Multiply the two results in the frequency domain.
4. Inverse transforming the frequency domain back to the spatial domain.

The computation of the second partial derivatives are shown in Eqs. (13-18)

$$LN_{xx}(p) = \sigma^2 I(p) \cdot \frac{\partial^2}{\partial x \partial x} g(p, \sigma) \quad (13)$$

$$LN_{yy}(p) = \sigma^2 I(p) \cdot \frac{\partial^2}{\partial y \partial y} g(p, \sigma) \quad (14)$$

$$LN_{zz}(p) = \sigma^2 I(p) \cdot \frac{\partial^2}{\partial z \partial z} g(p, \sigma) \quad (15)$$

$$LN_{xy}(p) = \sigma^2 I(p) \cdot \frac{\partial^2}{\partial x \partial y} g(p, \sigma) \quad (16)$$

$$LN_{yz}(p) = \sigma^2 I(p) \cdot \frac{\partial^2}{\partial y \partial z} g(p, \sigma) \quad (17)$$

$$LN_{zx}(p) = \sigma^2 I(p) \cdot \frac{\partial^2}{\partial z \partial x} g(p, \sigma) \quad (18)$$

The Hessian Matrix with at point p for scale σ , denoted $H_{\sigma}(p)$, is established using the following equation.

$$H_{\sigma}(p) = \begin{bmatrix} LN_{xx}(p) & LN_{xy}(p) & LN_{xz}(p) \\ LN_{yx}(p) & LN_{yy}(p) & LN_{yz}(p) \\ LN_{zx}(p) & LN_{zy}(p) & LN_{zz}(p) \end{bmatrix}. \quad (19)$$

3.2 METHOD

The method consists of three steps, the preprocessing step, Hessian Matrix analysis, and constructing a weighted graph and find the shortest path in the graph. Each of the steps is described in the following.

3.2.1 PREPROCESSING

Gradient in an image is generally used to compute the edge in the image. Using the Sobel operation, we compute the gradients, S_x , S_y , S_z along x , y , and z directions at point (i,j,k) . Let $p=(i,j,k)$ be a point, $(S_x(p), S_y(p), S_z(p))$, describes the direction that the intensity changes the most and $|S_x|+|S_y|+|S_z|$ gives the magnitudes that the intensity changes. If p is a boundary point, $|S_x|+|S_y|+|S_z|$ tends to be large and $(S_x(p), S_y(p), S_z(p))$ approximately points to the center of the tubular structure. Based on this observation, we shall look at the points that have the magnitude of the gradient greater than a given threshold T_g . And along the direction of the gradient, we shall determine the approximate boundaries and the estimated center point of the tubular structure. The estimated boundary points and center points are presented as the triple $\langle p,c,q \rangle$ where p and q are the estimated boundary points and c is the estimated center point.

The preprocessing step is different for finding centerline for vessel (2D case) and neuron (3D case). We separate the discussion in two parts.

2D Vessel Case

The triple is obtained by analyzing the intensity distribution along the direction of the gradient vector. Let p be a boundary point obtained from the Sobel filtering. Let \mathbf{v} be a directed line that has one end point anchor at p and emitting along the direction of the gradient. Along \mathbf{v} , the intensity distribution can be roughly divided into five intervals, namely, flat interval, upward interval, ridge interval, downward interval, and flat interval. The five intervals along \mathbf{v} indicate that the ray shooting into the vessel, passing the center of the vessel on the ridge interval, and exiting the vessel. The entering point, a point on the ridge interval, and the exiting points are respectively the points p , c , and q . The algorithm to determine the five intervals and the triple is described in the following.

Along \mathbf{v} , we discretize the directed line to a set of points, $\{s_i | s_i = p + i \cdot \varepsilon \cdot \mathbf{u}, i = 0, \dots, n\}$, where \mathbf{u} is the unit vector along \mathbf{v} . The intensity of each of the sampled point s_i is denoted $I(s_i)$. If s_i does not have integer coordinates, $I(s_i)$ is obtained using the bilinear interpolation from the four neighboring points. Each pair of $I(s_i)$, $I(s_{i+1})$, $i=1, \dots, n-1$ defines a slope m_i

$$m_i = \frac{I(s_{i+1}) - I(s_i)}{\varepsilon}. \quad (20)$$

Depending on the slope, we classify the intervals into four categories, namely, flat, upward, downward, and ridge. Given a threshold, T_{flat} , the i th interval is

- *flat*, if $|m_i| < T_{\text{flat}}$,
- *upward*, if $m_i > 3T_{\text{flat}}$,
- *downward*, if $m_i \leq -T_{\text{flat}}$, and
- *ridge*, if $|m_i| < T_{\text{flat}}$, and this interval is between an upward interval and a downward interval.

