

Measurement of Albumin Concentration Gradients  
within the Glomerular Capillary Wall Based on  
Active Contour Model

By

Bingshu Wang

Thesis

Submitted to the Faculty of the  
Graduate School of Vanderbilt  
University in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

May, 2016

Nashville, Tennessee

Approved:

D. Mitchell Wilkes, Ph.D.

H. William Fissell IV, M.D

# TABLE OF CONTENTS

	Page
<b>LIST OF FIGURES .....</b>	<b>iv</b>
<b>Chapter</b>	
<b>I. INTRODUCTION.....</b>	<b>1</b>
Background of Medical Image Processing.....	1
Background of the Study.....	1
Statement of the Problem .....	2
Content of the Paper .....	2
<b>II. IMAGE FILTERING .....</b>	<b>3</b>
Background .....	3
Linear Filter.....	3
Non-Linear Filter.....	5
Edge Preservation of Non-Linear Filter .....	7
<b>III. ACTIVE CONTOUR MODEL.....</b>	<b>8</b>
Active Contour Model.....	8
Balloon Force .....	13
GVF Snake Model.....	15
VFC Snake Model.....	19
<b>IV. ESTIMATION OF PROFILES.....</b>	<b>24</b>
Background .....	24
Preprocessing .....	25
Estimation of GBM Location.....	28
Estimation of Profile Across GBM.....	30
<b>SUMMARY .....</b>	<b>39</b>
<b>REFERENCES.....</b>	<b>40</b>
<b>MATLAB CODES .....</b>	<b>42</b>

## LIST OF FIGURES

1.	Mean filter mask .....	4
2.	Gaussian filter .....	5
3.	Edge preservation of median filter.....	7
4.	An active contour example .....	12
5.	Active contour model on an image with concave regions .....	15
6.	GVF snake model on an image with concave regions .....	18
7.	An example of discrete vector field kernel .....	21
8.	VFC snake model on an image with concave regions .....	22
9.	A comparison between GVF and VFC .....	23
10.	An example of pair of scanned images .....	24
11.	Profile estimation result on original image.....	26
12.	Profile estimation result on image filtered by $3 \times 3$ median filter .....	26
13.	Profile estimation result on image filtered by $5 \times 5$ median filter .....	27
14.	Both channels images after applying $3 \times 3$ median filter .....	27
15.	Initial active contour .....	29
16.	Final active contour on both channels of images.....	29
17.	An example of sampling position .....	30
18.	Sampling map .....	31
19.	Averaged sampling result .....	31
20.	Flow chart of estimation procedure .....	32
21.	Estimation results.....	33

# Chapter I

## INTRODUCTION

### **Background of Medical Image Processing**

Images are the main source for people to obtain and exchange information, in recent decades, image processing and pattern recognition technology were developed rapidly. Digital image processing technology has also been expanded. Medical image processing is one important branch. Along with the rapid development of medical imaging technology, computer-aided diagnosis becomes more viable. These technologies enable physicians and researchers to better understand the anatomy of the human body through virtual interaction, and to make better diagnoses and develop appropriate treatments.

Image processing techniques include image restoration, image reconstruction, image enhancement and image segmentation. Medical image processing, involving feature extraction, quantitative analysis and identification technology has become a popular research field. As electronic technology and image processing theory development, the use of image processing technology to achieve automatic identification and analysis of medical images become achievable.

### **Background of the Study**

According to [1], the glomerular basement membrane (GBM) is the extracellular matrix component of the selectively permeable glomerular filtration barrier that separates the vasculature from the urinary space. The passage of various endogenous proteins across the GBM was studied in normal Munich-Wistar rats. <sup>[5]</sup>

## **Statement of the Problem**

The exact passage of endogenous albumin in the GBM has not yet been determined. In [5], the site of the filtration barrier for various plasma proteins in normal Munich-Wistar rats was revealed. However, the method taken in [5] is by removing the kidney from the rats, which doesn't reflect the situation of living animals with renal blood flow.

## **Content of this paper**

In this paper, a method based on image processing is introduced to estimate the site of retention of endogenous albumin and low molecular weight dextran.

Chapter II introduces the preprocessing applied to the images, including basic concept of linear filter and non-linear filter and the differences between these two methods.

Chapter III introduces the active contour model and its variations, including the comparison of different models.

Chapter IV introduces a method for estimating the site of retention of endogenous albumin and low molecular weight dextran based on the active contour model.

## Chapter II

### IMAGE FILTERING

#### Background

Image noise is random variation of brightness or color information in images. Image noise may occur in the capturing and the transmission process of the image. Typical types of images noise are Gaussian noise, salt-and-pepper noise and Poisson noise. Usually image restoration technologies are applied to the images to reduce the effect of noise.

Filters play a major role in the image restoration process. The basic concept of the filter is digital convolution. Let  $f(i, j)$  is the input image subjected to filtering, and  $g(i, j)$  is the filtered output. The output can be expressed in form as

$$g(i, j) = \sum_{t=i-k}^{j+k} \sum_{u=j-l}^{j+l} f(t, u)W(i - t, j - u) \quad (1)$$

where  $W$  is the filter function. The function above represents a digital image convolution process. The filter acts like a moving mask replacing the center value in the mask with the result of convolution. The total output is created by a series of shift-multiply-sum operations.

#### Linear Filter

When  $W$  is a linear function, the filter becomes a linear filter. In general, a linear filter smooths an image by filtering out higher frequency of images, thus reduces the noise.

A typical linear filter is the mean filter. It reduces the intensity variation among adjacent pixels. The mean filter is a window spatial filter that acts as a mask. It is implemented on the basis of equation (1). The mask is a square. Usually the sum of the coefficients of the mask is set to 1 so that the average intensity of the image is not altered. Figure 1 is a mask for mean filter, whose sum of coefficients equals to 1.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Figure 1. A  $3 \times 3$  mean filter mask

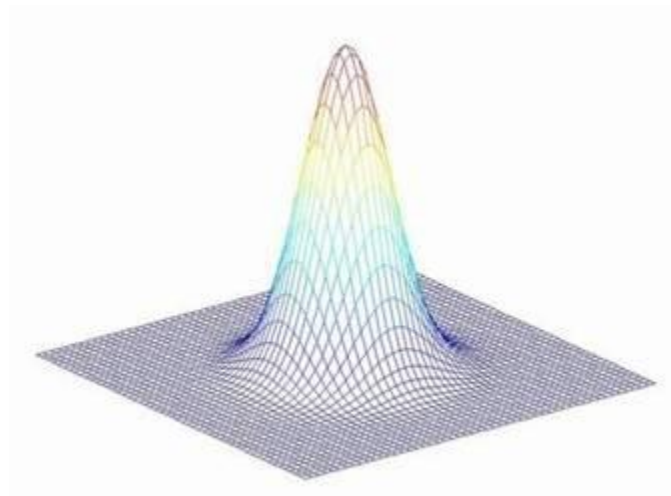
The mean filter is effective to the salt-and-pepper noise. A mean filter is a low pass filter that allows only the low frequent information to pass the filter and attenuates the high frequency information. Larger mask like  $5 \times 5$  or  $7 \times 7$  makes the image less noisy but more blurred. It's a tradeoff between noise reduction and image definition.

Another typical linear filter is the Gaussian filter. The Gaussian filter is similar to the mean filter, but it uses a different mask that represents the Gaussian function. <sup>[7]</sup>

In 2-D, an isotropic Gaussian filter has the form:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

Its distribution is shown in Figure 2.



*Figure 2. A Gaussian filter*

A Gaussian filter blurs an image, in a similar way to the mean filter. The degree of blurring is determined by the standard deviation  $\sigma$  of the Gaussian.

In contrast to the uniformly weighted property of the mean filter, a Gaussian filter provide a central weighted average of each pixel inside the mask. The closer the pixel is to the center of the mask, the more weight it gains. Thus, a Gaussian filter provides a lesser smoothing and preserves the edges better than a same sized mean filter.

### **Non-linear filter**

A non-linear filter is a filter based on a non-linear filter function. One of the popular non-linear filter is the median filter.

A standard median filter is a non-linear filter based on statistics. The center pixel of the mask is replaced by the median value of all pixels in the mask. A median filtering output can be expressed as



$$g(x, y) = \text{med}\{f(x - i, y - j), i, j \in W\} \quad (3)$$

where  $f(x, y)$ ,  $g(x, y)$  are the original image and the filtered image respectively,  $W$  is the mask function. The mask size is  $n \times n$  ( $n$  is usually an odd number like 5 or 7 and etc.). The mask shape may be linear, square, circular and so on.<sup>[9]</sup>

The unweighted median filter treats each neighbor equally, and may lead to morphological artifacts like rounding sharp corners and removing thin structures.<sup>[3]</sup> To address this problem, the weighted median filter has been introduced. The weighted median filter output is

$$g(x, y) = \text{med}\{w(i, j)f(x - i, y - j), i, j \in W\} \quad (4)$$

where the weight  $w(i, j)$  depends on an image  $I$  that can be different from  $f$ .

### **Edge Preservation of Non-Linear Filter**

The linear filter tends to blur the edges, while the non-linear filter like the median filter preserves the edge location. Because of this, linear filter are seldom used in practice for noise reduction.<sup>[2]</sup>

Figure 3 illustrates the edge preservation feature of the median filter. With increase in windows size, the mean filter blurs the edge, while the median filter preserves the edge location for both window sizes equally.

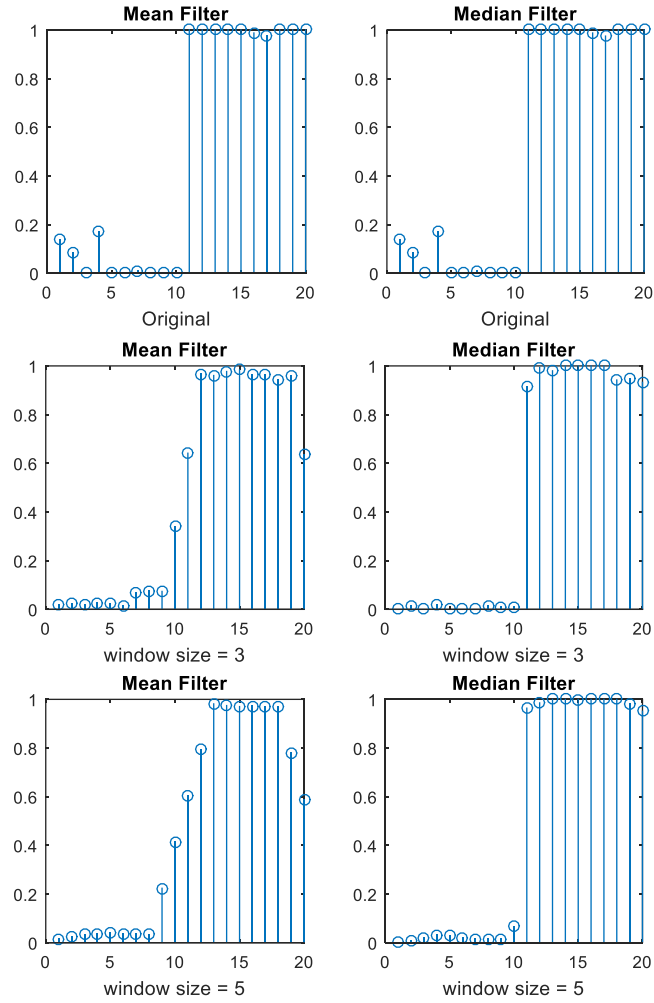


Figure 3. An Illustration of edge preservation feature of median filter. In the first two column, with increase in windows size mean filter blurred the edge, while median filter preserved the edge location for both window sizes equally

## Chapter III

### ACTIVE CONTOUR MODEL

#### Active contour model

The active contour model also known as snake, origin from the paper by Kass. et al. <sup>[10]</sup>, is a widely used method in computer vision for delineating an object edge from 2D image.

The main idea of the active contour is defining a continuity spline (surface) close to the region of interest (ROI), then deforming the spline under the forces of internal energy and external energy (minimizing the total energy) to obtain the ROI boundary. The internal forces, usually provided by the spline itself, tend to smooth the spline. The image forces, usually provided by the image, push the snake toward the desired image features like lines, edges. The external constraint forces, usually provide by user or advanced computer vision algorithm, are responsible for putting the snake near the desired local minimum. With  $v(x) = (x(s), y(s))$  representing the active contour location, the energy functional can be written as

$$\begin{aligned} E_{snake} &= \int_0^1 E_{snake}(v(s)) ds \\ &= \int_0^1 (E_{int}(v(s)) + E_{img}(v(s)) + E_{con}(v(s))) ds \\ &= \int_0^1 (E_{int}(v(s)) + E_{ext}(v(s))) ds \end{aligned} \quad (5)$$

where  $E_{int}$  represents the internal energy of the spline,  $E_{img}$  represents the image forces, and  $E_{con}$  represent externals constraint forces.

