An Image Analysis System for Measuring Shape and Motion of White Blood Cells from a Sequence of Fluorescence Microscopy Images

Luren Yang

Contents

Chapter 1

Introduction

The objective of the research is to develop an image analysis system to measure the motion of white blood cells from a temporal sequence of fluorescence microscopy images.

The human white blood cells provide a major defense against infection through a complex process in which the cells are attracted to and move towards the site of infection. Calcium ions (Ca^{2+}) plays an important role in this activation, and the relation between the cell motion and the calcium concentration thus becomes an interesting research topic. Examination of individual cell behavior by microscope is an important research method, and digital image processing and analysis techniques may offer an effective and efficient approach in this respect.

In the Department of Physiology, University of Oslo, a microscopy fluorescence cell imaging system has been developed, along with an image analysis system which measures the concentration of the calcium ions in the living cells.

The work described in this thesis is to measure the cell motion from a sequence of digital fluorescence images, so that the relation between the calcium concentration and the cell motion can be studied.

Like many other image analysis systems, the cell motion analysis system has three major parts: image segmentation, feature extraction and feature analysis. This thesis mainly concentrates on the first two parts, while the third part is also discussed. Noise reduction is used as a preprocess to the segmentation. A two-pass spatial-temporal segmentation system is proposed. In the first pass, an initial segmentation is applied to classify the pixels into cell and background pixels. Region labeling, correction and cell tracking are then done in the second pass to give a final segmentation. Motion measurement is based on shape features estimated from binary regions after the segmentation. The accuracy of the shape features is used to evaluate the segmentation results.

Including the current one, this thesis consists of 12 chapters, one appendix and a bibliography.

Chapter 2 gives the background knowledge associated with white blood cells and fluorescence cell imaging. Two aspects of cell motion are described.

Chapter 3 analyzes the problems in this image analysis task. Experiments in-

clude the examination of the gray level histograms and the topographical structures of the image intensity surfaces.

Chapter 4 presents the test results of several noise reduction methods used as a preprocess to the segmentation. The effects of the noise reduction to different segmentation algorithms are discussed.

Chapter 5 provides a review of edge-based segmentation methods and the experimental results of some of the methods for the fluorescence cell images.

Chapter 6 describes three types of gray level thresholding techniques: global, local and dynamic thresholding. A new dynamic thresholding method is presented.

Chapter 7 presents a two-pass spatial-temporal image segmentation system for the fluorescence cell image sequences. Techniques discussed in Chapter 5 and 6 are used in the first pass. Cell object tracking, region closing and boundary smoothing are done in the second pass. Chapter 4, 5, 6 and 7 lead to a complete segmentation system, which is evaluated in Chapter 11.

Chapter 8 contains a list of shape features. Cell motion description based on the features is discussed.

Chapter 9 discusses and tests several area and perimeter estimators applied to circles. The results should be useful for blob-like objects whose boundaries can be considered as chains of circular arcs.

Chapter 10 discusses the problem of fast and accurate computation of geometric moments, from which many shape features can be computed. A new moment computation method based on a discrete version of Green's theorem is presented. The precision of the computation is evaluated using Hu's moment invariants and several moment-based shape features.

Chapter 11 reviews methods of quantitative evaluation of image segmentation, and provides the evaluation results of the segmentation of the fluorescence cell images based on a supervised evaluation method. Some of the shape features are applied to the segmented cell objects, and the results are discussed. The time sequence of these features, i.e., the basis of the motion analysis methods given in Chapter 8, is also examined.

Chapter 12 contains a summary and discusses the relation between this work and the most closely related works by other authors.

Appendix A briefly describes an image processing software package, XITE, which was used in this work.

Chapter 2

Medical Background

In this brief introduction to the medical background, I first present some elements in the physiology of the human blood cells, especially of the white blood cells which concern this work. Then I describe a fluorescence cell imaging system built up in the Department of Physiology, University of Oslo. The fluorescence images recorded by this system are used to measure the intracellular calcium concentration ($[Ca^{2+}]$) and the cell motion simultaneously. The method to measure the calcium concentration is briefly described in this chapter; the method to measure the motion is the main part of the work and will be covered throughout this thesis.

2.1 Some Physiological Aspects of Blood Cells

The cellular constituents of human blood include red blood cells (erythrocytes), a variety of white blood cells (leukocytes) and platelets. Five classes of white blood cells have been recognized: neutrophils, eosinophils, basophils, monocytes and lymphocytes. The first three types are described collectively as granulocytes. The various types of the cells are distinguished in blood smears by their morphological and tinctorial characteristics when stained with a mixture of dyes. Pictures presented in the textbook of Berne and Levy [BL93] illustrate the morphology and tincture of the blood cells.

Of the white blood cells, 40% to 75% are neutrophils, which provide a major defense against infection by bacteria. In the defense process, the cells migrate towards the infection area. Two aspects of the motion have been studied: **locomotion** and **pseudopod**. It is known that the locomotion (moving from place to place) of the cells plays an important role in the defense mechanisms. Interest in cell locomotion is reflected in many papers, especially those dealing with the directed locomotion – chemotaxis [WH88]. The pseudopod means the changes in the membrane shape that occur during the locomotion. It has also been studied in order to understand its contribution to the locomotion and the sensing mechanisms located at the cell surface [LNY83].

Three major types of locomotion of the white blood cells were defined by Keller

et al. [KWA+80]:

Random locomotion "A type of locomotion that is random in direction. The axis of the moving cell or organism is not orientated in relation to the stimulus."

Chemokinesis "A reaction by which the direction of locomotion of cells and/or frequency of turning (change in direction) of cells or organisms moving at random is determined by substances in their environment. Chemokinesis is said to be positive if displacement of cells moving at random is increased and negative if displacement is decreased. Two forms of kinesis have been distinguished; orthokinesis, a reaction by which the speed or frequency of locomotion is determined by the intensity of the stimulus; and klinokinesis, a reaction by which the frequency or amount of turning per unit time is determined by the intensity of the stimulus."

Chemotaxis "A reaction by which the direction of locomotion of cells or organisms is determined by substances in their environment. If the direction is towards the stimulating substance, chemotaxis is said to be positive; if away from the stimulating substance, the reaction is negative."

The importance of studying the cell surface was stated by Wessells [Wes79]: "It has become increasingly evident that the cell surface plays a truly pivotal role in the life, development, and regulation of cells. On one hand, the surface functions in the transmission of information from the environment to the cell, and here I mean not only molecular signals, but also mechanical forces stemming from adhesions and junctions that affect the cytoskeleton and so intracellular activations. The surface is also, in a real sense, an expression of the cell's genetic information and developmental biologists must pay increasing heed to the cell surface and to its changing properties."

As one of the transmitted signals, calcium (Ca^{2+}) plays an important role in many cell activities, and is considered as a trigger or regulator. The relation between the calcium signalling and the cell motion was studied by many authors [Lew89, KCWL90, JTS+91, TYIW92]. To further this study is the purpose of this work.

2.2 Fluorescence Imaging

In this section, I describe a fluorescence imaging system for white blood cells, and the method to measure the calcium concentration from the fluorescence images. For details I refer to the report of Røttingen [Rø91].

2.2.1 Cell Preparation

Human blood was collected from healthy volunteers. The white blood cells were isolated by means of centrifugation. White cell pellets were then washed and resus-

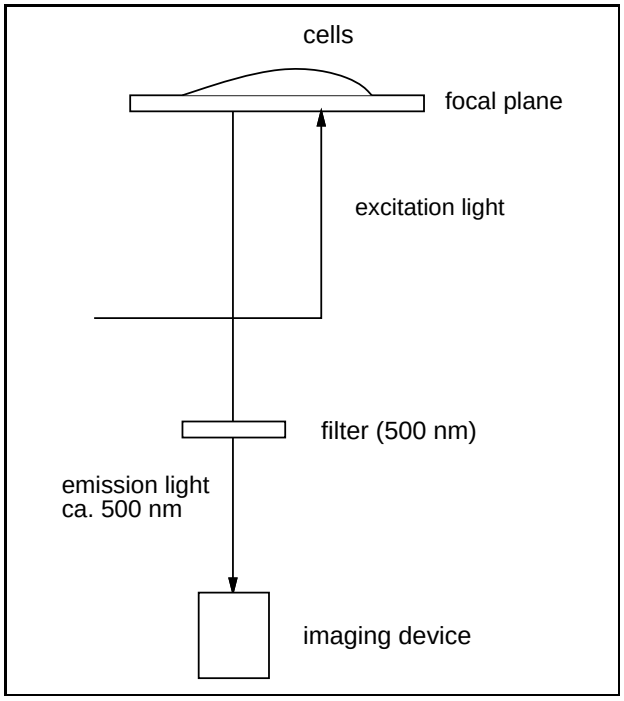


Figure 2.1: *A fluorescence imaging system.*

pended. Only neutrophils were investigated in this work. They were not separated from the other white blood cells, but could be identified later from the images.

2.2.2 A Fluorescence Imaging System

The fluorescence imaging system used in the Department of Physiology, University of Oslo, is illustrated in Figure 2.1. The white blood cells were placed on the image focal plane, dyed with a fluorescence indicator called Fura-2. Given an excitation light to the cells, the fluorescence emission was imaged. The system was capable of taking up to 5 images per second, while the living cells were moving on the image focal plane. The fluorescence images were digitalized to 256×256 pixels in spatial resolution and 256 gray levels. Each cell occupies 200 to 700 pixels. A sequence of static scene images represents a moving scene.

Figure 2.2 shows a fluorescence image where the cells have high intensity, while the background has low intensity. The intensity of a cell depends on the concentration of intracellular calcium ions and the thickness of the cell.

2.2.3 Measuring Intracellular Calcium Concentration

The fluorescence indicator Fura-2 is capable of binding to different ions inside the cells. Depending on the binding, the indicator emits fluorescence when exited by light of a particular wavelength. Free Fura-2 has an optimal excitation wavelength at 380 nm. Ca^{2+}-bound Fura-2 has an optimal excitation wavelength at 340 nm. The

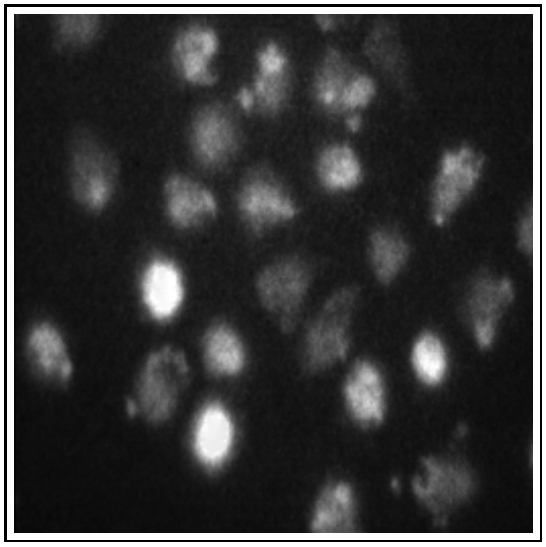


Figure 2.2: *A fluorescence cell image.*

two excitation wavelengths were applied alternately, and the ratio of the emission intensity at two different excitation wavelengths was used to measure the concentration of calcium ions. Let the intensity of fluorescence emitted during excitation at 380 nm be I_{380}, and the intensity of fluorescence emitted during excitation at 340 nm be I_{340}. The ratio $R = I_{340}/I_{380}$ depends on the concentration of calcium ions and not on the path length through the cytoplasm or the dye concentration. Calcium concentration is then measured as [RLH90]

$$[\mathrm{Ca}^{2+}] = K_d \beta \frac{R - R_{min}}{R_{max} - R} \tag{2.1}$$

where R_{max} and R_{min} are the ratios I_{340}/I_{380} in the presence of saturating calcium and zero calcium, respectively; β is the ratio of fluorescence intensity at 380 nm in zero and saturating calcium; and K_d is the dissociation constant of Fura-2 for calcium [SWF87].

The motion of the cells will be measured from the same image sequences from which the calcium concentration is measured so that the relation between them can be studied.

Chapter 3

Problem Analysis

The objective of this work is to develop an image analysis system which measures the motion of white blood cells from a sequence of fluorescence images. In this context, motion refers to the change in spatial mass distribution, size, position and orientation. Like many other image analysis systems, this system should have two major parts: image segmentation and feature extraction. The segmentation is to divide an image into meaningful regions [RK76, Pra91, GW92]. In this case, a cell occupies a two-dimensional connected region in each image frame. Some geometric features of the regions should be computed in order to measure the shape and the motion of the cell. In this chapter, I first describe a manual segmentation process which was used in order to provide test data for an automatic segmentation system, and then analyze the problems in the two parts according to a few experiments.

3.1 Test Data

Three fluorescence image sequences transferred from the Department of Physiology consist of about 50 frames (50 single images) each. A single frame has 256×256 pixels of 256 gray levels. The pictures are recorded at a frequency of 3 frames per second. The three sequences record the random locomotion and chemokinesis of the cells. Thus, in the present experiment, cell motion is not driven by a chemical gradient.

To provide test data for an automatic segmentation system, some cell physiologists[1] were invited to mark the cell boundaries manually on the digitalized fluorescence images. Software for marking the boundaries is available in XITE, an image processing software package (See Appendix A). The mark is of one pixel width. The manual segmentation was done as follows: First, we randomly chose a subsequence consisting of 10 adjacent image frames from each of the three image sequences we had. Then we chose some cells from the subsequences. For each of the cells, bound-

[1]They were Professor Jens-Gustav Iversen, and research fellows Jan Sigurd Røtnes and John-Arne Røttingen, Department of Physiology, University of Oslo, who participated in the development of the fluorescence imaging system and the calcium concentration measurement system.

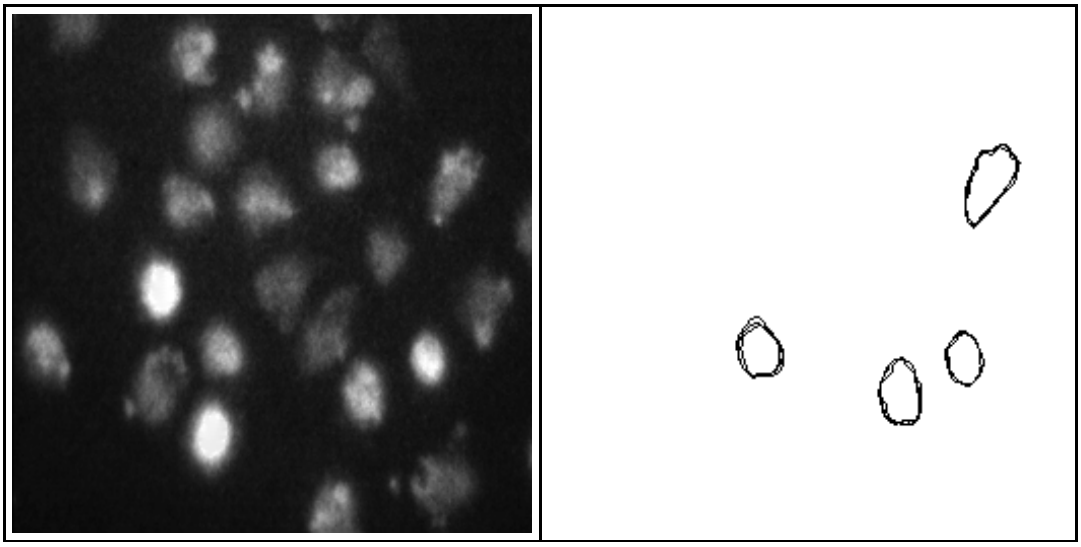


Figure 3.1: (left) *A gray level fluorescence cell image.* (right) *Boundaries of 4 cells in the image to the left, marked by three different persons.*

aries were drawn in the 10 frames. Three persons manually classified the same set of data described above. So for each cell we have three different boundary drawings. The manual segmentation results are represented as binary images. An example is shown in Figure 3.1.

3.2 Problems in Segmentation

Segmentation is one of the major parts in this image analysis system. I analyzed the problems in the segmentation according to a few experiments including the examination of the histogram and the topographical structure of the gray level surface.

3.2.1 Experiments and Results

From Figure 3.1 we can see that the cells generally have higher intensity than the background.

The histogram of the gray level values in an image is an important feature of the image [RK76, Pra91, GW92], and is usually examined, as a rule of thumb, in order to apply an image segmentation.

The normalized histogram of the image shown in Figure 3.1 is given in Figure 3.2. We see that the histogram has a main peak to the left, indicating the gray levels of the majority of the background pixels. To the right of the histogram there is a long tail, indicating that the gray levels of the cell pixels might be distributed over a long range. We can not find a clear valley which separates the two intensity classes, the background and the cells.

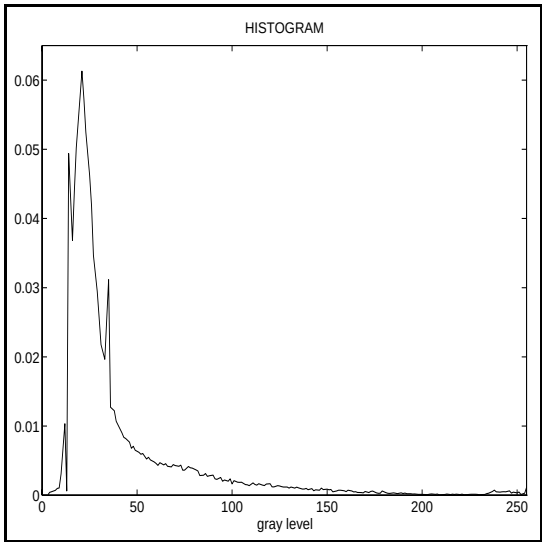


Figure 3.2: *The normalized histogram of the image shown in Figure 3.1(left).*

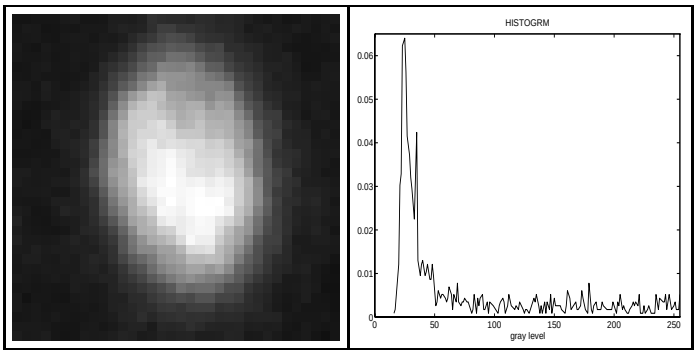


Figure 3.3: *A part of the image and its histogram.*

I also examined the histogram of a subimage containing only one cell, where the areas occupied by the cell and the background were nearly equal. Figure 3.3 shows a part of the image together with its histogram. We can still see a long tail to the right of the histogram.