With these four intervals, for a given point p , c and q are determine if they meet one of the following two conditions,

- q is in a downward interval and $I(p)=I(q)$. In this case, the centerline of the neuron, c , is $(p+q)/2$.
- c is the center of the ridge interval or at the boundary of the upward and downward intervals (ridge is empty). In this case, q locates at $c+|pc|$.

For each computed p , q , and c , we have a triple $\langle p,c,q \rangle$ which is a possible cross section of a vessel and c is a candidate for a point on the centerline. Note that, for each point p , there could be a pair of triples calculated. Figure 錯誤! 找不到參照來源。 show the triples (cross section) found according to the rules stated above. According to the rules, we find the set possible candidates for the centerline. The best possible result shall be determined by the shortest path finding algorithm. The set of points $C=\{c|c \text{ is a candidate for a point on the centerline}\}$ will be used for creating the weighted graph.

3D Neuron Case:

When the gradient analysis is applied to 3D volume case, the steps performed are the same as that in the 2D case. Since the directed line determined by the gradient may not pass through the true center of a neuron fiber, the candidates for the points on the centerline are determined differently. Let C' be the set of points of all points c found in gradient analysis and C be the set of points obtained by applying the dilation operation of C' . C contains most of the points on the centerline.

3.2.2 HESSIAN MATRIX ANALYSIS

The points in C is the set of candidate point for the centerline of the tubular structure. Note that the point in C is obtained through the gradient analysis that the magnitude of the gradient is greater than a given threshold. Noises can also produce points in C . To verify whether a point in C is truly on the centerline, we apply the Hessian Matrix analysis.

For each triple $\langle p,c,q \rangle$ in C , $|\overline{pq}|$ is the estimated width of the vessel. Let, $w=|pq|/2$ we expect that there is a tubular structure with radius equal to the half width of the cross section, W , passing through c . By using a Hessian Matrix with scale $\sigma = W$, there should be a detected tubular structure.

2D Vessel Case

For each point c in C , we compute the Hessian Matrix $H_\sigma = (c)$.

$$H_\sigma(c) = \begin{bmatrix} LN_{xx}(c) & LN_{xy}(c) \\ LN_{xy}(c) & LN_{yy}(c) \end{bmatrix}. \quad (21)$$

Note that, the σ used in obtained from the previous step. The two eigen values λ_1 and λ_2 ($\lambda_1 < \lambda_2$) are calculated. We define w_1 and w_2 ,

$$w_1 = \frac{|\lambda_1|}{|\lambda_2|^2} \quad (22)$$

and

$$w_2 = \frac{|\mathbf{v}_2 \cdot \overline{pq}|}{\|\mathbf{v}_2\| |\overline{pq}|} \quad (23)$$

where \mathbf{v}_2 is the eigen vector corresponding to λ_2 . Recall that, if c is indeed on the centerline of the vessel of width W , $|\lambda_1| \approx 0$ and $|\lambda_1| \ll |\lambda_2|$. That means w_1 should be very small. Secondly, if

$\bar{p}\bar{q}$ is truly orthogonal to the centerline of the vessel, \mathbf{v}_2 should have same direction as $\bar{p}\bar{q}$, thus w_2 should be close to 1. Combining w_1 and w_2 , we define the weight w

$$w = e^{\left(\frac{w_1^2}{-2a^2}\right)} e^{\left(\frac{(1-w_2)^2}{-2b^2}\right)} \quad (24)$$

w is in the range $[0,1]$. If w is large, there is significant line structure with width W and c is on the centerline.

3D Neuron Case

We compute the 3D Hessian Matrix as show in Eq. (19). Again, the three eigen values, λ_1 , λ_2 and λ_3 , are computed. Recall that if c is indeed on the centerline of a tubular structure, we have $\lambda_1 \ll \lambda_2$, $\lambda_1 \ll \lambda_3$, and $\lambda_2 \approx \lambda_3$. In this case, we define the weight w ,

$$w = \sqrt{\lambda_2 \lambda_3} - \lambda_1. \quad (25)$$

If w is large then c is likely on the centerline of a neuron fiber.

