Endosome Detection in Cell Images

Master Thesis

by
GAO JIONG

In
Department of Computer Science
School of Computing
National University of Singapore

Supervisor: Dr. Lee Mong Li

April 2006

Abstract

Detecting the movement of endosomes after the pharmacological treatment to cells is an interesting topic in pharmacology research. This study seeks to provide a comprehensive and objective characterization of the changes with respect to the intensity of cell cytoplasm and number of endosomes within a cell. Previous works have demonstrated that some automated methods can detect certain types of cells in fluorescence microscope images with high accuracy. However, cells in microscope images are tend to overlap with blur edges and noises. The existing methods are not effective enough to detect the endosomes and cell outlines for our cell images. Thus in this thesis, we defined a set of metrics to measure the endosomes in cells. Then we propose a method based on edge detection, machine learning and active contour modeling to detect the endosomes in the cells and locate those detected endosomes by cells. Based on our method, we implement a tool which can assist biologists to compute the metrics of each cell easily and quickly.

Table of Content

1 Introduction

Detailed knowledge of the changes of cells after pharmacological treatment is critical to a full understanding of its function. Fluorescence microscopy, with the method of fluorescence tagging, is the most active method to detect such changes. Biologists usually use microscopy images to discover diseases, protein changes, cell movements etc. However, there is an obvious problem of examining microscopy images by human. This is because when biologists examine the microscopy images, they are relying on their experience and knowledge. The result can not be repeated by other investigators. The process is also very time and labor consuming as the number of images increases. Therefore we aim to develop a method which can process such microscopy images quickly and effectively. The following figure shows an example cell image we are going to analyze.

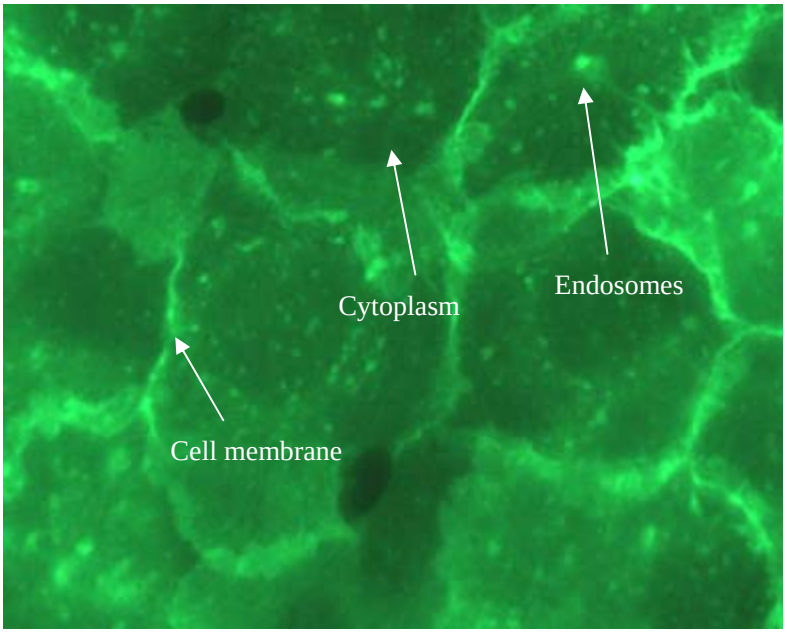


Figure 1: Cell image

Figure 1 shows a image with multiple cells. The proteins inside the cell are tagged by fluorescence techniques. Biologist puts drugs on the surface of cell. After

certain period of time, the drugs can move through the cell membrane which is a selectively permeable membrane into the cytoplasm. Then the tagged protein will become quite bright with the effect of drugs under the microscope. The microscopy images will show some relatively bright regions inside the cell, which are endosomes.

According to the pharmacological study, the intensity of cytoplasm (referred to microscopy image) and the number of endosomes in one single cell will be different under different treatments. Thus, our objective is to determine the intensity ratio of endosome and cytoplasm of a cell and the number of endosomes per cell.

1.1 Related Works

Endosome detection in cell images is a challenging task due to the complex nature of the cell tissue, and problems inherent to video microscopy. Object multiplicity, short range of grey levels, clutter, occlusion and non-random noise are some examples of the difficulties present in this kind of images. The diversity of cells also raises the difficulties of building up a universal solution in automatic cell segmentation problems. For example, the leukocyte and erythrocyte always have a consistent circle or elliptical shape with homogeneous intensity cytoplasm. Axon cells have very thick and clear cell membranes. Different neural cells have different protein sub-cellular patterns around their nucleolus. However, most of those cells on microscopy images share the following characteristics:

- No matter what kind of cell tissue is, there are cytoplasm and membrane for each cell. Cytoplasm has different intensity from membrane.
- The outline of all complete cells is an enclosed contour.
- The gradient at the edge of cell will sharply changed from the cell interior.

These characteristics are typically used as the basic features in cell image segmentation techniques. One common segmentation scheme is image thresholding [43, 48], which can be regarded as pixel classification. Other classical image segmentations include region-based segmentation, edge-based segmentation and etc. A good cell segmentation method always combines basic image segmentation techniques and achieves certain goals, such as track cell movement, monitor cell division, and etc.

Cell segmentation techniques for single cell analysis aim to classify the patterns of sub-cellular structures in fluorescence microscope images. Assessment of protein sub-cellular location is crucial to proteomics efforts since localization information provides a context for a protein’s sequence, structure, and function [50]. Therefore, an accurate recognition of the patterns of major sub-cellular structures is necessary to

biomedical researches. The purpose of single cell analysis is to classify different organic cells based on their interior proteins. Typically each image in single cell analysis has only one cell but with different sub-cellular protein structures presented in this cell. Therefore, several features of proteins are defined to classify different sub-cellular protein structures, such as the number of fluorescent objects in one cell, the average number of above-threshold pixels per object, etc. Since different organic cells have different sub-cellular protein structures, once the sub-cellular protein structure can be recognized, the cells can also be recognized. Many popular data mining techniques are applied in sub-cellular protein recognition, such as Support Vector Machine [11], neural networks [19], statistical classifier [38], etc. Our cell images are not directly applicable to the single cell analysis because there are multiple cells on each image. However, we can apply the protein recognition techniques used in single cell analysis to find out the endosomes on entire image, and then locate them by cells.

Cell segmentation techniques for multiple cells aim on cell tracking and cell outlining. The most systematic cell outlining method is Garrido's method [18], which uses the traditional morphological methodologies and Hough transform algorithm followed by deformable template model. Level-set [36] is another approach, which segments the cell images based on the intensity intervals and minimization energy functional. Another approach is to apply texture feature extraction method on cell images to get the texture information, and then followed by the thresholding to detect the abnormal regions [3, 23, 37]. Besides those main approaches, there are many other cell segmentation methods, such as mean shift [15], gradient vector [39], etc.

As we discussed in previous paragraphs, the protein recognition techniques, which are based on traditional image morphology and data mining techniques, can be applied to the endosome detection. On the other hand, the active contour algorithm used in multiple cell analysis can also be applied in our work to extract the cell outlines. Then we can locate the endosomes within a cell and compute the metrics for

each single cell.

1.2 Contribution

In this thesis, we propose a method which is based on Garrido's method. The first step is to apply Canny edge detector on cell image to get the Canny edge result. This Canny edge result contains two classes of edges: endosome edge and cell membrane edges, or cell boundaries. Then we define the features for those edges and apply the classification techniques to classify those edges into endosome edges and non-endosome edges. In the third step, we utilize the endosome edges to get the approximate cell locations. After we extract the approximate cell locations, we apply improved active contour algorithm to get the cell boundary for each cell. Finally, we can compute the metrics per cell.

In the following chapters, we first discuss the basic image segmentation techniques, such as edge-based segmentation, region-based segmentation, etc. Then we further analyze the details of some closely related previous research works done on cell image analysis. We will discuss Garrido's approach [18], level-set algorithm [36] and Gabor filter [14], and analyze these approaches. After related work discussion, we will describe our method which has 3 main steps:

1. Endosome detection with iterative training process.
2. Initial cell location detection.
3. Cell contour extraction.

In the experiment studies, we first show the performance of endosome detection with iterative training, and then compute the metrics by our method vs. the result obtained manually. Conclusion will be drawn after the experiment result, followed by the future work.

2 Related Works

Endosome detection in cell images is a quite new topic. There is no such literature found after a fair amount of search. However, many existing cell image analysis techniques can be utilized to solve this problem. Currently, there are a lot of works have been done on the cell image analysis [5, 13, 15, 17, 18, 25, 34, 36, 39, 40, 50], such as cell segmentation, cell tracking, sub-cellular recognition, tumor cell identification, etc. Those works involve traditionally image segmentation techniques, such as region-based or edge-based image segmentation, and advanced image segmentation techniques, such as texture extraction, pattern recognition, deformable template and etc.

In this chapter, we will first introduce the basic image segmentation techniques. After that, we will have a detailed discussion on the specific cell segmentation techniques.

2.1 Basic Image Segmentation Techniques

The principle goal of image segmentation is to partition an image into several regions that share some common features. Segmentation is very important in medical image processing and it has been used in many applications, such as vessel extraction, muscle measurements, bone classification, cancer pathology, tissue deformities, cell segmentations, etc. A wide variety of segmentation techniques has been proposed. However, there is no one standard segmentation technique can perfectly fit to all medical image problems. Different studies and different types of image data lead to different definition of the goal of segmentation. Therefore, different assumptions about the nature of images lead to different algorithm applied.

The most common used segmentation techniques can be classified as two classes: region-based algorithm and edge based algorithm. The former looks for the regions

that fit the requirement of segmentation, whereas the latter looks for the edges of target object.

2.1.1 Region-based techniques

Thresholding is a very common region segmentation method [43, 48]. In this technique, a threshold is selected and the image is divided into two groups. One group contains all the pixels with values higher than the threshold, and the other group is all pixels with lower values. However, direct thresholding approaches are not applicable to our cell images, because the grey level intensity of a cell image does not vary only on the boundary, but also within cells and throughout the background. In general, thresholding is not an effective method. The region-based thresholding is also not applicable, because not all of the parts of the same tissue are equally stained. Brighter background regions may be misclassified as endosomes and darker endosomes may be misclassified as background.

Region growing [1] is another commonly used region-based segmentation technique. It starts with a pixel or a group of pixels that belong to the structure of interest. Then the neighboring pixels are examined and “similar” pixels will be added to the growing region. The similarity can be defined in various ways, and the most common definition is the intensity homogeneity. The advantage of region growing is that it can correctly segment those regions that have the same properties and are spatially separated. However, this technique requires seeds for region growing, which can only be provided by an operator or some automatic seed finding procedure [53].

The watershed algorithm [7] is a region-based technique that utilizes image morphology. An initial seed for each object and the circle enclosing the area well outside the object are selected. The bright pixels can be considered as mountain tops and the dark pixels can be considered as valleys. Then some valleys are punctured and submerged with water. The water will start to fill the valleys until it flows outside the

circle or stops flow. In this technique, each point in the circle will be dropped by a drop of water, if this drop of water can flow to the exterior marker, then it will be considered as an exterior of object, otherwise, it is an interior.

The Tophat transform [16] is a morphological operation that uses the image opening or closing followed by subtraction. The endosomes actually are small bright regions on the relatively darker background. The shapes of endosomes are like circles or ellipses. Thus we can use a structure element that is larger than the extent of those regions to detect those endosomes. A structure element also called a kernel is a small rectangular grid that represents some basic shapes. For example, the structure element we used in the Tophat transform is a circle with radius of *n*. The following figure shows an illustration of a circle structure element with radius of 4 in 4x4 grids.

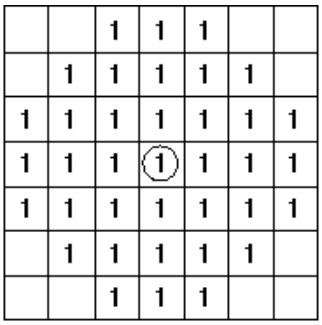


Figure 2: Structure element of circle with radius of 4

The image opening is a *Min* operation that removes those bright regions that are smaller in dimension than the structure element used in the operation. An opening is defined as erosion followed by a dilation using the same structure element for both operations. To compute the erosion of a binary input image by given structure element, we consider each of the foreground pixels in the input image in turn. For each foreground pixel (which we will call the *input pixel*) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates. If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background,

however, the input pixel is also set to background value. Dilation is the dual of erosion, i.e. dilating foreground pixels is equivalent to eroding background pixels. After applying image opening operation, we can just subtract the image with the thin peaks cut off from the original image and it gives you just those peaks plus some low amplitude noise.

2.1.2 Edge-based segmentation techniques

Region-based segmentation techniques are always based on pixel intensity, and edge-based segmentation techniques are based on local pixel intensity gradient. A gradient is defined as the approximation of the first-order derivative of the image. Since the digital images all consist of discrete pixels, the continuous differentiation is not applicable in digital images. However, most gradient operators use convolutions to differencing images in order to get the gradient map of original image. The most common used gradient operators are Roberts [21], Prewitt [24], Robinson [41], Krisch [41], and Frei-Chen [42].

Many edge detection methods use a gradient operator, followed by a threshold operation on the gradient, in order to decide whether a pixel is on the edge [4, 44]. Therefore, the output of the edge detector is always a binary image where the white pixels or lines indicate where the edges are. The edge-based segmentation techniques are computationally fast and do not require a priori information about image content. However, it requires the selection of threshold, which is a difficult task. On the other hand, thresholding will raise the problem of broken edges. This means the edges do not enclose the object completely due to the variety of object shape, color, light and etc. To form a closed boundary of an object, a post processing step is required, which is called edge linking.

The simplest approach of edge linking is to examine the neighboring edge pixels. If the edges have similar magnitude and direction, and the distance is close enough,

then a link can be established between these two edges. Generally speaking, edge linking is quite computationally expensive and not very reliable. One solution is to make the edge linking semiautomatic and ask a user to draw the edges when the automatic tracing becomes difficult. For example, Wang [46] developed a hybrid algorithm for MR cardiac cineangiography in which a human operator interacts with the edge tracing operation by using anatomic knowledge to correct errors.

The peaks in the first-order derivative correspond to zeros in the second-order derivative, therefore, people also can use second-order derivative to find the edges. The most common technique using second-order derivative is the Laplacian operator. It will make a transition through zero at the edge pixels. Therefore, it is also known as zero-crossing.

All edge detectors that are based on a gradient operator are very sensitive to noises. In most applications, a smoothing processing will be applied prior the edge detection in order to reduce the noise effect. Marr and Hildreth [33] proposed smoothing the image with a Gaussian filter before application of the Laplacian, also known as Laplacian of Gaussian. The advantage of Laplacian of Gaussian operator is that the edges of the objects are smoother and better outlined. Canny [6] proposed the same smoothing algorithm as Marr and Hildreth, but followed by a first-order derivative gradient operator.

2.2 Cell Segmentation Techniques

There are many cell segmentation techniques, such as Garrido [18], Mukeherjee [36], Ray [39], McInerney [34], Debeir [15], etc. Among those techniques, there are three main approaches, which are deformable template, level-set algorithm and texture feature extraction.

The deformable template model proposed by Garrido [18] is the most systematic method. The idea of this model is quite straightforward. Since every cell has membrane, and normally the cytoplasm inside the membrane appears darker or lighter than the outside environment. On the other hand, membrane also has different intensity from cytoplasm. So to extract single cells from a group of randomly distributed cells, they try to find the membranes first. After extracting the membranes, the cell outline can be drawn and approximate cell location can be found. A deformable template will be placed at each approximated cell location. With some preset criteria, those deformable templates will deform, grow and finally stop at the true membranes. In the end, each deformable contour will indicate a single cell.

Mukherjee *et al*. [36] detect and track leukocyte by applying level set algorithm. Level set algorithm segment the image into different regions according to the intensity at each pixel. Every pixel will fall in a region in which all the pixels have similar intensities. Thus, the image after level-set segmentation looks like a level map, which is where the term “level-set” comes from. Based on the layers, a minimization energy function is applied to each segment within one layer to get the segment with minimum energy value. After that, the segment with global minimum energy value will be selected as cell outline. However, the assumptions of this method are the leukocyte must be nearly circular and cytoplasm is almost intensity homogeneous.

Texture feature extraction is commonly used in the medical image feature extraction. One of the most popular signal processing based approaches for texture

feature extraction is the Gabor filters. Gabor filter enables texture feature filtering in the frequency and spatial domain. Turner [45] first implemented texture discrimination by using a bank of Gabor filters to analyze texture. A range of filters at different scales and orientations allows multi-channel filtering of an image to extract frequency and orientation information. Gabor filters are also used to model the response of the human visual system. Therefore, Gabor filter can be used to decompose the cell image into different sub-regions according to different texture features, such as different proteins, cell membranes, cell bond, etc.