A two-dimensional (2D) image can be viewed as a surface in the three-dimensional (3D) space $R^2 \times R^+$, where R^2 is the spatial space and R^+ is the space of the intensity values. The topographical structure of the surface gives valuable information for image analysis [Mus93].

The 3D surface of the intensity function of the cell shown in Figure 3.3 is plotted in Figure 3.4(left), where the origin corresponds the upper-left corner of the image, the x-axis is the vertical axis of the 2D image, and the y-axis is the horizontal one. I produced a 2D profile from the 3D surface by letting $y = 17$. The profile is shown in Figure 3.4(right) in the solid line. The vertical lines indicate the positions of the cell boundaries marked by the three manual segmentations. Note that the three persons marked the boundary differently.

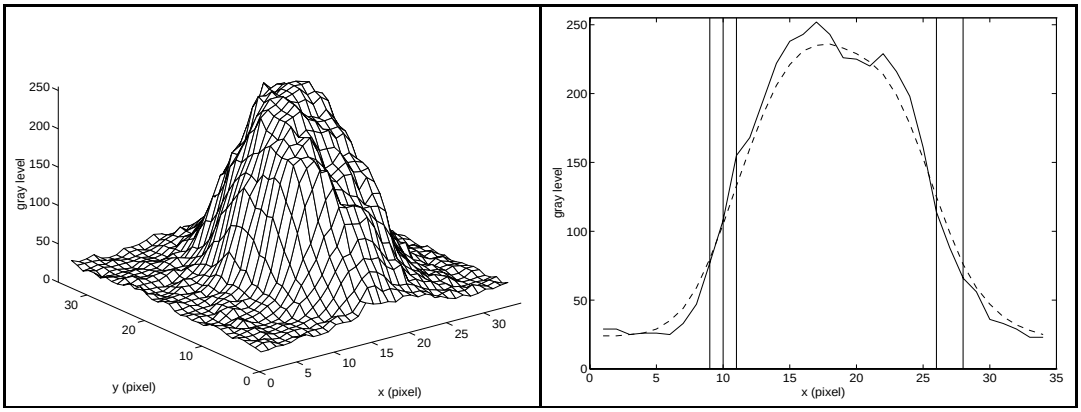


Figure 3.4: (left) *The 3D intensity surface of a cell.* (right) *The 2D profiles of the intensity function of the images with (dashed line) and without (solid line) noise reduction. The vertical lines indicate the positions the cell boundaries marked by the manual segmentation. (Two markings are overlapping.)*

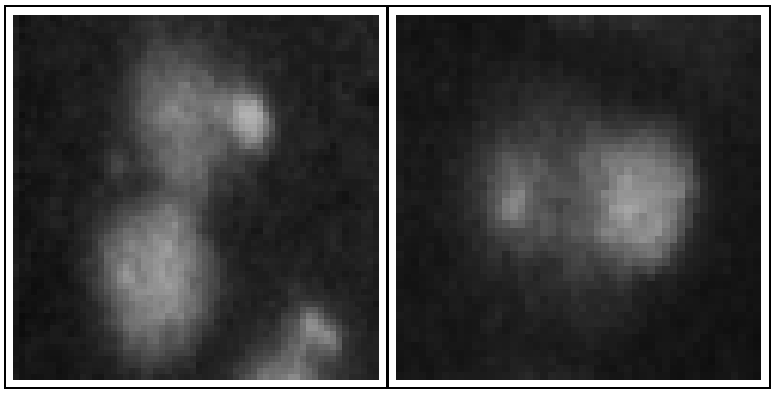


Figure 3.5: (left) *Two cells are closely located.* (right) *A cell having two light spots.*

The 3D intensity surface confirms that the intensity of a cell is distributed over a long range. A cell usually looks like a hill. Within the cell, the intensity is higher in the central part, and reduces gradually from the central part to the boundary. The boundary of the cell is in the hillside, where the intensity is smoothly changing. Step discontinuity of the intensity between the cell and the background is hard to observe. The intensity of the background near the cell is high. This is due to the fluorescence diffusion.

Noise reduction is often used as a preprocess of the segmentation [RK76, Pra91, GW92]. To show the effect of noise reduction, I used a Gaussian filter ($\sigma = 2.0$) to smooth the image. (See Chapter 4 for more about noise reduction.) The profile of the smoothed image is also shown in Figure 3.4(right), represented by the dashed line. The noise reduction removed small local intensity peaks. I also examined the histogram after the smoothing, and found that the noise reduction did not change the shape of the histogram significantly.

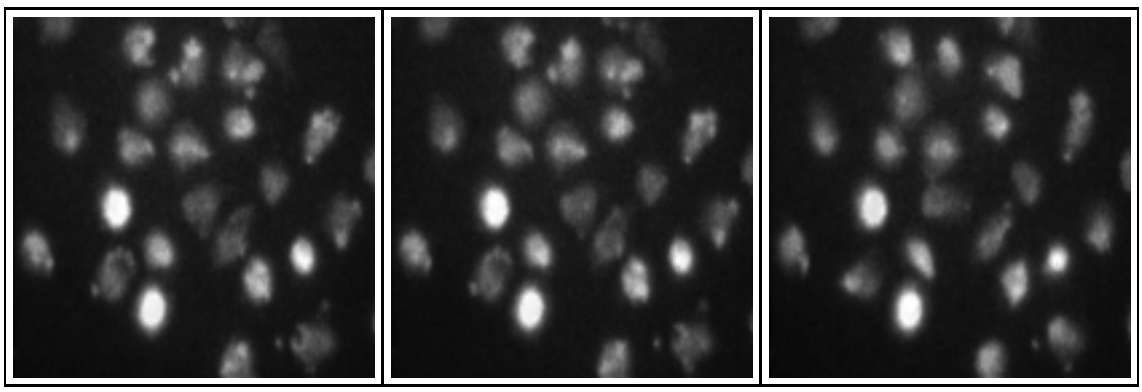


Figure 3.6: *Three frames of an image sequence. From the left to the right, the 30th, 31st and 45th frames of the sequence.*

Sometimes two cells can be closely located, as shown in Figure 3.5(left). However, the two light spots shown in Figure 3.5(right) are indeed of one cell. The segmentation system should be able to discriminate these two situations.

To illustrate the motion of the cells, Figure 3.6 shows three image frames from a sequence. The first two are subsequent (the 30th and 31st frames of the sequence), and the third is a frame at a later time (the 45th frame). From the image sequences I observed that the intensity, position and shape of a cell were all changing over the time. But the change was not fast. I found from the test data that the displacement of a boundary point could be at most 3 pixels between two frames. The displacement D_p of the boundary point p is defined by

$$D_p = \min_{p' \in B'} d(p, p') \tag{3.1}$$

where B' is the boundary in the last frame, and $d(p, p')$ is the distance between p and p'. The displacement of 3 pixels is a small compared to the diameter of a cell which is about 20 pixels. These images show the random locomotion and the chemokinesis of the cells. When a gradient of chemical stimulus is present, the motion will be directed, but will not be faster [Rø94].

3.2.2 Discussion

Segmentation is to divide an image in meaningful regions, i.e. the cells and the background. A variety of segmentation methods have been developed, as surveyed by Fu and Mui [FM81], Haralick and Shapiro [HS85] and Pal and Pal [PP93]. Different methods have different assumptions about the boundary and region. Edge detection, gray level thresholding, region growing and texture discrimination are commonly used techniques for single-channel gray level images. Fu and Mui [FM81] stated: "There are no general (segmentation) algorithms which will work for all images. One of the reasons that we do not have a general image understanding system is that a two dimensional image can represent a potentially infinite number of possibilities."

As a result, the segmentation has traditionally been an *ad hoc* process, and the selection of a segmentation method is often based on testing and evaluation.

We have seen that the boundary of a cell is located in the hillside if we consider the cell as a intensity hill. This implies that the edge detection can be used. An edge point refers to the position where the gradient of the intensity has a local maximum. Different digital edge detectors have been developed, and they may give different edge positions. We have to investigate how well the edge positions approximate the cell boundaries.

Gray level thresholding can also be applied since the cells have generally higher intensity than the background. There are many thresholding methods which can be classified as global, local, and dynamic thresholding (see Chapter 6.4). We have seen in global and local histograms that there is no clear valley which separates the two classes, the cells and the background. The intensity of the cells is distributed in a long range. Thus we have to look for a thresholding method which gives accurate results in this situation.

Most basic segmentation methods deal with 2D images. To segment a spatial-temporal image sequence, the objects should be tracked after a 2D segmentation.

We have seen the situation of two closely located cells, and a cell with more than one light spot. One way to detect such situations is to used the temporal information, e.g., the result of the last frame.

It is very common to apply noise reduction before the segmentation process. The purpose of the noise reduction is to reduce the white noise from the imaging channel, and reduce other noise including texture details. Small details are to be smoothed out to obtain information in a desired scale, and to make the segmentation easier. The noise reduction in this case is a preprocess of the segmentation, rather than a process of image restoration or enhancement. Therefore, to choose a noise reduction method, we need to consider the segmentation method to be used. To evaluate a noise reduction method, we have to consider the result of the segmentation. Many segmentation methods are designed to be optimal for a certain type of ideal boundary without noise. Some of them can be more sensitive to the noise than others.

The manually drawn cell boundaries can serve as test data for the evaluation of segmentation results. Of course, error may occur during the manual segmentation process. We may see from Figure 3.1 that the three persons marked the boundaries differently. If one assumes that the estimation is unbiased, then the error can be reduced by a certain averaging method. However, a bias is possible. The reason for that can be a mistaken understanding. It is therefore desirable to make the segmentation system trainable, so that the system parameters can be adapted to any new understanding of the boundary location.

3.3 Problems in Feature Extraction

The motion of an object refers to the change of the geometric properties including the spatial structure, size, position, and orientation. In order to measure the motion

of an object, we have to measure the geometric properties of the object in each static scene. Many shape features, such as area, perimeter, circularity, and image ellipse, are available to quantify the geometric properties of an object [Mar89, Jai89]. The motion can be quantified by using a temporal sequence of such shape features.

A cell looks like a fried egg when it lies on the image focal plane of a microscope. Although a cell moves in three dimensions, the vertical motion is much less important than the horizontal motion parallel to the focal plane. The motion of a cell to be considered is thus planar. Two aspects of cell motion, locomotion and pseudopod, are of interest.

After image segmentation, the regions of the cells are labeled. Shape features are then estimated from the discrete regions. The accuracy and the efficiency of the feature computation should be considered. The selection of the features for measuring the cell motion is dependent on the medical phenomenon to be studied.

Chapter 4

Image Smoothing

A smoothing process is usually required as a preprocess to image segmentation to reduce the noise, and remove the fine details in an image in order to obtain information in a proper scale.

I first give a review of image smoothing methods, and then present results of some of the methods. In this chapter, we will examine the effect of the smoothing visually according to the segmentation results, and to the changes of the topographical structure of the image gray level surface. A quantitative evaluation will be given in Chapter 11.

4.1 A Review of Methods

Many image smoothing filters are discussed in the literature [CY83, Mas85, AGL83, dBC90, WWL92]. Linear low-pass smoothing can be done in spatial domain by a discrete convolution of an input image with an impulse response array called spatial mask. Each pixel in the input image is replaced by a weighted sum of its neighborhood. Typical linear smoothing filters include the mean and Gaussian filter [Mar82].

Median filter [HYT79] is a popular nonlinear filter which is especially useful in removing isolated noise points. Many other nonlinear methods have been proposed for edge preserving noise filtering. They include K-nearest neighbor filter (KNN), K-nearest connected neighbor filter (KNCN), symmetric nearest neighbor filter (SNN) and maximum homogeneity filter (MAXH) [Lø88, dBC90]. Mode filter is a nonlinear filter often used for binary images and classification maps [Nib86]. Morphological operations, often used to smooth the boundaries of binary objects [SG91], can also be used to smooth the image gray levels [KAN91]. Mode filter and morphological operations will be discussed in Chapter 7 as a shape correction method.

Different filtering methods (linear or nonlinear) can be used sequentially to obtain a combined effect. Many filters can be iterated to obtain a further smoothing.

4.1.1 Mean Filtering

The mean filter is the simplest linear spatial filter. The value of each pixel in the input image is replaced by the average of the pixels inside a window centered at this pixel. The performance of this filter should be good on homogeneous regions. Edges will be blurred after the filtering. A few iterations of the mean filtering will closely approximate a Gaussian filtering [Wel86]. The mean filter is separable, so that the filtering can be done by row and column averaging separately. The row and column averaging can be done by updating, using a so-called running average technique [Mas85]. So the mean filter is computationally efficient.

4.1.2 Gaussian Filtering

In Gaussian filtering, an image is convolved by a Gaussian kernel:

$$H(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{4.1}$$

where σ is the standard deviation, determining the degree of blurring. The size of the Gaussian mask must be large enough to cover the pixels of a distance of two or three σ from the center.

The Gaussian is an important smoothing filter, since it is the only known filter which does not create any spurious features. This property was studied by many authors [YP86, BWBD86]. The Gaussian is therefore used in generating scale-space [Wit83], and is often used together with the gradient and Laplacian filters in edge detection [Can86, MH80].

The Gaussian filter can be implemented separably by two one-dimensional (1D) convolutions with kernels:

$$H(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad H(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \tag{4.2}$$

Kaiser [Kai87] described a fast recursive algorithm for the computation of equidistant samples of a 1D Gaussian function requiring only two multiplications per sampling point. Burt [Bur83] presented a pyramid technique to implement the Gaussian filtering efficiently in multi-resolution image processing. The technique exploit the fact that a Gaussian function can be factorized into a convolution of Gaussian functions with smaller σ. The Gaussian filtering can also be implemented efficiently using a cascade of mean filters, as proposed by Wells [Wel86]. This method relies on the central limit theorem which guarantees that under quite general conditions the repeated convolution of simple non-negative functions tends to a Gaussian function. Design of optimal Gaussian operators in small neighborhoods was discussed by Davies [Dav87].

4.1.3 Median Filtering

Median filtering is a nonlinear method used very often in impulse noise suppression. Each pixel in the input image is replaced by the median of its neighborhood. Window sizes of 3×3 and 5×5 are often used. Filters with cross- and X-shaped window were discussed by Bovik *et al.* [BHM87].

Median filter is useful for removing isolated pixels (impulse noise) while preserving spatial resolutions. Some analytical studies [YH81, BHM87] described the edge preserving property of this filter. The result of recursive application of these filters rapidly converges to a steady signal, the so-called root signal [dBC90].

In the most straightforward implementation of the median filter, all gray level values in the window are buffered, sorted, and the median is then selected. Sorting is time consuming, even when Quick sorting [Vei92] is used. A more efficient 2D running median algorithm based on histogram modification was suggested by Huang *et al.* [HYT79]. Another fast computation method, based on the manipulation of individual bits of the data, was proposed by Danielsson [Dan81].

Pratt *et al.* [PCK84] proposed a computationally simpler operator, called the pseudomedian filter, which maintains many of the properties of the median filter. There are many extensions of the median filter called median-based filters [CdB90, GHN90], possessing different characteristics for the noise smoothing.

4.1.4 Temporal Filtering

A straight and a motion-compensated temporal filtering process were presented by Huang and Hsu [HH81]. Let $g^k(x, y)$ denote the gray level of the position (x, y) in the kth frame of the image sequence. A nonrecursive straight temporal filter over $(2K + 1)$ frames is defined by

$$g^k_{out}(x, y) = F\{g^{k-K}_{in}(x, y), g^{k-K+1}_{in}(x, y), \cdots, g^k_{in}(x, y), \cdots, g^{k+K}_{in}(x, y)\} \tag{4.3}$$

where F is a filter, which can be mean, median or Gaussian. For white Gaussian noise, averaging over N frames will reduce the noise variance by a factor of N. Median filtering will reduce the variance of white noise by a factor of $2N/\pi$, and will be much more effective for impulse noise. To reduce the degrading effect, Huang and Hsu [HH81] proposed a motion-compensated temporal filtering, in which the direction of the motion has to be estimated. This method is suitable for rigid objects.

4.1.5 Quantitative Performance Measurement

Performance of the noise reduction is often assessed by using a supervised evaluation, in which a noiseless reference image is given. The root mean square error (RMS) between the filtered image and the reference image is often used as a measure for the performance. Chin and Yeh [CY83] proposed to partition the image into regions based on the spatial activity of the gray level. Given a threshold of the spatial activity, an image is partitioned into edges and homogeneous areas. The RMS values

are then measured in both two regions. This method was later used by many other authors [CdB90, dBC90, WWL92]. Two other supervised performance measures, the stability and the correct processing ratio, were applied by Wu *et al.* [WWL92].

Mean busyness is an image smoothness measure proposed by Hurt and Rosenfeld [HR87] and used in performance evaluation for noise reduction [WWL92]. This is an unsupervised performance measure since it is independent of a reference image. The busyness of each pixel is the median of the absolute vertical and horizontal intensity differences in a neighborhood. These median values are then averaged to get an overall busyness of an image, which is called the mean busyness. The execution time was also used as a performance measure [GAL88, WWL92].

4.2 Experiments and Results

Since the smoothing is used as a preprocess to the segmentation, its performance should be assessed according to the result of the segmentation. In this section, I show the segmentation results by combining different smoothing and segmentation methods. The modified Bernsen's dynamic thresholding method with an α-parameter and the Laplacian of Gaussian (LoG) edge detection were used in the testing. The two segmentation methods will be discussed in details in later chapters. Also the segmentation results presented here are not the final results, which will be obtained by further processing including a shape correction (see Chapter 7). Two smoothing filters, a mean filter[1] implemented in integer precision and a Gaussian filter implemented in 64 bit float precision, were tested. The cascaded mean was also tested.