3.2.3 GRAPH CPNSTRUCTION AND THE SHORTEST PATH CALCULATION

Neuron structure extraction is carried out through finding the shortest path in a weighted graph $G=(V,E)$. Each vertex in V corresponds to a point in C . Let u, v be two vertices in E . There is an edge between u and v if the points corresponding to u, v are 8-neighbor connected in 2D case or 26-neighbor connected in a 3D volume. There are costs associated with the edges in E . We used different way to define the weight of an edge.

2D Vessel Case

The weight between two connected vertices u and v is denoted $C(u,v)$.

$$C(u,v) = (C_{\text{weight}}(i,j) + C_{\text{width}}(i,j)) \cdot C_{\text{distance}}(i,j) \cdot C_{\text{direction}}(i,j), \quad (26)$$

where

$$C_{\text{weight}}(i,j) = \frac{\frac{1}{w_i} + \frac{1}{w_j}}{2} \quad (27)$$

$$C_{\text{width}}(i,j) = \left| (|w_i - w_j|) + 0.5 \right| \quad (28)$$

$$C_{\text{distance}}(i,j) = |\bar{u}\bar{v}| \quad (29)$$

$$C_{\text{direction}}(i,j) = \left(2 - \frac{(1 - \mathbf{v}_i \mathbf{v}_j \cdot \mathbf{dir}_i)^2}{2c^2}\right) \left(2 - \frac{(1 - \mathbf{v}_i \mathbf{v}_j \cdot \mathbf{dir}_j)^2}{2c^2}\right) \quad (30)$$

3D Neuron Case

For an edge e connecting u and v , we define

$$C(u,v) = w_1 + C_{\text{distance}}, \quad (31)$$

where $C_{\text{distance}}(u,v) = D(u_i, v_j)$.

The weights are designed so that the centerline of a piece of vessel or neuron fiber occur on the shortest path between a pair of vertices on the neuron. Given the weighted graph, $G=(V,E)$, we calculate the all pairs shortest paths using the Dijkstra Algorithm. The structure extraction algorithm is obtained by a graphics user interface system. The user provides a pair of points on the centerline as the input. The shortest path between the pair of points are calculated that corresponds to the centerline between the points. The process iterates until the whole tubular structure is extracted.

3.3 RESULTS

In Figure 6, a computer generated phantom test data is shown. The centerline obtained by the proposed method is shown on the right side.



Figure 6: A computer generated phantom data set. The original image is shown on the left and the center line obtained is show on the right.

In order to test the robustness of the proposed method, we used computer to generate low contrast phantom image. We also generated the condition that the background intensity changes gradually (Figure 7). The image containing noises were also generated by computer for testing robustness reason(Figure 8). In Figure 7, since the background intensity changes, the centerline shift to the right of the true centerline. But the centerline falls in the interior of the vessel. That means the centerline can still serve as a reference to find the boundary of vessel. In the noisy image shown in Figure 8, the proposed method can still accurately calculate the centerline.

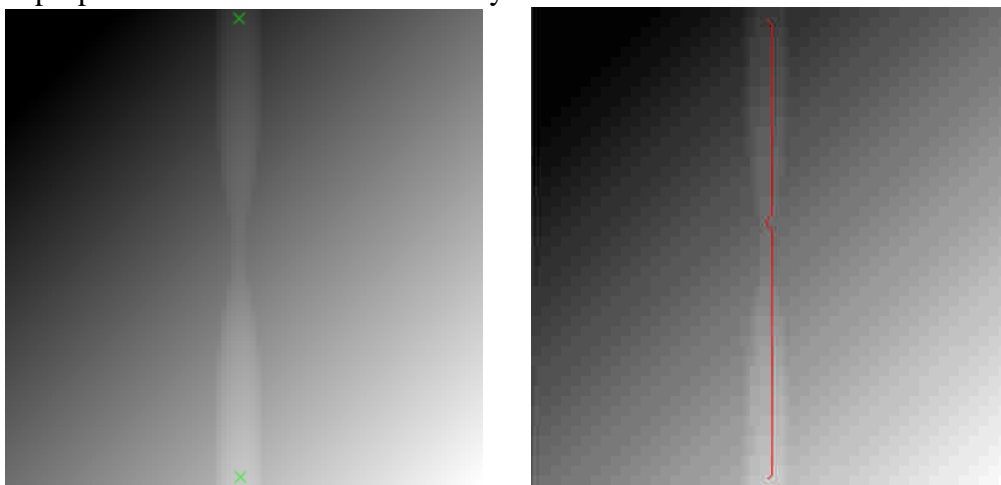


Figure 7: The test data containing low contrast image and the background intensity change gradually. The original image is shown on the left and the center line obtained is show on the right.