Neural network is another popular approach of sub-cellular structures recognition in recent years. The proteins in cell can be considered as patterns. Since different proteins will have different features, therefore, those patterns in the microscopy images will have different appearances. Those features can be extracted by some classical image segmentation or morphology methodologies, such as thresholding, watershed, edge detector, etc. Some texture feature extraction techniques are also used to extract the object features, such as Gabor filter, Wavelet transform, etc. With those features, researchers can build up a neural network classifier by applying the latest data mining techniques. Besides neural network classifier, Support Vector Machines (SVM), decision tree, Bayesian classifier, statistical classifier, almost all the popular classifiers have been integrated into cell image analysis, and achieve quite good performance in certain fields.

There are some other methods which are proposed to solve certain cell image problems. Mean-shift algorithm is used to capture the changes of center point of a given region. An approach based on mean-shift algorithm is proposed by Debeir *et al*. [15], which is to track the process of migrating cell trajectories establishment through in vitro phase-contrast video microscopy. Fok *et al*. [17] use an elliptical Hough transform to roughly identify all the axon centers of nerve cells, and then apply active contour model to extract the boundaries of each axon. Ray uses a modified gradient vector flow, which is called motion gradient vector flow to track rolling leukocytes in

microscope.

In this section, I will first go through three main approaches, which are Garrido's method, level-set algorithm and Gabor filter approach. A full comparison and discussion on the Pros and Cons of those existing methods will be drawn in the end of this chapter.

2.2.1 Garrido's Method

To address the automatic cell segmentation problem, Garrido presented a novel method, which is based on the deformable template. The images used in this paper are cytology images, which are acquired through a CCD camera adapted to an optical microscope and stained with the Papanicolau technique. There are three main characteristics are presented in this paper:

- *An absence of high contrast*. It is well know that microscopical biomedical images have a short range of grey levels.
- *Many cluttered objects in a single scene*. A high number of overlapping objects makes image segmentation difficult
- *Low quality*. Traditional staining techniques like that of Papanicolau introduce a lot of in homogeneities into the images, where not all of the parts of the same tissue are equally stained.

Garrido designed an automatic, complete and systematic segmentation method for those cell images with problems such as a short range of grey levels, clutter, occlusion and non-random noises. There are three steps, cell edge detection, cell location detection and deformable template evolution. Figure 4 shows the flow chart of Garrido's method.

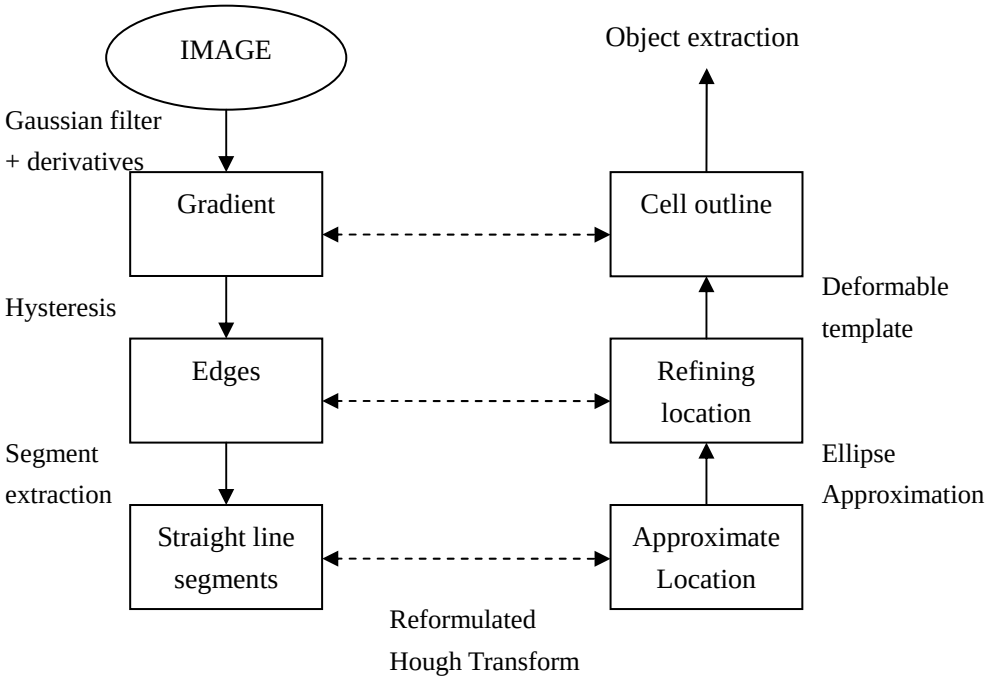


Figure 3: Flow chart of Garrido's method

The first step is to detect cell edges. The purpose of this step is to obtain the evidence of the cell locations. They use Canny edge detector [6], which is designed to be the optimal edge detector. It works in a multi-stage process. First of all the image is smoothed by Gaussian convolution, then Roberts Cross, which is a simple 2-D first derivative operator, is applied to the smoothed image. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks those ridges with control of two thresholds. The detail of Canny edge detector will be further discussed in next chapter.

Before starting the locating process, they do a post-process to the edges. The post-process consists of preparing the chains and determining the location of the straight line segments. Both processes are quite straightforward. They just remove the joint point of every edge. Then if the maximum distance between each of the points along the chain and the given straight line segment is less than a given threshold, this chain is considered as corresponds to this straight line segment.

In step 2, Hough transform [2, 26] is applied to the edge image to estimate the location of cell center. They use an octagon with equal length of sides as the segment to define a circle, which is shown in the following figure:

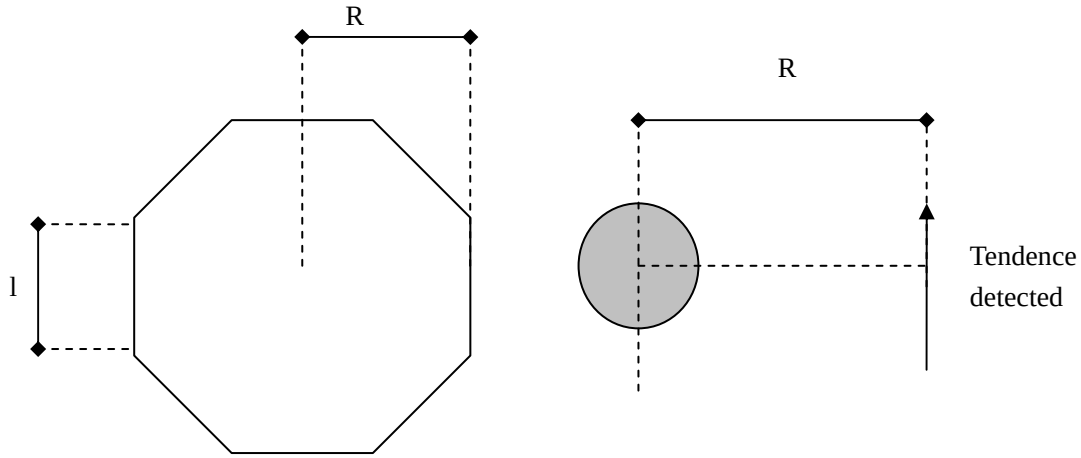


Figure 4: Segments to define a circle

With a shape defined by n segments r_i of length l_i $(0 < i < n+1)$. If m_i segments of length L_{i_j} detected corresponding to tendence r_i and referencing position p in the Hough transform accumulator, the value of evidence $E(p)$ can be calculated as follows:

$$E(p) = \sum_{i=1}^{n} a_i \max_{1 \le j \le m_i} \left\{ \frac{\min\{L_{i_j}, l_i\}}{l_i} \right\} \tag{1}$$

$$\sum_{i=1}^{n} a_i = 1 \tag{2}$$

where p is any pixel in the image. l_i is the length of octagon's side, which is equal to each other from i to n. m_i is the chains considered as corresponding to a given tendence r_i. Thus this formula is saying to get the evidence value at pixel p, we can draw a octagon centered at p, then find the chain segments detected in first step corresponding to the eight sides of this octagon. After that, we find the longest matched chain segment for each side, and times the coefficient a_i and sum up them to get the evidence value. Those evidence values constitute the parameter space. After setting simple threshold to the parameter space, the estimated cell center can be obtained.

The last step is to apply a deformable template model to find the real cell boundary. They use a deformable template with global shape constraints, which was proposed by Grenander [22, 31]. They define an external function involves of the stable edges and image gradients.

This model is effective to the images with homogenous intensity in cytoplasm and with elliptical shapes of cell. However, for our cell images, there are a lot of endosome regions inside the cell, thus after applying canny edge detector, there will be many false edges detected inside cells. Those false edges actually are endosomes, and they can confuse Garrido's model. Another problem of this model is the Hough transform they used in this paper. They will calculate every pixel to construct parameter space, which takes a lot of time to process. Fok [17] uses the same procedures as Garrido, but the difference is Fok's image contains some interior noises and a very sharp and thick cell boundary. Therefore Fok do not need to concern cell boundary detection very much, and he just uses the standard active contour algorithm. So we are not going to discuss Fok's model in details.

2.2.2 Level Set Algorithm

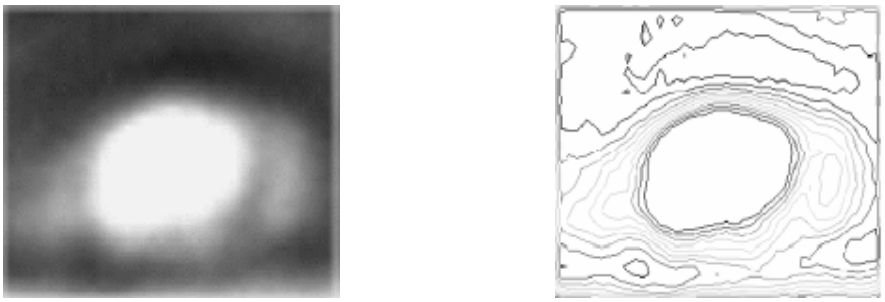
Level-set algorithm is a new approach in cell segmentation field. In mathematics, a level-set of a real-valued function *f* of *n* variables is a set of the form:

$$\{(x_1,...,x_n) \mid f(x_1,...,x_n) = c\} \tag{25}$$

where *c* is a constant. That is, it is the set where the function takes on a given constant value. When the number of variables is two, it is called level curve or contour line. It is a curve connecting points where the function has a same particular value. The advantage of the level set method is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects. Also the level set method makes it very easy to follow shapes which change topology, for example when a shape splits in two, develops

holes, or the reverse of these operations. All these make the level set method a great tool for modeling the geographical objects. The medical images are always in grey level. Therefore people also can apply level-set algorithm by assuming those medical images as geographical images.

Mukeherjee's proposed a level-set based method [36], which is designed to detect the leukocyte and also track the movement of detected leukocyte. Since our images are not live cell images, so we do not need to concern about the tracking part, the interest part is only the detection of leukocyte. Level set morphology in leukocyte image segmentation refers to the binary umbra extracted from the image using a threshold decomposition of particular image intensity level. The leukocyte and level lines of this leukocyte are shown in Figure 8. Naturally, the binary umbra contains of collection of connected components that constitute objects in the image. The boundaries of these connected components are referred to as level lines. Each intensity level may have several connected components. Certainly, the leukocyte shape profile is embedded in any one or many of these level lines.



(a) (b)

Figure 5: (a) leukocyte. (b) level lines of leukocyte

Mukeherjee proposed level-set based algorithm is because they assume two specific features of their leukocyte's cell intensity profiles always hold:

1) a typical boundary envelope in which the intensity profile is different from the cell cytoplasm and from the background, if not the entire boundary but at least for a significant part of the border;

2) the leukocyte shapes are nearly circular, except for teardrop-like deformation

encountered when in contact with the endothelium [13].

Therefore, it is necessary to define an energy functional which can find the shape embedded in the level lines. To achieve this target, they consider detecting homogeneous regions with distinct boundary as the placement of a closed curve that maximizes image gradient at its boundary and intensity homogeneity for its interior. Given a parameterized curve $C_i(s) = [X(s), Y(s)]$, $s \in [0,1]$, that separates objects from the background, the energy functional for leukocyte capture should minimize the following function:

$$E(C_i) = -\int_0^1 g(|\nabla I|)ds - \kappa \iint_{\wp(C_i)} H(x,y)dxdy + \iint_{\wp(C_i)} \left(\sum_{j=1, j\neq i}^{N} \chi_j(x,y) \right) dxdy \qquad (26)$$

Here the first term $\int_0^1 g(|\nabla I|)ds$ integrates image gradient along the curve C_i. If this value is high, then it means the gradients on the curve are high. High gradient means sharp changes of the intensity, which is an indication of cell boundary. With a negative sign, this term can be minimized.

The second term represents the homogeneity of the image region $\wp(C_i)$, where *H(x, y)* is defined as following:

$$H(x,y) = -\frac{(I(x,y)-\mu)^2}{2\sigma^2} \qquad (27)$$

(x, y) is the coordinate of pixels inside the closed curve, *I(x, y)* represents the intensity of this pixel, and μ is the intensity mean of this curve, σ is the intensity variance of this curve. If the cell interior is not homogenous, then the variance of interior should be high. Therefore the accumulated intensity difference between each pixel and average intensity value will also increase. With a negative sign, this value also can be minimized.

They also assume the leukocytes are not overlapping to each other, therefore the

curves representing leukocytes can neither be intersecting nor circumscribed into one another. This assumption is represented as the third term in equation (27). The function X_j is the characteristic function for the j^{th} curve representing a leukocyte boundary and is defined as:

$$\chi_j(x,y) = \begin{cases} 1, & if\ (x,y) \in \wp(C_j) \\ 0, & otherwise \end{cases} \tag{28}$$

$\wp(C_j)$ is the region bounded by curve C_j and N is the total number of leukocytes detected in the image. If a pixel (x, y) belongs to multiple curves delineating potential cells, $\sum \chi_j$ increases. The summation is minimized in the case that there is no overlap between cell boundaries. Small value means highly possibility of this component being on top of all other overlapping component.

After define the energy functional, it is time to design the minimization algorithm. Since the image is segmented by level-set algorithm, so each layer represents an image that contains a lot of connected components. If we superimpose one layer on top of another layer, we can find a lot of overlapping connected components. For the overlapping components, Mukeherjee assigned them same label. So the problem became how to find the minimum energy functional component with same label.

The algorithm proposed by Mukeherjee is designed as follows:

1. First eliminate subscale and above-scale components from original image.
2. A set of level sets that contains all connected components are extracted from the image got from step 1.
3. For different level sets, label the overlapping components with the same index.
4. Calculate energy functional value for each component.
5. For components with same label, find the one with minimum value.

Thus those components with minimum values are the cells they wanted.

This method can quickly find the leukocyte in microscope. It is because of the low calculation complexity and fast minimization process. The image used in this method has the following features:

1. Elliptical shape.
2. Homogeneous interior and low noises.
3. No cell occlusion and clutter.

2.2.3 Gabor Filter

Gabor filter is defined by harmonic functions modulated by a Gaussian distribution. It has received considerable attentions because it can approximate some functions of certain cells in the visual cortex of some mammals [14]. In addition, these filters have shown to posses optimal localization properties in both spatial and frequency domain and thus are well suited for texture segmentation problems [27, 28]. Investigators have successfully employed Gabor filters in a wide range of image-processing applications, including texture segmentation, document analysis, image coding, retina identification, target detection, fractal dimension measurement, edge detection, line characterization, and image representation [47]. Our endosome detection in cell image can also be considered as texture segmentation problem. This is because the endosomes and cytoplasm can be treated as two different textures, and Gabor filter is the optimal method for texture segmentation. Therefore, utilize Gabor filter to segment our cell images could be another approach.

A Gabor filter can be viewed as a sinusoidal plane of particular frequency and orientation, modulated by a Gaussian envelope. It can be written as:

$$G_{\sigma,\psi,\theta}(x,y) = s_{\psi,\theta}(x,y)g_{\sigma}(x,y) \tag{29}$$

Where $s(x, y)$ is a complex sinusoid, known as a carrier and $g(x, y)$ is a 2-D Gaussian shaped function, known as envelope. X and y are the coordinates or pixel on image, so the pair (x, y) means one point on image. The complex sinusoid and the

Gaussian envelope are defined as follows,

$$s_{\psi,\theta}(x,y) = \exp\left(j2\pi\psi\left(x\cos\theta + y\sin\theta\right)\right) \tag{30}$$

$$g_{\sigma}(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{31}$$

$$j = \sqrt{-1}$$

where ψ is frequency, θ is orientation and σ is bandwidth.

Therefore, $G_{\psi\theta\sigma}(x, y)$ can be transferred to a complex number, which is defined as the following formula.

$$G_{\sigma,\psi,\theta}(x,y) = G_R(x,y) + jG_I(x,y) \tag{32}$$

The real part and imaginary part can be defined as:

$$G_R(x,y) = g_{\sigma}(x,y)\cos\left[2\pi\psi\left(x\cos\theta + y\sin\theta\right)\right] \tag{33}$$

$$G_I(x,y) = g_{\sigma}(x,y)\sin\left[2\pi\psi\left(x\cos\theta + y\sin\theta\right)\right] \tag{34}$$

After define the Gabor filter, we can apply it to the sample image. This process is similar to the convolution. First set the size of Gabor filter, which is $2k+1$. Then convolve the image with this Gabor filter pixel by pixel, which is defined as follows:

$$G(x,y) = \sum_{j=-k}^{k}\sum_{i=-k}^{k} f\left(x+i, y+j\right)G_{\sigma,\psi,\theta}\left(i,j\right) \tag{35}$$

The real part and imaginary part of the above convolution formula can be written as:

$$G_R\left(x,y,\sigma,\psi,\theta\right) = \sum_{j=-k}^{k}\sum_{i=-k}^{k} f\left(x+i, y+j\right)G_R\left(i,j\right) \tag{36}$$

$$G_I\left(x,y,\sigma,\psi,\theta\right) = \sum_{j=-k}^{k}\sum_{i=-k}^{k} f\left(x+i, y+j\right)G_I\left(i,j\right) \tag{37}$$

where $f(x, y)$ means the intensity of pixel (x, y).