In the modified Bernsen's thresholding, I used a 21×21 square window, a contrast parameter 15, and $\alpha = 0.65$. Applying the mean filters of different window sizes W_M, the segmentation results of the image in Figure 4.1 are given in Figure 4.2. We see that with $W_M = 3$ the boundaries of the cells are still jagged. The degree of blur was increased with the increasing of the window size.

I also applied the Gaussian filters with different standard deviations σ. The size of the Gaussian window is $2\lceil 3\sigma \rceil + 1$, where $\lceil x \rceil$ is the smallest integer larger than x. The results are given in Figure 4.3. We see that the boundaries are still jagged when $\sigma = 1.0$, and are smoothed with larger standard deviations.

The Laplacian edge detector is usually combined with the Gaussian smoothing. Such combination is called the Laplacian of Gaussian (LoG). The results of the LoG of different standard deviations (σ) are given in Figure 4.4. We can see that this method is very sensitive to the noise, so that $\sigma = 1.0$ and 2.0 give very jagged boundaries. The boundaries are more smooth with larger σs.

The mean filter was also combined with the Laplacian. The results were very noisy. The reason might be the integer precision of the mean filter output. The Gaussian filter can be approximated by cascading the mean filters [Wel86]. In this

[1]The implementation in XITE by Otto Milvang was used.

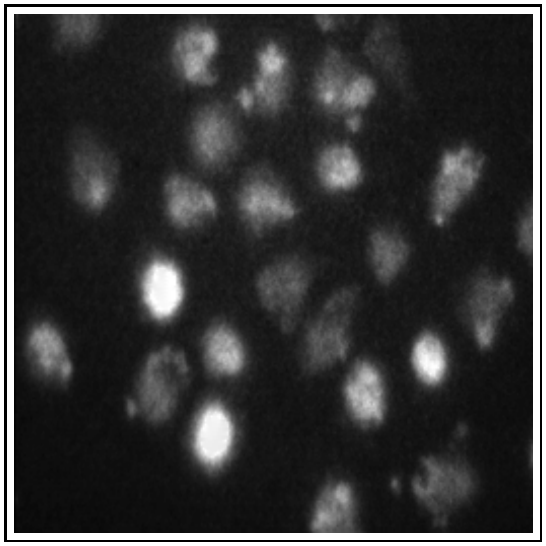


Figure 4.1: *A fluorescence cell image.*

case, the precision of the mean filtering can also be important to the results. I tested the cascaded mean filters. The results of the modified Bernsen's thresholding were very similar to those with the Gaussian filtering. But the results of the Laplacian were much more noisy than those with the Gaussian filtering. The results should be improved when float precision were used.

For the two segmentation methods, the smoothing can increase the size of the objects. This effect is especially clear in Figure 4.3 where the modified Bernsen's method and the Gaussian filtering were used.

As discussed in Chapter 3, the topographical structure of the image gray level gives useful information for image analysis. Musin [Mus93] defined the topographical structure of an image as a set of singular points and lines of the gray level surface, and demonstrated that such a structure could also be defined for a discrete space. A method for detecting the singular points and lines such as peak, pit, pass, ridge, and ravine was proposed by Peucker and Douglas [PD75]. (Johnston and Rosenfeld [JR75] developed a method for some of these features.) Multi-resolution behavior of the structural features has been studies by many authors [Koe84, GP93].

I observed the change of the topographical structure after smoothing. As an example, I show the gray level peaks in the image after the Gaussian filtering with different standard deviations. The method of Peucker and Douglas [PD75] was used to detect the peaks. The results are given in Figure 4.5. We can see that the amount of the peaks reduces with the increasing of the Gaussian blur. Similarly, the mean filter also reduced the amount of the peaks. This amount might be used as a measure of image smoothness which is useful in automatic determination of the smoothing filter parameters.

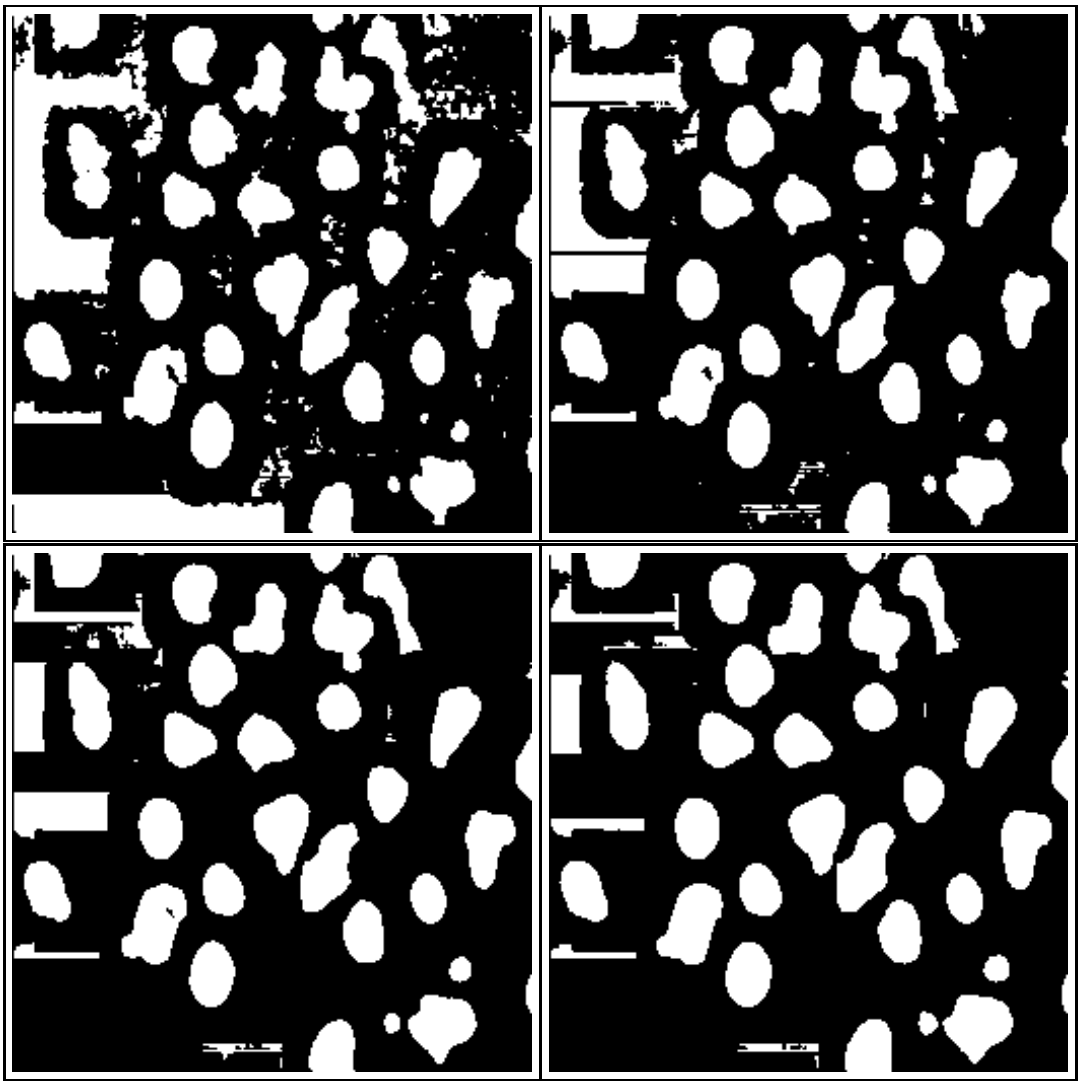


Figure 4.2: *Results of the modified Bernsen's thresholding after a mean filtering with window size* W_M. (upper-left) $W_M = 3$. (upper-right) $W_M = 5$. (lower-left) $W_M = 7$. (lower-right) $W_M = 9$.

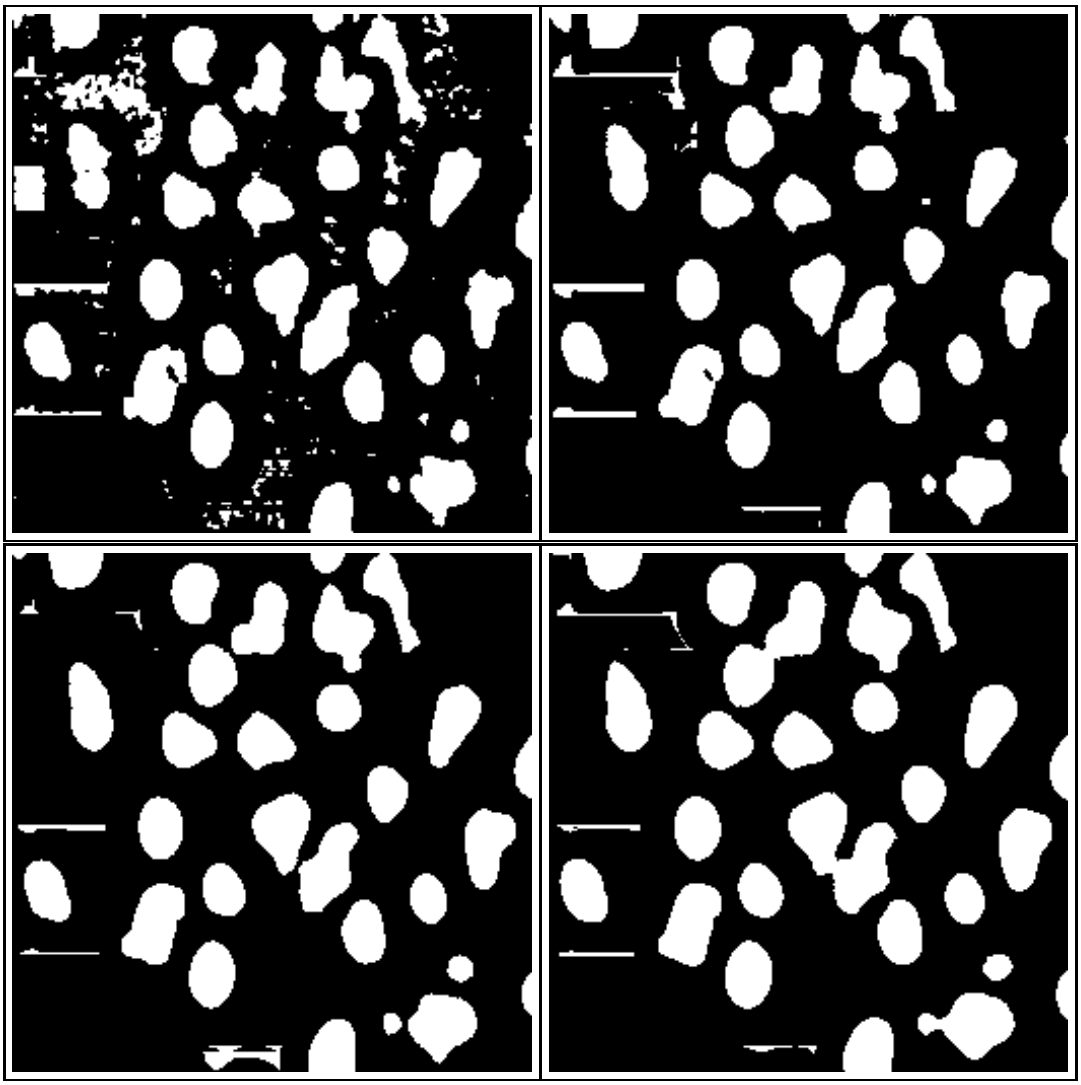


Figure 4.3: *Results of the modified Bernsen's thresholding after a Gaussian filtering with standard deviation* σ. (upper-left) $\sigma = 1.0$. (upper-right) $\sigma = 2.0$. (lower-left) $\sigma = 3.0$. (lower-right) $\sigma = 4.0$.

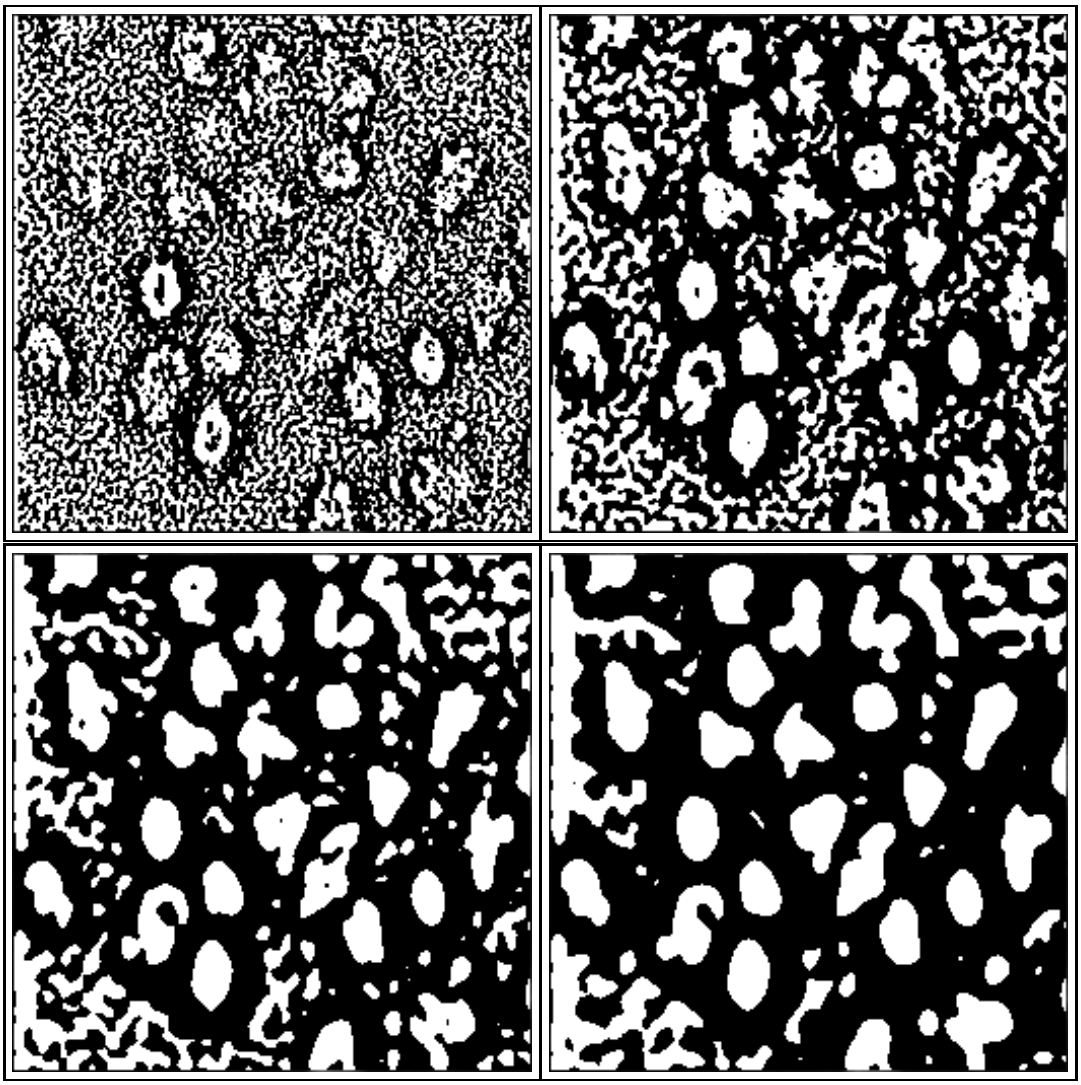


Figure 4.4: *Results of Laplacian of Gaussian. The white and black pixels are used to represent positive and negative responses, respectively.* (upper-left) $\sigma = 1.0$. (upper-right) $\sigma = 2.0$. (lower-left) $\sigma = 3.0$. (lower-right) $\sigma = 4.0$.

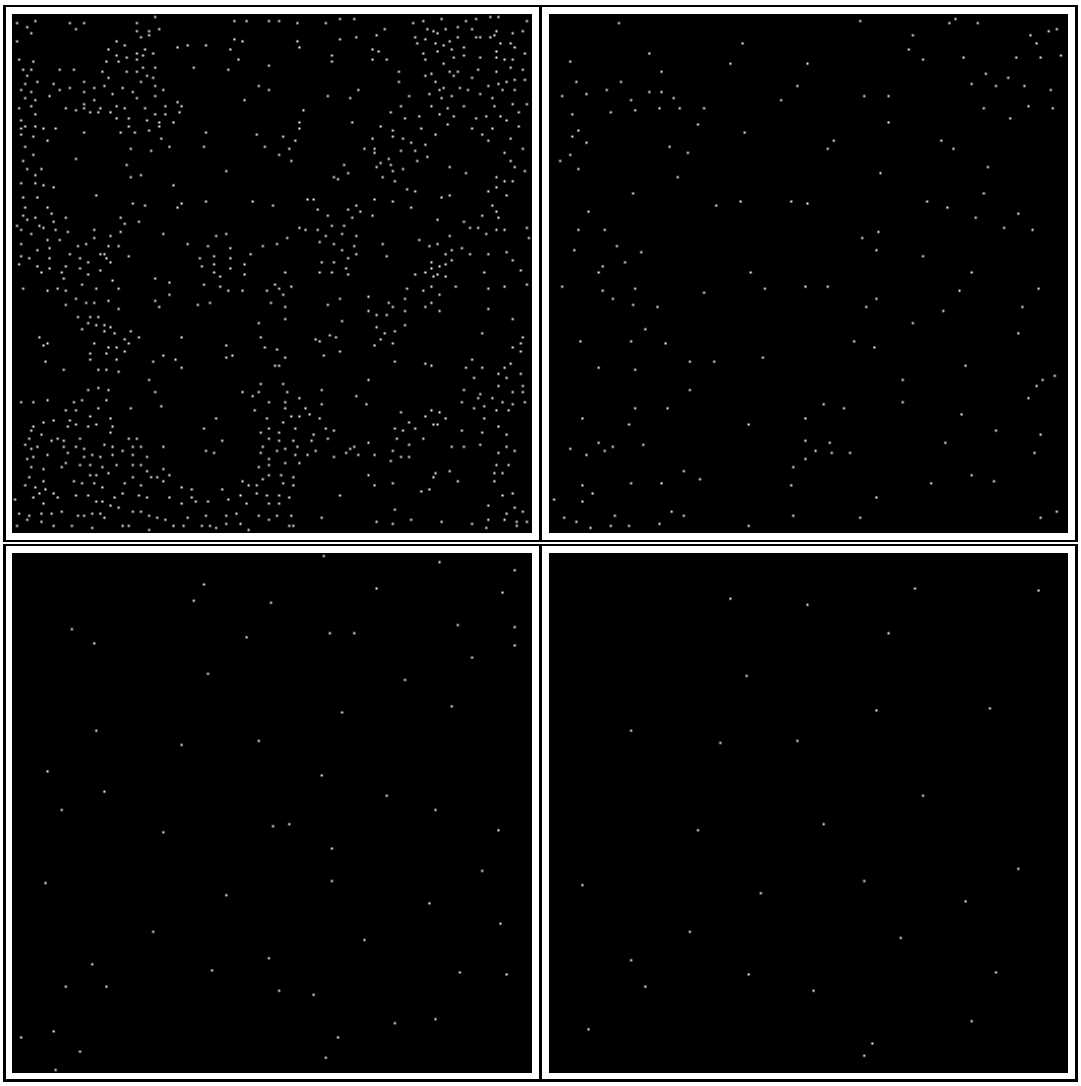


Figure 4.5: *Gray level peaks in the image after Gaussian filtering.* (upper-left) $\sigma = 1.0$. (upper-right) $\sigma = 2.0$. (lower-left) $\sigma = 3.0$. (lower-right) $\sigma = 4.0$.