The internal spline energy can be written as

$$E_{int} = \frac{1}{2}(a(s)|v'(s)|^2) + \frac{1}{2}(\beta(s)|v''(s)|^2) = \frac{1}{2}\left(\alpha(s)\left\|\frac{d\bar{v}}{ds}(s)\right\|^2 + \beta(s)\left\|\frac{d^2\bar{v}}{ds^2}(s)\right\|^2\right) \quad (6)$$

The spline energy is composed of a first-order term controlled by  $\alpha(s)$  and a second-order term controlled by  $\beta(s)$ . The first-order term controls the continuity of the spline. The second-order term controls the curvature of the spline. Setting  $\alpha(s)$  to zero makes the spline the first-order discontinuous with intermittent. Setting  $\beta(s)$  to zero makes the snake second-order discontinuous and developing a corner.

We define the image force of the active contour as  $F = \nabla E_{img}$ . The image force is the driving force of active contour, generally provided by the image itself. When the contour is close to the ROI, such as edges, the total energy  $E_{snake}$  reaches the minimum value.

The simplest useful image energy function is the image intensity itself. If we set  $E_{img} = I(x, y)$  then the active contour will be attracted to lines.

If we set  $E_{img} = -|\nabla I(x, y)|^2$  then the active contour is attracted to contours with large image gradients such as edges.

If we set  $E_{img} = -|G_\sigma * \nabla^2 I(x, y)|^2$ , where  $G_\sigma$  is a Gaussian function of standard deviation  $\sigma$ , then the active contour is attracted to the zero-crossings of  $G_\sigma * \nabla^2 I(x, y)$  which defines edges as per Marr–Hildreth theory.

An active contour that minimizes energy  $E_{snake}$  must satisfy the Euler equation

$$a(s)v''(s) - \beta(s)v''''(s) - \nabla E_{ext}(v(s)) = 0 \quad (7)$$

This can be viewed as a force balance equation

$$F_{int} + F_{ext}^1 = 0 \quad (8)$$

where  $F_{int} = a(s)v''(s) - \beta(s)v''''(s)$  and  $F_{ext}^1 = -\nabla E_{ext}(v(s))$ .

To find a solution to (7), the snake is made dynamic by treating  $v$  as function of time  $t$  as well as  $s$ .

$$\frac{\partial v(s, t)}{\partial t} = a(s)v''(s, t) - \beta(s)v''''(s, t) - \nabla E_{ext}(s, t) \quad (9)$$

Then, the partial derivative of  $v$  with respect to  $t$  is then set equal to the left hand side of (7) as follows

$$\begin{aligned} & \alpha_i (v_i - v_{i-1}) - \alpha_{i+1} (v_{i+1} - v_i) + \beta_{i-1} (v_{i-2} - 2v_{i-1} + v_i) \\ & - 2\beta_i (v_{i-1} - 2v_i + v_{i+1}) + \beta_{i+1} (v_i - 2v_{i+1} + v_{i+2}) + (F_x(i), F_y(i)) = 0 \end{aligned} \quad (10)$$

where  $F_x(i) = \partial E_{ext}/\partial x$ ,  $F_y(i) = \partial E_{ext}/\partial y$ ,  $i = 0, 1, K, \dots, N - 1$ .

The Euler equation can be transform to a matrix form as:

$$Ax + F_x = 0 \quad (11)$$

$$Ay + F_y = 0 \quad (12)$$

where  $A$  is diagonal matrix as in figure 4 defined by coefficient  $\alpha$  and  $\beta$ :

$$A = \begin{bmatrix} c & b & a & 0 & L & L & 0 & a & b \\ b & c & b & a & 0 & L & L & 0 & a \\ a & b & c & b & a & 0 & L & L & 0 \\ 0 & a & b & c & b & a & 0 & L & M \\ M & O & O & O & O & O & O & O & M \\ M & L & 0 & a & b & c & b & a & 0 \\ 0 & L & L & 0 & a & b & c & b & a \\ a & 0 & L & L & 0 & a & b & c & b \\ b & a & 0 & L & L & 0 & a & b & c \end{bmatrix} \quad (13)$$

where  $a = -\beta/h^4, b = (4\beta + h^2 \alpha)/h^4, c = (-6\beta - 2h^2 \alpha)/h^4, h$  is sampling interval in  $x$  and  $y$  direction in image space.

The equation 10 and 11 can be represented as

$$Ax_{t+1} + F_x = -\gamma(x_{t+1} - x_t) \quad (14)$$

$$Ay_{t+1} + F_y = -\gamma(y_{t+1} - y_t) \quad (15)$$

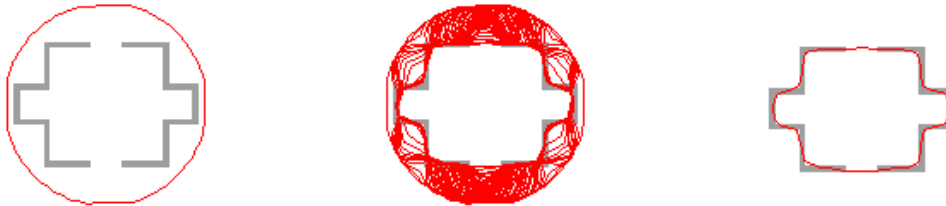
where  $\gamma$  is the damping factor, equation can then be simplified by finite-difference methods into the final contour:

$$x_{t+1} = (A + \gamma I)^{-1}(\gamma x_t - F_x) \quad (16)$$

$$y_{t+1} = (A + \gamma I)^{-1}(\gamma y_t - F_y) \quad (17)$$

where  $I$  is identity matrix.

Figure 5 shows the iteration process of an active contour. The left figure is the initial contour. The middle figure is the iteration result every 5 steps. The right figure is the result of 250 iterations.



*Figure 4. Iteration process of active contour model. The left figure is the initial contour. The middle figure is the iteration result every 5 step. The right figure is the result of 250 iteration.*

Comparing to other image edge detection methods, the active contour model has the following advantages.

- 1) The constraints of image data, initial estimation, target contour and a priori knowledge can be unified into one feature extraction process. After properly initialized, it can converge to the energy minimum automatically.
- 2) Due to the robustness to noise and intermittent border point of active model, a complete, closed target contour can be obtained without the target being divided into several small areas.
- 3) Integration of advanced information that at the beginning of the extraction process will focus on the desired target extraction and from coarse to fine to minimizing energy can greatly expanded the capture zone and reduce the computational complexity, thus it can dramatically reduce the amount of computation.

- 4) The active contour model is unrestricted to target trait and is adaptive to target traits change. It can also be used in target tracking and complex target contour extraction.

Due to its own characteristics, the traditional active contour model has some limitations.

- 1) The profile curve can easily to fall into a concave region, thus fails to detect a higher curvature concave contours.
- 2) Gradient force field is sensitive to the initial contour, thus it requires an initial contour provided close to the target contour.
- 3) Computational complexity is relatively high.
- 4) It cannot handle profile splitting and merging and does not have topology change capability.

### **Balloon Force**

In the traditional active contour model, without an external force, the counter will eventually shrink to a point or a line due to the internal force. Also the traditional active contour model has a relatively small capture range, that is, an active contour which is not close enough to targets is not attracted by them. Suppose that an initial contour is placed inside a black rectangle with white background. Even though the black rectangle provides perfect edges, the initial active contour will still shrink and vanish due to the lack of image force inside the black rectangle. One method to overcome the problem of capture range in traditional active contour model is to use the Gaussian blur with a larger standard deviation  $\sigma$ , however it blurs and distorts the edges. Also due to the balance of the internal force, only points with large image energy are able to affect the evolving of the active contour.



Due to these behaviors, a more dynamic active contour model is introduced in [4]. The active contour is now considered as a balloon that we inflate. A pressure force pushing outside was added to the previous force from an initial oriented contour. The force  $F$  now becomes

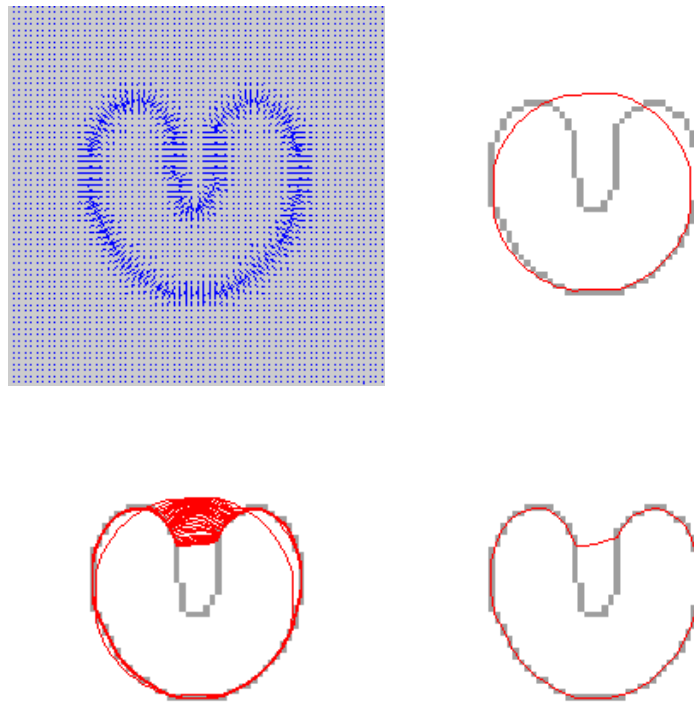
$$F = k_1 \vec{n}(s) - k \frac{\nabla E_{img}}{\|\nabla E_{img}\|} \quad (18)$$

where  $\vec{n}(s)$  is the normal unitary vector to the curve at point  $v(s)$  and  $k_1$  is the amplitude of the balloon force. If  $k_1$  is negative, it will deflate instead of inflating. Parameters  $k_1$  and  $k$  are chosen such that they are of the same order, which is smaller than a pixel size, and  $k$  is slightly larger than  $k_1$  so an edge point can stop the inflation force. The active contour is still attracted to the edges as before. But since there is a pressure force, the contour now will expand instead of shrinking even if there is weak or no image forces. Also with the help of the internal force, the contour will pass singular edge points with regard to the rest of the contour is being inflated.

As described in [11], balloon force model added a pressure force to the traditional potential force. The extra force covers the image points where traditional potential force is insufficient, thus increases the capture range of active contour model. However, the balloon force can only be set to one direction, either shrinking or expanding. Furthermore, the strength of the pressure force is difficult to set. For target with non-uniform edges, if the pressure force is too small, it may not overcome the weak edges and noises, and if the pressure force is too large, it may overwhelm legitimate edge forces.

## GVF Snake Model

One of the problem of the traditional active contour is that, without proper initialization, it may fails to capture the concave regions. Figure 6 shows an example where the traditional active contour method fails to capture the concave region of the target. The left top figure is the gradient field. The right top figure is the initial contour. The left bottom figure is the iteration process. The right bottom figure is the final contour.



*Figure 5. An example of traditional active contour model. The left top figure is the gradient field. The right top figure is the initial contour. The left bottom figure is the iteration process. The right bottom figure is the final contour.*

An external force field named gradient vector flow (GVF) field is introduced by Chenyang et al. <sup>[11]</sup> to solve the problems stated above. As described in [11], the GVF field points toward the

object boundary when very near to the boundary, but varies smoothly over homogeneous image regions, extending to the image border. The GVF field improves the capture range of the traditional active contour model, and unlike balloon force, the initial contour can be set at either side of or even across the object boundary. Also GVF field can lead the contour into concave regions.

An edge map  $f(x, y)$  with larger values near the image edges is derived from the image  $I(x, y)$ . Accordingly, we can use

$$f(x, y) = -E_{ext}^i(x, y) \quad (19)$$

where  $i = 1, 2, 3, \text{ or } 4$ ,  $E_{ext}^i(x, y)$  is one of the external force defined as

$$E_{ext}^{(1)}(x, y) = -|\nabla I(x, y)|^2 \quad (20)$$

$$E_{ext}^{(2)}(x, y) = -|\nabla[G_\sigma(x, y) * I(x, y)]|^2 \quad (21)$$

$$E_{ext}^{(3)}(x, y) = -I(x, y) \quad (22)$$

$$E_{ext}^{(4)}(x, y) = G_\sigma(x, y) * I(x, y) \quad (23)$$

The edge map  $f(x, y)$  has three basic properties. First, the gradient of the edge map is normal vectors pointing to the edge. Second, these normal vectors are larger near the edges. Third, in homogeneous regions,  $I(x, y)$  is constant, thus  $\nabla f$  is zero.