After convolution with Gabor filter, each point will have a complex number calculated by Gabor filter. The energy for each point then can be defined as the square

of modulus, which is as follows:

$$E(x,y) = \left[G_R(x,y,\sigma,\psi,\theta)\right]^2 + \left[G_I(x,y,\sigma,\psi,\theta)\right]^2 \quad (38)$$

Thus, to get the optimal solution of Gabor filter is to minimize $E(x, y)$. There are three variables in this energy function, ψ, θ, σ. So the combination of those three variables which leads to the minimum value of $E(x, y)$ is the optimal solution. After get the optimal solution from the sample image, this Gabor filter can be applied to the testing images. The similar textures in testing image will have same energy value as those in sample image. The noises or other textures in testing image will generate relatively higher energy value. Therefore, in the end of process, the textures in testing image which are different from sample image will show abnormal high intensity in the grey level result. So people can easily use some thresholding technique to find out those different textures.

2.3 Initial Study on Canny, Level-set Gabor & Tophat Methods

To better understand the cell segmentation approaches, we implemented the Canny, Level-set Gabor and Tophat methods and apply them to the cell images. We also compare these methods with the straightforward thresholding, which is based on the intensity histogram. Let us look at the image intensity histograms first. The following figure shows the image intensity histogram for three types of treated cells.

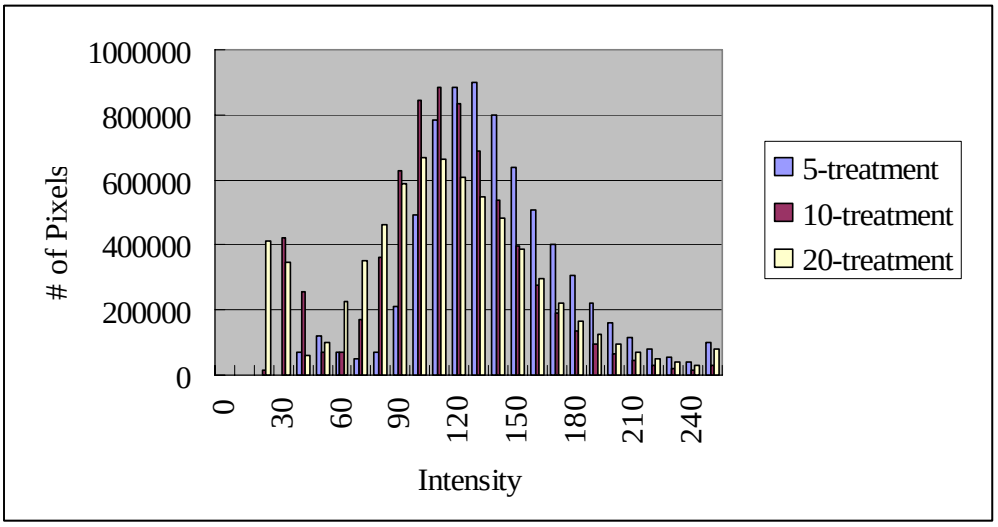
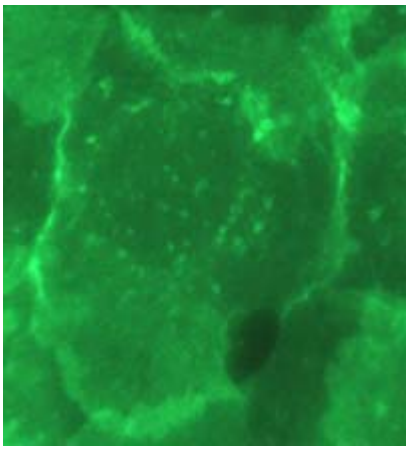


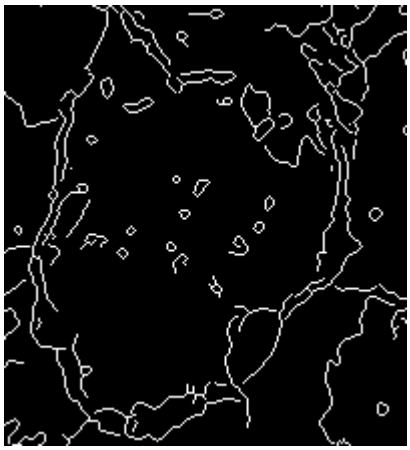
Figure 6: Histogram of number of pixels per intensity

From this histogram, we can see that the distributions of these three types of cells are quite similar to each other. That is why the simple thresholding technique will not work well on the cell images. The interesting thing is the low intensity bars. For 5-treatment cells, there are no pixels under the intensity of 20. However this cannot be used as a feature to classify 5-treatment cells from other treatments. It is because in our 5-treatment images, no background was taken into the microscopy images, but for 10-treatment and 20-treatment images, they both have quite large areas contain the background.

Besides the thresholding method, we also implemented Canny detector, level-set method, Gabor approach and Tophat transform. The following figures show the result of those three initial approaches.

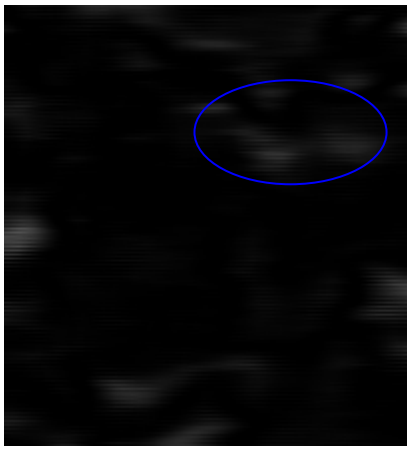


(a) Original Image

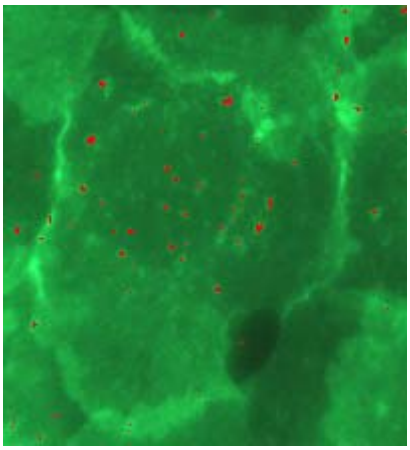


(a) Canny Result

(c) Level-set Result



(d) Gabor Result



(e) Tophat Result

Figure 7: Different approaches to cell segmentation problem

Figure 7 (a) is the cell cropped from the original image. Obviously, this cell is an elliptical cell, but cell top is occluded by another cell. Figure 7 (b) shows the result of

Canny detector. We can see that endosomes are captured nicely, and the cell outline is almost there. The only problem is the cell boundary is not well formed by straight lines. Figure 7 (c) shows the result of level-set algorithm. The red ellipse shows there are a lot of endosomes inside that region and no cell boundary over there. However, when we look at the original image, there is no endosomes there but a very clear cell edge. Figure 7 (d) shows the result of Gabor filter. The blue region indicates there are some obvious endosomes there, but actually they are just overlapping cell membranes. Figure 7 (e) shows the result of Tophat transform. Red regions indicate the endosome detected by Tophat transform.

We findthat the red region in level-set algorithm and the blue region in Gabor filter do not match. This because these two methods look for different features of images. Let us look at the intensity map of the original image first.

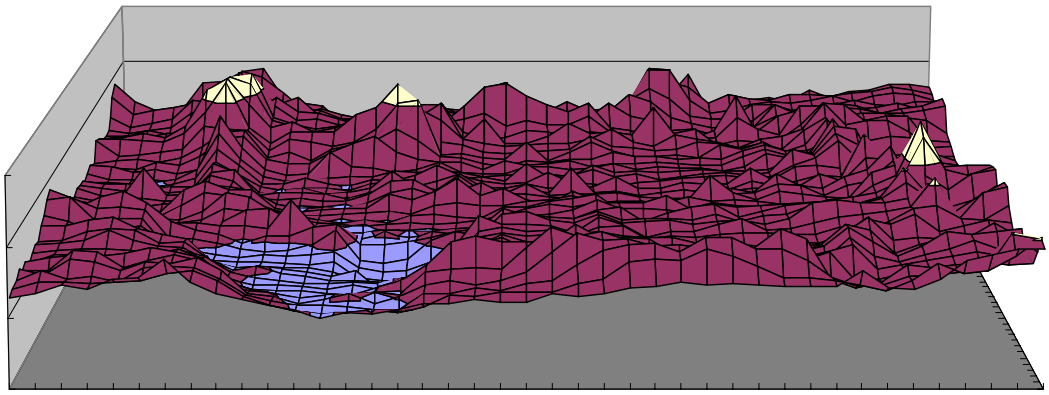


Figure 8: Intensity map of original cell image

Figure 8 shows the intensity map of original image. We found that the cell interior is much smoother than the cell edges. The endosomes are even lower than edge peaks. Therefore, when we apply Gabor filter to this cell. If we choose the cytoplasm as sample texture, the cell edge will give higher energy value than endosomes. This tells us the reason why Gabor filter gives us the cell occlusion part instead of endosomes.

From the intensity map, we draw a horizontal line from left to right. The points along this horizontal line have different intensities, so we can draw a curve where the x-axis and y-axis are the x coordinate and intensity of those points respectively. Suppose we have the following curves:

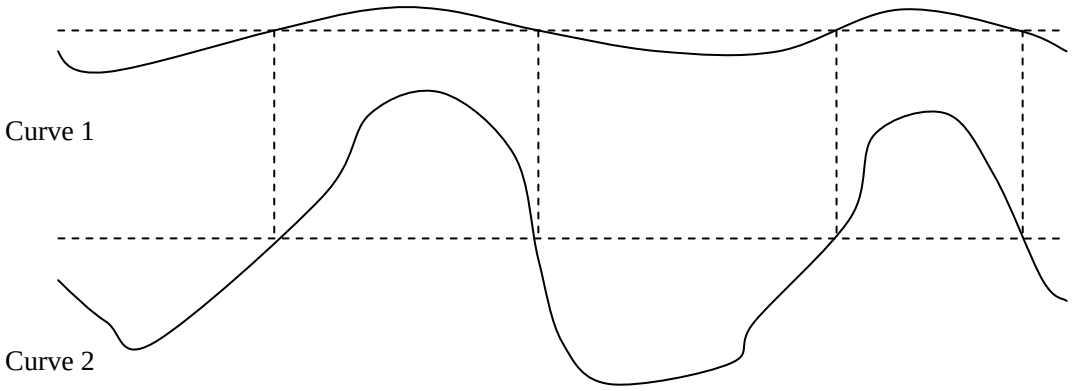


Figure 9: Different curve vs. same level set image

Curve 1 and curve 2 represent different textures. The texture of curve 1 is quite smooth, but the texture of curve 2 is quite rough. However, if the intensity level is set as like what Figure 9 shows, then these two textures will have exactly same level-set images, which is not true. The reason of this error is because level-set algorithm is highly depends on the intensity intervals. If we set the interval too large, then the level-set image cannot present the real texture information. But if we set the interval too small, a lot of fake objects will be generated. Therefore, in the initial result of level-set algorithm, there are a lot of fake endosomes detected. It is because the cytoplasm is just cross two level intensity intervals.

For Tophat transform, there are two drawbacks. The first drawback is although it can find the location of endosomes, but the region detected cannot cover the entire endosome region. Many endosome pixels are missing. The second drawback is it contains a lot of tiny noises but misses some obvious endosomes. This is because many tiny noises are smaller than the structure element we used and some obvious endosomes have larger size than our structure element; therefore the Tophat transform

cannot remove the noises effectively but missed some big endosomes.

Canny detector's result looks like the best one among those three initial results. It can capture most of the endosomes and cell edges. Since Canny detection is the first step of Garrido's method, so we believe based on this edge segment image, Garrido's method could be quite effective in next steps. Then we are going to use this method as the blueprint of our method.

However, there are also two weak points of Garrido's methods. Although the homogeneity of interior is not a critical requirement for this method, Garrido's method is lack of the endosome detection, which is the first weak point of this method. Garrido's image does not have endosomes in cells, so there are not many noises generated by canny detector. Most of the noises on Garrido's image lie on edges or outside the cell, which will not affect the cell location approximation in the next step. But in our cell images, the number of endosomes is competitive to the number of cell edges, and those endosomes are treated as "noises" in Garrido's method. So to fit Garrido's method into our cell image, the first task is to temporally "remove" endosomes inside cells, after we get the approximate cell location, and then move them back.

The second weak point of Garrido's method is the active contour algorithm. Garrido just apply the standard active contour algorithm, which works perfect on their images. This is because cells on their images all have smooth and clear boundaries, so the standard active contour algorithm works very nicely. However, our cells normally do not have such clear and smooth boundaries. Instead, they always cluttered, with broken boundaries, blur edges, etc. This will lead the improper active contour evolution. So our second task is to improve the active contour algorithm to fit our cell characteristics.

To overcome these two weak points, we need to improve Garrido's method. For first weak point, we first tried two different methods, which are Tophat transform and

Canny detector. Then we use training process to improve the classification of endosomes and non-endosome objects. For second weak point, we propose a new energy term which can restrict the growing and shifting of active contour. The details will be presented in the following chapters.

3 Proposed Method

Our cell images are taken by fluorescence microscope, which represent cytology stained with the fluorescent tagging technique. Compared with the cells images used in Garrido's method, beside the characteristics of low contrast, cluttered objects and low quality, our cell images have the following features:

1. *Full of endosome*.
2. *Irregular shape of cells*.
3. *Broken cell edges*. The cell edges are always broken and not smooth.
4. *Intensities are non-uniformly distributed*. Due to the reflection of light, some parts of image are very bright, and some parts are very dark.
5. *Absence of inter-cell background regions*. That is, cells are tightly cramped.

The objective of our application is to calculate the intensity ratio of endosomes (summation & average) and cytoplasm in a single cell and count the number of endosomes for each cell. We formalize our metrics in the following table:

Index	Name	Formula / Symbol	Description
1	Intensity Sum Ratio	$R_i = \frac{\sum I(x,y)\chi_e(x,y)}{\sum I(x,y)\chi_c(x,y)}$	Sum of the endosome intensity over the Sum of cytoplasm intensity.
2	No. of Endosome	N_E	Count of endosome regions.
3	Average Intensity Ratio	$R_a = \frac{\sum I(x,y)\chi_e(x,y) / N_e}{\sum I(x,y)\chi_c(x,y) / N_c}$	Average intensity of endosome over the average intensity of cytoplasm
4	Average Endosome Intensity	$\sum I(x,y)\chi_e(x,y) / N_e$	Average intensity of endosomes
5	Average Cytoplasm Intensity	$\sum I(x,y)\chi_c(x,y) / N_c$	Average intensity of cytoplasm

Table 1: Cell Metrics

The first metric gives the intensity sum ratio of endosomes and cytoplasm per cell. (x, y) is the coordinate of a pixel. $p(x, y)$ defines a pixel with (x, y) as its coordinate. $I(x, y)$ defines the intensity of $p(x, y)$. N_e and N_c are the number of pixels of endosomes and cytoplasm per cell respectively.

$\chi_e = \begin{cases} 1, p(x,y) \in E \\ 0, p(x,y) \notin E \end{cases}$ and $\chi_c = \begin{cases} 1, p(x,y) \in C \\ 0, p(x,y) \notin C \end{cases}$, E is the set of endosome pixels and C is the set of cytoplasm pixels.

In the previous chapter, we show that Garrido's method is the most systemic method so far to analyze cell images. Therefore, we are going to design our method based on Garrido's method. However, since the Garrido's method is designed for cell segmentation and not endosome detection. Thus we need to apply some enhancements on Garrido's method:

1. Garrido only uses canny detector to get the cell boundaries. Our objective is to get the endosomes, so we can apply other pattern detector on the images to extract endosomes, for example, Tophat transform.
2. Garrido's method uses fixed cell template to match the cell edges detected by Canny detector to get initial cell locations. Since we are not going to utilize cell edges to detect initial cell locations due to the numerous endosomes, we cannot use Garrido's approach. A new cell location approximation method is needed.
3. Garrido's method works on cells, whose interiors are almost homogenous. When they apply the active contour algorithm, there is no need to consider the noises inside cell. In our work we need to remove endosomes first before applying active contour algorithm.
4. The Hough transform used in Garrido's method is too expensive, because each pixel on the image will be examined whether there is a potential cell outline around it. Therefore we need to find some simple but effective enough method to find out the approximate cell locations.