4.3 Discussion and Conclusion

In this chapter, I gave a brief review of image smoothing methods, and presented the results of some of the methods for the fluorescence cell images. A quantitative evaluation will be given in Chapter 11.

The smoothing is usually used as a preprocess to the segmentation. A small degree of smoothing can give a noisy segmentation result, and a large degree of smoothing can degrade the shape. A proper smoothing method with a proper parameter is therefore important to a good segmentation result. This is, however, dependent on the segmentation method. The Laplacian method is more sensitive to the noise than the thresholding method. This method is often used with a Gaussian filter. I found that the Gaussian should be computed in high precision if the LoG is implemented in two steps. (This will be further discussed in the next chapter.) To obtain smoothed object boundaries, the LoG required $\sigma = 3.0$ or larger. For the modified Bernsen's method, a Gaussian smoothing with $\sigma = 2.0$ already gave smoothed boundaries. The Bernsen's method can also be used with the mean or the cascaded mean computed in an integer precision. In this case, the cascaded mean gave similar results as if the Gaussian were used.

There are many other smoothing methods which have not been tested. The median can possibly be used before the mean or the Gaussian filtering to reduce impluse noise. The temporal method can be combined with the spatial method to obtain a spatial-temporal smoothing. In this case, a proper length of the filter (in temporal dimension) is important.

The topographical structure of the gray level surface was used to show the effect of the smoothing. The amount of the gray level peaks reduced with the increasing of the blur, and thus might be used as a measure of image smoothness.

Chapter 5

Edge-based Segmentation

Segmentation can be obtained through detection of edges, defined as points with abrupt changes in gray level [RT71]. Edge-based segmentation assumes that the boundaries of regions are located at the edges. One of the motivations of this approach is that biological visual systems appear to make use of edge detection [RK76]. The first step in an edge-based segmentation is edge detection. Detected edge elements are combined, and regions are then determined. Many edge detection methods have been tested for the fluorescence cell images, as presented in this chapter.

5.1 A Review of Methods

Davis [Dav75] classified edge detection techniques into two categories: sequential and parallel. In the sequential technique, the decision whether a pixel is an edge pixel or not is dependent on the result of some previously examined pixels. The performance of a sequential method is therefore dependent on the choice of an appropriate starting point and how the results of previous points influence the decision on the next point. Levine [Lev85] gave a review of many sequential edge detection methods.

In the parallel method, the decision whether a point is an edge or not is based on the point under consideration and some of its neighboring points. Parallel differential operators of first and second order are often used in edge detection. If one assumes that the intensity is a continuous function over the (x, y) plane, then in the edge positions the intensity gradient magnitude is large and the Laplacian of the intensity crosses zero. Many edge detectors [Can86, MH80] are therefore based on the digital approximation of the gradient or Laplacian operation. On the other hand, Hueckel [Hue71] considered the edge detection as an approximation problem, in which a circular section of an image was matched to an ideal step edge. The matching methods based on two-dimensional moments were proposed to allow more edge accuracy and better noise immunity [RAM83, GM93]. Haralick [Har84] proposed a facet model approximating the image function by a two-dimensional polynomial, from which second order directional derivatives were computed. The facet model was comprehensively discussed by Haralick and Shapiro [HS93]. Nalwa and Binford

[NB86] proposed a surface-fitting method to detect step edges. The edge angle is first estimated by a sequential least square fitting. Then the image data along the edge direction is fit to a hyperbolic tangent function.

In the following, I present some of the edge detection methods based on differential operations.

5.1.1 Gradient Operators

Let the gradients of the intensity function in the x and y-direction be $G_x(x,y)$ and $G_y(x,y)$ respectively. The gradient magnitude is given by

$$|G(x,y)| = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \tag{5.1}$$

which can be approximated [RK76] by

$$|G(x,y)| \approx |G_x(x,y)| + |G_y(x,y)| \tag{5.2}$$

A high value of the gradient magnitude indicates the presence of an edge. The gradient direction

$$\theta(x,y) = \tan^{-1}\left[\frac{G_y(x,y)}{G_x(x,y)}\right] \tag{5.3}$$

also gives useful information for edge-based segmentation.

The gradients in the x and y-direction ($G_x(x,y)$ and $G_y(x,y)$) can be estimated by linear filtering, often done by convolving with a 3×3 impulse response mask. Such operators include the Prewitt and Sobel masks, which are described in many textbooks [RK76, Nib86, Pra91].

In noisy environment, a larger mask can be used to obtain a built-in smoothing effect. A 7×7 Prewitt-type operator called boxcar and a 7×7 truncated pyramid operator were described by Pratt [Pra91]. Based on the structure of the Sobel masks, Danielsson and Seger [DS90] developed a method to generate masks of any sizes. Generally, large size gradient operators can be considered to be compound operators in which a smoothing operation is performed on a noisy image followed by a differentiation operation. A well-known example of a compound gradient operator is the derivative of Gaussian (DroG) [Can86] in which Gaussian smoothing is applied before the differentiation. The impulse response functions of this operator in the x and y-directions are

$$H_x(x,y) = -\frac{x}{\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad H_y(x,y) = -\frac{y}{\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{5.4}$$

where σ is the standard deviation of the Gaussian smoothing. Canny [Can86] developed an analytic approach to the design of such operators, based on three criteria: good detection, good localization and single response. An efficient approximation of Canny's detector is the DroG.

Instead of computing the gradients in two orthogonal directions, one can consider many directions by convolution of an image with a set of masks, called compass gradient masks. The edge angle is determined by the direction of the largest gradient. This approach was used by Prewitt [Pre70], Robinson [Rob77] and Nevatia and Babu [NB80].

5.1.2 Laplacian Operators

In the continuous domain, the Laplacian of an image $g(x, y)$ is

$$\nabla^2 g(x,y) = \left[\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right] g(x,y) \tag{5.5}$$

In digital images, the Laplacian can also be approximated by a linear operation. The four-neighbor and eight-neighbor Laplacian masks [Pra91] are two commonly used operators. Both of them are separable. Consequently, the convolution can be computed separably along the rows and columns of an image. The zero-crossing of the Laplacian indicates the presence of an edge.

Since the Laplacian is very sensitive to noise, it is often used after a Gaussian noise filtering. The combination is referred to as Laplacian of Gaussian (LoG) [MH80]. Let σ be the standard deviation of the Gaussian filtering. The impulse response of the LoG is

$$H(x,y) = \frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{5.6}$$

It can be shown [HM86] that the LoG is also separable. The LoG is often called the *Mexican hat* filter.

A discrete domain version of the LoG can be obtained by sampling the continuous domain impulse response function over a $W_s \times W_s$ window. In order to avoid deleterious truncation effects, the size of the mask should be such that $W_s = 3c$ or greater, where $c = 2\sqrt{2}\sigma$ is the width of the positive center lobe of the LoG function [HM86]. Techniques for fast and accurate convolution with LoG masks were discussed [CHM87, SB89].

Marr and Hildrith [MH80] found that the LoG could be closely approximated by a Difference of Gaussian (DoG), and the ratio $\sigma_2/\sigma_1 = 1.6$ provided a good approximation, where σ_1 and σ_2 were the standard deviations of the two Gaussian filters.

5.1.3 Edge Element Selection

After a gradient operation, one obtains the gradient magnitude and direction for each pixel. The edge elements are usually selected by a thresholding and a thinning process. The thresholding is applied on the gradient magnitude in order to select

pixels with high edge strength. More about the thresholding methods will be discussed in Chapter 6. Thresholded edges are then thinned. Methods for thinning binary patterns were comprehensively surveyed by Lam *et al.* [LLS92]. Thinning can also applied on gray level images [DR79].

After a Laplacian operation, one obtains a positive or a negative response for each pixel. The zero-crossings between the positive and the negative responses are the edge points [MH80]. The Laplacian tends to give a lot of small edges representing details. Before the Laplacian operation, it is common to apply a Gaussian smoothing, which however reduces the accuracy of edge location. One can apply a thresholding on the gradient magnitude to select strong edges from the zero-crossings. However, the thresholding operation will normally introduce some parameters.

Colchester *et al.* [CRK90] detected the edge elements using maximum gradient profiles (MGP). An MGP is formed by linking the pixels in their gradient directions. Let the intensity be a function of the pixel positions along the MGP. The intensity function is monotonic. The maxima and minima of the first order derivative of the intensity function can be used as edge elements. In the method of Griffin *et al.* [GCR92], the MGPs are used to connect singular points, and a segmentation is then based on the partition of the image by the MGPs.

5.1.4 Edge Element Linking

The detected edge elements often need to be linked to form a closed boundary of the object. Many early methods were reviewed by Fu and Mui [FM81]. Niblack [Nib86] suggested that neighboring edge elements were first linked together to form some edge segments, these edge segments were then linked together to form the contour of the object. The decision were based on the gradient magnitudes and the angles of the edge elements. Tavakoli and Rosenfeld [TR82] proposed a method to link edge segments based on the geometric configuration of the segments and the gray levels associated with them. Hayden *et al.* [HGP87] used temporal information to link edges in time-varying imagery. By summing up image frames after a position correction, a new image is created that tends to have connected edges. The Hough transform [DH72, OC76, IK88] can be used to find parametric curves from a set of points. It performs specially well in detecting lines and circles, and has also been applied to detect ellipses [YIK89].

An alternative to the edge linking is the local thresholding method of Yanowitz and Bruckstein [YB89], which determines a thresholding surface based on the gray levels of the edge points. More about this method is given in Chapter 6.

5.1.5 Scale Space and Edge Focusing

A smoothing process is often required as a preprocess to the edge detection. The basic conflict encountered in edge detection is to eliminate the noise without distorting the shape of the edges. By smoothing a gray level image, we remove the noise and unnecessary details, but at the cost of poorer localization. The degree of smoothing

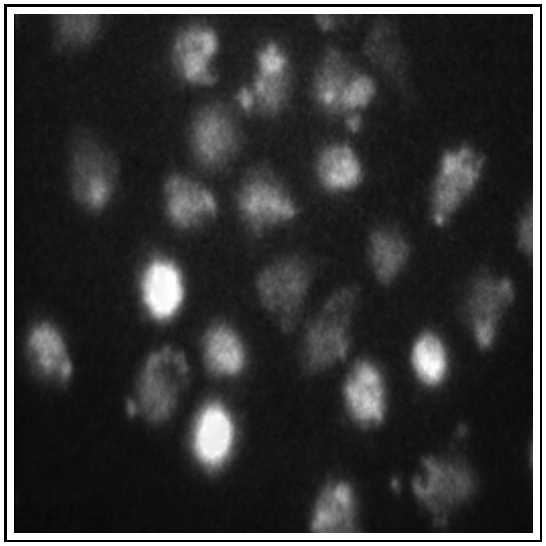


Figure 5.1: *A fluorescence cell image.*

is essentially determined by the size of the smoothing filter. The standard deviation of a Gaussian kernel is usually used as a scale parameter, denoting the degree of smoothing. Witkin [Wit83] proposed to use the scale space as a description of what happens with a gray level image during the change of the scale parameter. It can be shown [Koe84, YP86] that the Gaussian is the only filter which prohibits the generation of new features at a coarse level of resolution that do not exist at a finer level.

To combine a high edge accuracy and a good noise reduction, Bergholm [Ber87] proposed a method called edge focusing. First apply edge detection on an image of a large scale obtained by using a Gaussian smoothing with a large standard deviation. Then reduce the scale, and apply edge detection in the regions where there are edges detected in the previous scale. The old edge points are thrown away, and the edge focusing goes on until the Gaussian blurring is quite weak.

5.2 Experiments and Results

Edge detection methods were applied to segment the fluorescence cell images. In this section, I show the results for the test image in Figure 5.1.

After a Gaussian smoothing with standard deviation $\sigma = 3.0$, the Sobel edge operator[1] gave gradient magnitude of the test image, shown in Figure 5.2(left). We can see that the operator gave weak response for the vertical and horizontal edges. This might be due to the approximation by Equation (5.2). The problem should be solved by using Equation (5.1). It is not easy to find a good global threshold of the gradient magnitude for the determination of edge elements. The dynamic thresholding method of Bernsen [Ber86] gave a relatively good result shown

[1]The implementation in XITE by Otto Milvang was used.

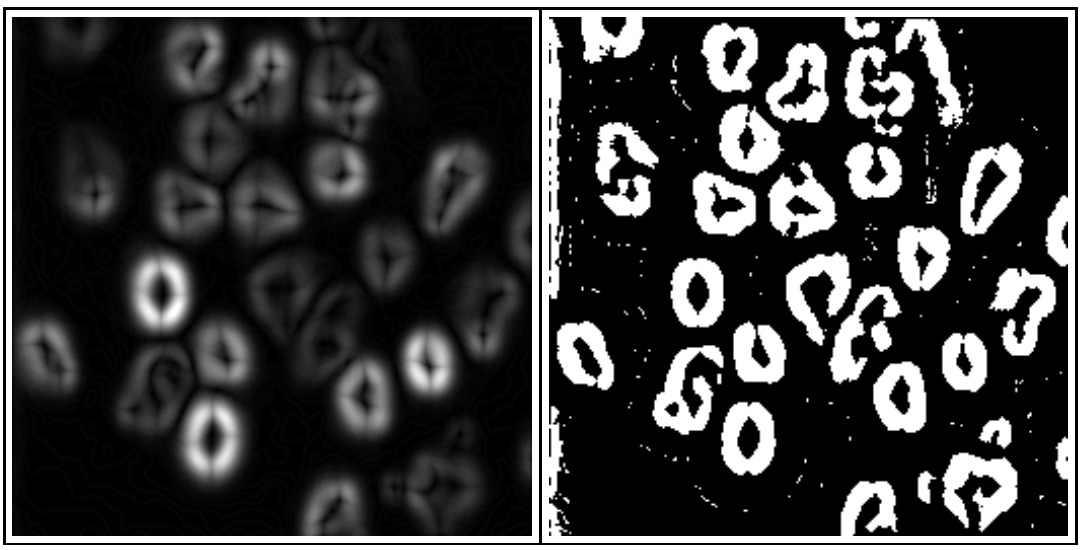


Figure 5.2: (left) *Sobel gradient magnitude of the test image in Figure 5.1.* (right) *The high gradient pixels obtained by a thresholding using Bernsen's method.*

in Figure 5.2(right). The thresholded edge objects have to be thinned and linked to give a final segmentation.

Instead of applying a thresholding after the Sobel operation, I used the MGP method of Colchester *et al.* [CRK90] to finish the segmentation. I implemented the Colchester method in a pixel classification manner. For each pixel, if its next pixel along the MGP in the gradient direction has a lower gradient magnitude, then the pixel is classified as the background, otherwise the pixel is classified as the object. The result is shown in Figure 5.3. Note that ghost figures were obtained inside the cells where the consistency of the gradient directions was low.

I also tested the LoG method, implemented by using a four-neighbor Laplacian after a Gaussian filtering. Given standard deviation $\sigma = 3.0$, the result is shown in Figure 5.4(left) where the white pixels represent the positive responses and the black ones represent the negative responses. (By using 64 bit float numbers, there is no zero response for this test image.) The zero-crossings can then be detected as the boundaries between the positive and the negative regions, as shown in Figure 5.4(right). Other σ values were also tested. The results were given in Figure 4.4 in the previous chapter where the parameter of the smoothing process was discussed.

Obviously, the results of the MGP and the LoG method have to be further processed (corrected) to obtain a final segmentation, since there are holes and gaps in the cell objects. Also, I found that the two methods gave different edge positions, although the same smoothing method was applied. Generally the MGP method gave smaller objects than the LoG method. Method to quantitatively evaluate the segmentation results will be presented in Chapter 11.