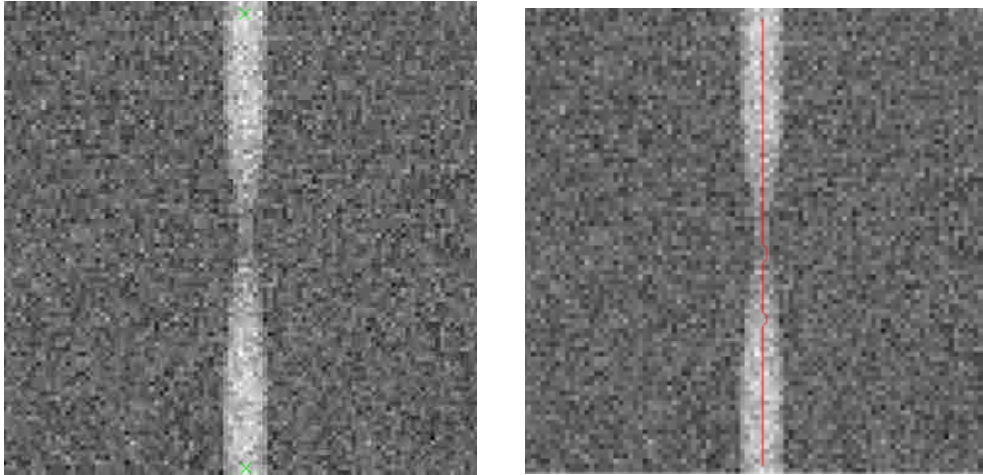


Figure 8: Image containing noises. The original image on the left and the result shown on the right.

In Figure 9, the retinal images is shown. The centerlines for the vessels in the low contrast area could still be well identified.

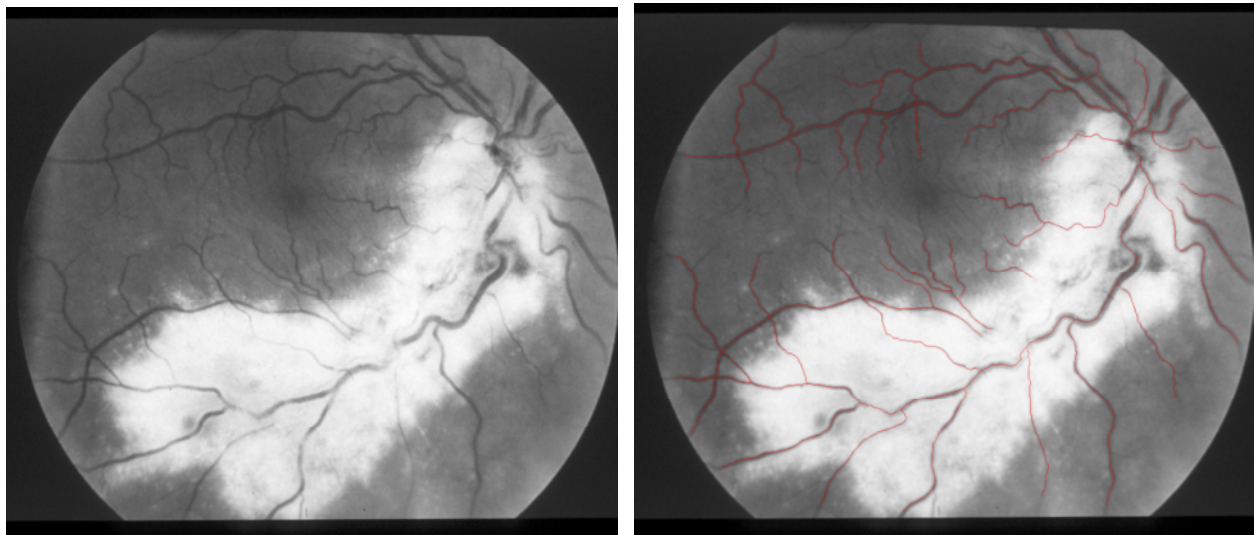


Figure 9: The result obtained for the retinal image. The original image is shown on the left and the centerlines found is shown on the right.