Therefore, we propose our method as following:

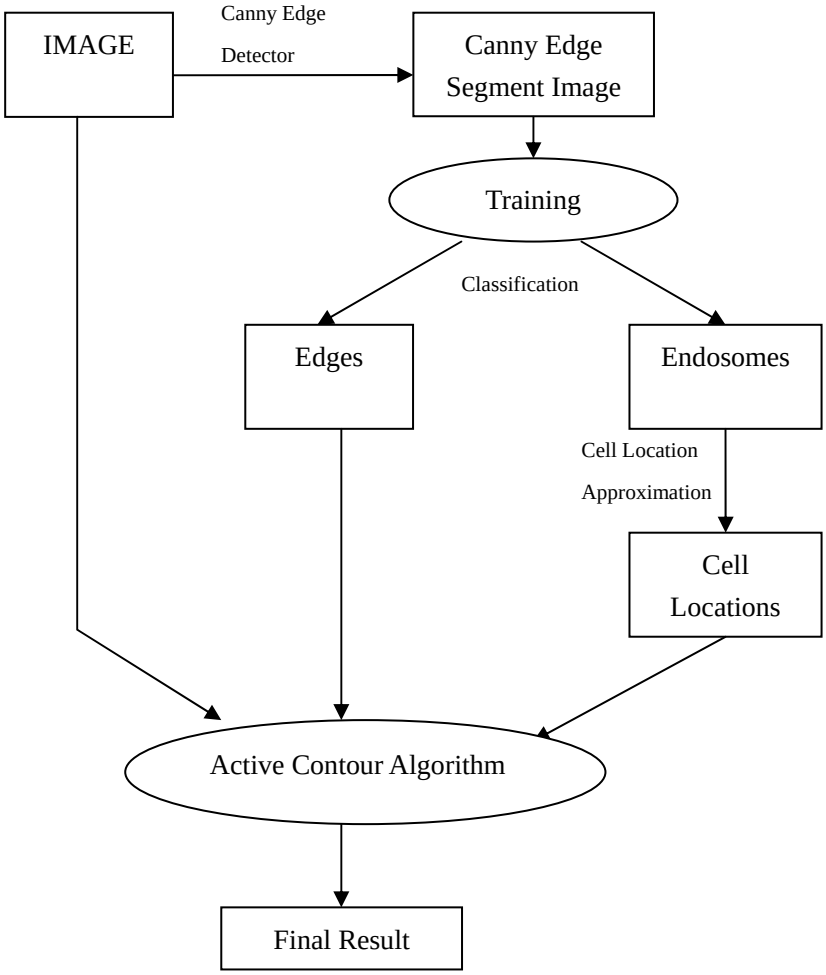


Figure 10: Flow chart of our method

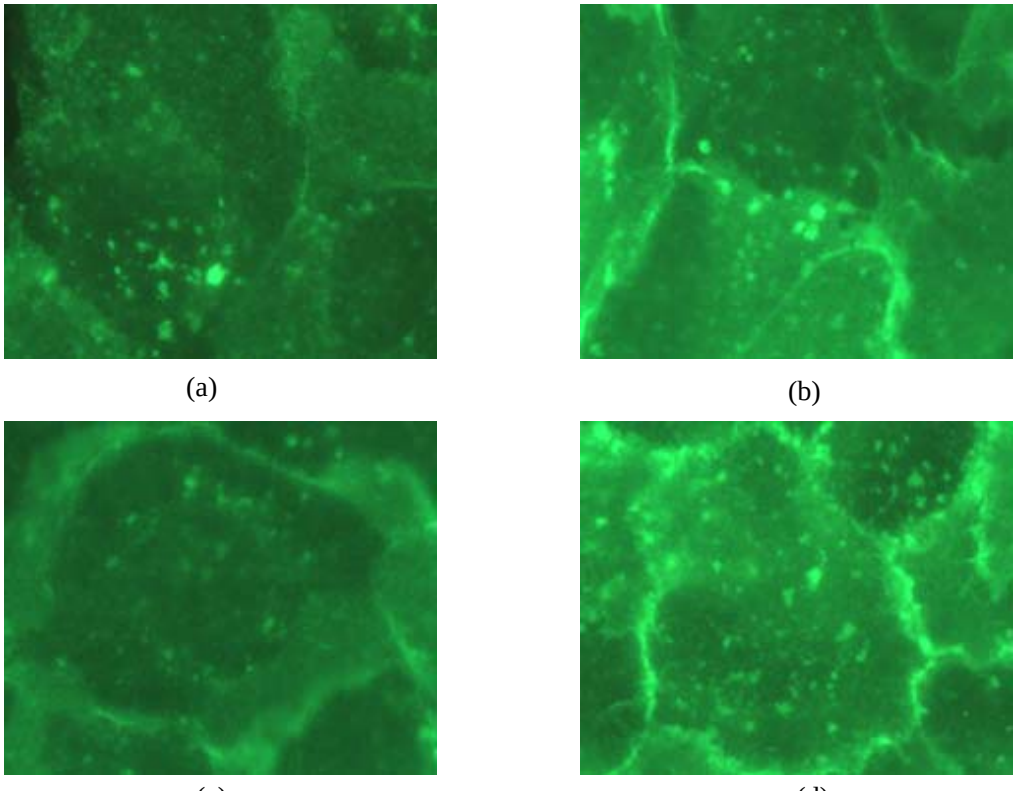
First, we apply Canny edge detector on original image to extract the outlines of cell edges and endosomes. Then we use iterative training process to classify cell edges and endosome segments from the line segments obtained in Canny edge detector. The third step is to utilize the endosomes we obtained after training to generate initial location of cells. Since the Hough transform used in Garrido's method is too expensive, we propose our improved method to obtain the initial location efficiently. The last step is to apply active contour algorithm on the initial seeds to get the closed cell boundaries. When we have the endosomes and cell boundaries, we can easily compute the metrics.

In the first subsection, we will discuss how to get endosomes by applying Canny edge detector on the original images and how to classify those detected edges into endosome segments and non-endosome segments. In the second subsection, we will try to utilize the result of previous step to get the approximate cell locations. In the third subsection, we will start from the approximate cell locations to search for the complete cell boundaries by applying active contour based algorithm.

3.1 Endosome Detection

Endosomes are the bright spots regions distributed in the cytoplasm. The endosomes are tagged proteins, and normally will reflect more lights from the microscope, thus the intensity is higher than the cytoplasm. There are also some bright spots located at the edge of cells. Those bright regions are not endosomes, they are just noises.

The intuitive method of endosome detection is image thresholding, which is also a very common method in most image segmentation problems [24, 43]. However, the simple thresholding cannot give effective result to our cell images. This is because when microscope takes images of cells, normally there are some reflection regions in the scope. Therefore some regions appear very bright and some are very dark. The endosomes are usually not uniformly distributed and the intensity of endosomes is also not fixed within certain range. From the observation of the cell images, endosomes can be located anywhere in a cell. The following figures show the different locations of endosomes in cells:



(a) (b)

(c) (d)

Figure 11: Four different endosome distribution

Figure 11 (a) shows that the endosomes are cramped at a small region of a cell, and are quite closed to the cell membrane. Figure 11 (b) shows the cells are overlapping, thus the endosomes appears just right on the cell edges. Figure 11 (c) shows the endosomes form a circle and Figure 11 (d) shows the endosomes are uniformly distributed in the cell.

3.1.1 Endosome segments detection

The endosomes have these characteristics: shape is circular or elliptical; intensity is higher than surrounding cytoplasm pixels and gradient around endosome is higher than background. Therefore, we can utilize these two characteristics to separate endosomes from cytoplasm and cell membranes. As discussed in the previous chapter, we adopt Canny detector for the pre-processing step. The Canny operator [6] takes as input a grey scale image, and produces as output an image showing the positions of tracked intensity discontinuities. First of all, the image is smoothed by Gaussian convolution, and then followed by 2-D first derivative operator, like Roberts Cross.

Gaussian convolution, also called Gaussian smoothing operator is a 2-D convolution operator that is used to “blur” images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (bell-shaped) hump. The following equations show the 1-D and 2-D forms of Gaussian distribution:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \tag{1}$$

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

The following figure shows a 2-D Gaussian distribution example:

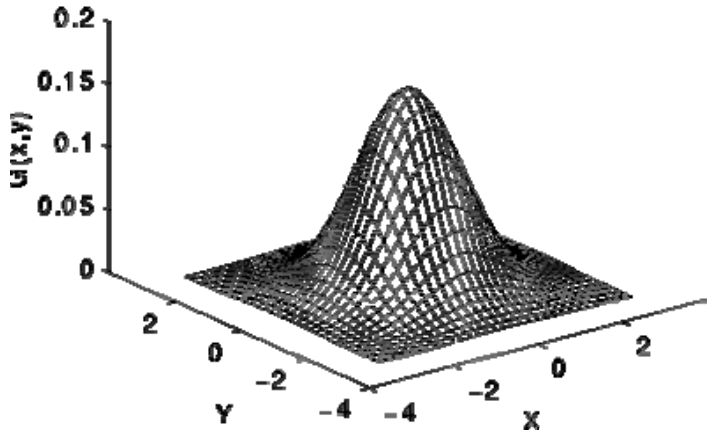


Figure 12: 2-D Gaussian distribution with mean (0, 0) and $\sigma = 1$

Once a suitable kernel has been calculated, then the Gaussian smoothing can be performed using standard convolution methods, which is given as the following equation:

$$O(i,j) = \sum_{k=1}^{m}\sum_{l=1}^{n} I(i+k-1, j+l-1)K(k,l) \qquad (3)$$

Where *M* and *N* are the width and height of input image, and the kernel *K* has *m* rows and *n* columns, then the size of the output image will have *M-m+1* rows, and *N-n+1* columns. Therefore, in equation (18), *i* runs from 1 to *M-m+1* and *j* runs from 1 to *N-n+1*. The 2-D Gaussian convolution can in fact be performed by first convolving with a 1-D Gaussian in the *x* direction, and then convolving with another 1-D Gaussian in the *y* direction. In fact, the Gaussian is the only completely circularly symmetric operator which can be decomposed in such a way.

Roberts Cross operator performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It consists of a 2x2 convolution kernels as shown in following figure:

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Figure 13: Roberts Cross convolution kernels

The kernels can be applied separately to the input image with standard convolution method, to produce separate measurements of the gradient component in each orientation, and then combine together to find the absolute magnitude and orientation of the gradient at each point. The formulas are shown in the following equations:

$$|G| = \sqrt{Gx^2 + Gy^2} \tag{4}$$

$$\theta = \arctan(Gy / Gx) - 3\pi / 4 \tag{5}$$

After these two convolutions, the regions of the image with high first spatial derivatives will be highlighted. Edges give rise to ridges in the gradient magnitude image. Then the algorithm will track along the top of these ridges and set to zero all pixels that are not actually on the ridge top so as to give a thin line in the output. The tracking process is controlled by two thresholds: *T1* and *T2*, with *T1* < *T2*. If the magnitude is below *T1*, it is set to zero. If the magnitude is above *T2*, it is set as an edge. And if the magnitude is between the two thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above *T2*. This tracking process helps to ensure that noisy edges are not broken up into edge fragments. Figure 14 shows the segment detected by Canny detector.

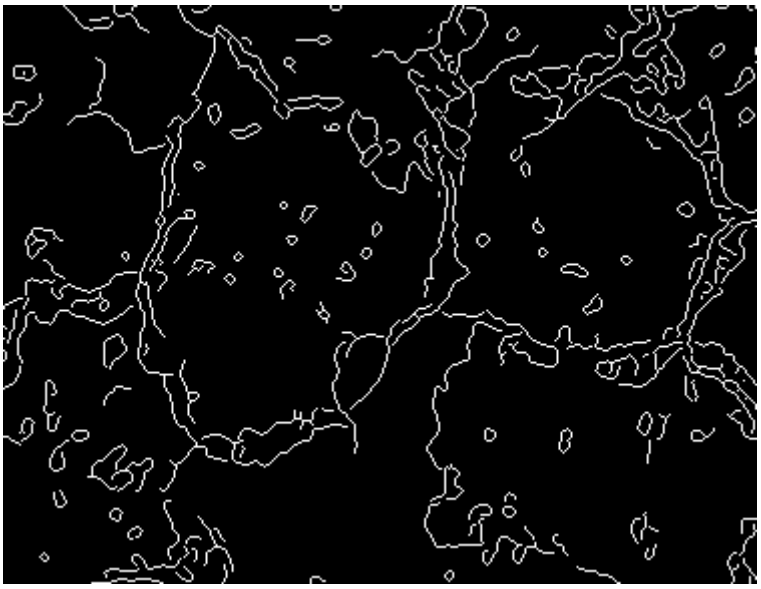


Figure 14: Result of Canny detector

With those detected edges, a post-processing is applied. This step is to remove the small segment and pruning the small branches from some big edges. If the length of segment is shorter than threshold, it will be discarded, and otherwise, it will be remained. Therefore, after first step, there are only those straight edge segments without branches left on the image. The next step is how to utilize those edge segments to estimate the approximate cell locations.

3.1.2 Analyze segment features

From Figure 14, we observe that most of endosomes are captured and the cell boundaries are also detected on the image. Thus the next step becomes how to separate endosome segments from boundary segments. In order to classify those segments into two classes: endosome segments and non-endosome segments, we propose six features that can be used in the data mining technique. The following figure shows how we define those features:

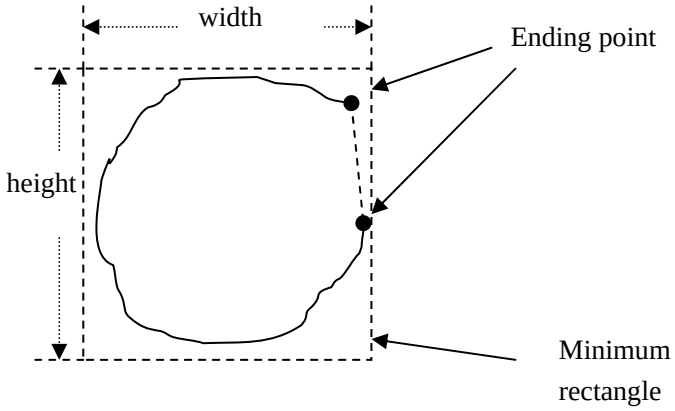


Figure 15: Endosome segment of canny edge detector result

1. *Number of pixels in segment (size)*. The boundary segment always has more pixels than endosome segment. Thus we can set a threshold on the number of pixels per segment.
2. *Distance between ending point (gap)*. Some endosomes are open, like most boundary segments. However, the gap between ending points of endosomes are much less than of boundary segments.

3. *Average minimum distance from each pixel to rectangle boundary (difference)*. The boundary segments are line-like segments, and on the other hand, endosomes are circle-like segments. So we can find a minimum rectangle just fit the segment. Then calculate the average minimum distance from each pixel on segment to the four sides of rectangle. Obviously, endosomes will have smaller average distance than boundary segments.
4. *Rectangle width/height ratio (ratio)*. Sometimes, the boundary segments are just orthogonal to the horizontal line or vertical line. For this case, we can calculate the width and height ratio of rectangle defined in previous feature. Low such ratio indicates an evidence of boundary segment.
5. *Average Co-distance (co-distance)*. Sometimes, the endosomes are tightly grouped together. This makes the endosome segments detected by canny detector ramose. There are always several branches in edge segments. But compare with those ramose endosome segments, edge segments always have a main branch which lasts quite long. On the other hand, the branches of endosome segments are quite close to each other, but edge segments do not have such characteristics. Figure 16 shows the different appearances of these two segments.

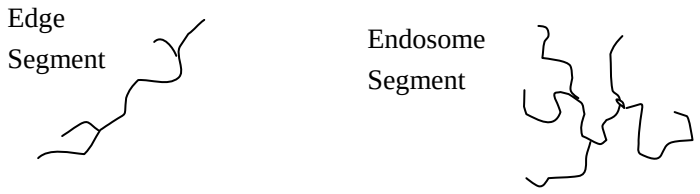


Figure 16: Illustration of edge segment and endosome segment

6. *Neighbor Count*. Some endosomes are like cotton wool, and they are the most difficult endosomes to capture by program due to their irregular shapes. However, they are quite close to those circle or ellipse shape endosomes. So we can count the number of circle or ellipse shape endosomes around cotton wool like endosomes. If there are many circle or ellipse shape endosomes near a cotton wool like endosome, we can say it is highly likely an endosome.

We notice that not every endosome has all six features. For example, some endosome do not have two ending points, but has a tiny tail. On the other hand, we found that 90% of endosome segments in our cell images only have at most two ending points. Therefore, we can differentiate the endosomes by counting their ending points first. For endosomes with one or less than one ending points, we use ratio, difference and size as features. For endosomes with exactly two ending points, we use ratio, difference, gap and size as features. For endosomes with three or more than three ending points, we use ratio, neighbor count, co-distance and size as features. The following table shows the distribution of three endosome classes.