I implemented the LoG in two steps, applying Gaussian and then Laplacian. (Note that the LoG can be implemented in one step by using Equation (5.6).) When the two-step approach is used, the output of the Gaussian filter should have

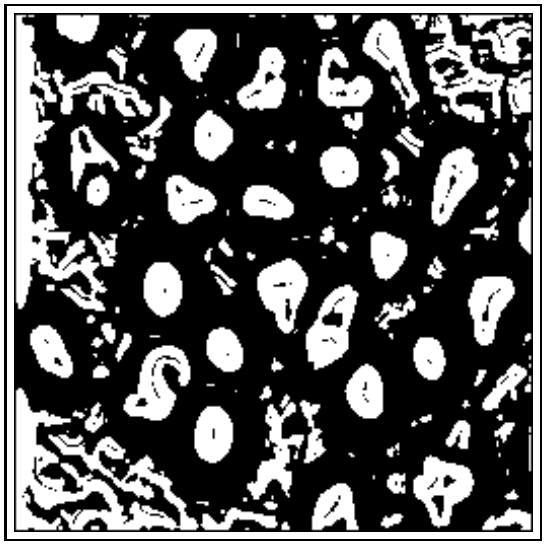


Figure 5.3: *The result of a MGP algorithm.*

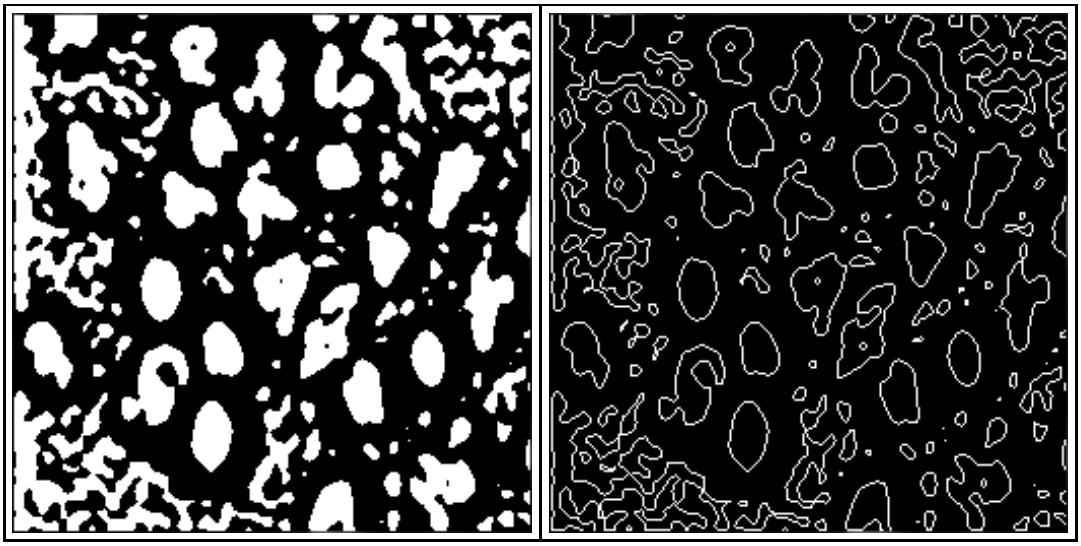


Figure 5.4: *Results of Laplacian of Gaussian.* $\sigma = 3.0$. (left) *Positive (white) and negative (black) response regions.* (right) *Zero-crossings.*

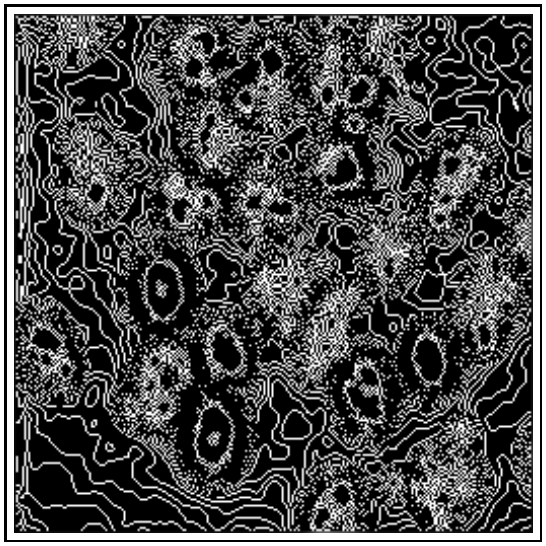


Figure 5.5: *Result of the LoG when the integer precision is used for the output of the Gaussian filtering.*

a good precision. I used 64 bit float numbers for the Gaussian output. Usually, one pixel in a gray level image is stored in one byte, representing an integer value from 0 to 255. Using the byte pixel for the output of the Gaussian filtering, however, I obtained a very noisy result shown in Figure 5.5. Note that this is the result one would get if the XITE package is used as it is at present.

5.3 Discussion and Conclusion

In this chapter, I gave a review of edge-based methods for image segmentation. The edges can often be detected by gradient or Laplacian operators. The gradient-based MGP method and the Laplacian of Gaussian method were tested. Their results have to be further processed to obtain a final segmentation (see Chapter 7) since there are holes, gaps and thin gulfs in the cell objects. Different edge detectors may give different edge positions since they are different approximations to the two-dimensional derivatives. Visual examination is hard to evaluate these two methods. A quantitative evaluation will be given in Chapter 11.

A smoothing is usually required as a preprocess. The Gaussian filter prohibits the generation of new features, and is therefore used to generate the scale space. Torre and Poggio [TP86] have shown that numerical differentiation of images is an ill-posed problem because its solution does not depend continuously on the data. They have found that the Gaussian filter has strong regularizing properties which can solve this problem. In fact, the Gaussian is associated with many edge detection methods such as LoG, DoG and GroG. I have shown that when the LoG is implemented in two steps (Gaussian and then Laplacian), the precision of the Gaussian output is important to the result. The DoG is an approximation of the LoG. In this method,

float precision of the Gaussian output should also give a better result than integer precision, since many zero responses are avoided.

I have discussed the methods for edge element selection and edge linking. Both processes will involve some parameters and can be complex. However, these two processes can be avoided in some edge detection methods by using a pixel classification. In the MGP method, I proposed to classify each pixel according to the gradient magnitude of the next pixel. In the LoG method, each pixel can be classified as a positive or negative response pixel. The pixel classification simplifies the process of the segmentation.

Chapter 6

Gray Level Thresholding

Thresholding is often used in image segmentation due to its simplicity and efficiency. Quite a number of thresholding methods have been proposed over the years, and surveyed by several authors [Wes78, SSWC88, Alb93, TT93]. To segment an image into regions of two classes, a threshold $T(x, y)$ can be computed for each spatial position (x, y). The segmentation is then done by letting

$$g_{out}(x, y) = \begin{cases} 0 & \text{if } g_{in}(x, y) < T(x, y) \\ 1 & \text{otherwise} \end{cases}$$

where g_{in} is a gray level input image and g_{out} is a binary output images.

Automatic thresholding is to select the threshold value $T(x, y)$ without any human intervention. In this chapter, I give a review of automatic thresholding techniques and discuss their applications in the segmentation of the fluorescence cell images. A new method [YALG94] is presented, giving a better robustness under the change of object size.

6.1 A Review of Methods

6.1.1 Global Thresholding

In global thresholding, one threshold T is applied for the entire image. This approach is suitable for images where the classes are well separated. That means the overlapping between the distributions of the background and the object pixel values should be small. The amount of overlapping thus determines the minimal classification error for a global thresholding.

Global thresholding methods can be divided into parametric and non-parametric methods. The parametric thresholding is essentially a special case of a supervised classification [DH73, Nib86] where each pixel has a scalar value rather than a feature vector. In parametric thresholding, the parameters (means, standard deviations and a priori probabilities of the two distributions) should be either known, or extracted from the a posteriori (observed) mixture of the two distributions. Usually they

are first estimated by a supervised learning process, and then updated under the classification process [TSM85, JB94].

The non-parametric methods are more frequently used than the parametric methods. In non-parametric thresholding, one separates the two classes in an optimum manner according to some a posteriori criteria, without estimating the parameters of the two distributions. Usually an iterative or a sequential search method is used to find the optimal threshold T. A survey and evaluation of the methods was given by Albregtsen [Alb93]. Non-parametric methods of Ridler and Calvard [RC78] minimizes the classification error by assuming a bimodal distribution where the variances and the a priori probabilities of the two classes are equal. The method of Otsu [Ots79] maximizes a between-class variance, assuming equal intra-class variances. The minimum error method of Kittler and Illingworth [KI86] selects the threshold based on a criterion function, continuously updating estimates of the means and standard deviations. This method works well for images with low object-to-background ratios [Alb93]. Brink [Bri89] proposed a method based on maximizing the correlation between the original gray level image and the thresholded image. The method of Kapur *et al.* [KSW85] maximizes a global entropy. Tsai [Tsa85] used the preservation of moments to obtain a threshold value without iteration or search. The threshold selection can also be done through an analysis of the convex deficiency of the histogram [Wha91]. Sasakawa *et al.* [SKI91] proposed a method which maximizes a mean adjacent pixel number. In contrast to other methods, this method does not use the histogram of the gray levels. The method is meant to find a smooth boundary, and is suitable for images containing small objects with noisy and low-contrast background. Recently, Pal and Bhandari [PB93] proposed some new global thresholding techniques, including two algorithms based on a new conditional entropy measure and a minimum error method based on the Poisson distribution.

6.1.2 Local Thresholding

In some images, the background intensity is variable, and the overlapping between the two classes is therefore large. In such case, the threshold T should be determined locally. In local thresholding, the thresholds are first determined in some selected spatial positions. These thresholds are then interpolated to form a thresholding surface applied thereafter to the entire image. Local thresholds can be evaluated from bimodal subimages or in high gradient positions.

Chow and Kaneko [CK72] proposed a simple local thresholding method in which an image is divided into non-overlapping windows. Thresholds are selected for those windows that have bimodal histograms. These thresholds are then interpolated to define a thresholding surface. Nakagawa and Rosenfeld [NR79] tested this method and provided an extension which allowed histograms to be either bimodal or trimodal.

The method of Taxt *et al.* [TFJ89] first estimates the global distribution parameters for the two classes, by a supervised training or a clustering. The image is then divided into non-overlapping windows. For each window, the parameters

are updated using an expectation-maximization (EM) algorithm [TSM85], and the pixels are classified using the quadratic Bayes classifier. This method was improved by Eikvil *et al.* [ETM91] using two windows. The pixels inside a small window are thresholded on the basis of clustering of the pixels inside a larger concentric window. This avoids the abrupt change of threshold value at the window borders. In the method of Eikvil *et al.*, Otsu's method [Ots79] is used to divide the pixels into two clusters.

The method of Yanowitz and Bruckstein [YB89] assumes that the boundary of an object is located at the edge. An edge detection is first applied. For each edge point, the threshold is the gray level of that point. A thresholding surface is constructed by an iterative interpolation scheme. A similar method based on an edge criterion was proposed by Medina-Rodriguez *et al.* [MFD92].

6.1.3 Dynamic Thresholding

A dynamic thresholding method determines a threshold value for each spatial position in the image, according to some local features. The threshold $T(x,y)$ is often computed as a linear combination of the feature values

$$T(x,y) = \mathbf{w}^T \mathbf{f}(x,y) \tag{6.1}$$

where $\mathbf{f}(x,y)$ is a vector of local features, and $\mathbf{w}$ are the weights.

The dynamic thresholding method of Bernsen [Ber86] uses the local grey level minimum and maximum as the features. The threshold T is evaluated locally as

$$T(x,y) = 0.5 \min_{(i,j)\in W(x,y)}[g(i,j)] + 0.5 \max_{(i,j)\in W(x,y)}[g(i,j)] \tag{6.2}$$

where $\min_{(i,j)\in W(x,y)}[g(i,j)]$ and $\max_{(i,j)\in W(x,y)}[g(i,j)]$ are local grey level minimum and maximum inside a window W of fixed size, centered at (x,y). Bernsen proposed to consider the local contrast, defined as

$$C(x,y) = \max_{(i,j)\in W(x,y)}[g(i,j)] - \min_{(i,j)\in W(x,y)}[g(i,j)] \tag{6.3}$$

The thresholding given by equation (6.2) is applied when the local contrast is higher than a certain threshold T_C. Otherwise the pixel at the position (x,y) is classified by another method, such as using a default or interpolated threshold. In Bernsen's method, it is classified as the background. The contrast threshold T_C can be considered as a scale parameter which eliminates details with small contrast. In fact these details are often artifacts.

Bernsen used simply 0.5 as the weights for both of the two features. This may not be an optimal choice for all types of images. More generally a parameter $\alpha \in (0,1)$ can be used and the threshold is then estimated as [YALG94]

$$T(x,y) = \alpha \min_{(i,j)\in W(x,y)}[g(i,j)] + (1-\alpha) \max_{(i,j)\in W(x,y)}[g(i,j)] \tag{6.4}$$

where α may be estimated from training data. This extension will be referred to the modified Bernsen's method in this thesis.

White and Rohrer [WR83] proposed to use a biased local mean as a feature. Niblack [Nib86] used a local mean, together with a local standard deviation as a bias. The threshold at pixel (x, y) is calculated as

$$T(x, y) = \mu(x, y) + k\sigma(x, y) \tag{6.5}$$

where $\mu(x, y)$ and $\sigma(x, y)$ are the local mean and standard deviation respectively, and k is a parameter.

In Equation (6.1), I give the general form of a linear dynamic thresholding, in which a thresholding using N local features is to find an $N - 1$ dimensional hyperplane which partitions the N dimensional feature space. Classical methods such as Bernsen's method and the method of White and Roher use a small amount of features. When many features are used together, it can be difficult to find a set of optimal weights by trial. Supervised learning methods can then be used to determine the weights, when a training set is provided. One can approximate the thresholding process by a logistic function and then use a logistic regression [HL89], or solve the problem by a backpropagation neural network [HKP91].

6.1.4 A Validation Step

Due to uneven illumination and noise, the thresholding surface might intersect background or object regions, yielding *ghost figures* [YB89] in the segmented image. As a method to eliminate the ghost figures, Yanowitz and Bruckstein [YB89] proposed a validation step, which could be conveniently incorporated into other thresholding methods. The process was formalized by Trier and Taxt [TT93]. For each object, the average gradient magnitude along the boundary is computed. If it does not exceed a certain threshold, the process eliminates it. The problem is then to find a gradient magnitude threshold. A solution to this problem has not been found by Trier and Taxt [TT93].

6.2 A New Dynamic Thresholding Method

Dynamic thresholding is an important segmentation method applied to images with an uneven background. Previous methods can fail to give precise segmentation due to the use of a window with fixed shape and size. Furthermore the results are sensitive to the choice of window. To overcome this problem, a new dynamic threshold selection method [YALG94] was proposed by using a connected region of consistent gradient direction (CRCGD) instead of a fixed window. The CRCGD has been used together with the thresholding method of Bernsen. Note that it can also be incorporated with other dynamic thresholding methods as well.

In Bernsen's method, a correct estimation of the local features $\min_W[g]$ and $\max_W[g]$ plays a very important role in getting a precise segmentation. Like many

authors [Koe84, GCR92, GP93] previously did, we consider an image as a 3D grey level landscape. The local features should be evaluated according to the topographical structure of the image landscape. If one assumes that the image has a dark background and light objects, then the boundaries of the objects are located in the hillsides. In a hillside region, the local minimum and local maximum should be estimated according to the height of the local valley and ridge, respectively. If the intensity changes smoothly in a high gradient hillside region, then the gradient direction in the hillside region always points from the object towards the background. Thus the value of $\min_W[g]$ and $\max_W[g]$ should be estimated within a connected region with similar gradient directions. The advantage of using a connected consistent region is that the algorithm adapts to the geometric scale and shape of the local hillside, and thus we can exclude irrelevant information which might be included if we were using a fixed window.

For each pixel p, a connected region of consistent gradient direction (CRCGD) is defined as a region in which

1. p is included

2. the region is connected (using 4 or 8-connection)

3. the gradient direction difference between p and any other pixel is less than θ.

A CRCGD can be bounded or unbounded. For a given pixel p, an unbounded CRCGD includes all the pixels which meet the above criteria, and a bounded CRCGD includes pixels which meet the criteria, and which are within a given distance from p. If the maximum absolute value (chess) distance metric is used, this is equivalent to bounding the CRCGD by a square window centered at p.

An unbounded CRCGD can be constructed for each spatial position (x, y) as follows. The pixel at position (x, y) is first included in the region. The gradient direction ϕ of the pixel is evaluated. The gradient directions of all its 8-neighbors are examined. The neighbors with gradient directions similar to ϕ are then included. Here the similarity of gradient direction is defined as a difference less than θ. The algorithm is applied recursively on all the included points to include more pixels with gradient directions similar to ϕ, until there are no more pixels to be included. A bounded CRCGD can be constructed in a similar way.

Using the CRCGD with Bernsen's method, local features $\min_W[g]$ and $\max_W[g]$ are computed as the minimal and maximal intensity values inside the connected consistent region instead of a window. In the bounded CRCGD method, although a window centered at (x, y) is used so that the construction of the region is done inside the window, the local features are not computed from all the pixels within the window. Thus, the result is not as sensitive to the size of the window as in Bernsen's method.

The contrast parameter T_C defined by Bernsen can also be used. This parameter is used to eliminate the ghost figures and does not affect the position of boundaries. When C is less than T_C, many different methods can be applied to determine the

local threshold, or classify the pixel directly. One way is to use an interpolated threshold surface. When the threshold surface is very rough, a smoothing can be applied. In the experiment detailed in the next section, we used a very simple interpolation method. More sophisticated interpolation methods were proposed by Yanowitz and Bruckstein [YB89].

A disadvantage of the new method is its computational ineffectiveness as compared to Bernsen's method, in which the local minimum and maximum can be updated when the window is moved from one position to the next.

6.3 Experiments and Results

Some thresholding methods were tested for the segmentation of the fluorescence cell images. In this section, I show the results for the test image in Figure 6.1.

I first applied the global thresholding. The result of Otsu's method[1] is shown in Figure 6.2(upper-left), and that of the method of Kittler and Illingworth[2] is shown in Figure 6.2(upper-right). Otsu's method obviously did not give a good result since many cells were shrunken. The cell objects produced by the method of Kittler and Illingworth are however much larger than in the manual segmentation results. We can also see that the two global methods give considerably different results. I also applied manual thresholding and found that no single threshold value could give good results for all cells. This means that the global thresholding method is not suitable to the cell images. The reason might be an uneven background, and that the intensity values of object pixels are distributed over a long range, as discussed in Chapter 3.

The local thresholding method of Eikvil *et al.*[3] was tested. The result is shown in Figure 6.2(lower-left). The size of the large window was 29 and the small window was 9. Ghost figures were eliminated by examining the distance (gray level difference) of the two estimated cluster means in each larger window. The thresholding was applied when the distance was larger than 15, otherwise all the pixels inside the small window were assigned to the class with the closest updated mean value [ETM91].

The result of the dynamic thresholding method of Niblack is given in Figure 6.2(lower-right). A window size 21 was used, and the parameter k was given 0.1. In order to eliminate the ghost figures, the pixels with local standard deviation less than 10.0 were classified as background.