The gradient vector flow (GVF) field is then defined to be the vector field  $v(x, y) = (u(x, y), v(x, y))$  that minimizes the energy functional

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |v - \nabla f|^2 dx dy \quad (24)$$

where  $f$  is the edge map,  $\nabla f$  is the gradient field of  $f$ . When  $\nabla f$  is small, a smooth field is generated by the first term of the energy function. When  $\nabla f$  is large, the second term dominates the energy function, and is minimized by setting  $\mathbf{v} = \nabla f$ . The parameter  $\mu$  is a weight adjust parameter between the first term and the second term.  $\mu$  should be increased when the noise level increases.

GVF can be found by solving the following Euler equations

$$\mu \nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0 \quad (25)$$

$$\mu \nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \quad (26)$$

where  $f_x = \partial f / \partial x$ ,  $f_y = \partial f / \partial y$ . In homogeneous regions (where  $f(x, y)$  is constant), gradient of  $f(x, y)$  is zero, which makes the second term zero. Therefore, within these regions, GVF fill in from the boundaries of the region. This is the reason that GVF points into the concave regions.

Equations (25) and (26) can be solved by treating  $u$  and  $v$  as functions of time and solving

$$u_t(x, y, t) = \mu \nabla^2 u(x, y, t) - (u(x, y, t) - f_x(x, y)) \cdot (f_x(x, y)^2 + f_y(x, y)^2) \quad (27)$$

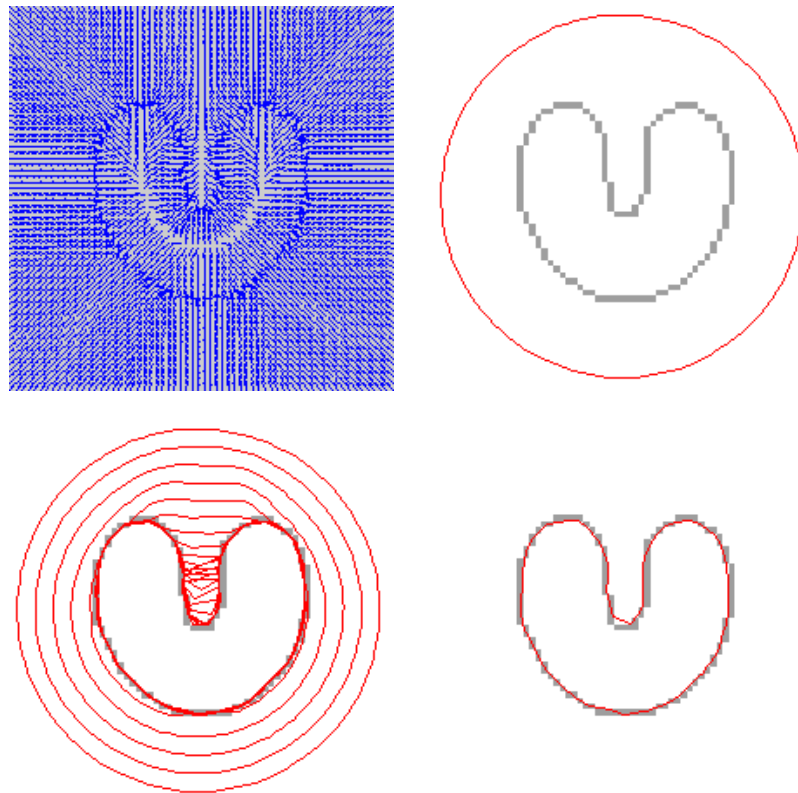
$$v_t(x, y, t) = \mu \nabla^2 v(x, y, t) - (v(x, y, t) - f_y(x, y)) \cdot (f_x(x, y)^2 + f_y(x, y)^2) \quad (28)$$

The steady-state solution (as  $t \rightarrow \infty$ ) of equation (27) and (28) is the desired solution of the Euler equations (25) and (26).

After we compute  $\mathbf{v}(x, y)$ , potential force  $-\nabla E_{ext}$  in equation (9) is then replaced by  $\mathbf{v}(x, y)$ , yielding

$$x_t(s, t) = \alpha x''(s, t) - \beta x''''(s, t) + \mathbf{v} \quad (29)$$

The following example shows the advantage of the GVF snake module in capturing concave regions comparing to traditional active contour model. Figure 7 shows the same object we tried to capture using traditional active contour method. The left top figure is the gradient field. The gradient force extended to the entire image. The right top figure is the initial contour. The initial contour is defined relatively far from the target object. The left bottom figure is the iteration process, it takes 250 times iteration to capture the object. The right bottom figure is the final contour and it captures the recessed area of target object.



*Figure 6. An example of GVF snake model. The left top figure is the gradient field. The gradient force extended to the entire image. The right top figure is the initial contour. The initial contour is defined relatively far from the target object. The left bottom figure is the iteration process, it takes 250 times iteration to capture the object. The right bottom figure is the final contour and it captures the recessed area of target object.*

## VFC Snake Model

Although the GVF model solves the problem that the traditional active contour fails to converge to concave regions, it requires a large amount of computation and is sensitivity to noise. Also the parameters are difficult to set up.

Vector field convolution (VFC) active contour model introduced in [8] is an active contour model using the VFC field as the external force. The VFC field is the convolution result of the edge map and the Vector Field Kernel which is similar to the gravity vector field. The VFC active contour model has a larger convergence area than the traditional active contour model and is able to converge to the concave areas. Also the VFC active contour model requires less amount of computation than GVF model and is robust against noise.

By replacing the standard external force  $f_{ext}(\mathbf{v}) = -\nabla E_{ext}(\mathbf{v})$  by the VFC field  $f_{vfc}(\mathbf{v})$ , the iterative active contour solution becomes

$$\frac{\partial v(s,t)}{\partial t} = a(s)v''(s,t) - \beta(s)v''''(s,t) - f_{vfc}(v(s,t)) \quad (30)$$

To get  $f_{vfc}(v(s,t))$ , we first define a vector field kernel  $k(x,y) = [u_k(x,y), v_k(x,y)]$  as

$$k(x,y) = m(x,y)n(x,y) \quad (31)$$

where  $m(x,y)$  is the magnitude of the vector at  $(x,y)$  and  $n(x,y)$  is the unit vector pointing to the kernel origin  $(0,0)$

$$n(x, y) = \left[-\frac{x}{r}, -\frac{y}{r}\right] \quad (32)$$

except that  $n(0,0) = [0,0]$ , where  $r = \sqrt{x^2 + y^2}$  is the distance from the origin. A free particle placed in the kernel field is able to move to the kernel origin. Note that the kernel origin is not the origin of the image or the edge map.

The VFC external force  $f_{vfc}(x, y) = [u_{vfc}(x, y), v_{vfc}(x, y)]$  is given by calculating the convolution of the vector field kernel  $k(x, y)$  and the edge map  $f(x, y)$  generated from the image  $I(x, y)$

$$f_{vfc}(x, y) = f(x, y) * k(x, y) = [f(x, y) * u_k(x, y), f(x, y) * v_k(x, y)] \quad (33)$$

Since the edge map  $f(x, y)$  has larger values near the target edges while is close to zero in the homogenous regions, the VFC external force will attract free particles to the edges.

Usually the effect of VFC field force should decrease as the distance between the particle and edges increases, thus  $m(x, y)$  should be a decreasing positive function of distance from the origin. Two types of magnitude functions are given in [8] as

$$m_1(x, y) = (\gamma + \varepsilon)^{-\gamma} \quad (34)$$

$$m_2(x, y) = \exp\left(-\frac{\gamma^2}{\xi^2}\right)^{-\gamma} \quad (35)$$

where  $\gamma$  and  $\xi$  are positive parameters controlling the decreasing speed,  $\varepsilon$  is a small positive constant to prevent division by zero at the origin  $m_1(x, y)$ .

An example discrete vector field kernel is demonstrated in Figure 8.

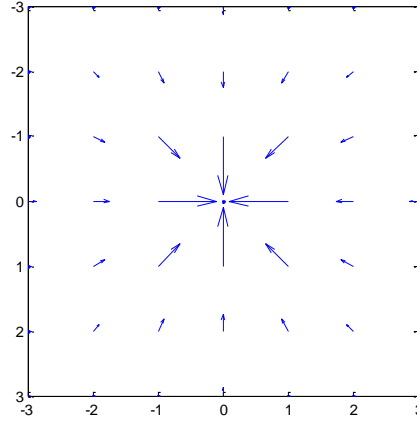
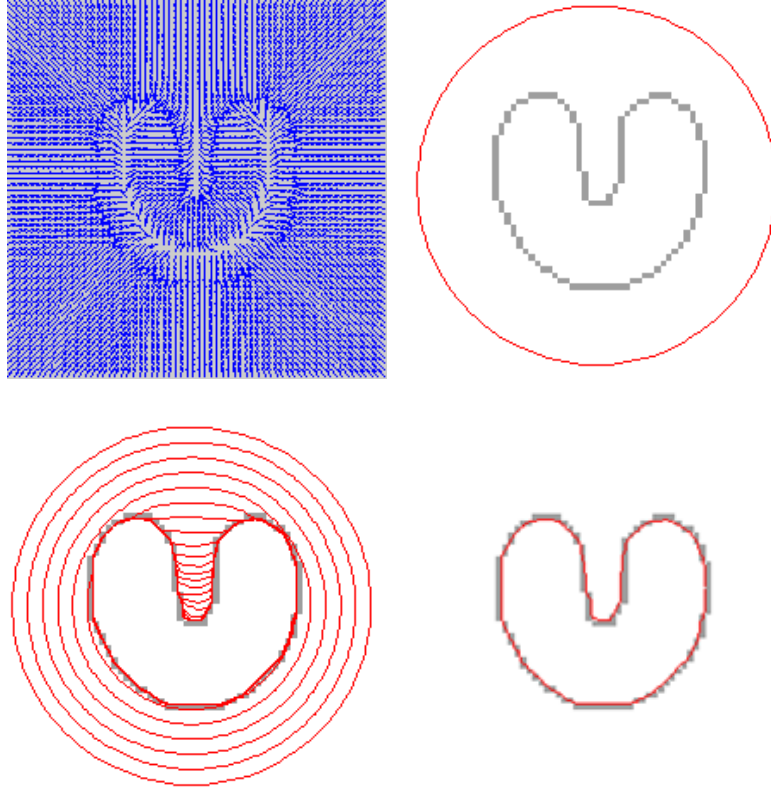


Figure 7. An example discrete vector field kernel

After we find the VFC external force  $f_{vfc}(x, y)$ , we calculate the equation (30) as in the traditional active contour model

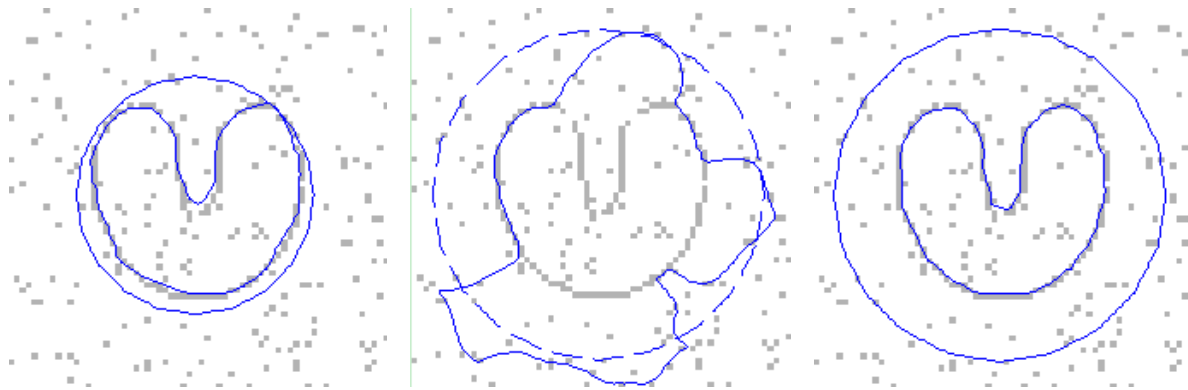
The VFC snake model converges faster than the GVF snake model. In figure 9, the VFC snake model is applied to the object we discussed in the traditional active contour and GVF snake model. The left top figure is the gradient force of the image. The right top figure is the initial contour. The left bottom contour is the iteration process, it takes 125 iterations for the VFC snake model to capture the target object compared to 250 for the GVF active snake model. The right bottom figure is the final result. The VFC snake model works well with the concave regions as does the GVF snake model.