	< 2 ending points	**= 2 ending points**	**> 2 ending points**
# of endosomes segments per image	283 (61.9%)	123 (27.0%)	51 (11.1%)

Table 2: Distribution of different endosomes per image

3.1.3 Training process

After extracting the edge segments, we can apply the existing classification techniques to classify those edge segments into endosome segments and non-endosome segments. There are two steps in data classification, which are model building and data testing. In the first step, a model is built up which describes the characteristics of each pre-defined classes. Therefore, each tuple is assumed to belong to a predefined class, and an attribute called class label will indicate which class it belongs to. There are two types of classification, supervised and unsupervised. Supervised learning means the class label of each training sample is provided and it contrasts with unsupervised learning, also known as clustering, in which the class label of each training sample is not known. Obviously, we should adopt supervised learning, and we have two classes: endosome and non-endosome segment. The second step is quite straightforward, which is just applying this model to classify data. The predictive accuracy is always estimated to test the effectiveness of the classifier.

Accuracy of a classifier on a given test set is the percentage of test set samples that are correctly classified by the classifier.

Since our feature sets are not very complicated, and the training data size is not very big, we choose the most common technique, decision tree. The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner [51, 52]. It is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions. In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. A path is traced from the root to a leaf node that has the class prediction for that sample.

The following figure shows the initial result of endosome and non-endosome classification by using decision tree. Red circles indicate the endosomes detected by program. The full experiments and comparisons of training process will be discussed in next chapter.

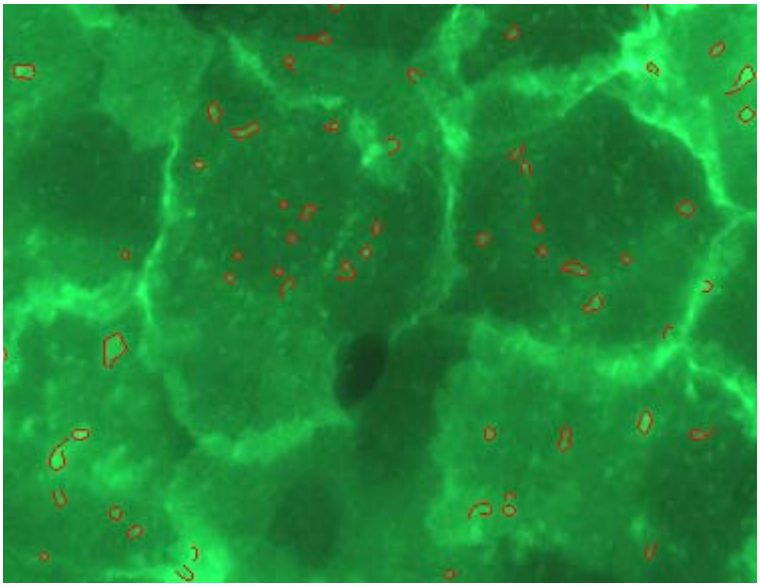


Figure 17: Result of endosome detection

3.2 Approximate Cell Location

The cell boundaries in the image are quite blur and tortuous, which can be seen from the result of canny edge detector. Thus we cannot trace the edge to get the cell locations directly. However, some characteristics of endosomes give us some hints to locate cell automatically. The first characteristics is that most endosomes are inside cells, and only a small portion of endosomes located at the cell boundaries due to cell overlapping. The second feature of endosomes is that usually endosomes from same cell are quite close to each other. Therefore, we can utilize the endosomes detected in the previous step to get the initial location of cells.

First of all, we calculate the gravity point for each endosome. The gravity point is defined as the mean spatial value of *x* and *y* axis respectively. By given an object *C*, the gravity point is given by the following formula:

$$G(C) = \frac{\sum_{\wp(C)} p(x, y)}{N(C)} \tag{41}$$

Where *p*(*x, y*) is the spatial vector of pixel *p* inside *C*, *N(C)* is the total number of pixels in *C*.

Then we draw a circle around this gravity point, and assign different values to each point inside this circle. The value varies by the distance from the gravity point, which is defined as the following formula:

$$V(p) = \frac{\lambda}{1+R}(R-d) \tag{42}$$

Where *p* is the given point, *R* is the radius of circle, *d* is the distance from gravity point, and λ is the coefficient. Therefore, to a particular circle, the farther of the point is, the lower value will be assigned. For the overlapping circles, we just sum up the values. Since endosomes from same cell are more likely to be grouped together, so the value around the cell center will be relatively higher than the edges. The following

figure shows the simple illustration of this process:

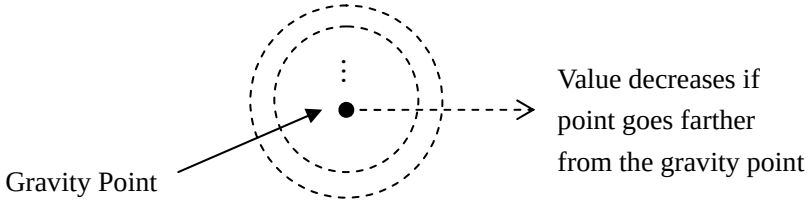


Figure 18: Illustration of assigning different values around gravity point

After accumulating all the values, we can apply a simple thresholding to the valued image. Those regions with higher values will be separated from the lower value regions. The last step is just to calculate the gravity point for those higher value regions again to get the approximate cell locations. The following figure shows the result.

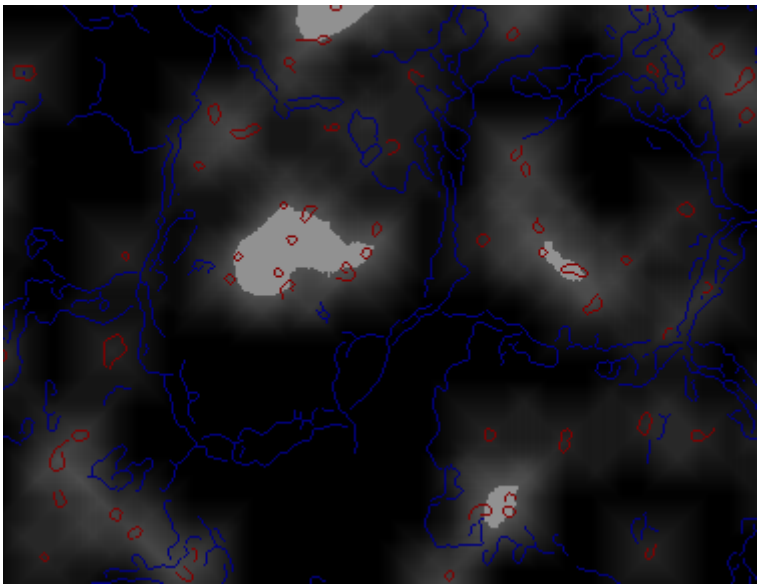


Figure 19: Approximate cell locations

The gray regions are the possible regions of cell center. The red circles are the detected endosomes and the blue lines are non-endosome segments. Some grey regions will be very close to each other. This is because the endosomes are not uniformly distributed among cytoplasm. So it is possible there are two groups of endosomes in one cell. However, we will put the active contour seed on every grey region. After the evolution of active contour, those cells with two or more than one

active contour seeds will be covered by overlapping contours. Therefore, we provide the user interface to allow user to remove the duplicate contours.

3.3 Cell Boundary

Cell boundary detection is the most complicated part in this cell segmentation problem. The edges between cells are very blur and cluttered. Almost every edge is shared by two or even three cells due to cell overlapping. Some cell's membrane even looks like broken. Thus when we apply the traditional edge detector, for instance, Canny detector, most of the cell boundaries are broken. There are few complete closed cell contours in Canny segmentation result. However, since we already have the approximate cell centers, and know the most endosomes, with some evidence of cell boundaries, we can apply active contour algorithm.

3.3.1 Standard active contour algorithm

Deformable template or active contour, known as "snake" was proposed by Kass et al [30], in 1987. It deforms a contour to lock onto features of interest within an image. Usually the features are lines, edges, and/or object boundaries. Kass et al [30] named their algorithm as "snakes" because the deformable contours resemble snakes as they move.

Given an approximation of the boundary of an object in an image, an active contour model can be used to find the actual boundary. An active contour is an ordered collection of *n* points in the image plane:

$$V = \{v_1, ...v_n\} \tag{10a}$$

$$v_i = (x_{i,} y_i), i = \{1, ..., n\} \tag{10b}$$

The points in the contour iteratively approach the boundary of an object through the solution of an energy minimization problem. For each point in the neighborhood of v_i, an energy term is computed:

$$E_i = \alpha E_{\text{int}}(v_i) + \beta E_{ext}(v_i) \tag{11}$$

Where $E_{int}(v_i)$ is an energy function dependent on the shape of the contour and $E_{ext}(v_i)$ is an energy function dependent on the image properties, such as the gradient, near point v_i. α and β are constants providing relative weighting of the energy terms. E_i, E_{int} and E_{ext} are matrices. The value at the center of each matrix corresponds to the contour energy at point v_i. Other values in the matrices correspond to the energy at each point in the neighborhood of v_i. Each point, v_i, is moved to the point, v_i', corresponding to the location of the minimum value in E_i. If the energy functions are chosen correctly, the contour, V, should approach, and stop at, the object boundary.

There are two terms used in equation (11), internal energy and external energy. Internal energy function is intended to enforce a shape on the deformable contour and to maintain a constant distance between the points in the contour. We adopt two internal energy functions for our model, continuity energy and balloon force energy.

$$\alpha E_{\text{int}}(v_i) = cE_{con}(v_i) + bE_{bal}(v_i) \tag{12}$$

The continuity energy term coerces a closed deformable contour into a circle. The formulation of the continuity energy

$$e_{jk}(v_i) = \frac{1}{\Gamma(V)} \left\| p_{jk}(v_i) - \gamma(v_{i-1} + v_{i+1}) \right\|^2 \tag{13}$$

$p_{ik}(v_i)$ is the point in the image that corresponds spatially to energy matrix element $e_{jk}(v_i)$. γ in (13) is the distance scale coefficient of v_i and v_{i+1}. For our case, the cell boundary is a closed contour, therefore we adopt the γ definition as follow:

$$\gamma = \frac{1}{2\cos\left(2\pi/n\right)} \tag{14}$$

Here the $\Gamma(V)$ is the average distance between points in V, which is defined by :

$$\Gamma(V) = \frac{1}{n}\sum_{i=1}^{n}\left\|v_{i+1} - v_i\right\|^2 \tag{15}$$

This term is a normalization factor, which is the average distance between points. The normalization can make $E_{con}(v_i)$ independent of the size, location, and orientation of V.

A balloon force can be used on a closed deformable contour to force the contour to expand or shrink in the absence of external influences. A contour initialized within a uniform image object will expand under the influence of a balloon force until it nears the object boundary. Chalana et al. suggest an adaptive balloon force [8] that varies inversely proportionally to the image gradient magnitude. The adaptive balloon force is strong in homogeneous regions and weak near object boundaries, edges and lines. The energy term for each element, $e_{jk}(v_i)$, in the matrix, $E_{bal}(v_i)$ is expressed as a dot product:

$$e_{jk}(v_i) = n_i \bullet (v_i - p_{jk}(v_i)) \tag{16}$$

n_i is the outward unit normal of V at point v_i, which can be found by rotating the tangent vector, t_i, by 90^o:

$$t_i = \frac{v_i - v_{i-1}}{\left\|v_i - v_{i-1}\right\|} + \frac{v_{i+1} - v_i}{\left\|v_{i+1} - v_i\right\|} \tag{17}$$

The External energy function attracts the deformable contour to interesting features, such as object boundaries. Image gradient and intensity are obvious characteristics to look at. Shape and size are also commonly used as external energy functions. However, our cells have irregular shapes and different sizes. Therefore, we are not going to adopt these two constraints. The following external energy function is investigated:

$$\beta E_{ext}(v_i) = mE_{mag}(v_i) + gE_{grad}(v_i) \tag{18}$$

where $E_{mag}(v_i)$ is an expression that attracts the contour to high or low intensity regions and $E_{grad}(v_i)$ is an energy term that moves the contour towards edges. m and g are provided to adjust the relative weights of the terms.

Intensity energy and gradient energy are quite straightforward, which is given as the following equations. Intensity energy function returns the intensity of the given point, and gradient energy function returns the gradient magnitude of the given point.

$$e_{jk}(v_i) = I(p_{jk}(v_i)) \tag{19}$$

$$e_{jk}(v_i) = -\left|\nabla I(p_{jk}(v_i))\right| \tag{20}$$

The energy functions should be scaled so that the neighborhood matrices contain comparable values. This process is referred to as regularization.

Continuity energy is simply scaled by taking in the minimum value and maximum value in the matrix.

$$e'_{jk}(v_i) = \frac{e_{jk}(v_i) - e_{\min}(v_i)}{e_{\max}(v_i) - e_{\min}(v_i)} \tag{21}$$

where $e_{min}(v_i)$ and $e_{max}(v_i)$ are the minimum and maximum valued elements in the matrix of E_i.

Balloon force energy function is scaled by adapted to the image gradient magnitude. The formula is given as follows:

$$e'_{jk}(v_i) = \frac{e_{jk}(v_i) - e_{\min}(v_i)}{e_{\max}(v_i) - e_{\min}(v_i)} * \left(1 - \frac{\left|\nabla I(v_i)\right|}{\left|\nabla I\right|_{\max}}\right) \tag{22}$$

where $\left|\nabla I\right|_{\max}$ is the maximum gradient magnitude in the entire image. The principle of the balloon force regularization behind this equation is to reduce the power when the contour approaches the object edge. When the contour is near the

edge, image gradient magnitude will increase dramatically compared to the center region of cell. Therefore, the second term of (36) will be reduced dramatically, and balloon force will have less power at the edge pixels.

The regularization of intensity energy and gradient energy are quite similar. Both of them are scaled by the local maximum value or the global maximum value, depending on the parameter δI and δG set. The formulas are shown as following equations:

$$e'_{jk}(v_i) = \frac{e_{jk}(v_i) - e_{\min}(v_i)}{\max(e_{\max}(v_i) - e_{\min}(v_i), \delta I \cdot I_{\max})} \tag{23}$$

$$e'_{jk}(v_i) = \frac{e_{jk}(v_i) - e_{\min}(v_i)}{\max(e_{\max}(v_i) - e_{\min}(v_i), \delta G \cdot |\nabla I|_{\max})} \tag{24}$$

In general, active contour algorithm can be considered as an EM process, where E means “estimation” and M means “minimization”. Each point on the active contour will estimate the energy value in its neighbor points first. Then it chooses the point with minimum energy value as its new position. The standard active contour has the internal forces which push the contour outward and the external forces which pull back the contour. The internal forces include continuity force and balloon force. The external forces include the gradient and intensity barrier. The problem in our cell images is some cell has broken boundary. The standard active contour works on the closed object, but will fail on our cells with broken boundary. Thus we need find some other external forces to stop the active contour growing at the broken boundary.

3.3.2 Gap leaking

Gap leaking means the cell has a gap or many gaps on its boundary, sometimes quite small, but sometimes can be very big one. The gradient and intensity at the gap changes slightly. Thus the external energy, which is supposed to have greater power at the edge to stop the contour growing, will have limited power compared to internal

energy. In such case, the internal energy, especially the balloon force will have relatively strong power to push the contour outwards through the gap. We call this scenario gap leaking.

To prevent such problem occurring, we introduce a new constraint to the internal energy function, which is the velocity vector. We define the velocity vector as follows: from previous position of active contour point *i'* to current position *i*, draw a straight line. The *i'* is the start point and *i* is the end point, and length of this straight line is the magnitude. When we need to render the next position of a contour point, we displace the rest contour points' velocity vector to this point with magnitude divided by the distances. Then just do a simple vector summation to get reference velocity vector. After that, we assign increasing values from this vector. This process is shown as the following figure:

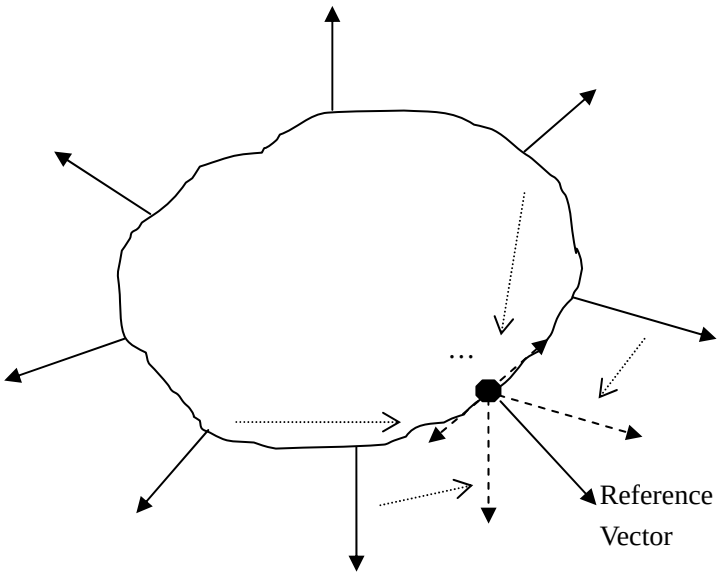


Figure 20: Velocity vetor

The following formula gives the formal definition of velocity vector:

$$v_i = \sum_{j=0, j\neq i}^{n} v_j \frac{1}{d_j} \tag{43}$$

d_j is the distance between *j-th* point to *i-th* point. We just assign different values to the pixels in the matrix according to this reference vector, which is shown in the following figure.