In the results of both methods, we can see that there are holes, gaps and thin gulfs in the cell objects. Therefore the results have to be further processed (corrected) in order to obtain a final segmentation (see Chapter 7).

When applying Bernsen's method, I used $T_C = 15$ so that most of the ghost figures were removed and the cells could appear. A very simple interpolation method was used to determine the threshold of a pixel where the local contrast C was less

[1]The implementation in XITE by Tor Lønnestad was used.
[2]The implementation in XITE by Olav Borgli was used.
[3]The implementation in XITE by Øivind Due Trier was used.

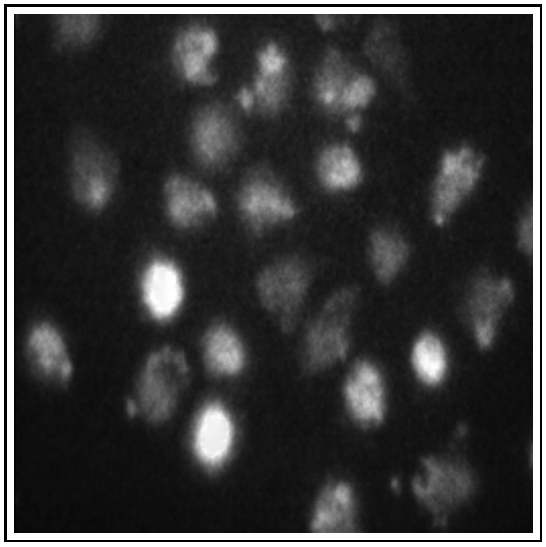


Figure 6.1: *A grey level fluorescence image.*

than T_C. I simply used the threshold of the last processed pixel as the threshold of this pixel. The interpolation resulted in some horizontal bar-shaped artifacts, but did not affect the positions of the boundaries. I modified Bernsen's method by setting $\alpha = 0.65$ according to the manual segmentation results. A square window of size $Ws \times Ws$ was used. It seems that $Ws = 21$ is a good choice giving a result shown in Figure 6.3(upper-left). However, the result is sensitive to the window size. I show the results of $Ws = 31$, 41 and 51 in Figure 6.3(upper-right), (lower-left) and (lower-right). We see that a larger window will shrink the object.

The method of CRCGD is a solution to this problem. When applying the CRCGD method incorporated into Bernsen's method, I again used $T_C = 15$, $\alpha = 0.65$ and the same interpolation method when C is less than T_C. The result of unbounded CRCGD method is shown in Figure 6.4(upper-left). As we can see, the boundary is very coarse. To solve this problem I smoothed the thresholding surface by a 7×7 mean filter. Applying the smoothed thresholding surface, I obtained the result shown in Figure 6.4(upper-right). In order to smooth the thresholding surface, however, a new parameter is introduced. As an alternative to smoothing the thresholding surface, one can smooth the boundaries of the objects directly in the binary image by morphological operations or mode filtering (see Chapter 7).

The CRCGD method introduce a new parameter θ. However, this parameter is not sensitive to the image contents. (The window size of Bernsen's method is.) In the unbounded CRCGD method, I chose θ to be $\pi/8$. This value can be applied to the resampled images of which the sizes are much larger or smaller than the original one. But, in Bernsen's method, the window size must be changed when the image is resampled.

When we consider θ as a parameter, Bernsen's method becomes a special case of our method using a bounded CRCGD, in which $\theta = \pi$. It can be shown that as θ increases, the result becomes more and more dependent on the window, and

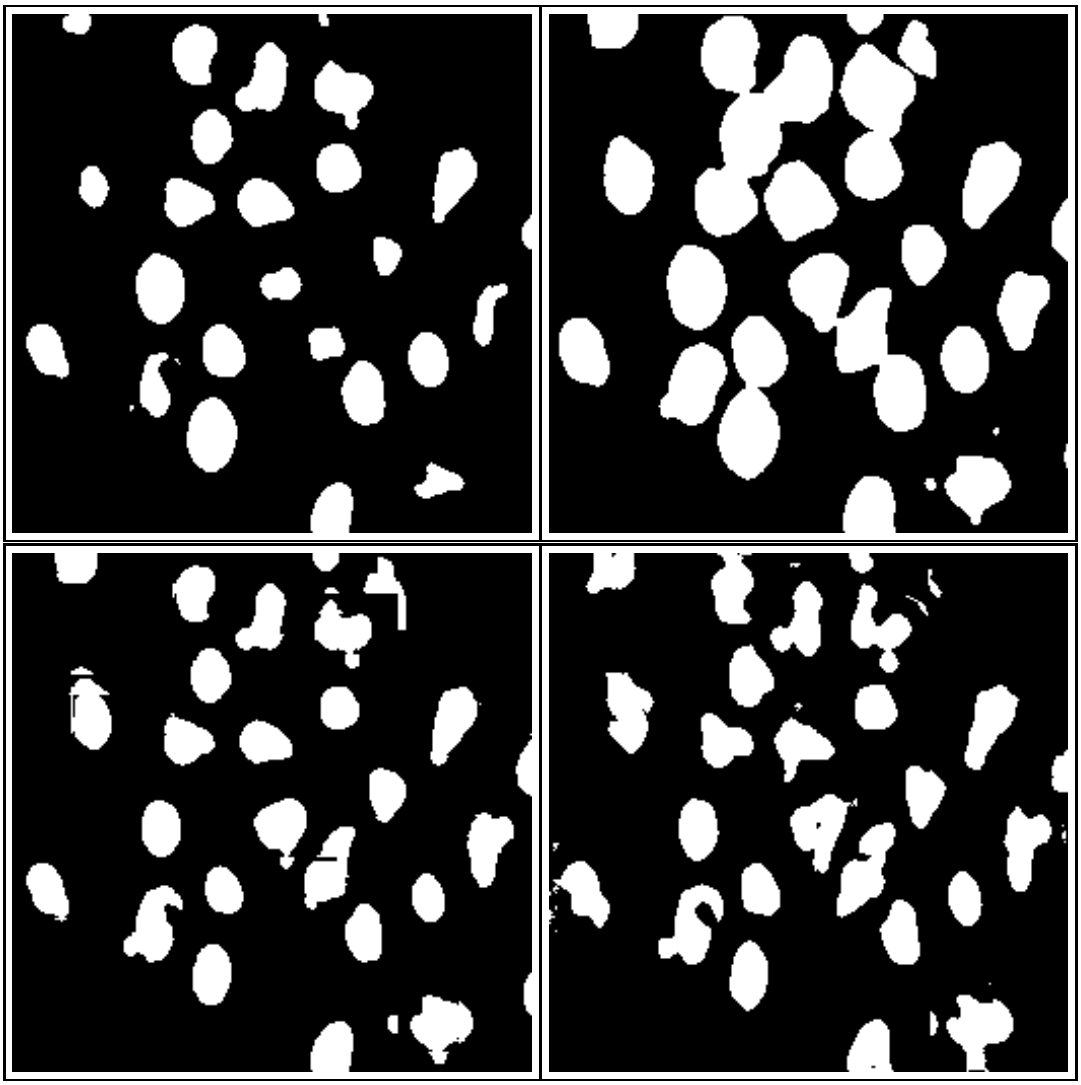


Figure 6.2: (upper-left) *Result of the method of Otsu.* (upper-right) *Result of the method of Kittler and Illingworth.* (lower-left) *Result of the method of Eikvil* et al. (lower-right) *Result of the method of Niblack.*

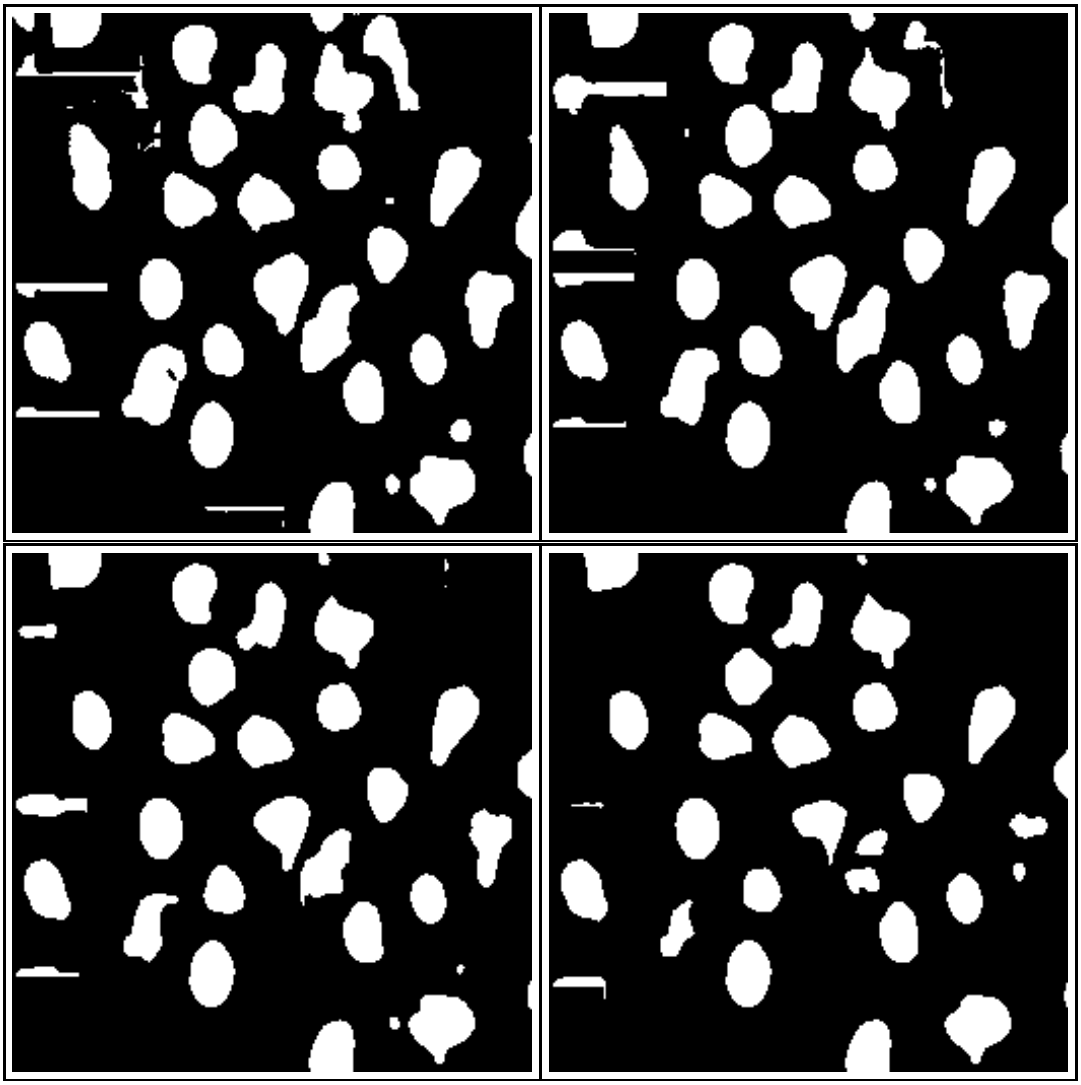


Figure 6.3: *Results of the Bernsen's method using square windows of different sizes.* (upper-left) $Ws = 21$. (upper-right) $Ws = 31$. (lower-left) $Ws = 41$. (lower-right) $Ws = 51$.

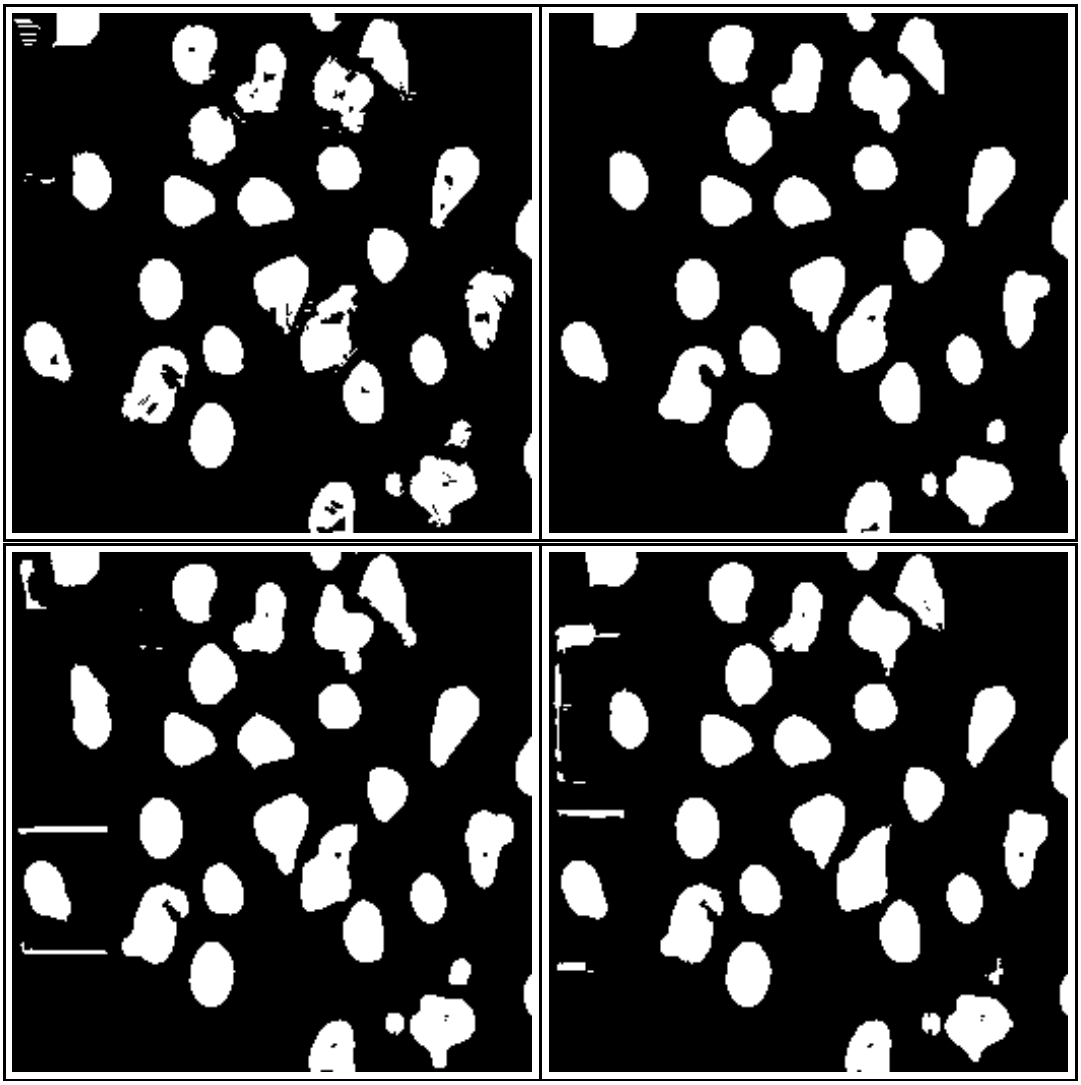


Figure 6.4: *Results of the CRCGD method.* (upper-left) *Unbounded CRCGD with* $\theta = \pi/8$. (upper-right) *Unbounded CRCGD with* $\theta = \pi/8$. *The thresholding surface is smoothed.* (lower-left) *Bounded CRCGD with* $\theta = \pi/3$ *and* $Ws = 21$. (lower-right) *Bounded CRCGD with* $\theta = \pi/3$ *and* $Ws = 41$.

the threshold surface becomes more and more smooth. In Figure 6.4(lower-left) and (lower-right), I show the results when $\theta = \pi/3$ and the window sizes are 21 and 41. Here, the threshold surfaces are not smoothed. We can see that the boundaries of the objects are not so coarse when $\theta = \pi/3$. Compared with the results of Bernsen's method in Figure 6.3, the results of the new method are obviously more stable under changing window size.

6.4 Discussion and Conclusion

In this chapter, I gave a review of gray level thresholding methods, including global, local and dynamic thresholding, and presented the test results of some of the methods.

Thresholding is often an efficient and effective method for image segmentation. Many global thresholding methods have been proposed. These methods apply a single threshold on the entire image, and therefore require that the two intensity classes are well separated. Unfortunately, the cell images do not meet this requirement, and no global threshold can give good results for all cells. Some local and dynamic thresholding methods work for these images. However, a shape correction process should be applied to fill the holes and gaps in the cell objects. This will be discussed in the next chapter. A quantitative evaluation of the segmentation results will be given in Chapter 11.

In dynamic thresholding, some local features are computed from a window (often with fixed size and shape). The threshold is then obtained as a linear combination of the feature values. The result can be sensitive to the choice of the window. To overcome this problem, a new method, using a connected region of consistent gradient direction (CRCGD) instead of a window, is proposed. I incorporate the CRCGD into Bernsen's method, and find that the new method is more robust under changing object size. Using a bounded CRCGD, the previous method of Bernsen becomes a special case of the new method.

Chapter 7

A 3D Segmentation System

In the preceding chapters, we discussed various basic techniques of image segmentation, including edge-based methods and gray level thresholding. In this chapter, I describe a spatial-temporal segmentation system (STSS) for the fluorescence cell image sequences.

Segmentation is to divide an image into regions representing the objects. A cell object in an image sequence occupies a three-dimensional (3D) spatial-temporal region. The STSS is supposed to find such a region for each of the cell objects.

The STSS first applies an initial segmentation, in which each pixel is labeled by either 1 or 0 indicating whether it belongs to a cell or the background. Techniques described in the preceding chapters can be used in the initial segmentation. After the initial segmentation, some two-dimensional (2D) regions can be identified according to the connectivity of the pixels (4- or 8-connection). However, the initial segmentation does not finish the job of segmentation, since there are three problems which remain:

1. Two or more cells may appear in one connected 2D region. The problem is referred to as touching, since it happens when two cells are touching or closely located.

2. There may be holes, gaps and thin gulfs in a segmented cell.

3. 3D regions are not formed since the temporal linking between the 2D spatial regions is not established.

The first two problems can be easily observed from the segmentation results in the preceding chapters. The processes described in this chapter is to solve the three problems.