*Figure 8. An example of VFC snake model. The left top figure is the gradient force of the image. The right top figure is the initial contour. The left bottom contour is the iteration process, it takes 125 iterations for the VFC snake model to capture the target object compared to 250 for GVF snake model. The right bottom figure is the final result.*

The VFC snake model is also robust to noise. In figure 10, a comparison of GVF and VFC is made between the GVF and the VFC snake model in a noisy image. In the left figure, the initial contour and contour after 60 iterations using the GVF snake model. The mid figure is the initial contour (the dashed line) and the contour after a 60 iterations using the GVF snake model. The right figure is the initial contour and the contour after 60 iterations using the VFC snake model. We can see that the VFC snake model successfully captures the target after 60 iterations while the GVF snake model failed to converge under the influence of the noise.



*Figure 9. A comparison between GVF and VFC under the influence of the noise. The left figure is the initial contour. The mid figure is the initial contour (the dashed line) and the contour after a 60 iterations using the GVF snake model. The right figure is the initial contour and the contour after 60 iterations using the VFC snake model.*

## Chapter IV

### ESTIMATION OF PROFILES

#### Background

The images of Glomerular Capillary Wall (GCW) were obtained by confocal microscopy operating on living rats including both proteinuric and non-proteinuric groups. In the scanning procedure, two channels of images (low molecule channel and high molecule channel) were obtained. Images are in  $512 \times 512$  tif format with gray level from 0 to 4095. We want to estimate the intensity profile along the GBM.

Figure 11 is a pair of sample scanned images. The left image is the low molecule channel result and the right is the high molecule channel result.

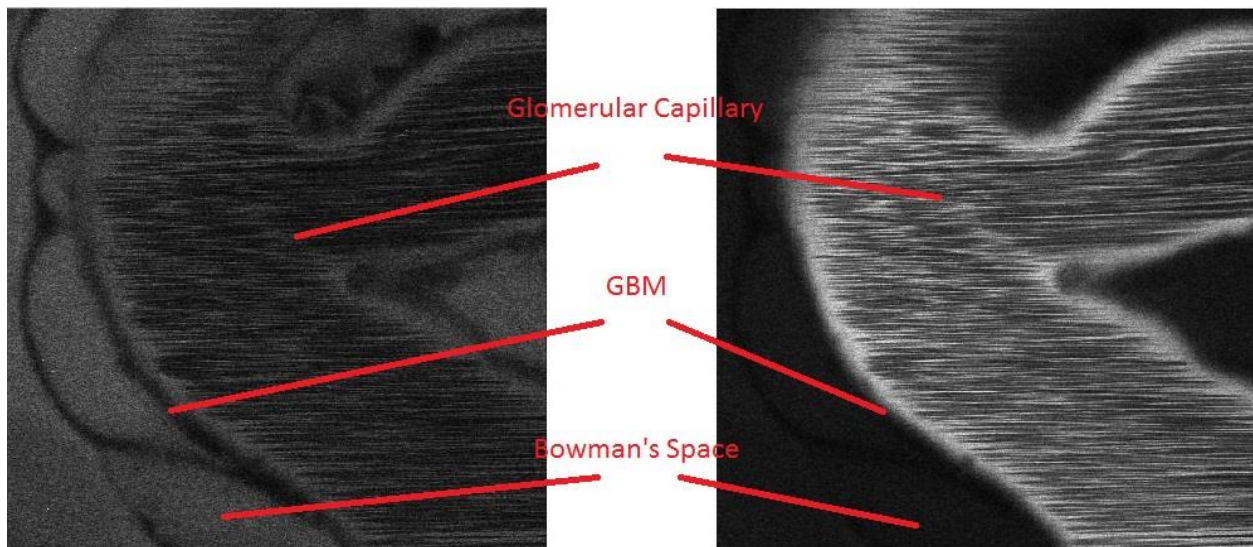


Figure 10. A pair of sample scanned images. The left image is the low molecule channel result and the right is the high molecule channel result.

## Preprocessing

The scanned GCW images contain significant line artifacts. As in figure 11, the bright horizontal line artifacts only appears at the inner boundary of the capillary wall. These artifacts occur during the scanning process due to the blood flowing inside the capillary wall.

The scanned images are degraded by horizontal line artifacts and other noise. These artifacts are caused by blood flow in Glomerular Capillary during image acquisition process. A filter needed to be applied to reduce the noise and smooth the image. Since we want to estimate the profile along the GBM, the edge information is vital to the accuracy of the result. Thus, a non-linear filter with its edge preservation advantage will reduce the noise level and preserve the edge information of GBM. In our case a median filter was applied to reduce the image noise.

We applied different sizes of median filter. Figure 12 is the image brightness cross the GBM estimated on the original image. Figure 13 is on a  $3 \times 3$  median filtered image. Figure 14 is on a  $5 \times 5$  median filtered image. Where CH2 represents the low molecule channel, and CH3 represent the high molecule channel. We can see that with  $3 \times 3$  median filter, the spikes in Bowman's space (the right part of the curve) are smoothed. To minimize the effect of the median filter on the image while reducing the noise, we chose the  $3 \times 3$  median filter to preprocess scanned images.

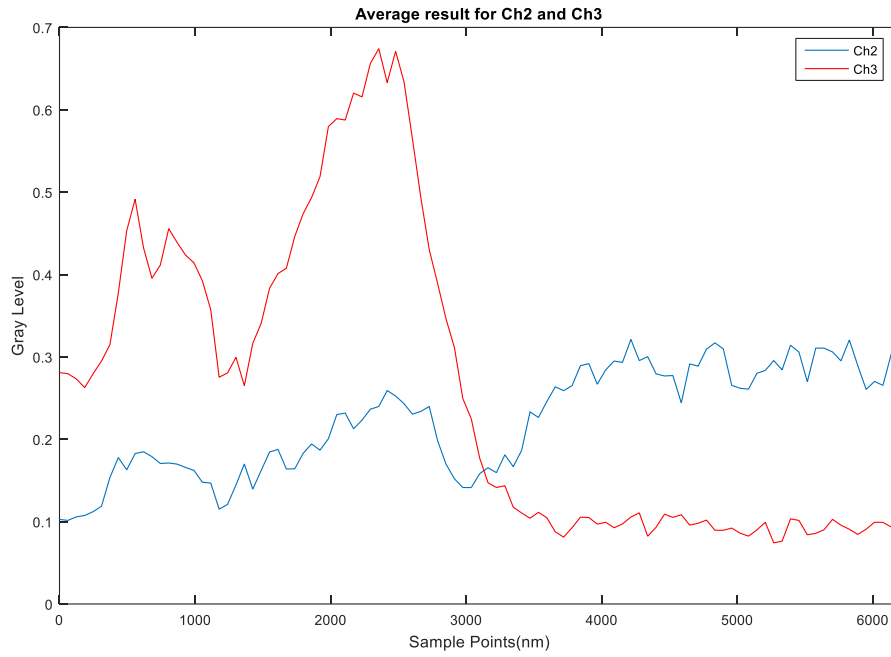


Figure 11. Profile cross the GCW estimated on original image at a single point

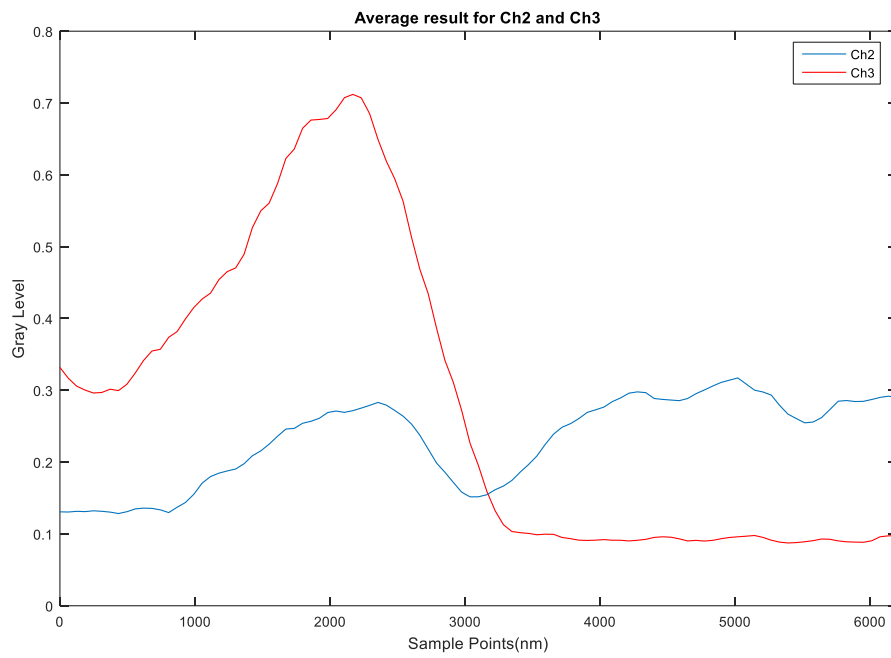


Figure 12. Profile cross the GCW estimated on  $3 \times 3$  median filtered image at a single point

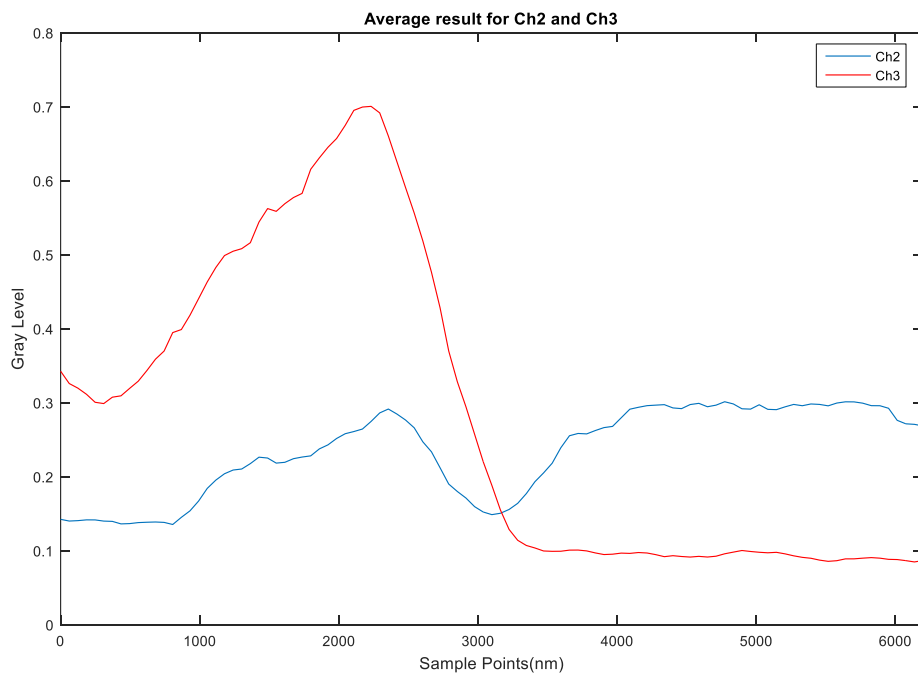


Figure 13. Profile cross the GCW estimated on  $5 \times 5$  median filtered image at a single point

Figure 15 is the images of the two channels after applying the  $3 \times 3$  median filter. The left figure is the low molecule channel image, the right figure is the high molecule channel image.

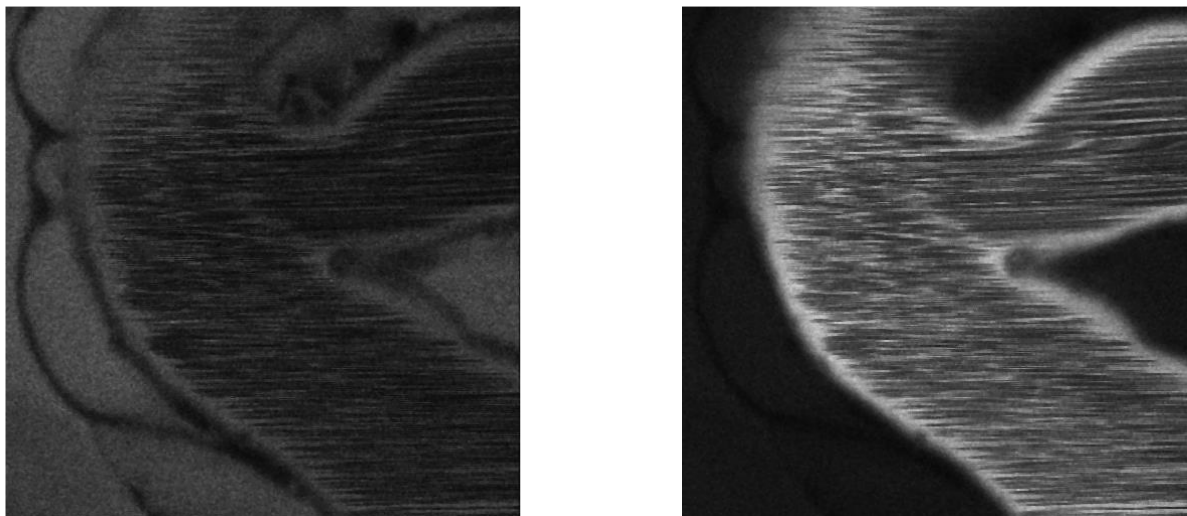


Figure 14. Both channels images after applying  $3 \times 3$  median filter. The left is the low molecule image, the right is the high molecule image.