The experiments using the coronary artery angiogram was also performed (Figure 10). The arterial tree can be extracted fairly complete.

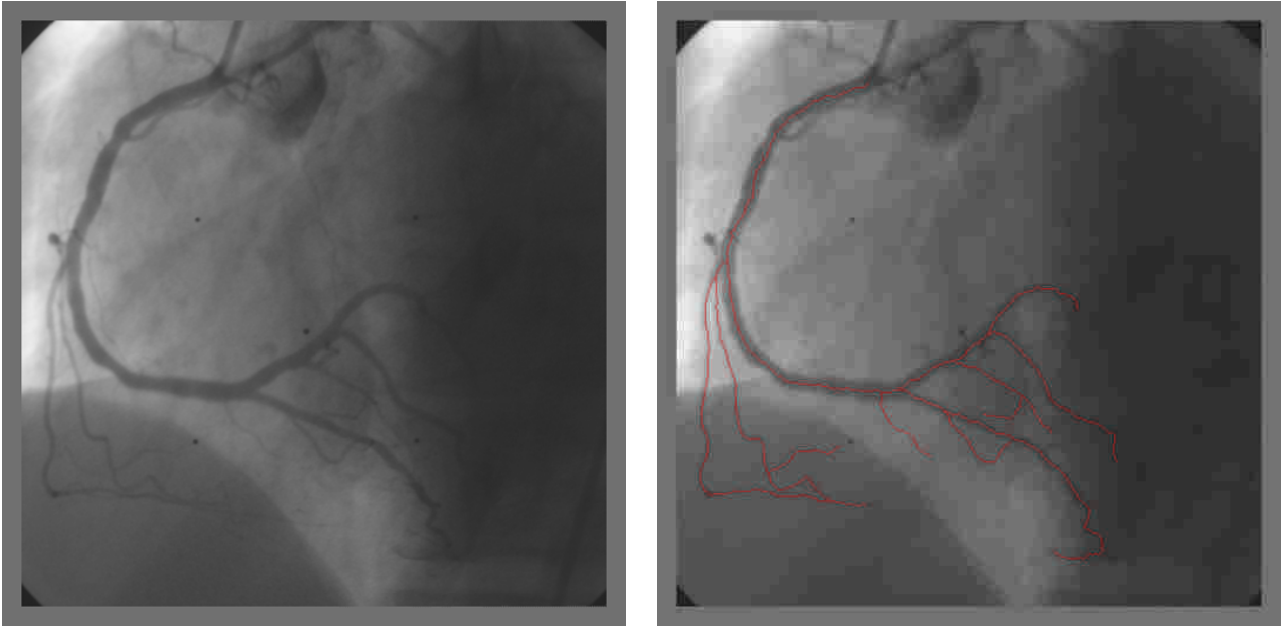


Figure 10: Center lines identification for the coronary artery in the angiogram. The image on the right shows the extracted centerlines.