The STSS is a semi-automatic system, requiring a manual segmentation in the first frame. Given a segmented cell object in the previous frame and the result of the initial segmentation of the current frame, the STSS applies a cell tracking process, which establishes a correspondence between the previous and the current frame, and then applies a region closing process, filling the gaps and gulfs in the cell

objects. Results of the STSS will be shown in this chapter and will be quantitatively evaluated in Chapter 11.

7.1 Method

7.1.1 Morphological Operations

Mathematical morphology [Ser82, GD88] provides an approach to the processing of digital images based on shape. Morphological operations tend to simplify image data preserving their essential shape characteristics and eliminating irrelevances [HSZ87].

Dilation and erosion are two basic morphological operations. There are many different ways to define the dilation and the erosion operations for binary images [HSZ87, Pra91]. Let $g(x, y)$ be a binary image where the pixel value is either 1 or 0, and $H(x, y)$ be a structuring element which is also an array of 1s and 0s. The definition of the dilation operation $g_{out} = g_{in} \oplus H$ given by Pratt [Pra91] is:

$$g_{out}(x, y) = \bigvee_m \bigvee_n g_{in}(m, n) H(x - m + 1, y - n + 1) \tag{7.1}$$

and that of the erosion operation $g_{out} = g_{in} \ominus H$ is

$$g_{out}(x, y) = \bigwedge_m \bigwedge_n g_{in}(m, n) \bar{H}(x - m + 1, y - n + 1) \tag{7.2}$$

where $\bigvee$ denotes a sequence of binary OR operations, $\bigwedge$ is a sequence of binary AND operations, and $\bar{H}$ is the complement of H obtained by applying binary NOT operations to all the points in H. A structuring element H usually contains a small object, often a disk or another simple shape. The size of the object determines the degree of the dilation or the erosion. Objects in a binary image will be thickened after a dilation and shrunken after an erosion by a disk structuring element. From Equations (7.1) and (7.2), we see that the two operations can be implemented as convolution operations.

Dilations and erosions are often used in pairs to obtain morphological opening and closing. The opening of an image g by a structuring element H, denoted by $g \circ H$, is defined as [Pra91]

$$g \circ H = (g \ominus H) \oplus H \tag{7.3}$$

The closing of an image g by H, denoted by $g \bullet H$, is defined as [Pra91]

$$g \bullet H = (g \oplus H) \ominus H \tag{7.4}$$

Opening by a disk structuring element smoothes the boundary, breaks narrow parts, and eliminates small objects. Closing by a disk structuring element smoothes the boundary, fills narrow bays, and eliminates small holes.

Morphological dilation, erosion, opening and closing were also defined for gray level images [HSZ87]. A set of morphological operations were used as gray level smoothing filters with preservation of crucial image structures [SG91]. Some algorithmic techniques of mathematical morphology were presented by Vincent [Vin91].

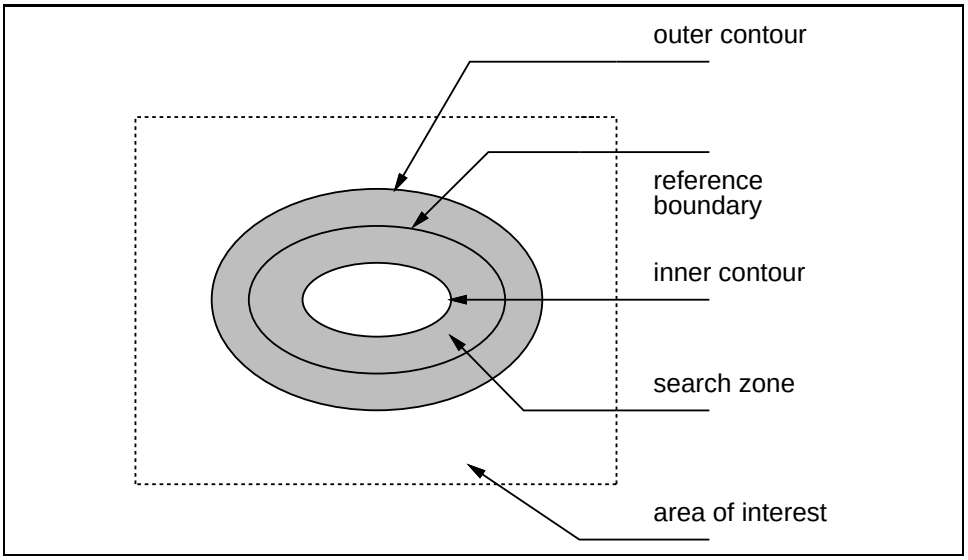


Figure 7.1: *The geometry of a search zone.*

7.1.2 Cell Tracking

In the segmentation of time-varying images, one has to establish temporal correspondence between features in a sequence of image frames. Three major techniques used to determine the correspondence are pixel-based, token-based and object-based methods. In pixel-based method [Jai81], the motion of each pixel is estimated. Tsai and Huang [TH84] used this method to estimate motion parameters of rigid objects, whose shape did not change over time. In token-based method [ADM81, SJ84, DOSA91], a set of tokens is derived from the boundary of the object. The tracking of tokens results in a flow or a trajectory field which can be used to study the motion of the object. Waks and Tretiak [WT90] used an object-based method, in which the correspondence between the objects in a sequence of images was established. Since the cell motion is very complex, I will use an object-based method for cell tracking.

Waks and Tretiak [WT90] presented a method to track an object using a search zone, assuming that the boundary changed in a small amount from frame to frame. The search zone predicts where the boundary of the object is most likely to exist. The geometry of the search zone is illustrated in Figure 7.1. The reference boundary is the boundary of the object in the previous frame. The outer contour and the inner contour are obtained by applying morphological dilation and erosion operations on the object in the previous frame. The search zone is the region between the outer and inner contours. The boundary of the object in the current frame is then searched for in the ring-like search zone. The authors also suggested to locate an area-of-interest, so that the operations to find the search zone are done inside the area-of-interest.

I propose an object-based cell tracking operation based on the following properties of the cell image sequences: The motion of a cell is not large from one frame to the next. The boundary of a cell is located in a hillside where the intensity in

the cell is larger than that in the background. After smoothing, the intensity of a cell tends to have peaks in the center of the cell. In this cell tracking operation, I use a search zone which is similar to the one of Waks and Tretiak [WT90], but is improved so that it also solves the problem of touching cells.

A simple structuring element

$$H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

is used for the morphological operations. Let a boundary point be a pixel in an object with a least one 4-connected background neighbor. By using this structuring element, the dilation operation can be implemented by adding all the 4-connected background neighbors to the object, and the erosion can be implemented by removing all the boundary points from the object.

When two cells are touching or closely located, there should be a valley in the intensity surface between the two cells. In order to solve the touching problem, I define a conditional dilation, in which a neighbor point is added if its intensity is lower than the boundary point, and not added otherwise. I also define a conditional erosion, in which a boundary point is removed if there is at least one 4-connect background neighbor whose intensity is higher than the boundary point, otherwise the point is not removed.

Given the segmented cell object in the previous frame and the initial segmentation result of the current frame, the cell tracking process labels the pixels of the cell object in the current frame. This process is stated as follows:

1. Locate a rectangular area-of-interest whose boundaries are either the boundaries of the image, or $n + 1$ pixels away from the closest point in the object in the previous frame. The following operations are then done inside the area-of-interest.

2. Apply n iterations of erosion on the object in the previous frame and obtain a region bounded by the inner contour. Label all the pixels inside the region as object pixels.

3. Apply the conditional erosions on the object in the previous frame iteratively until there is no change in the image, using the intensity values from a smoothed image.

4. Apply n iterations of the conditional dilation on the result of step 3 and obtain a region bounded by the outer contour, using the intensity values from a smoothed image. The region between the outer and inner contour is the search zone.

5. Remove all the isolated single object pixels inside the search zone. The remaining object pixels are labeled.

There is a parameter n which determines the size of the search zone. We know that the displacement of the cell boundary $D_p \leq 3$ (see Chapter 3). So the value of n should be at least 3. Since the conditional erosion and the conditional dilation are used, the search zone will not grow farther when its boundary reaches an intensity valley, although a large n is used. This is the reason why the touching problem can be solved by using this search zone. For the same reason, the process should be robust for a large n. An illustration of the search zone will be given in Section 7.2.

7.1.3 Region Closing

After the cell tracking process, some pixels are labeled to be a particular cell. The labeled pixels form one or more regions called the labeled regions. If there are more than one such region, the regions have to be linked. There can also be thin gulfs in a labeled region which need to be filled. A region closing operation is therefore required.

One possible approach to the region closing is to compute the convex hull of the set of the labeled pixels [Kim82]. However, the cell objects are not always convex. Another approach is to use the morphological closing operations. But it is not easy to determine the suitable size of the structuring element. If the structuring element is too small, then the gaps will remain unclosed. A large structuring element leads to a large degradation of the cell shape. A better approach might be the active contour [KWT88, Coh91, WS92], which is an elastic deformable model. The model is placed on the image and is subject to the action of external or internal forces which move and deform it from its initial position to best fit it to the desired features in the image. Recently a group of Japanese researchers proposed an alternative called active nets [YKA93, Bro94]. The advantage of the active nets compared to the active contour is the large number of internal nodes which stabilizes the nets. By using the active contour or the active nets, gray level information can be utilized in the region closing process.

I propose a conceptually simple method to deal with the region closing problem of the cell images. This method is easy to explain for analogue (continuous) objects. From each point in the boundary, we can draw a normal line, perpendicular to the boundary, towards the background. If the boundary of the object is smooth, the normal lines will cover all the background. If there are two separate objects, then some normal lines from one object will intersect the other one (See Figure 7.2). If an object has holes or gulfs, then the normal lines will intersect the object itself. The region closing can be done by letting the intersecting normal lines be a part of the object. The closed region will not be a convex hull, but any thin gulfs will be removed.

To implement this method reliably for digital objects can be complex. I use a very simple approach where a normal line is an 8-directional line of connected pixels. The direction of the normal line depends on the background-to-object transitions of the boundary pixel. Consider a boundary pixel as a square. A background-to-object transition is one side of the square which divides the object and the background.

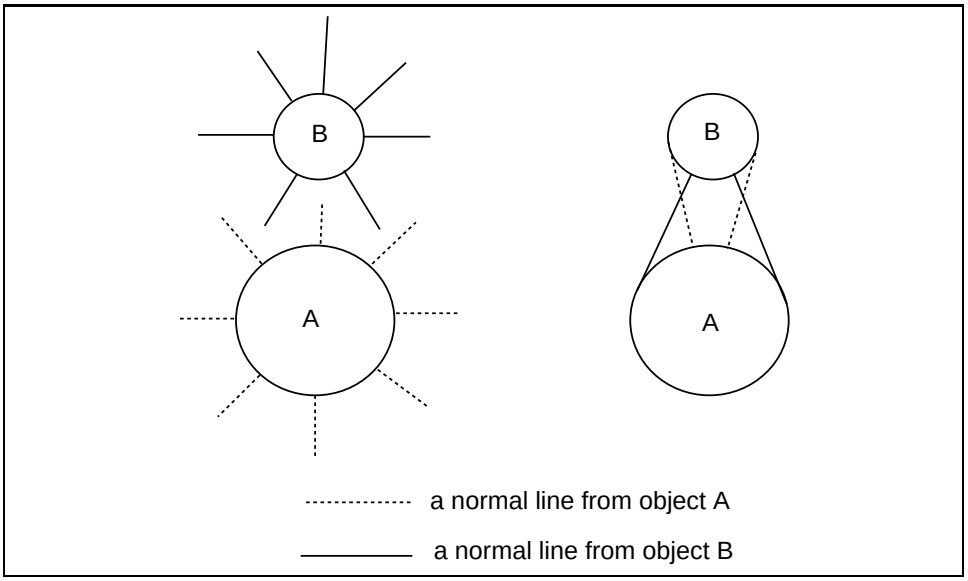


Figure 7.2: (left) *Two separate objects and their normal lines.* (right) *Some normal lines intersect other objects.*

The transitions can be detected by a contour following algorithm (see Chapter 9). The number of the transitions of each boundary pixel can be up to three since the isolated points are removed. As illustrated in Figure 7.3, a boundary pixel can have one or two main normal lines, and some additional normal lines. The main normal lines are drawn first. An additional normal line is drawn if and only if its direction is between those of the two neighboring main normal lines along the transitions. A normal line starts from a boundary pixel and stops at another boundary pixel or at a pixel in the boundary of the area-of-interest. The contour following is applied two times to draw all the normal lines. In the first time the main normal lines are drawn, and in the second time the additional normal lines. The pixels in a normal line are labeled as the object if the normal line stops at a boundary points. Otherwise the pixels are not labeled. After the pixels in the normal lines are labeled, holes in the labeled regions are filled. The region closing process is then finished. To illustrate the effect of the region closing, I show an example in Figure 7.4.

This simple digital realization of the normal line method works well for the cell images since the gaps to be filled is not large. To deal with more complex problems one will have to consider the normal lines of arbitrary directions. Leymarie and Levine [LL89] proposed a method to smooth the discrete boundary orientations by a Gaussian filtering. Methods to draw a digital line of arbitrary direction can be found in literature [FvDFH90].

7.1.4 Boundary Smoothing

After the region closing, the boundary of the objects can still be jagged. Methods to smooth the boundary of a binary region include mathematical morphological

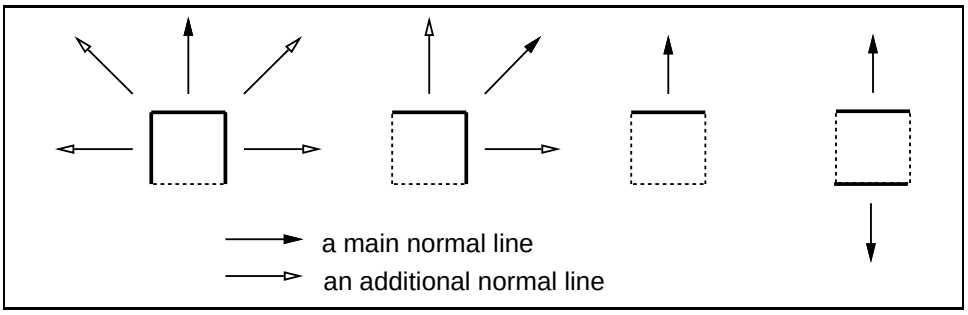


Figure 7.3: *The directions of the normal lines. The squares represent boundary pixels. The solid line sides of the squares represent the background-to-object transitions.*

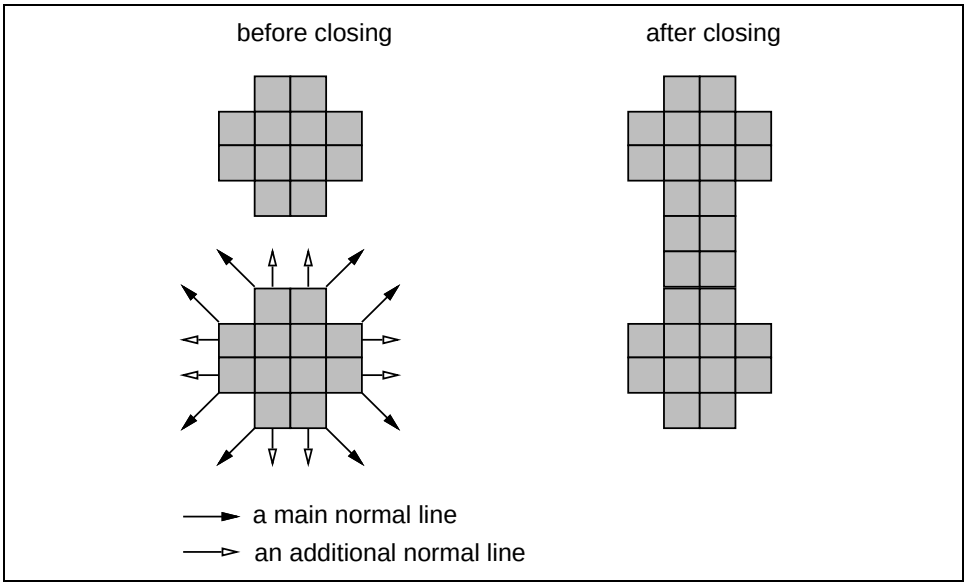


Figure 7.4: *An example of the region closing process.*

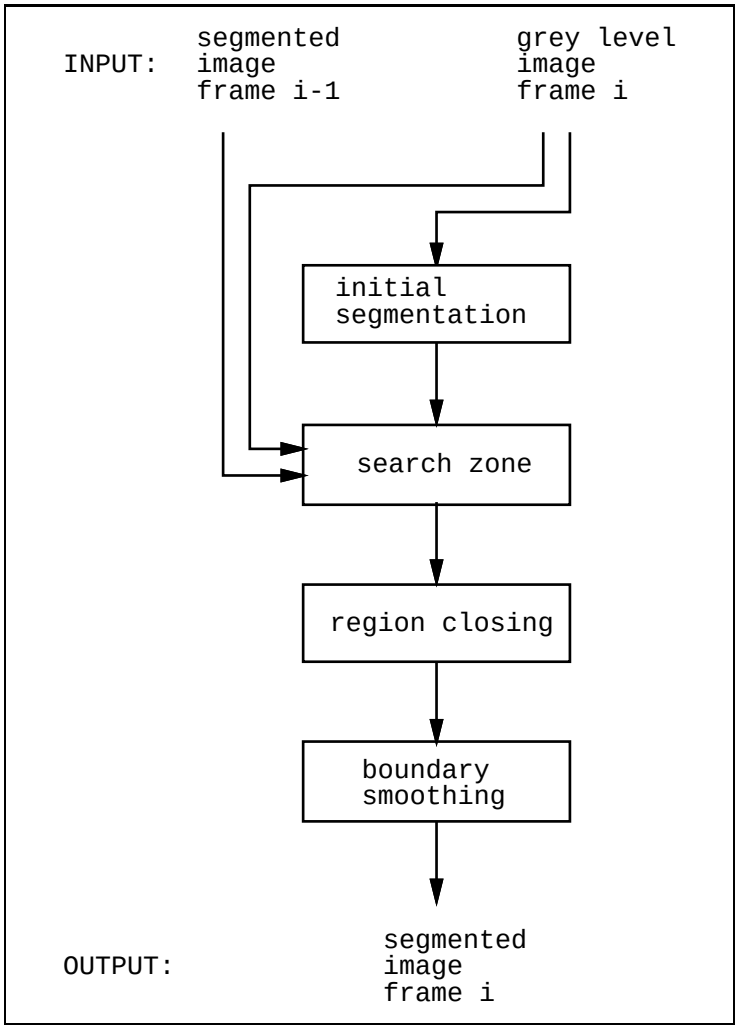


Figure 7.5: *STSS: A temporal-spatial segmentation system for the fluorescence cell image sequences.*

operations, median filtering and mode filtering. The mode filtering is the simplest method in which a pixel is replaced by its most common neighbor inside a window [Nib86]. Its effect is the same as the median in binary images. The shape and the size of the window are the parameters of this approach. The result of recursively applying of a mode filter rapidly converges to a steady signal.