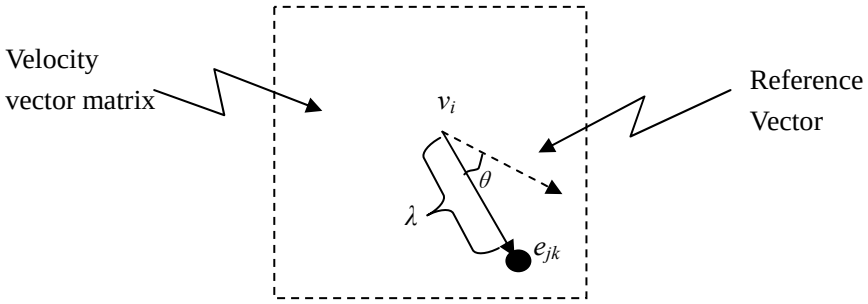


Figure 21: Velocity vector matrix

The formula is given as the follows:

$$e_{jk} = \alpha(\theta_r - \theta_{jk}) + \beta(\lambda_r - \lambda_{jk}) \tag{44}$$

Where the θ_{jk} is the spatial orientation of pixel e_{jk}, θ_r is the spatial orientation of reference vector. λ_r is the magnitude of reference vector, and λ_{jk} is the magnitude of pixel e_{jk}. α and β are balancing coefficient between the two terms.

With this new constraint, we can have more power on the edges, even sometimes there is no physical edge existing. The reason we do not adopt shape constraint is cells in those images have ambiguous shapes, thus shape constraint is not applicable.

3.3.3 Resample points

One important aspect of parametric snake evolution is the discretization of the continuous contour. These discrete points on the snake are called snaxels. During the course of evolution the snaxels either grow apart or come close to each other. To avoid nonuniform sample spacing, one needs to resample the contour intermittently during the evolution. This resampling is usually done explicitly by choosing sample points uniformly during the snake evolution [49]. The cost of such explicit sampling is *O(n)*, with n being the number of snaxels.

Since the resampling process represents a significant computational expense.

Therefore, some researchers introduce shape, size or position constraints to active contour algorithm to eliminate the need for the explicit resampling and reparameterization of the snaxels. Those constraints will give higher penalty to those points which violate the preset template shape, normally circle or ellipse. Therefore, the points will not go further from each other. The active contour algorithm used in Garrido's method also has this constraint, which is based on cyclic Markovian Gaussian density formula. However, for our cell images, we cannot apply shape, size and position constraints. This is because our cells do not have fix shape, size or positions. Therefore, we can only adopt the resampling method to solve the discretization problem.

There are two resampling point methods. First method is quite trivial. We can always check the distances between each neighbor point pair. If the distance between neighboring points is less than the tolerance, we just remove one point. If the distance goes greater than the upper bound, we just insert one more point between these two points. This method is easy to implement but a bit time and memory consuming.

The second method is called T-snake. McInerney's utilizes T-snakes [35] to effectively segment complex anatomic structures from medical volume images as well as 2-D medical images. There are three components to the T-snakes formulation [35].

The first component is a discrete form of a standard parametric snake. That is, a T-Snake behaves like a standard parametric snake between reparameterization and it is free to deform in any direction.

The second component of T-snakes is the affine cell image decomposition (ACID). This ACID will decompose images into triangular grids, which is shown as the following figure:

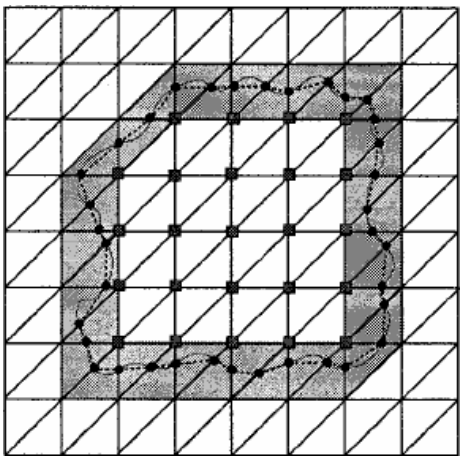
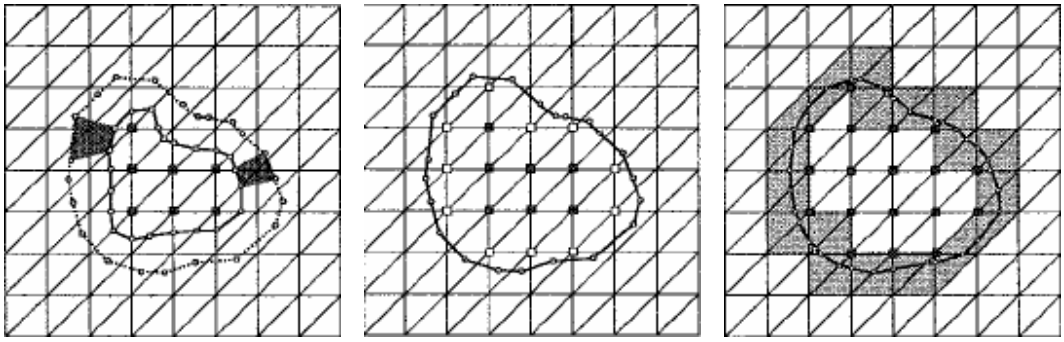


Figure 22: T-Snake

This figure shows a simplicial approximation (dashed line) of an object contour (solid line), using a Freudenthal triangulation. The model nodes (intersection points) are marked and the boundary triangles are shaded.

The third component of T-snakes is the reparameterization process. This process is illustrated in the following figures:



(a) (b) (c)

Figure 23: T-Snake reparameterization process

Figure 23 (a) shows the initial state of reparameterization process, the shaded regions show examples of grid vertices that are turned on by the expanding contour. Figure 23 (b) shows the new inside grid vertices (white) added to the current inside vertices (dark). Figure 23 (c) shows the new contour after one deformation step showing new grid intersections, inside grid vertices, and boundary grid cells (gray shaded). Note that the contour points in (b) changed to those intersection points of

contour and grid edges in (c).

T-Snake is very useful when applying active contour algorithm to medical imaging problems, like retinal blood vessel detection and MR brain image segmentation. Due to the irregular shape of our cell images, this technique will also be useful during active contour evolution.

The principles behind the two methods are same. After every evolution, check the point distribution along the contour. If points are getting further and further, then add new points; if points are getting closer and closer, and then merge them. In our method, we just adopt the first method, because it is easy to implement, and will not add complexity to snake's implementation and unpredictability to its performance.

3.4 Summary

In this chapter, we propose a new method based on Garrido's method. There are three steps in this method. First we adopt Canny edge detector to retrieve object information from cell images. The segments generated by Canny edge detectors can be classified into two classes, which are endosomes and non-endosomes, according to their features. Therefore, we use the most common techniques in data mining, which is the iterative training based decision tree to set proper threshold dynamically.

In the second step, Garrido uses reformulated general Hough transform to get the approximate cell locations. We will not use Hough transform in this step because of two reasons:

1. Time consuming. Hough transform will be applied to each pixel on the image which is very expensive.
2. From the first step, we capture the endosomes already. An important feature for our cell image is endosomes are inside cells and grouped around the nucleus.

Therefore, we can utilize the endosomes detected in the first step to get the approximate location of cells. This process will accumulate the pixel energy segment by segment, which will not involve a lot of calculation.

The third step is to get the cell outlines. In this step, active contour algorithm will be applied to the intermediate result from previous step. Each approximate cell location detected in second step will be put on an active contour seed. To overcome the gap leaking and discretization problem during the active contour evolution, we add two constraints to the standard "snake". The first constraint is called velocity constraint, which will adjust the growing speed of each contour point to make it not go too far away from other contour points. The second constraint is the resampling constraint, which will adjust the distances between neighbor contour points to avoid points stretch or compression.

The following figure shows the final result of our method. Blue contours are the outlines of cells and red circles are the endosomes detected. Although we cannot capture all the cells, we can get enough cells to calculate the metrics presented in the very beginning of this chapter.

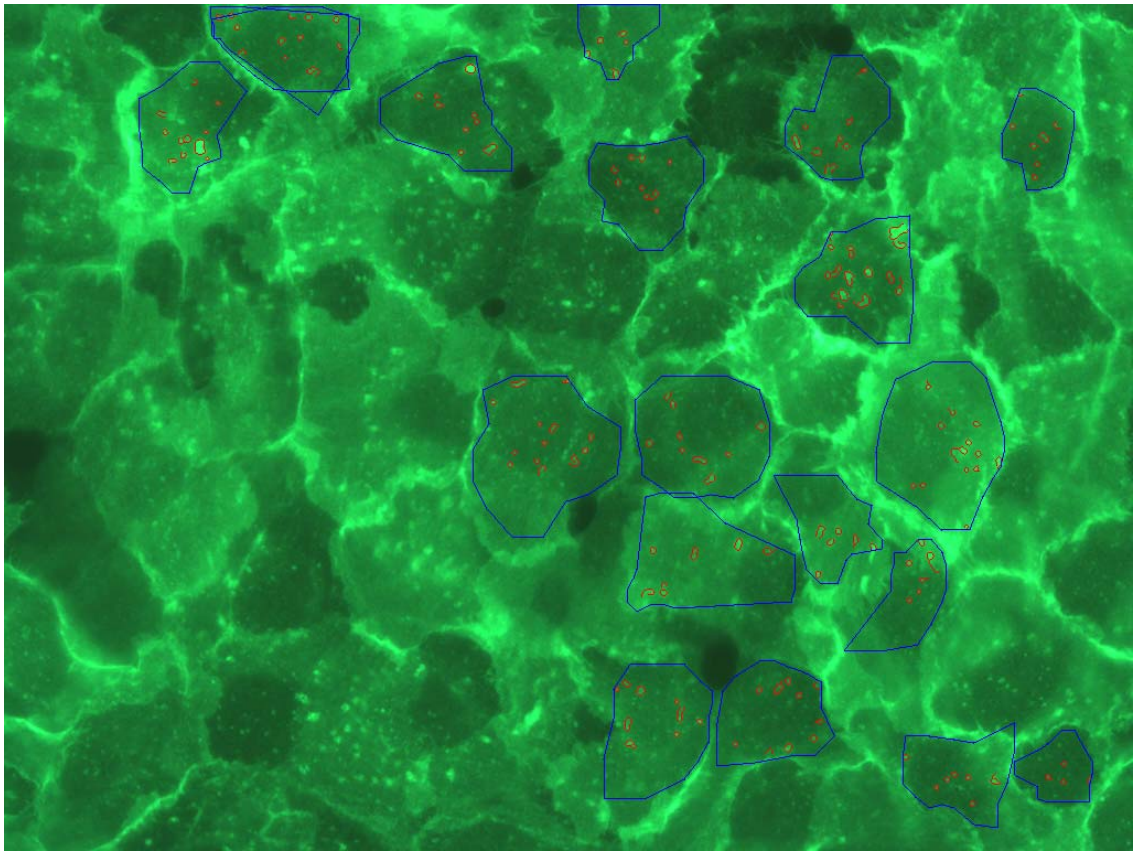


Figure 24: Final result of our method

4 Experiments and Discussion

We conducted a series of experiments to illustrate our methodology. The images we used are in bitmap format, and file size is 1024x768 bit by bit. There are 35 images in all. 8 images are cells without treatment, 9 images are cells with level-5 treatment, 9 images are cells with level-10 treatment and 9 images are cells with level 20 treatment. There are two sets of experiments. The first set of experiment demonstrates how the iterative training process increases the accuracy of endosome detection. The second set of experiments computes the set of metrics per cell and compare them with the metrics obtained from the manually markup cells.

Before we run the experiments, we need a ground truth endosome image as the reference image. Therefore we manually mark out the endosomes in each image, and compare endosome detected by the program with the manually mark out endosomes. The following figure shows the manually mark out endosome image. The black regions are the manually mark out endosomes. We superimpose our detected endosomes to this manually mark out image. If the detected endosomes have most pixels covered by black regions, then we set this endosome as true endosome, otherwise it is a false positive case.

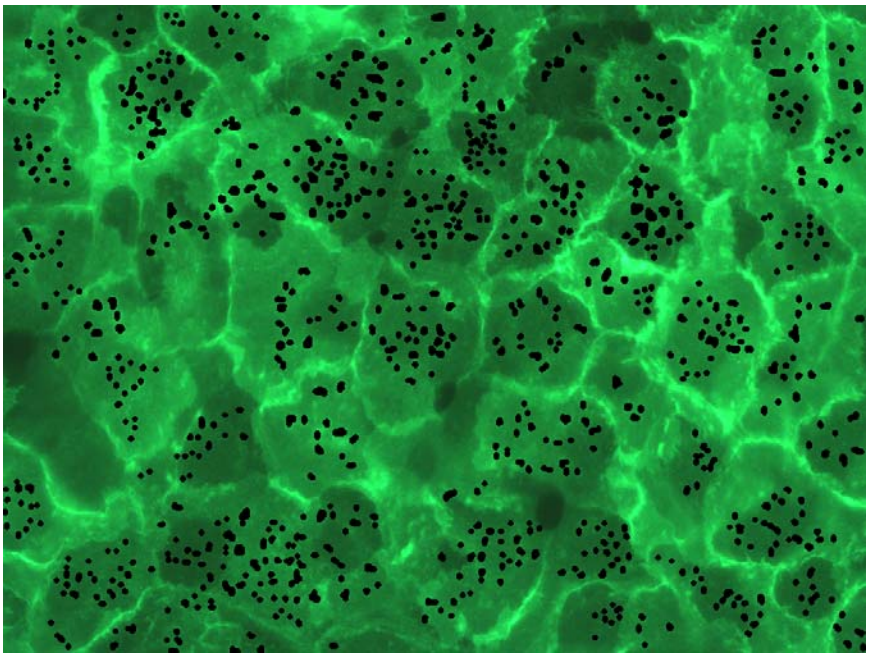


Figure 25: Manually mark out image

4.1 Endosome Detection Training

As we discussed in the previous chapter, due to the fuzzy cell boundaries and similar intensity profiles of endosomes and cell boundaries, the default classifier for endosome detection may not be ideal for all cell images. Therefore, we propose the iterative training process to improve the performance of endosome detection. We define two metrics to measure the endosome performance. The first metric is sensitivity, which is the ratio of true endosomes captured out of the total number of the manually mark out endosomes. The second metric is accuracy, which is the ratio of true endosomes captured out of the total detected endosomes.

The iterative training process consists of three steps:

1. Initial classification based on the default classifier.
2. Iterative learning from user and rebuild classifier for each round of learning.
3. Process whole batch of image with the rebuilt classifier.

The first step is to apply the default classifier to the image and get the initial detection of endosomes. Then during each round of learning, user indicate the positive and negative sample endosomes based on the previous round leaning result. The program will read in the user input and rebuild the classifier. The adjusting of parameters is defined as follows:

1. Size: Because endosomes are typically smaller than cell boundary edges, we choose the average value of minimum negative sample and maximum positive sample as the new threshold for the size.
2. Gap: Endosomes have smaller gap than cell boundary edges, therefore, we choose the new gap as the same way as size
3. Difference: Endosomes are like circles, and cell boundaries are like lines. Therefore, endosomes have smaller minimum distance to the coverage rectangle, so we choose the average value of minimum negative sample and maximum positive sample as the new difference value.

4. Ratio: This feature is to differentiate the vertical or horizontal cell boundaries from the endosomes. Therefore, endosomes have greater ratio than cell boundaries, because the coverage rectangles of endosomes are mostly like squares. Thus we choose the average value of maximum negative sample and minimum positive sample as the new threshold.
5. Co-distance: Endosomes edges are more compact than cell boundaries edges, so we choose the average value of maximum negative sample and minimum positive sample as the new threshold.
6. Neighbor count: More neighboring existing endosomes indicates high possibility of being endosomes. Therefore, we choose maximum negative sample and minimum positive sample as the new threshold.

This training process will continue until the user is satisfied with the results of the classification. In the third step, the final classifier will be used for the endosome detection for the whole batch of images. We randomly pick one image as training image each time. We use the same default classifier for the initial classification. For each training image, we first select one cell to train the program. Then for each round of training, we add one more cell to the training data set, and process the rest of the images with the rebuilt classifier. Figure 26 shows the plots of the average accuracy and sensitivity after each round of training.