## Estimation of GBM Location

To estimate the location of the GBM, two methods were proposed. One of them is to extract the edge of the Glomerular Capillary interior wall. The other one is to extract the position of the darkest point of the GBM which can be done in low molecule channel images, and then apply the edge location to high molecule channel images.

From figure 15 we can see that even after the median filter, the line artifacts in the Glomerular Capillary is still affecting the boundary. So the alternative method which is to extract the position of the darkest point of the GBM would work well in our case.

The position of darkest point of the GBM is not considered as an edge, but due to the flexibility of active contour model, we can still define the image force as  $-I(x, y)$ , so the active contour model can converge to the local minimum pixel. First we define an initial active contour. This contour is sampled about every 2 pixels. As in figure 16, we manually set the initial contour along the GBM.

We let the active contour iterate 500 times. For the internal force, parameter  $\alpha$  is set to 0.05,  $\beta$  is set to 0.01. Every iteration we resample the contour to make sure it is evenly distributed.

A result of estimating the position of darkest point of the GBM is shown in figure 17. The left figure is the active contour result after 500 iterations on the low molecule channel image. The right figure is the same result displayed on high molecule channel image.

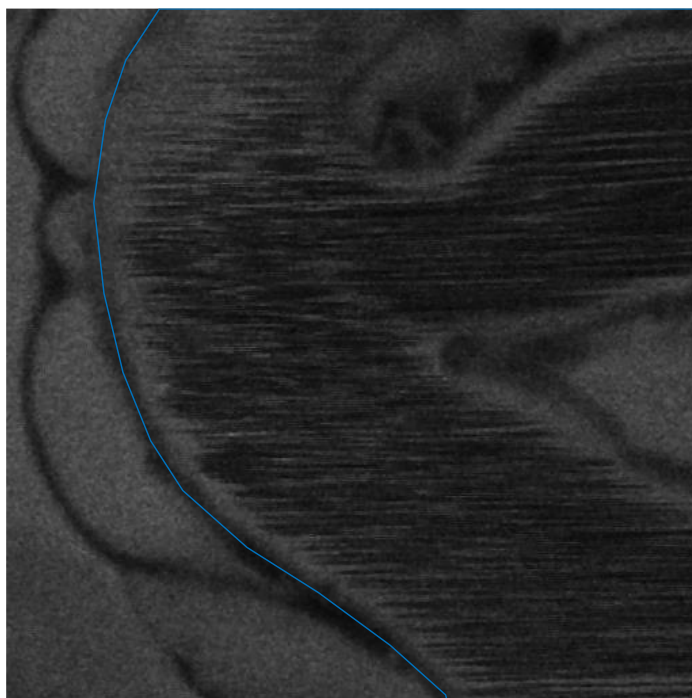


Figure 15. Initial contour along the GBM

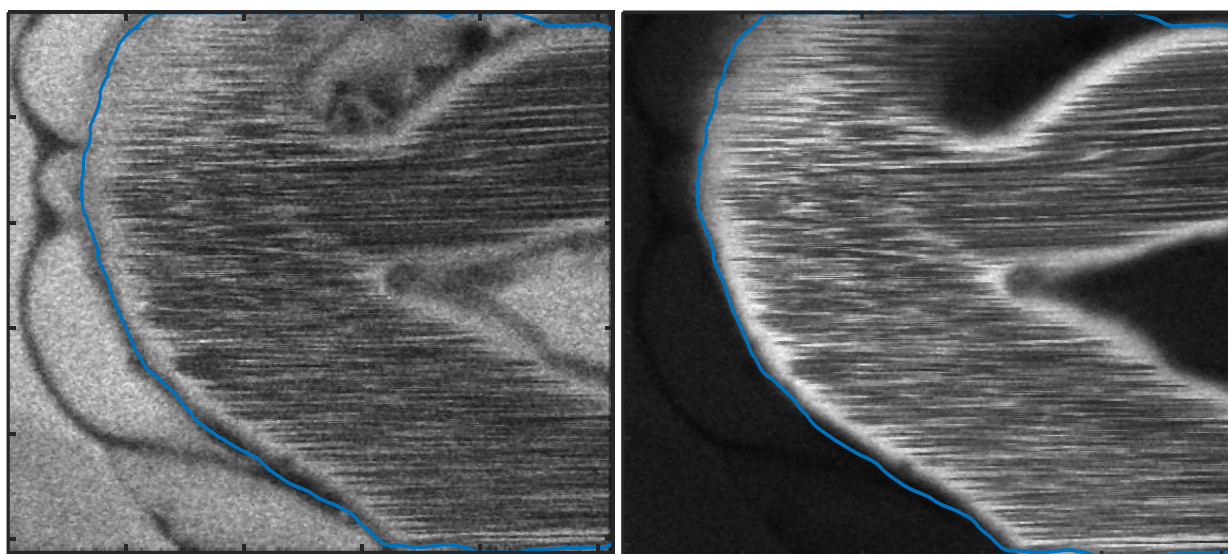


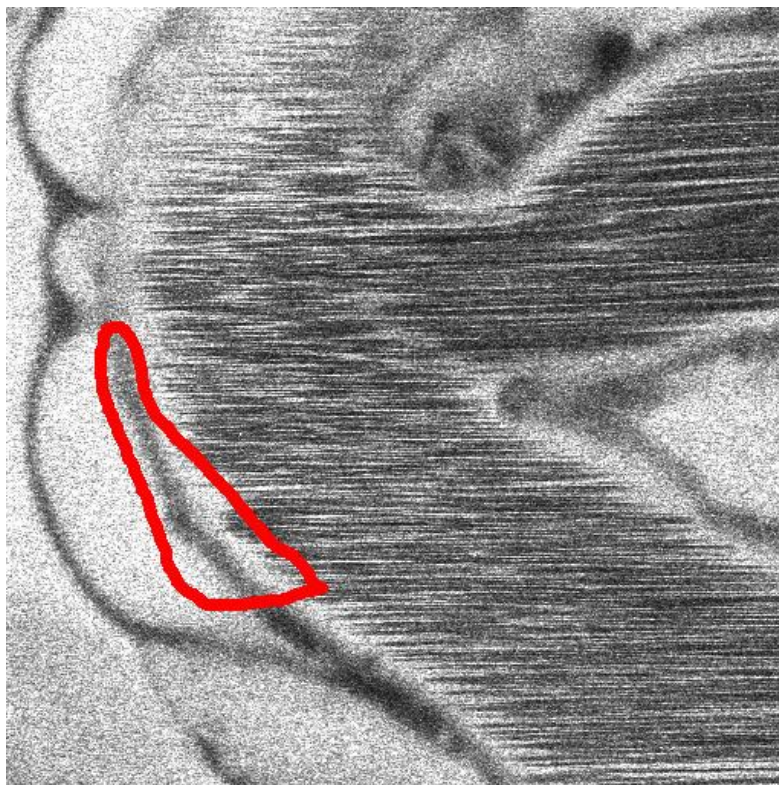
Figure 16. A result of estimating the position of darkest point of the GBM. The left figure is the active contour result after 500 iterations on the low molecule channel image. The right figure is the same result displayed on high molecule channel image.



### Estimation of Profile across the GBM

We sampled around the estimated location of GBM to obtain the density profile on both high and low molecule images. The profile is obtained by interpolating normally to the final active contour on both direction. The interpolation is 1-pixel-width for a total 100 points. The sampling procedure was applied on a certain area of the GBM and the results were averaged.

Figure 18 shows an example of a good sampling position.



*Figure 17. An example of sampling position*

Figure 19 shows a sampling map.

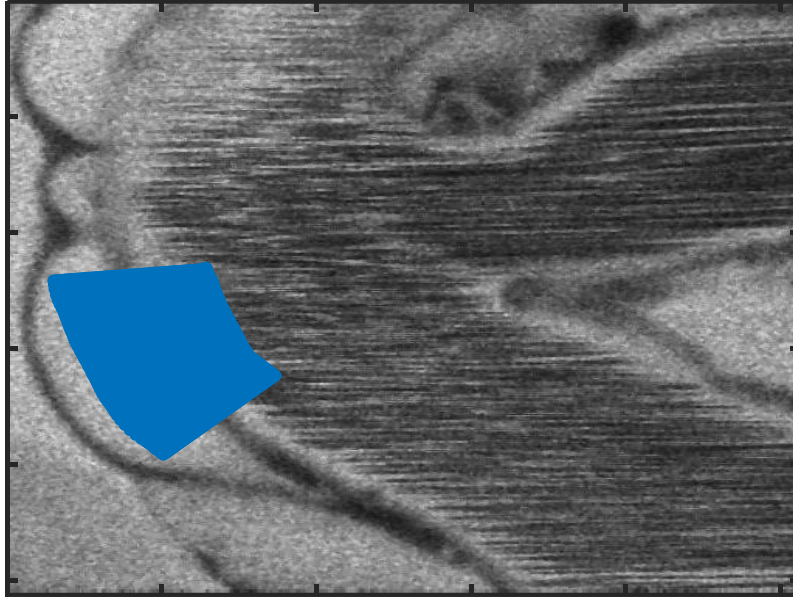


Figure 18. Sampling map. It was sampled normally to the final active contour on each direction for 50 points.

The average result in figure 19 is shown in figure 20. The blue curve is the low molecule channel (CH2) result and the red curve is high molecule channel (CH3) result.

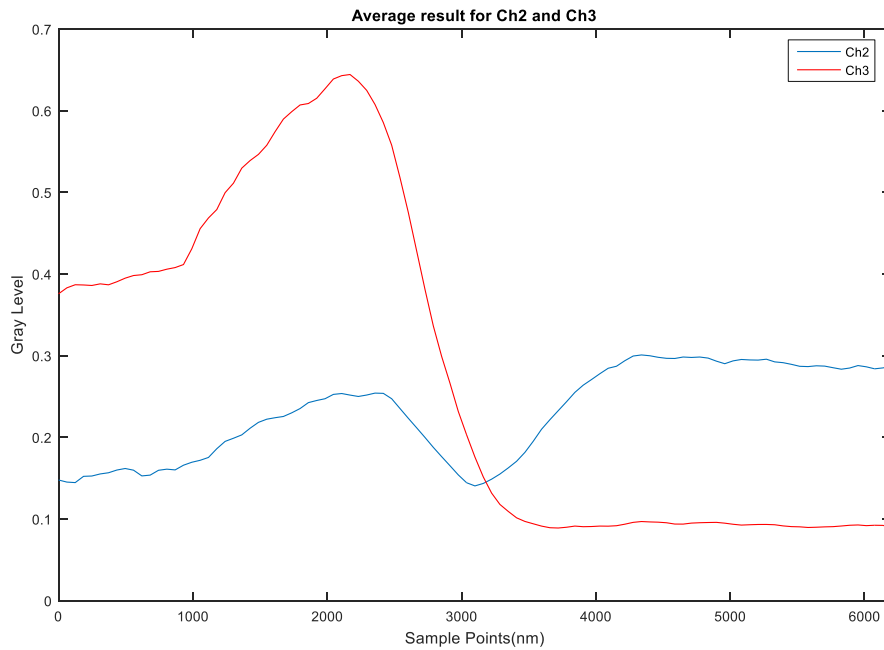


Figure 19. Average of the sampling result

A flow chart of the extraction procedure is described in figure 21.

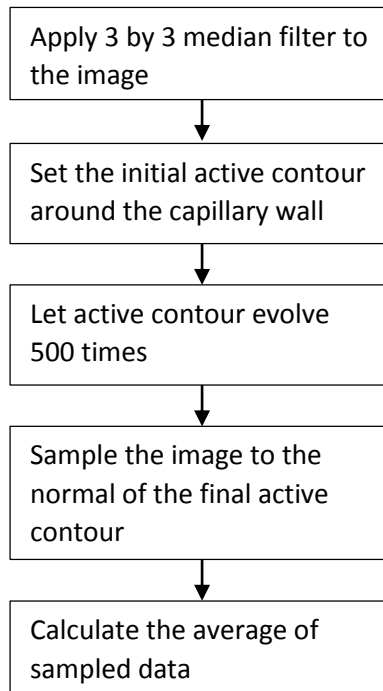


Figure 20. The flow chart of estimating the profile across GBM

## Estimation Result

We measured 5 images from a non-proteinuric set and 5 images from a proteinuric set. The average result of each image is shown in figure 22 to figure 32. In each figure the right top figure is the final active contour on low molecule channel (CH2), the left top figure is the final active contour on high molecule channel (CH3), the right bottom figure is the sampling map, and the left bottom figure is the average result.