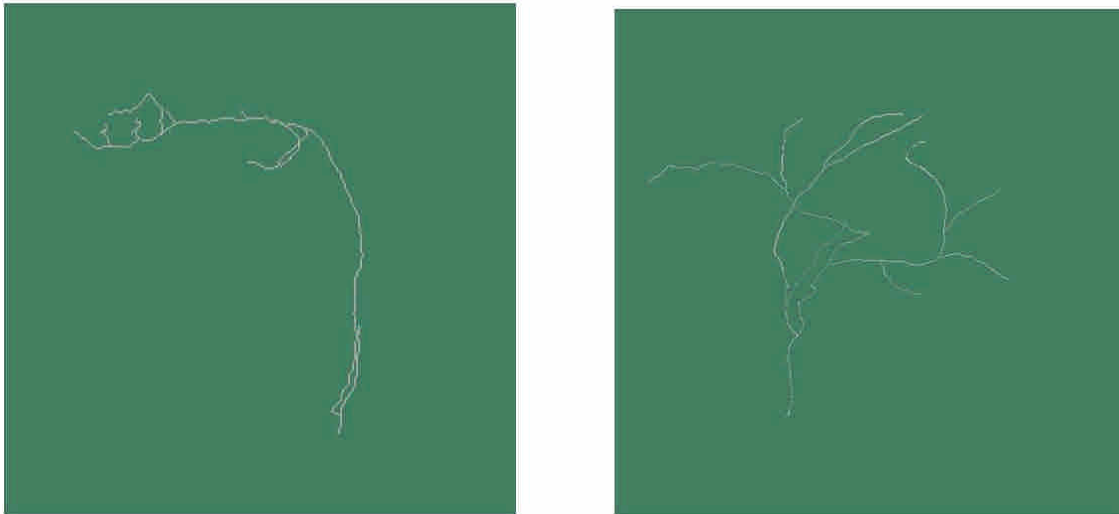


Figure 11: Two examples show the results of tracing neuron fibers. The images are 3D confocal microscopic images.

In Figure 11, we present the result obtained by tracing neuron in 3D confocal microscopic volume data. The image on the right was obtained by tracing neuron fibers in a volume data of size

512×512×120. This problem can be fitted into a personal computer with 2G memory. The preprocessing time took about two hours.

3.4 CONCLUSION AND DISCUSSION

We presented a semi-automatic method to find the centerline of tubular structure in 2D or 3D volume images. A problem with this approach is the long preprocessing time. However, the method worked well even for 3D neuron tracing case in confocal microscopic images. The future work will be segmentation of vessels or neuron fibers based on the centerline extracted.

4 ON GOING AND FUTURE WORK

Based on the previous mentioned result, we shall develop algorithms and software system to extract and analysis the fine structures in the *Drosophila* confocal microscopic images. The goal is to employ computer methods to solve important biological problems. We try to extract useful quantitative or geometric information from the segmented fine structures such as the neural fibers, bouton and cell bodies. We expect that using the developed software system can obtain important information that is not available by using commercially available software system such as Amira.

REFERENCES

1. Chiang AS., Liu YC., Chiu SL., Hu SH, Huang CY, and Hsieh CH. (2001) Three-dimensional mapping of brain neuropils in the cockroach, *Diploptera punctata*. *J. Comp. Neurol.* 440, 1-11.
2. Rein K, Zockler M, Mader MT, Grubel C, and Heisenberg M. (2002) The *Drosophila* standard brain. *Curr. Biol.* 12, 227-231.
3. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, pp.321-331, 1987.
4. D. Terzopoulos and K. Fleischer, "Deformable models," *Vis. Comput.*, vol. 4, pp. 306-331, 1988.
5. F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 617-634, 1993.
6. D. Terzopoulos and R. Szeliski, "Tracking with Kalman snakes," in *Active Vision*, A. Blake and A. Yuille, Eds. Cambridge, MA: MIT Press, 1992, pp. 3-20.
7. Chenyang Xu and Jerry L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Transactions on Image Process*, vol. 7, no. 3, Mar. 1998.
8. J. Canny., "A computational approach to edge detection," *IEEE Transactionis on Pattern Analysis and Machine Intelligence.*, Vol 8, pp670-698, 1986.
9. R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Interscience, 1953.
10. D.J. Willianms and M. Shuh, "A fast algorithm for active contours and curvature estimation", *CVGIP:Image Understanding*, Vol. 55, No. 1, pp. 14-26, 1992.
11. S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," *IEEE Trans, Med. Imag.*, vol. 14, pp. 12-24, Mar. 1995.
12. H. Freeman, "On the encoding of arbitrary geometric configuration," *IRE Transactions on Electronic computers*, 1961.
13. T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*. MIT Press, 1989.
14. A. H. Charles and T. A. Porsching, *Numerical Analysis of Partial Differential Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1990, reading.
15. W. F. Ames, *Numerical Methods for Partial Differential Equations*, 3rd ed. New York: Academic, 1992, reading.