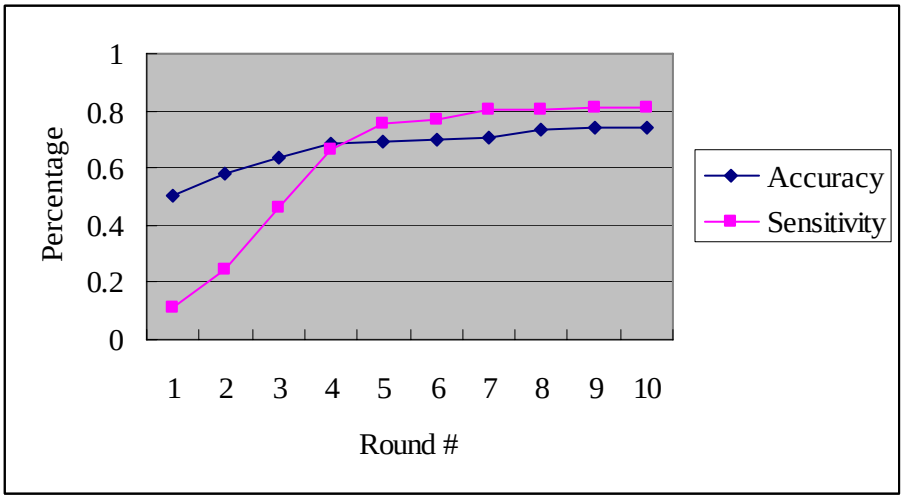


Figure 26: Avg. Accuracy vs. Sensitivity

Figure 26 shows that the average accuracy and sensitivity increases after three rounds of training. Then the accuracy and sensitivity stabilize at 74% and 80% respectively.

4.2 Cell Boundaries Detection

From the Figure 24, we noticed that the segmented cell boundaries contain many sharp corners. This is because fixed number of active contour pixels is used in the experiments. As an improvement, T-Snake can be implemented to detect cell boundaries. T-Snake highly depends on the affine cell image decomposition (ACID). The smaller the triangular grids are the more accurate contours can be detected by ACID.

Another observation is some contours do not cover the entire cell but just half. This is due to the use of velocity vector constraint. The effect of velocity vector constraint is like a chain reaction. If the contour slows to grow at one point, the points beside this point will also be affected, i.e. slow down as well. If the initial seed is quite close to the cell boundary, the contour points near the boundary will stop after just a few rounds of evolution. The stop of those points will be accumulated quickly and spread to other points, so that the whole evolution process will be stopped unexpectedly.

4.3 Metrics Computation

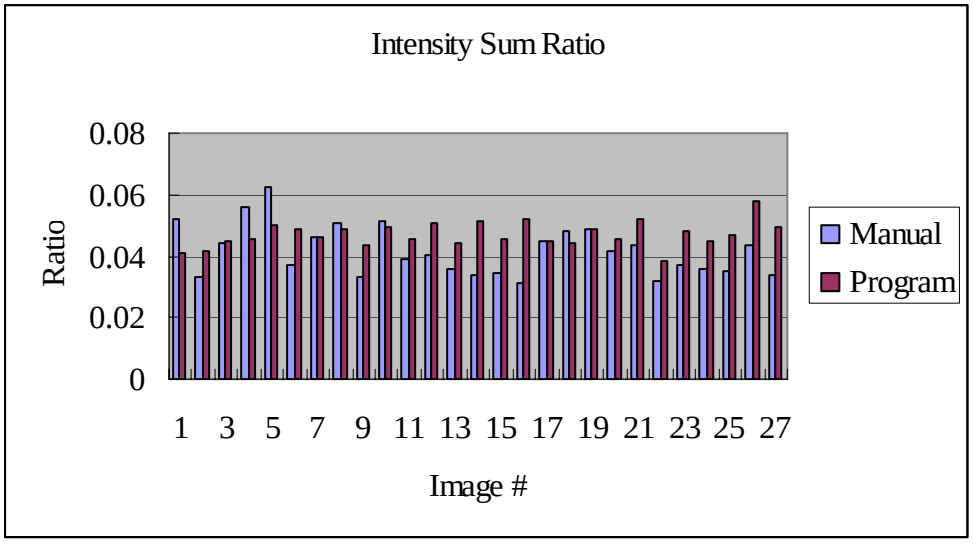
In this section, we will compute the metrics which have been defined in the previous chapter. The ultimate target for our cell segmentation method is to compute the metrics per cell to help the biologists in biological research. According to the pharmacological study, the intensity of cytoplasm (referred to microscopy image) and the number of endosomes in one single cell is an indication of the effect of a drug treatment. Thus, we defined those five metrics to measure the changes among different drug treatment in Table 1 (Chapter 3).

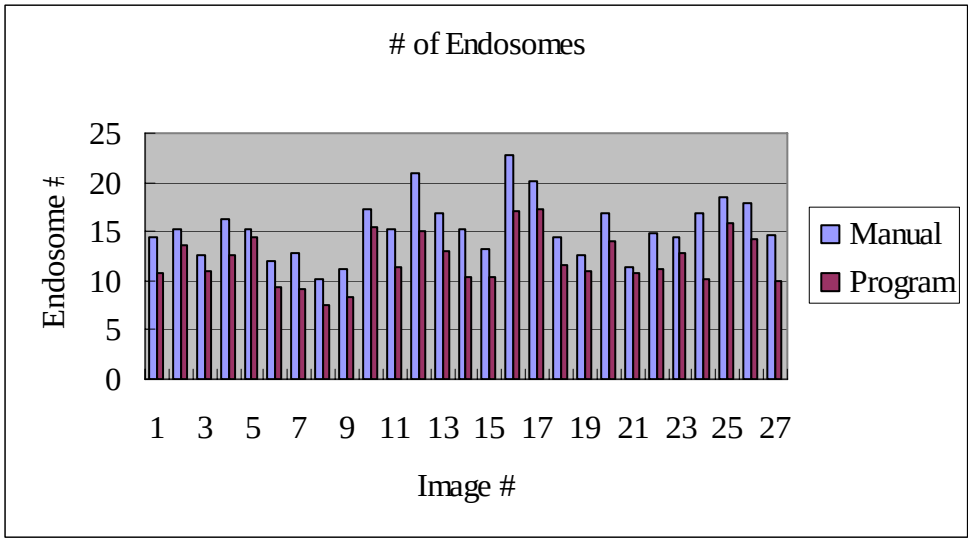
We first compute the intensity sum ratio and number of endosomes in each cell. For each detected cell, the intensity of each endosome pixel is accumulated, and then divided by the sum of the intensity of each cytoplasm pixel. Table 3 shows the result of our method and the manually mark out result:

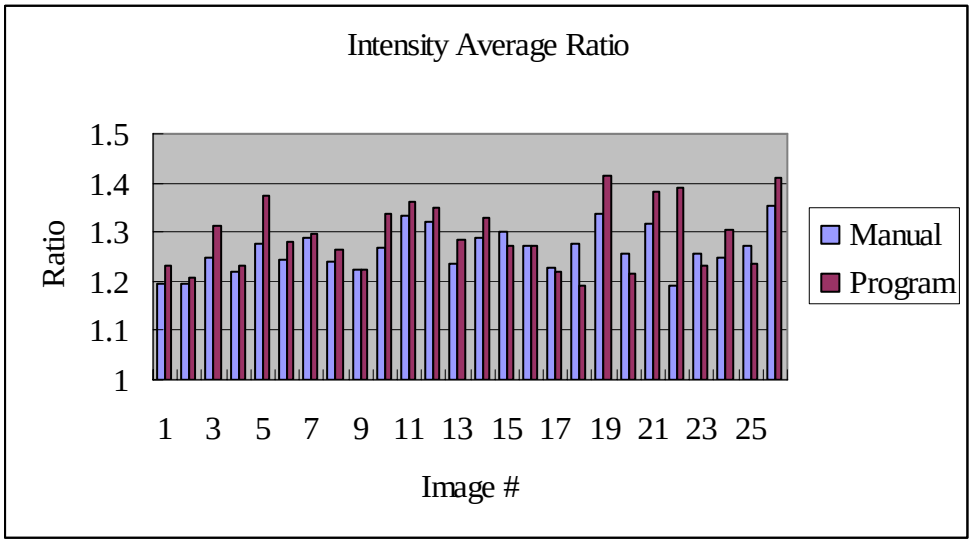
Methods	**Avg. Intensity Sum Ratio per cell**	**Avg. # of Endosomes per cell**	**Avg. Intensity Ratio per cell**	**Avg. Endosome Intensity per cell**	**Avg. Cytoplasm Intensity per cell**
Automatic Cell Segmentation	0.046993	12.62	1.259811	165	130
Human Measurement	0.041685	15.35	1.293262	159	127

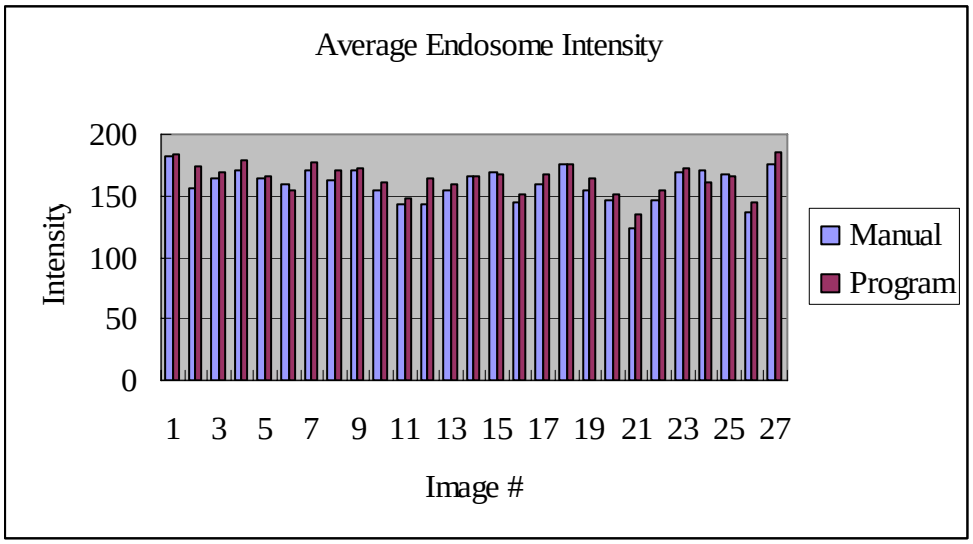
Table 3: Metrics per cell

Besides the metrics per cell, we also compute the metrics per image. The following figures show the individual metrics per image:




# of Endosomes
Endosome #
25
20
15
10
5
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27
Image #
Manual
Program


Intensity Average Ratio
Ratio
1.5
1.4
1.3
1.2
1.1
1
1 3 5 7 9 11 13 15 17 19 21 23 25
Image #
Manual
Program


Average Endosome Intensity
Intensity
200
150
100
50
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27
Image #
Manual
Program

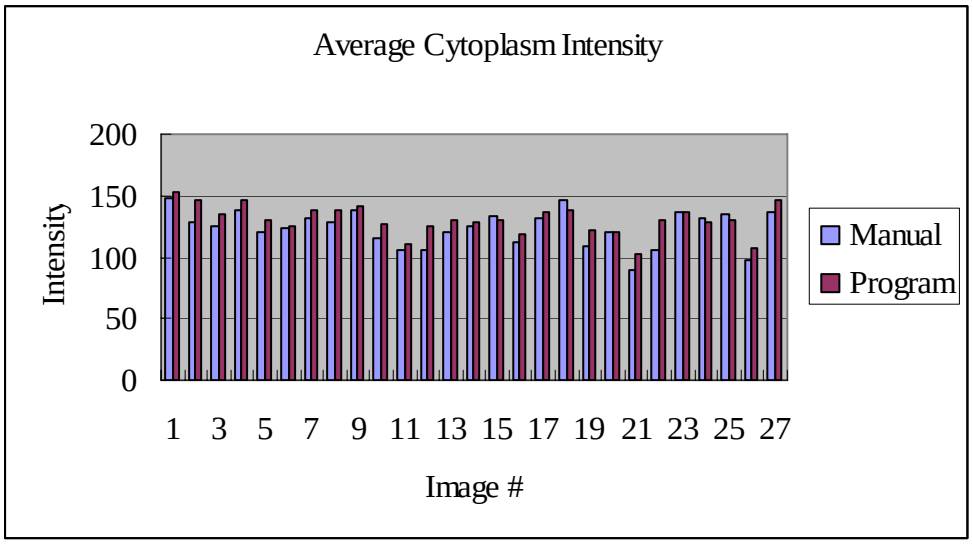


Figure 27: Column chart for metrics per image

Table 4 shows the average value, standard deviation for program computed and human computed metrics per image and the average difference between them.

Methods	**Avg. Intensity Sum Ratio per image**	**Avg. # of Endosomes per image**	**Avg. Intensity Ratio per image**	**Avg. Endosome Intensity per image**	**Avg. Cytoplasm Intensity per image**
Automatic Cell Segmentation	0.047084	12.14	1.264672	166	131
Human Measurement	0.040513	15.33	1.303577	160	125

Methods	**Std. Dev. of Intensity Sum Ratio per image**	**Avg. # of Endosomes per image**	**Std. Dev. of Intensity Ratio per image**	**Std. Dev. of Endosome Intensity per image**	**Std. Dev. of Cytoplasm Intensity per image**
Automatic Cell Segmentation	0.004042	2.61	0.066376	11.88055	12.00035
Human Measurement	0.008275	3.02	0.043908	13.66166	14.81202

Avg. Diff. of Intensity Sum Ratio per image	Avg. Diff. of Endosomes per image	Avg. Diff. of Intensity Ratio per image	Avg. Diff. of Endosome Intensity per image	Avg. Diff. of Cytoplasm Intensity per image
0.008493	3.19	0.044882	6.77	8.39

(c) Avg. Difference between automatic cell segmentation and human measurement

Table 4: Metrics per image

From the above tables, we find that our result is quite similar to the human measurement. Therefore, we conclude that our method is effective for our cell image segmentation problem.

5 Conclusion

Cell image segmentation is an essential technique in medical imaging analysis. There are a lot of existing algorithms to solve this problem. However, there is no standard solution that can fit to all cell segmentation problems due to the variety of cells and microscopes. The most difficult part of our problem is the image provided tends to overlap with blur edges and noises. Besides those noises, there are a lot of endosomes within cells, which bring more difficulties to segmentation. Most of the traditional edge detector or active contour algorithms are not so effective to this particular image. Therefore we describe a method that based on Garrido's method but with many improvements to solve the new cell segmentation problem. According to the biologists, the cell images with drug treatment will have the endosomes inside cell. How to give a proper measurement of the endosomes is a critical problem for biological studies. Therefore we proposed a method to provide assistance to biologists to obtain those measurements automatically.

The future work of this research work could be very challenging. There are two problems to solve. The first problem is in this batch of cell images, although the cells do not have clear boundaries, all the endosomes have quite clear and stable shapes. However, for some other cell images, the endosomes are not so clear. They always group very tightly and just located around the nucleus. Those endosomes look like cotton wool in the cells. Therefore the Canny detector cannot capture the endosomes directly. An image transform must be applied before the endosome detection, which is to make those "cotton wool" like endosomes clearer.

The second problem is how to differentiate the cell images under different drug treatments. Theoretically speaking, more drugs will produce more endosomes in cells. However, due to the fuzzy situations, such as overlapping cells, clutter cell boundaries, similar intensity profiles of endosomes and cell boundaries, etc. it is very difficult to make the program automatically classify the different drug treatments.

6 References

[1] R. Adams, L. Bischof, “Seeded region growing,” *IEEE Trans. Pattern. Recogn. Mach. Intell.* Vol. 16, no. 6, pp. 641-647, 1994.

[2] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes”, *Pattern Recognition*, vol. 13, pp. 111-122, 1981.

[3] O. Basset, Z. Sun, J.L. Mestas, G. Gimenez, “Texture analysis of ultrasonic images of the prostate by means of co-occurrence matrices,” *Ultrasonic Imaging*, 15, 218-237, 1993.

[4] M. Bomans, K. H. Hohne, Y. Tiede, M. Riemer, “3-D segmentation of MR images of the head for 3-D display,” *IEEE Trans. Medical Imaging*, Vol. 9, no. 2, pp. 177-183, 1990.

[5] M. V. Boland, R. F. Murphy, “A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of Hela cells,” *Bioinformatics*, vol. 17, pp. 1213-1223, Dec. 2001.

[6] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8, pp. 679-698, Jun. 1986.

[7] K. R. Castleman, *Digital Image Processing*. Upper Saddle River: Prentice Hall, 1996.

[8] V. Chalana, W. Costa, Y. Kim. “Integrating region growing and edge detection using regularization,” *Proceddings of the SPIE Conference on Medical Imaging*, SPIE, 1995.

[9] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790-799, Aug. 1995.

[10] T.F. Cootes, C.J. Tayylor, D.H. Cooperm, J. Graham, “Active shape models – their training and application,” *Comput. Vision Image Understanding*, vol. 61, pp. 38-59, 1995.

[11] C. Cortes, V. Vapnik, *Support Vector Networks*, Machine Learning 1995, 20:I-25.

[12] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, vol. 1. New York: Interscience, 1953.

[13] E. R. Damiano, J. Westheider, A. Tozeren, and K. Ley, “Variation in the velocity,

deformation, and adhesion energy density of leukocytes rolling within venules," *Circ. Res.*, vol. 79, pp. 1122-1130, 1996.

[14] J. G. Daugman, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. Acoust,, Speech, Singal Processing*, vol. 36, no. 7, pp.1169-1179, July 1988.