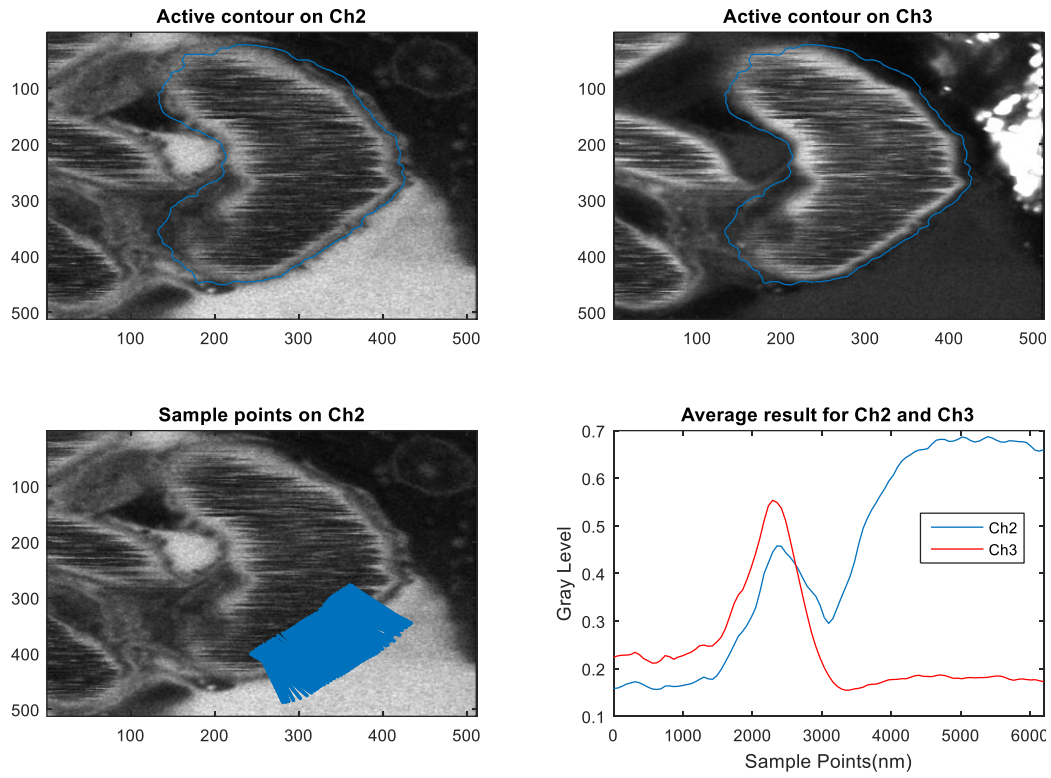


Figure 21

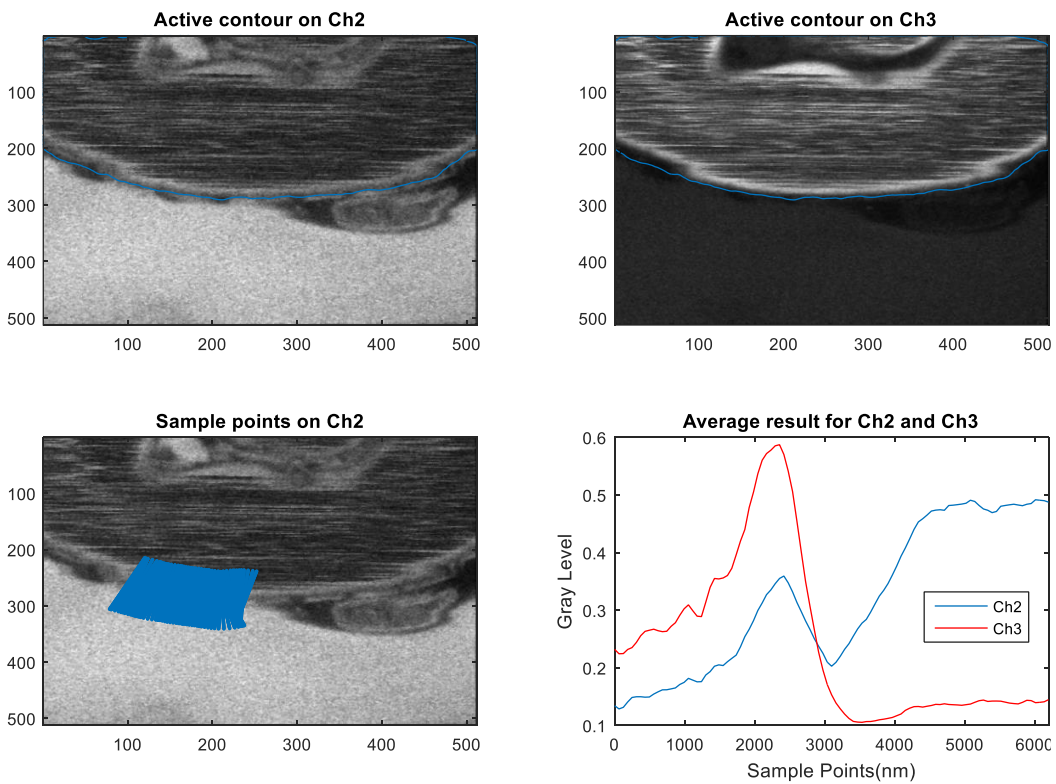


Figure 22

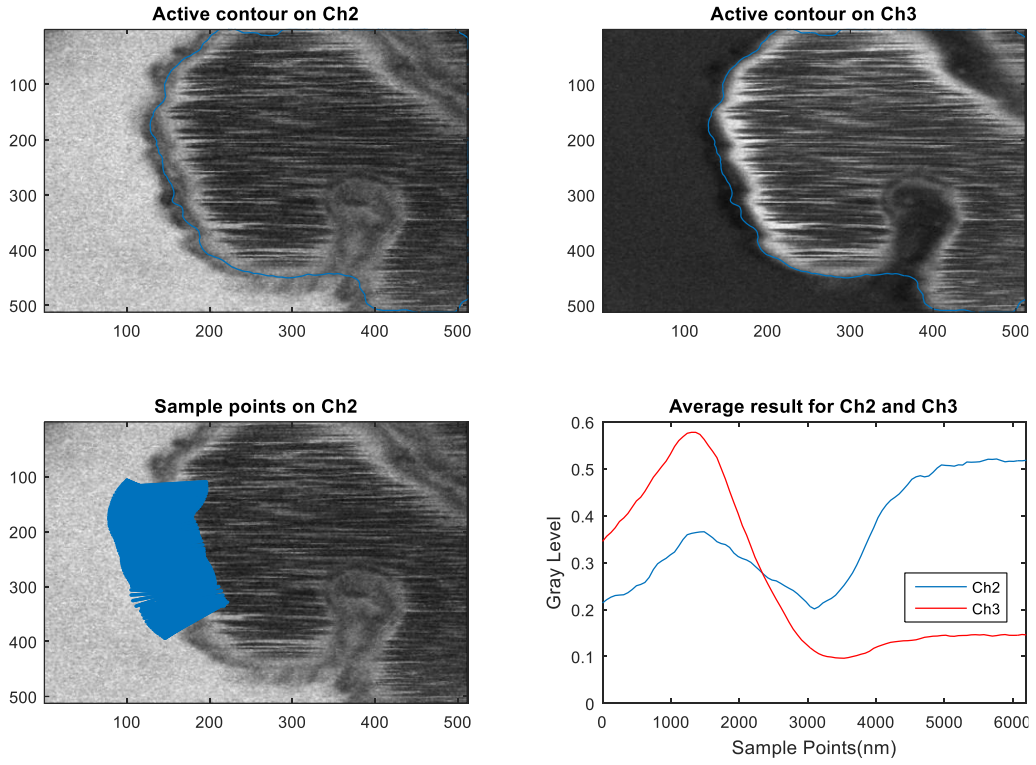


Figure 23

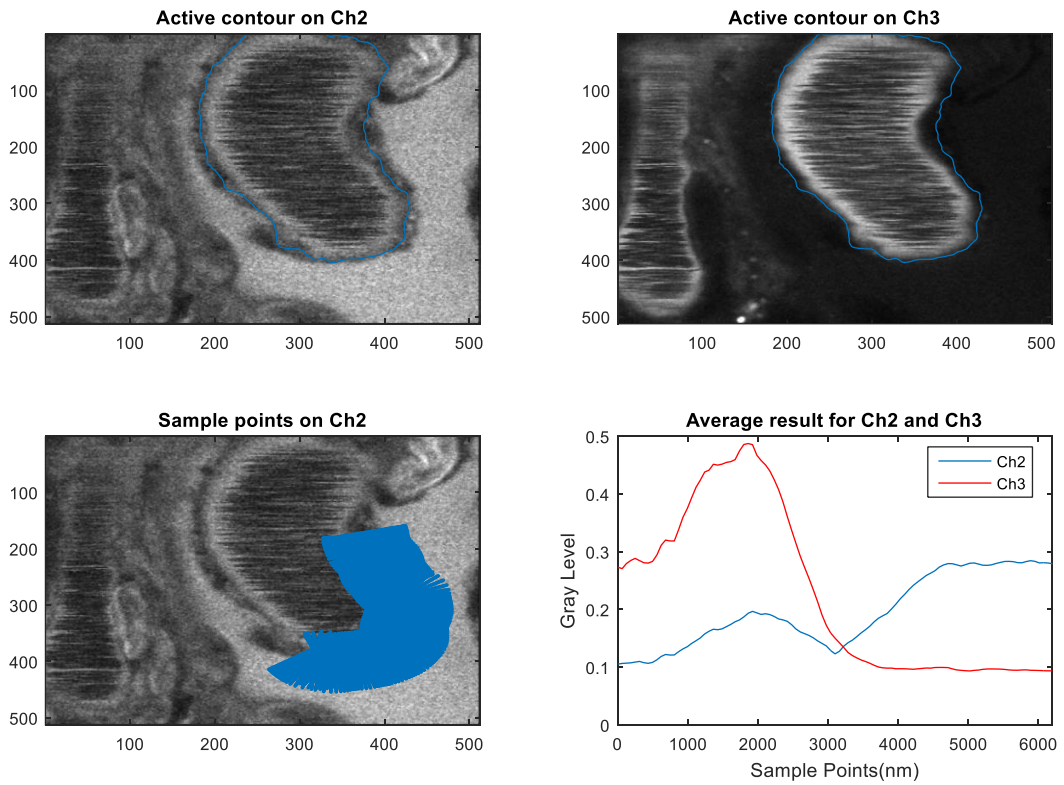


Figure 24

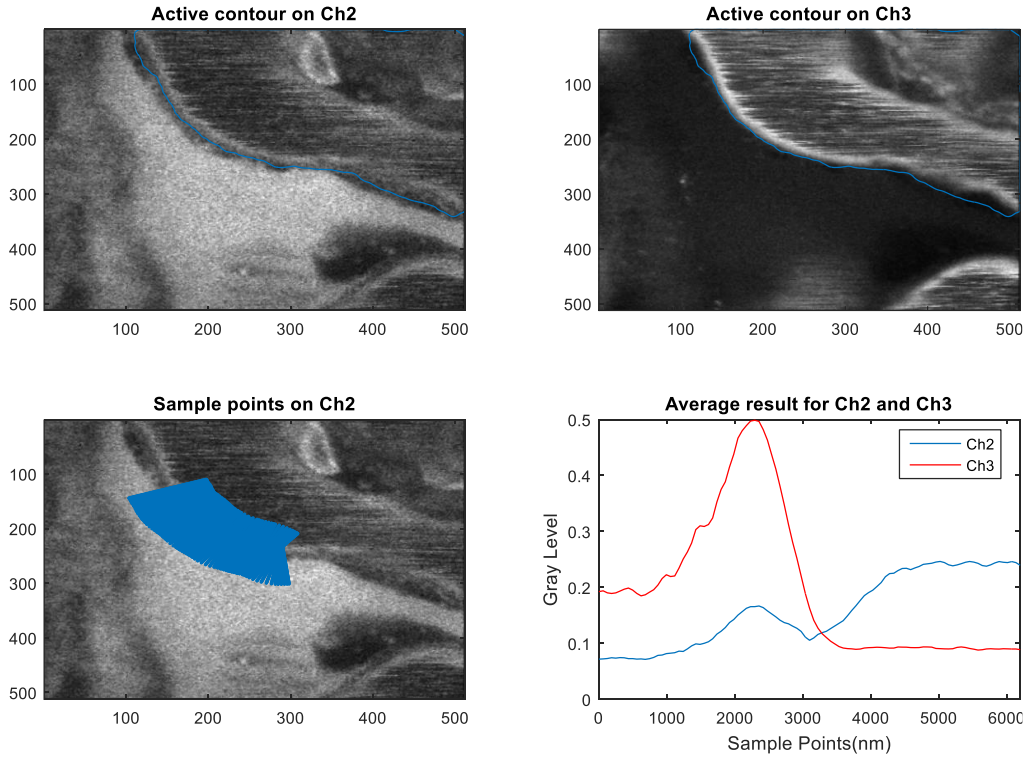


Figure 25

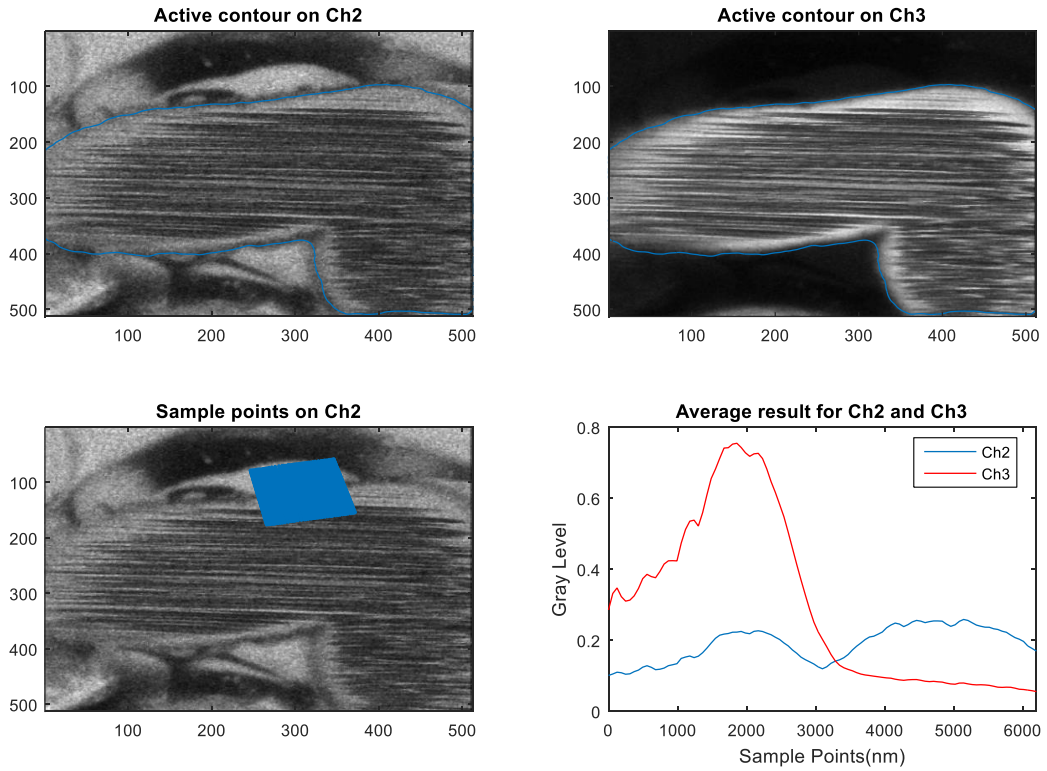


Figure 26

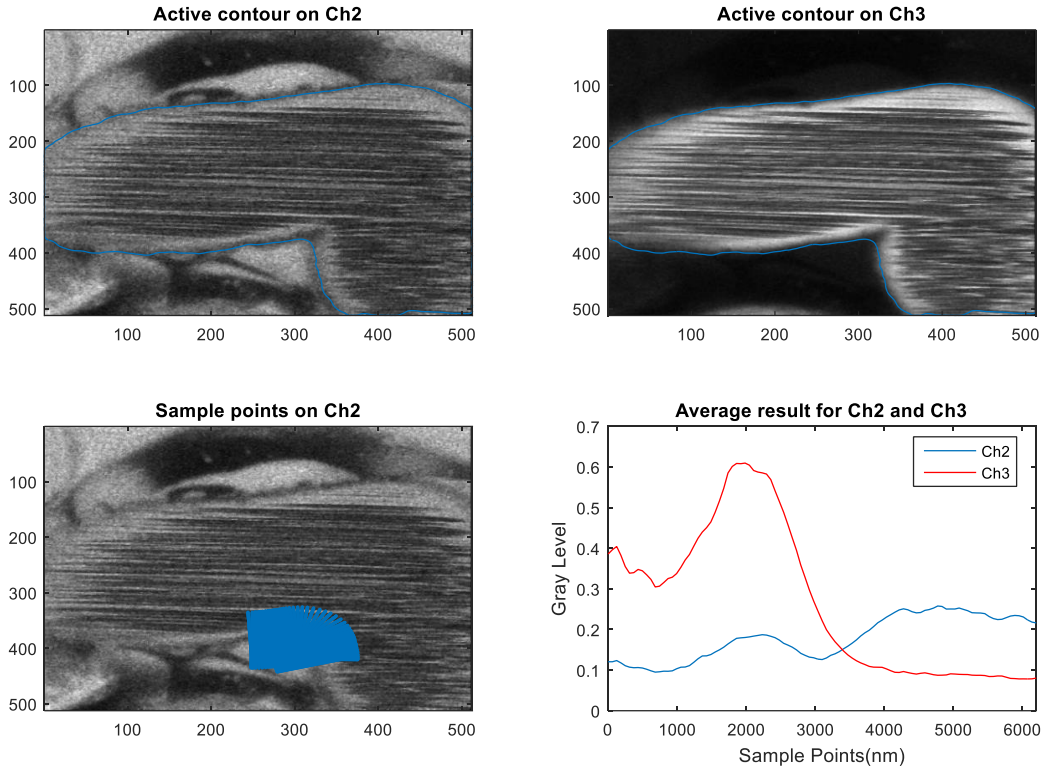


Figure 27