16. D. Marr and D. Hildreth, "Theory of edge detection." Proc. Roy Soc, Ser. B, vol. 207, pp. 287-217, 1980
17. J. Canny, "A Computational Approach to Edge Detection," IEEE Trans. PAMI, vol. 8, pp. 679-697, 1986.
18. T. Elfving, J.O. Eklundh, and S. Nyberg, "Edge Detection Using the Marr-Hildreth operator with different sizes," Int. Conf. On Pattern Recognition, pp. 1109-1112, 1982.
19. W. Frei and C.C. Chen, "Fast boundary detection: A generalization and a new algorithm,," IEEE Trans. Comput., vol. 8, pp. 988-998, Oct. 1977.
20. F.M. Dickey, K.S. Shanmuugam, and J.A. Gree, "An optimal frequency domain filter fir edge detection in digital pictures," IEEE Trans. Anal, Machine Intell., pp. 37-49, Jan. 1979.
21. K.W. Mondestino and R.W. Fries, "Edge detection in noisy images using recursive digital filtering," Comput. Graphics and Image Processing, vol. 6, pp. 409-433, 1911.
22. V. Torre and T.A. Poggio, "One edge detection," IEEE PAMI., vol PAMI-8, pp. 147-163, 1979.
23. P. H. Eichel, E.J. Delp, K. Koral, and A. J. Buda, "A method for fully automatic definition of coronary arterial edges from cineangiograms," IEEE Trans. Med. Imaging, vol. 18, pp. 313-320, 1988.
24. K. Haris, S. N. Efstratiadis, N. Magkaveras, C. Pappas, J Gourassas, and G. Louridas, "Model based morphological Segmentation and labeling of coronary artery angiograms," IEEE Trans. Med. Imaging, vol. 18, pp. 1003-1015, 1999.
25. N. Ezquerro, S Capell, L. Klein, and P. Duijves, "Model-Guided Labeling of Coronary Structure," IEEE Trans. Med. Imaging, vol. 17, no. 3, jun. pp 429-441, 1998.
26. R. C. Chan, W. C. Karl, and R. S. Lees, "A New Model-Based Technique for Enhanced Small-Vessel Measurements in X-Ray Cine-Angiograms," IEEE Trans. Med. Imaging, vol. 19, pp. 243-255, Mar., 2000.
27. Y. Sun, "Automated identification of vessel contours in coronary arteriograms by an adaptive tracking algorithm," IEEE Trans. Med. Imaging, vol. 8, pp. 78-88, Mar., 1989.
28. S. Tamura, K. Tanaka, S. Ohmori, K. Okazaki, A. Okada, and M. Hoshi, "Semiautomatic leakage analyzing system for time series fluorescein ocular fundus angiography," Pattern Recognition, vol. 16, no. 2, pp. 149-162, 1983.
29. Y. Toliias and S. Panas, "A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering," IEEE Trans. Med. Imaging, vol. 17, pp. 263-273, Apr., 1998.
30. S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. GOLDBAUM, "Detection of blood vessels in retinal images using two-dimensional matched filter," IEEE Trans. Med. Imaging, vol. 8, pp. 263-269, 1989.
31. A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating Blood Vessels in Retinal Images by Piecewise threshold probing of a matched filter response," IEEE Trans. Med. Imaging, vol. 8, pp. 203-210, 2000.
32. Y. Sun, R. J. Lucariello, and S. A. Chiaramida, "Directional Low-Pass Filtering for Improved Accuracy and Reproducibility of Stenosis Quantification in Coronary Arteriograms," IEEE Trans. Med. Imaging, vol. 14, No. 2, pp. 242-248, Jun.,2000.
33. T. Lindeberg, Scale-Space Theory in Computer Vision: Kluwer Academic Publishers, 1994.
34. T. Lindeberg, "Feature detection with automatic scale selection," International Journal of Computer Vision, vol. 30, pp. 79-116, 1998.
35. T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," International Journal of Computer Vision, vol. 30, pp. 117-154, 1998.

36. Y. Sato, S. Nakajima, H. Atsumi, T. Koller, G. Gerig, S. Yoshida, and R. Kikinis, "3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Cvrmcd-Mrcas'97*, vol. 1205, pp. 213-222, 1997.
37. A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," *Medical Image Computing and Computer-Assisted Intervention - Miccai'98*, vol. 1496, pp. 130-137, 1998.
38. O. Wink, A. F. Frangi, B. Verdonck, M. A. Viergever, and W. J. Niessen, "3D MRA coronary axis determination using a minimum cost path approach," *Magnetic Resonance in Medicine*, vol. 47, pp. 1169-1175, 2002.
39. O. Wink, W. J. Niessen, and M. A. Viergever, "Multiscale vessel tracking," *Medical Imaging, IEEE Transactions on*, vol. 23, pp. 130-133, 2004.