[15] O. Debeir, P. Van Ham, R. Kiss, C. Decaestecker, "Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes," *IEEE Trans. Med. Imag.*, vol. 24, pp. 697-711, Jun. 2005.

[16] E. R. Dougherty. *Mathematical Morphology in Image Processing*. Marcel Dekker, New York, 1992.

[17] Y.L. Fok, C. K. Chan, R. T. Chin, "Automated analysis of nerve-cell images using active contour models," *IEEE Trans. Med. Imag.*, vol. 15, pp. 353-368, Jun. 1996.

[18] A. Garrido, N. Perez de la Blanca, "Applying deformable templates for cell image segmentation," *Pattern Recognit.*, vol. 33, pp. 821-832, 2000.

[19] F. Girosi, T. Poggio. *Networks and the best approximation property*. A. I. Memo 1164, Massachusetts Institute of Technology, 10, 1989.

[20] J. M. Gonzalez-Linares, N. Guil, E. L. Zapata, "An efficient 2D deformable objects detection and location algorithm," *Pattern Recognition*, vol. 36, pp. 2543-2556, 2003.

[21] R. C. Gonzalez, R. E. Woods, *Digital Image Processing.* Reading, MA: Addison-Wesley Publishing Company, 1993.

[22] U. Grenander, "Pattern Synthesis," *Lectures in Pattern Theory*, vol. 1, *Appl. Math. Sci.* vol. 18, Springer, Berlin, 1976.

[23] H. Handels, T. Ross, J. Kreusch, H.H. Wolff, S.J. Poppl, "Computer-supported diagnosis of melanoma in profilometry," *Methods of Information in Medicine*, 38, 43-49, 1999.

[24] R. Haralick, L. Shapiro, "Image segmentation techniques," *Comput. Vision. Graph & Image Process*, vol. 29, pp. 100-132, 1985

[25] Kai Huang, Robert F. Murphy, "Boosting accuracy of automated classification of fluorescence microscope images for location proteomics," *Bioinformatics*, May. 2004.

[26] J. Illingworth, I. Kittler, “A survey of the Hough transform,” *Comput. Vision. Graph & Image Process*, vol. 44, pp. 87-116, 1988.

[27] A. Jain, S. Bhattacharjee, “Address block location on envelopes using Gabor filters,” *Pattern Recognition*. vol. 25, no. 12, 1992.

[28] A. Jain, N. Ratha, and S. Lakshmanan, “Object detection using gabor filters,” *Pattern Recognition*, vol. 30, pp. 295-309, 1997.

[29] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[30] Michael Kass, Andrew Witkin, Demetri Terzopoulos. “Snakes: Active contour models,” *International Journal of Computer Vision*, pages 321-331, 1988.

[31] A. Knoerr, Global Models of Natural Boundaries: Theory and Applications. Report in Pattern Theory 148, Brown University, 1988.

[32] B. S. Manjuanath, R. Chellappa, “A unified approach to boundary perception: edges, textures, and illusory contours,” *IEEE Trans. Neural Networks*, vol. 4, pp. 96-108, Jan. 1993.

[33] D. Marr, E. Hildreth, “Theory of edge detection,” *Proc. Roy. Soc. London*, vol. 27, pp. 187-217, 1980.

[34] T. McInereny, D. Terzopoulos, “Topology adaptive deformable surfaces for medical image volume segmentation,” *IEEE Trans. Med. Imag.*, vol. 18, pp. 840-850, Oct. 1999.

[35] T. McInerney and D. Terzopoulos, “Topologically adaptable snakes,” in *Proc. Fifth Int. Conf. computer Vision (ICCV’95)*. Los Alamitos, CA: IEEE Computer Society, 1995, pp. 840-845.

[36] D.P. Mukherjee, N. Ray, S.T. Acton, “Level set analysis for leukocyte detection and tracking,” *IEEE Trans. Image Processing*, vol. 13, pp. 562-572, Apr. 2004.

[37] J. Parkkinen, K. Selkainaho, E. Oja, “Detecting texture periodicity from the co-occurrence matrix,” *Pattern Recognition Letters*, 11, 43-50, 1990.

[38] J. C. Rajapakse, J. N. Giedd, J. L. Rapoport, “Statistical approach to segmentation of single channel cerebral MR images,” *IEEE Trans. Med. Imag.*, Vol. 16, no. 2, pp. 176-186, 1997.

[39] N. Ray, S.T. Acton, “Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours,” *IEEE Trans. Med. Imag.*, vol. 23, pp. 1466-1478, Dec. 2004.

[40] N. Ray, S.T. Acton, K. Ley, “Tracking leukocytes in vivo with shape and size constrained active contours,” *IEEE Trans. Med. Imag.*, vol. 21, pp. 1222-1234, Oct. 2002.

[41] A. Rosenfeld, A. C. Kak, *Digital Image Processing*. New York: Academic Press, 1982.

[42] J. C. Russ, *The Image Processing Handbook*. Boca Raton: CRC Press, 1999.

[43] P.K. Sahoo, A.K. Saltani, A.K.C. Wong, Y.C. Chen, “A survey of thresholding techniques,” *Comput. Vision, Graph & Image Process*, vol. 41, pp. 233-260, 1988.

[44] V. Toree, T. A. Poggio, “On edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell*., Vol 8, pp. 147-163, 1986.

[45] M. R. Turner, “Texture discrimination by Gabor functions,” *Biological Cybernetics*, vol. 55, pp. 71-82, 1986

[46] J. Z. Wang, D. A. Turner, M. D. Chutuape, “Fast, interactive algorithm for segmentation of series of related images: application to volumetric analysis of MR images of the heart,” *JMRI*, vol. 2, no. 5, pp. 575-582, 1992.

[47] T. P. Weldon, W. E. Higgins, and D. F. Dunn, ``Gabor filter design for multiple texture segmentation," *Optical Engineering*, vol. 35, no. 10, pp. 2852-2863, Oct. 1996.

[48] J. S. Weszka. “A survey of threshold selection techniques,” *Computer Graphics and Image Proc.*, vol. 7, pp. 259-265, 1978.

[49] C. Xu and J.L. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE, Trans. Image Processing*, vol. 7, pp. 359-369, Mar. 1998.

[50] T. Zhao, M. Velliste, M. V. Boland, R. F. Murphy, “Object type recognition for automated analysis of protein subcellular location,” *IEEE Trans. Image Processing*, vol. 14, pp. 1351-1359, Sept. 2005.

[51] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[52] S.K. Murthy. “Automatic construction of decision trees from data: A

multi-disciplinary survey," *Data Mining and Knowledge Discovery*, 2(4):345-389, 1998.

[53] S. W. Zucker, "Region growing: childhood and adolescence," *Computer Graphics Image Proceedings*, vol. 5, pp. 382-399, 1976.

Appendix A: Cell Analysis Tool

We use Visual C++ 6.0 to implement our tool. There are three main functions provided in our tool: (1) Endosome training, (2) Cell contour extraction, (3) Metrics computation. The following figure shows the user interface of our tool:

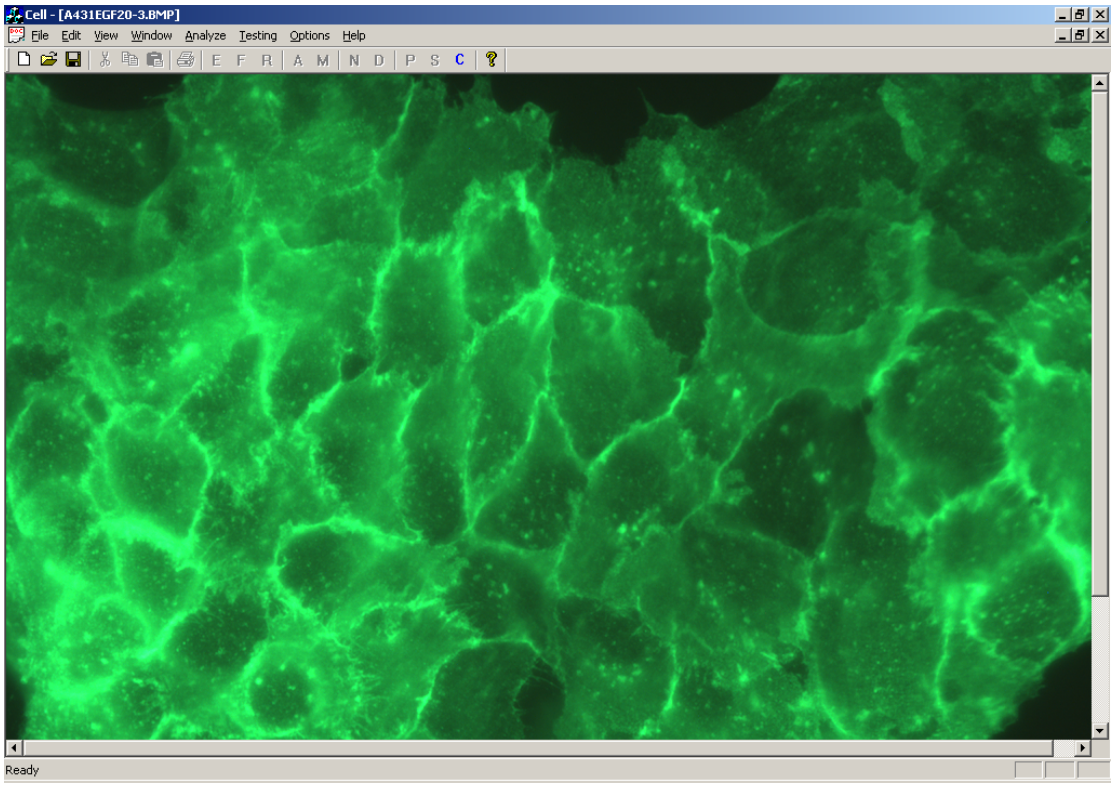


Figure A1: Program interface

User can use the default classifier to detect endosomes by selecting "Analyze->Training", which is shown in Figure A2:

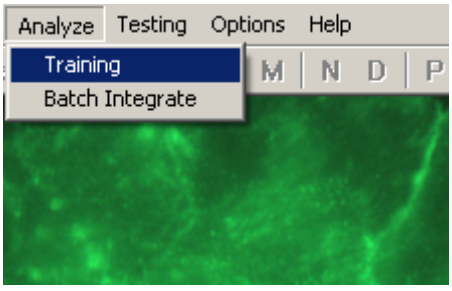


Figure A2: Initial image processing

This method will detect the endosomes by the default classifier, and show them on the display panel. In Figure A3, the red segments are detected endosomes, purple

segments are confirmed edge segments and white segments are unknown segments.

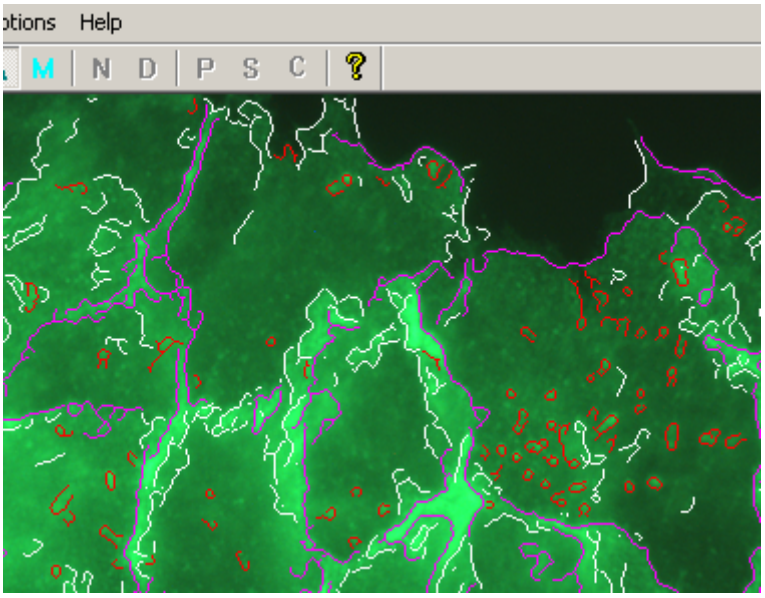


Figure A3: Result of endosome detection by default classifier

User can enable the endosome editing mode by selecting “Options->Endosome training->Enable training mode” to iteratively train the endosome detection method, which is shown on the Figure A4.

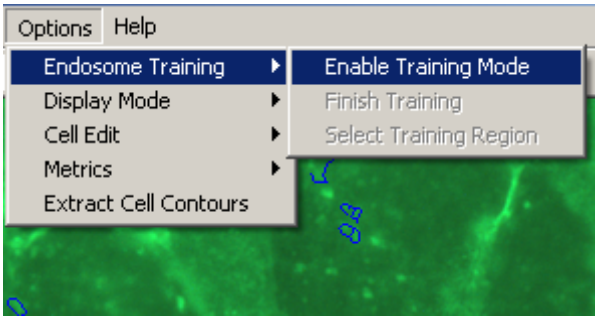


Figure A4: Enable training mode

Once the user has enabled the training mode, he can add new endosomes or delete detected false endosomes by clicking on the display panel, which is shown on the Figure A5. The red dots are the newly added endosomes. User can also specify the size of newly added endosomes. After editing, user needs to drag the mouse to draw a

rectangle to tell the program where the learning region is. After specifying learning region, user can select “Options->Endosome training->Finishing Training” to let the program start classifier rebuilding process.

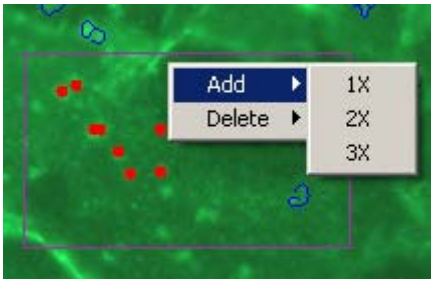


Figure A5: Add and delete endosomes

Once user is satisfied with the result endosome detection, he can extract the cell contour by selecting “Options->Extract cell contours”, which is shown in Figure A6:

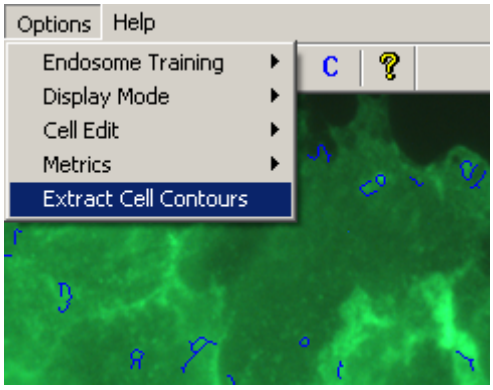
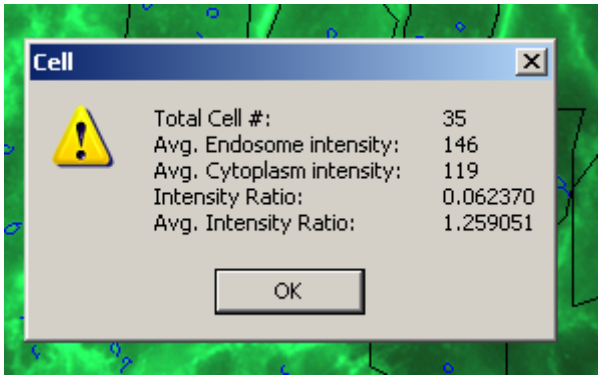
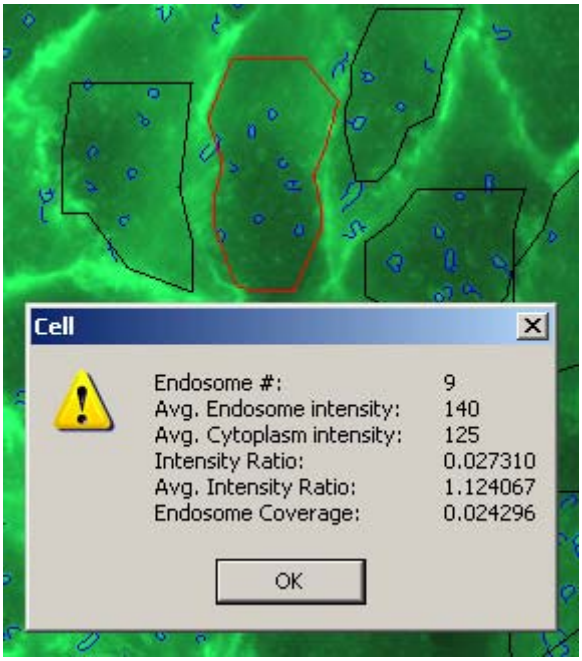


Figure A6: Cell contour extraction

The cell contours will be shown on the display panel, which are closed black contours on Figure A7. If user is not satisfied with the cell contour detection, he can delete that cell and add a new one manually. There are two types of metrics computation, metrics per cell and metrics per image. User can select “Options->Metrics->All Cells” to compute the metrics per image, or select single cell to compute metrics per cell. Figure A7 shows the metrics per image and per cell:



(a) Metrics per image



(b) Metrics per cell

Figure A7: Metrics per image and per cell

We also provide Save and Load function in our tool. Save function can save the endosome and cell information of current image into save file. Load function can load the endosome and cell information from the save file to resume previous cell analysis work.