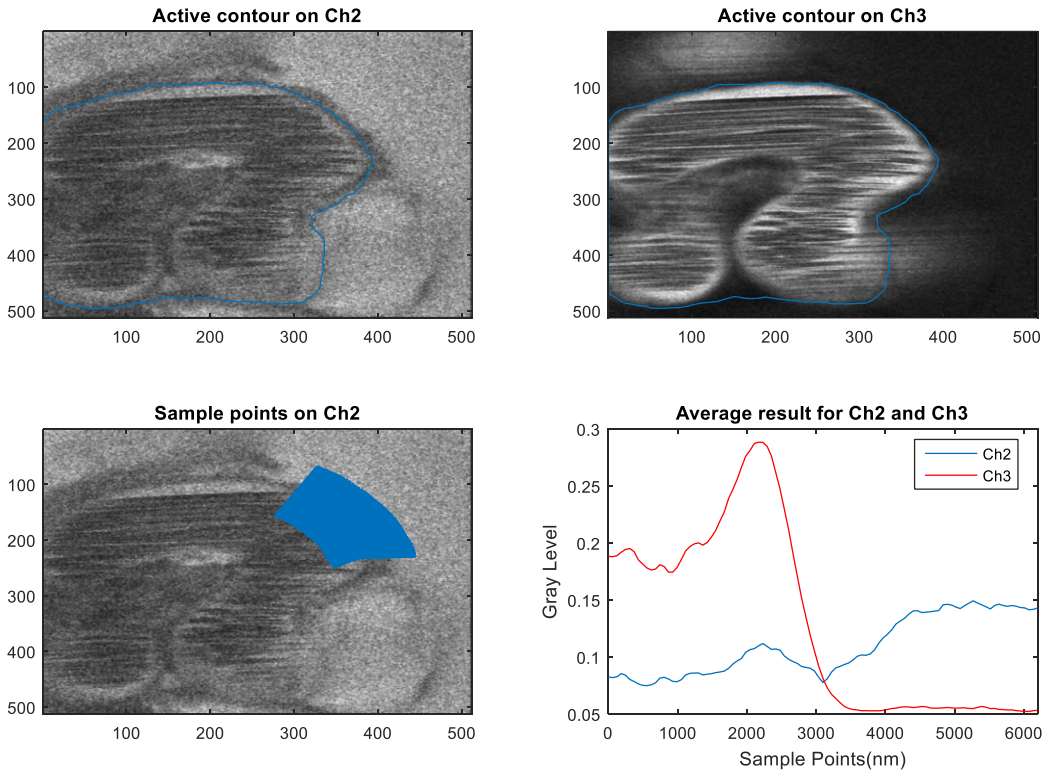


Figure 28

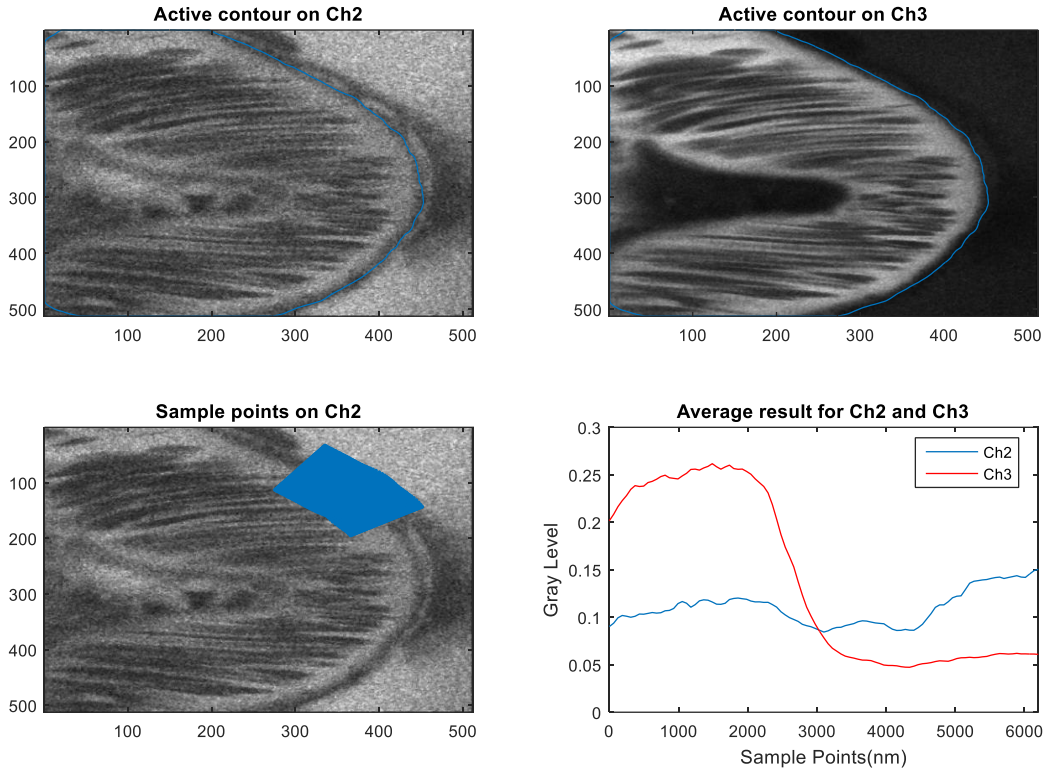


Figure 29

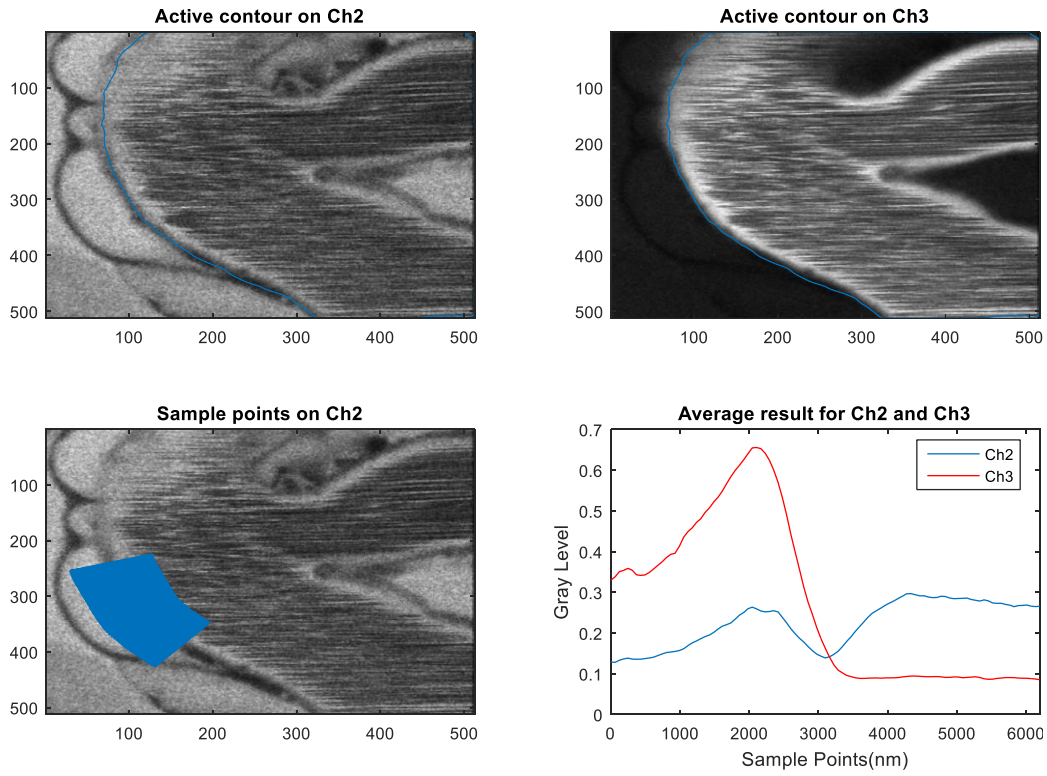


Figure 30



## **Chapter V**

### **SUMMARY**

Medical image processing has become a popular research field in the computer vision. This paper provides a method based on computer vision to measure the passage of various endogenous proteins across the GBM.

We introduce the basics about the linear filter, including the mean filter and the Gaussian filter, and the non-linear filter, including the unweighted median filter and the weighted median filter. We also compare the edge preservation capability between the linear-filter and the non-linear filter.

We introduce the basics about the traditional active contour model and some variants including the Balloon force model, the GVF snake model and the VFC snake model. We also compare the capability of capturing the concave regions, convergence speed and noise resistance between these models.

We select an appropriate sized median filter to preprocess the images obtained by confocal microscopy, thus reduces the noise and preserve the edge information.

We use the active contour model to capture the location of the GBM. By setting the external energy to image itself, we are able to capture the darkest point inside the GBM.

We select desired area after we get location of the GBM, then sample to the normal of the GBM in both directions. After averaging the sample result, we successfully obtain the profile of both low and high molecule channel for future study.

More images need be studied to get a reliable result for future study.

## REFERENCES

- [1] Miner, Jeffrey H. "The glomerular basement membrane." *Experimental cell research* 318.9 (2012): 973-978.
- [2] Arias-Castro, Ery, and David L. Donoho. "Does median filtering truly preserve edges better than linear filtering?." *The Annals of Statistics* (2009): 1172-1206.
- [3] Ma, Ziyang, et al. "Constant time weighted median filtering for stereo matching and beyond." *Proceedings of the IEEE International Conference on Computer Vision*. 2013.
- [4] Cohen, Laurent D. "On active contour models and balloons." *CVGIP: Image understanding* 53.2 (1991): 211-218.
- [5] Fujigaki, Yoshihide, et al. "Intra-GBM site of the functional filtration barrier for endogenous proteins in rats." *Kidney international* 43.3 (1993): 567-574.
- [6] Bhumika Gupta<sup>1</sup>, Shailendra Singh Negi. "Image Denoising with Linear and Non-Linear Filters: A REVIEW." *IJCSI International Journal of Computer Science Issues* (2013): 1694-0784
- [7] R. Gonzalez and R. Woods *Digital Image Processing*, Addison-Wesley Publishing Company, 1992, p 191.
- [8] Li, Bing, and Scott T. Acton. "Active contour external force using vector field convolution for image segmentation." *Image Processing, IEEE Transactions on* 16.8 (2007): 2096-2106.
- [9] Zhu, Youlian, and Cheng Huang. "An improved median filtering algorithm for image noise reduction." *Physics Procedia* 25 (2012): 609-616.
- [10] Kass, Michael, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models." *International journal of computer vision* 1.4 (1988): 321-331.

[11] Xu, Chenyang, and Jerry L. Prince. "Gradient vector flow: A new external force for snakes." *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on.* IEEE, 1997.

## Appendix I

### Matlab Codes

```
Function [x,y]=snake(image,Niter,nsample,alpha,beta,gamma,extcoef)
    [n,m] = size(image);
    fx=zeros(n,m);
    fy=zeros(n,m);
    [dGx,dGy]=gradient(image); % define external forces
    fx=-extcoef*dGx;
    fy=-extcoef*dGy;
    [xi,yi] = getline ('closed'); % Get initial curve
    [N,~]=size(xi);
    t = 1:N;
    ts = (1:(N-1)/nsample:N)';
    x = interp1(t,xi,ts);
    y = interp1(t,yi,ts); % Interpolate initial curve
    x=x(1:end-1);
    y=y(1:end-1); % Erase the last point cause it is the same point with
    x(1)
    [N,~]=size(x);
    % Compute diagonal matrix A in snake algorithm,
    A=diag((6*beta+2*alpha)*ones(N,1))+...
        diag((-4*beta-alpha)*ones(N-1,1),1)+...
        diag((-4*beta-alpha)*ones(N-1,1),-1)+...
        diag((beta)*ones(N-2,1),2)+...
        diag((beta)*ones(N-2,1),-2)+...
        diag((beta)*ones(2,1),N-2)+...
        diag((beta)*ones(2,1),-N+2)+...
        diag((-4*beta-alpha)*ones(1,1),N-1)+...
        diag((-4*beta-alpha)*ones(1,1),-N+1);
    for i=1:Niter
        j = find(x<=1);
        x(j)=1;
        j = find(x>=m);
        x(j)=m;
        j = find(y<=1);
        y(j)=1;
        j = find(y>=n);
        y(j)=n;
        % Codes above makes set the curve out of the image at the boundary of
        the
        % image
        fxc = interp2(-fx,x,y);
        fyc = interp2(-fy,x,y); % Interpolate the external energy at current
        % curve
        dx=[x(2:end);x(1)]-[x(end);x(1:end-1)];
        dy=[y(2:end);y(1)]-[y(end);y(1:end-1)]; % Compute the distance in each
        % direction
        x=(gamma*A+diag(ones(1,N)))\ (x+gamma*(fxc)+balcoef*(-
        dy)./sqrt(dx.*dx+dy.*dy));
        y=(gamma*A+diag(ones(1,N)))\ (y+gamma*(fyc)+balcoef*dx./sqrt(dx.*dx+dy.*
        dy));
        % Compute new curve
        x = [x;x(1)];
        y = [y;y(1)]; % Make the curve close
```

```

d = sqrt(dx.*dx+dy.*dy); % Compute the distance between nearby points
d = [0;d];
d = (d'*triu(ones(length(d),length(d))))'; % Compute the distance of
each point to the first point
curvelength=max(d);
p = 0:curvelength/nsample:curvelength; % Define the resolution
x = interp1(d,x,p);
y = interp1(d,y,p); % Interpolate the points
x= x(1:end-1)';
y= y(1:end-1)';
% the codes above make the points uniformly distributed when the snake
% involving

% Image2 is the low molecule image, image3 is the high molecule image
Image2=double(image2);
Image2=image2/4095; % The original image1 is .tif file which means its gray
% level is from 0 to 4095, normalized it from 0 to 1
image2=medfilt2(image2,[3,3]); % Smoothed by 3 x 3 median filter
image3=double(image3);
image3=image3/4095;
image3=medfilt2(image3,[3,3]);

[m,n]=size(image2);

% parameters
Niter=500; % This is the number of iteration
nsample=1000; % This is how many points you get in snake curve
alpha=0.05; % Increase alpha will increase the internal energy of the snake
% as it stretches more and more
beta=0.01; % Increase beta will increase the internal energy of the snake as
% it develops more curves
gamma=10; % This is the step length of snake computation. The larger the
% snake converges faster

extcoef=-1; % This is the external force
[x,y]=snake(image2,Niter,nsample,alpha,beta,gamma,extcoef);
% Above is snake function

theta=pi/2; % this is the angle to perpendicular direction you want to sample
% with.
interv=7; % this is how many points of snake curve you wish to use in
% calculating the sample direction
dx=[x((interv+1)/2:end);x(1:(interv-1)/2)]-[x(end-(interv-3)/2:end);x(1:end-
(interv-1)/2)];
dy=[y((interv+1)/2:end);y(1:(interv-1)/2)]-[y(end-(interv-3)/2:end);y(1:end-
(interv-1)/2)];
% dx,dy is the tangents of the curve which will use in computing the sample
% points
for i=1:length(x)
    if dx(i)>0&&dy(i)>0
        xi(i,:)=dx(i)./abs(dx(i)).*(cos(atan(dy(i)./dx(i))+pi-theta))*(-
50:1:50)+x(i)*ones(1,101);
        yi(i,:)=dy(i)./abs(dy(i)).*(sin(atan(dy(i)./dx(i))+pi-theta))*(-
50:1:50)+y(i)*ones(1,101);
    end
end

```

```

elseif dx(i)>0&&dy(i)<0
    xi(i,:)=-dx(i)./abs(dx(i)).*(cos(atan(dy(i)./dx(i))-theta))*(-
    50:1:50)+x(i)*ones(1,101);
    yi(i,:)=dy(i)./abs(dy(i)).*(sin(atan(dy(i)./dx(i))-theta))*(-
    50:1:50)+y(i)*ones(1,101);
elseif dx(i)<0&&dy(i)>0
    xi(i,:)=-dx(i)./abs(dx(i)).*(cos(atan(dy(i)./dx(i))-theta))*(-
    50:1:50)+x(i)*ones(1,101);
    yi(i,:)=dy(i)./abs(dy(i)).*(sin(atan(dy(i)./dx(i))-theta))*(-
    50:1:50)+y(i)*ones(1,101);
elseif dx(i)<0&&dy(i)<0
    xi(i,:)=-dx(i)./abs(dx(i)).*(cos(atan(dy(i)./dx(i))-theta))*(-
    50:1:50)+x(i)*ones(1,101);
    yi(i,:)=dy(i)./abs(dy(i)).*(sin(atan(dy(i)./dx(i))-theta))*(-
    50:1:50)+y(i)*ones(1,101);
elseif dx(i)==0&&dy(i)>0
    xi(i,:)=-(-50:1:50)+x(i)*ones(1,101);
    yi(i,:)=y(i)*ones(1,101);
elseif dx(i)==0&&dy(i)<0
    xi(i,:)=(-50:1:50)+x(i)*ones(1,101);
    yi(i,:)=y(i)*ones(1,101);
elseif dy(i)==0&&dx(i)>0
    xi(i,:)=x(i)*ones(1,101);
    yi(i,:)=(-50:1:50)+y(i)*ones(1,101);
elseif dy(i)==0&&dx(i)<0
    xi(i,:)=x(i)*ones(1,101);
    yi(i,:)=-(-50:1:50)+y(i)*ones(1,101);

    end
end
% Above xi, yi is the sample points which are 500 points on each side of
% snake curve. It is calculated in 4 cases which the curve goes
% different directions

% Above codes will delete points whose sample points go out of the image2.

[px,py]=ginput(2); % You select two points here and get the sample results
between these two points
dpx1=(x-px(1)).^2+(y-py(1)).^2;
dpx2=(x-px(2)).^2+(y-py(2)).^2;
pn1=find(dpx1==min(dpx1));
pn2=find(dpx2==min(dpx2));
% Above codes find the nearest points of your selected points on the snake
% curve
if pn2>pn1
    xi=xi(pn1:pn2,:);
    yi=yi(pn1:pn2,:);
else
    xi=[xi(pn1:end,:);xi(1:pn2,:)];
    yi=[yi(pn1:end,:);yi(1:pn2,:)];
end
xin=xi;
yin=yi;
[i,~]=find(xi>n);
xin(i,:)=[];
yin(i,:)=[];

```

```

[i,~]=find(xi<1);
xin(i,:)=[];
yin(i,:)=[];
[i,~]=find(yi>m);
xin(i,:)=[];
yin(i,:)=[];
[i,~]=find(yi<1);
xin(i,:)=[];
yin(i,:)=[];
% Above is the codes solve the problem of the x(1) and y(1) lies between your
% start and end points
fgh=length(xin(:,1));
xp=xin(:);
yp=yin(:);

% interpolation and average.
interpimage3=interp2(image3,xin,yin);
interpimage3=mean(interpimage3);

interpimage2=interp2(image2,xin,yin);
interpimage2=mean(interpimage2);

% interpimage2 and interpimage3 are the average result

```