7.1.5 STSS: a Spatial-temporal Segmentation System

The STSS segmentation system designed for the fluorescence cell image sequences is illustrated in Figure 7.5. I implemented the system in two passes, i.e., two sequential visit to an image sequence. The initial segmentation is done in the first pass, and the other processes are done in the second pass. The initial segmentation takes a gray level image as input, and produces a binary result where each pixel is classified

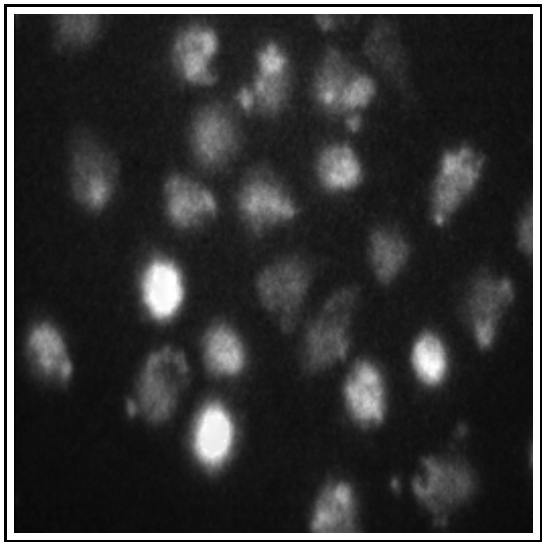


Figure 7.6: *A gray level fluorescence cell image.*

as either the cell or the background. Methods described in the preceding chapters can be used in the initial segmentation. Then a search zone is determined for each cell according to its boundary in the previous frame and the gray level image in the current frame. The boundary in the current frame is then searched for inside the search zone. The search zone is able to solve the touching problem since gray level information is considered. The search zone process has five steps as described in Section 7.1.2. A region closing process is then applied to fill the holes and gaps, and to link the separate parts of a cell. A normal line approach is used to the region closing. Finally, a mode filtering is applied to smooth the boundary of the cell objects.

7.2 Experiments and Results

The STSS system was tested for the three image sequences we had. In this section, I show some results of the testing. Quantitative evaluation of the system will be given in Chapter 11 where the shape features are used in the evaluation.

In the initial segmentation, I used the LoG method with $\sigma = 3.0$ and the modified Bernsen's method with a Gaussian filtering of $\sigma = 2.0$. The modified Bernsen's method used a contrast parameter $T_C = 15$, a 21×21 square window, and a parameter $\alpha = 0.65$. As discussed in preceding chapters, the results of these methods were good according to visual examination. The modified Bernsen's method could also be used with the CRCGD, giving more robust results. By using these initial segmentation methods, the STSS system successively did the cell tracking and region closing for all the cells of which we had manual segmentation, and all other cells having reasonable good data. According to the testing, I found that the system was not sensitive to the parameter n in the search zone process, provided the value of

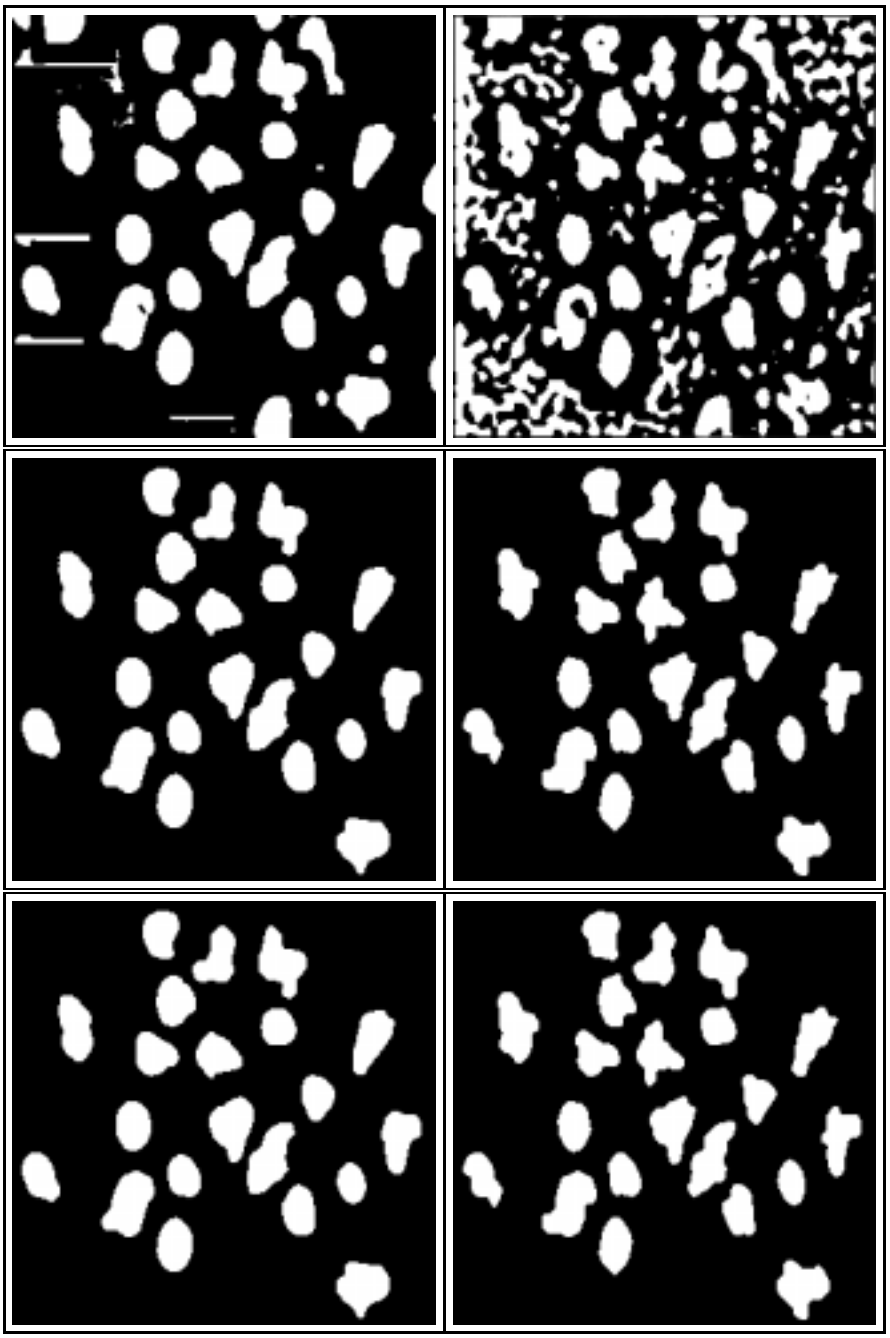
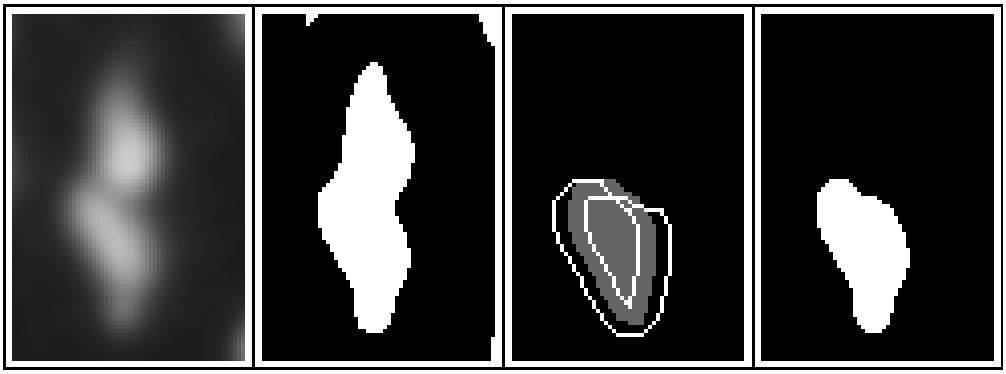


Figure 7.7: *Results of the STSS segmentation system.* (left) *The modified Bernsen's method is used in initial segmentation.* (right) *The LoG method is used in initial segmentation.* (top) *Results after initial segmentation.* (middle) *Results after region closing.* (bottom) *Results after boundary smoothing by a* 3×3 *mode filter.*



(a) (b) (c) (d)

Figure 7.8: *Result of the STSS segmentation system dealing with the touching problem.* (a) *A gray level image after a smoothing.* (b) *After initial segmentation using the modified Bernsen's method.* (c) *Segmentation result in the previous frame (gray) and the search zone (white boundary).* (d) *The segmentation result of one of the cells.*

the parameter was large enough. According to the maximum displacement of the cell boundary (see Chapter 3), $n = 4$ should be large enough. This parameter determines the size of the search zone. However, the search zone will not grow farther when its boundary reaches an intensity valley. I also found that a 3×3 mode filtering worked well for the boundary smoothing. Iteration of this filter did not further blur a boundary which had already been smoothed, However, a larger window size would give a larger degree of blur.

For the test image in Figure 7.6, the segmentation results are shown in Figure 7.7. To illustrate the results of this system dealing with the touching problem, I give an example in Figure 7.8. In these examples, a search zone of $n = 4$ and a 3×3 mode filter were used.

7.3 Discussion and Conclusion

In this chapter, I presented a two-pass spatial-temporal segmentation system (STSS) for the fluorescence cell image sequence. The first pass deals with each 2D image frame separately, and classifies the pixels into cell and background. Temporal linking is established in the second pass. Two main processes in the second pass are the cell tracking and the region closing. Some results are presented in this chapter. Further evaluation will be given in Chapter 11.

Many practical problems in image analysis require the segmentation of time-varying image sequences. An important step in image sequence segmentation is to establish correspondence between image frames. To track a motion object one has to make some assumptions about the motion. Researchers have studied rigid objects [TH84] and non-rigid objects where some tokens can be defined and tracked [DOSA91]. I assumed [Rø94] that the displacement of the cell boundary is small

between two successive frames, and thus used a search zone to track the cell object. The search zone is established according to the gray level structure of the image so that the problem of touching cells is also solved.

I also presented many techniques to correct the shape of an object. The methods include morphological operations, mode filtering, median filtering, and active contour/active nets. The morphological operations are used in this system to form the search zone. In the region closing process, I propose a simple and effective shape correction method called a normal line method, which fills the gaps in the cell object. Mode filter is used to smooth the boundary of a cell in the final step.

Chapter 8

Shape Features and Motion Analysis

The motion of an object refers to the change of the geometric properties including the spatial structure, size, position, and orientation. The geometric properties of an object can be quantified by a set of shape features $\mathbf{F} = [F_1\ F_2\ \cdots\ F_n]^T$, such as area, perimeter, circularity, centroid and so on. The motion can thus be quantified by a temporal sequence of the shape features $\mathbf{F}(i)$.

Although a blood cell is a three-dimensional object, the motion of a two-dimensional (2D) silhouette of the cell will be studied. In this chapter, I will first give a review of 2D shape features used in description tasks, and then present some techniques used in analyzing the motion of the white blood cells.

8.1 Shape Features

"Shape is a concept which is widely understood yet difficult to define" [Mar89]. In our daily language, shape often refers to the geometric properties of an object which are invariant to the change of size, position and orientation. However, a shape feature, which is a parameter to measure a geometric property of an object, does not need to be invariant. A number of shape features can be found in the literature [LNY83, NL86, Mar89, Jai89], where they are used in shape representation and shape and motion analysis. In this section, I limit the discussion on shape features to planar regions which are connected and without holes.

8.1.1 Some Simple Features

Here I list some simple shape features of a region R in a 2D (x, y)-plane:

1. The **area** A of a region R is given by

$$A = \int\int_R dxdy \tag{8.1}$$

2. The **perimeter** P of a region R is the length of the bounding contour of R.

3. The **circularity**, also known as **compactness** or **roundness**, is defined as

$$C = \frac{4\pi A}{P^2} \tag{8.2}$$

The area and the perimeter depend on the size of the object. But the circularity is a scale invariant shape feature. It is also rotation invariant. A circle has a circularity of unity. The circularities of all other shapes are less than one.

4. The **centroid** $(\bar{x}, \bar{y})$ is the center of mass of a region R. Assume that the region is homogeneous (binary). The centroid is given by

$$\bar{x} = \frac{1}{A}\int_R x dx dy, \qquad \bar{y} = \frac{1}{A}\int_R y dx dy \tag{8.3}$$

5. The **length** l is the distance between the two farthest points on the boundary.

6. The **width** w is the maximum extension of the object normal to the length.

7. The **elongation** EL of a region can be measured in a variety of ways. One technique is to use the ratio of the length l to the width w. It can also be measured as the ratio of the length to the width of the bounding rectangle (see Section 8.1.4), or the ratio of the lengths of the semimajor and semiminor axes of the image ellipse (see Section 8.1.4). Sometimes the ratio of the maximum and minimum radius (R_{max}/R_{min}) is used as a measure of the elongation [Jai89]. R_{max} and R_{min} are the maximum and minimum distances from the centroid to the boundary.

The estimation of area and perimeter from a discrete object will be discussed in Chapter 9.

8.1.2 Bending Energy

The tangential direction $\theta(t)$ of a curve can be considered as a function of path length t. The **curvature** at t, denoted as $\kappa(t)$, is defined as [Jai89]

$$\kappa(t) = \frac{d\theta(t)}{dt} \tag{8.4}$$

We have

$$|\kappa(t)|^2 = \left(\frac{d^2y}{dt^2}\right)^2 + \left(\frac{d^2x}{dt^2}\right)^2 \tag{8.5}$$

The curve is convex when the curvature is positive and concave when the curvature is negative, assuming that we traverse the contour counterclockwise.

The bending energy of a contour [YWB74] is defined conceptually as the minimal physical energy required to bend a rod to the shape represented by the given contour.

It corresponds to the square of the curvature summed over the length of the contour. Normalization of contour length may be applied to detect shapes at different scales. Let P be the length of the contour. The **normalized bending energy** E of the contour is

$$E = \frac{1}{P}\int_0^P |\kappa(t)|^2 dt \tag{8.6}$$

Accurate estimation of the curvature in digital images is not a simple task. Leymarie and Levine [LL89] introduced an estimation method based on the 8-connected Freeman chain code representation of the contour. (See Chapter 9 for the Freeman chain coding.) The chain codes are first multiplied by 45 degrees to obtain a discrete orientation representation, which is then smoothed by a Gaussian filter. The curvature is then computed as the differences of the smoothed orientations. Many other curvature estimation methods were surveyed in a recent paper by Worring and Smeulders [WS93].

8.1.3 Polygon-based Features

A closed discrete boundary can be approximated with arbitrary accuracy by a polygon. The approximation is exact when the number of line segments in the polygon is equal to the number of points in the boundary so that each pair of adjacent points defines a line segment of the polygon. Techniques for polygon approximation were discussed by Levine [Lev85] and Vokáč [Vok92]. One approach relies on the determination of critical points which then define a polygon. Freeman [Fre78] discussed algorithms for computing critical points based on the discontinuity of the curvature and some other criteria. Another approach uses an iterative approximation. The typical algorithm due to Ramer [Ram72] subdivides a segment of the boundary successively into two parts until a given criterion is satisfied. A stop criterion might be that the maximum perpendicular distance from a boundary segment to the line joining its two end points is less than a given threshold. If not, the farthest point in the boundary segment becomes a new vertex, thus the initial boundary segment is subdivided into two segments.

The following shape features are based on a polygon approximation:

1. The **number of polygon sides** N_p.

2. The **number of reflex angles** N_r is the number of vertices which have an internal angle greater than 180.

3. The **angle regularity** AR is a measurement of the sum of the differences between the angles of a given polygon and a regular one (with equal angles) having the same number of sides. It can be used instead of the circularity as a shape complexity measure [NL86]. Levine [Lev85] computed the angle regularity as

$$AR = \frac{1}{K}\sum_{k=1}^{N_p}\left|\theta_k - \frac{360}{N_p}\right| \tag{8.7}$$

where

$$K = \begin{cases} 360(N_p + 2)/N_p & \text{for } N_p \text{ even} \\ 360(N_p + 1)/N_p & \text{for } N_p \text{ odd} \end{cases}$$

and θ_k is the kth internal angle of the polygon.

4. The **side regularity** SR is a measurement of the sum of the differences between the lengths of the sides of a given polygon and a regular one having the same number of sides. Levine [Lev85] computed the side regularity as

$$SR = \left[\frac{N_p}{2(N_p - 2)P}\right] \left[\sum_{k=1}^{N_p} (l_k - L)^2\right]^{1/2} \tag{8.8}$$

where P is the total length of the polygon sides, l_k is the length of the kth side, and L is the length of side of the regular polygon with the same perimeter.

8.1.4 Moment-based Features

Theory of the Moments

Cartesian geometric moments (for short moments) have been widely used in shape analysis and pattern recognition [Hu62, TC88, PR92, GW92]. The $(p + q)$'th order moment of a function $g(x, y)$ is defined as

$$m_{pq} = \int_{y=-\infty}^{\infty} \int_{x=-\infty}^{\infty} g(x, y) x^p y^q dx dy \tag{8.9}$$

The infinite set of the moments $(p, q = 0, 1, 2, ...)$ uniquely determines $g(x, y)$ and vice-versa [Hu62].

The double integral in Equation (8.9) is often replaced by a double summation in discrete images

$$m_{pq} = \sum_y \sum_x g(x, y) x^p y^q \tag{8.10}$$

In binary images, assuming the object has pixel value 1 and the background has pixel value 0, Equation (8.10) becomes

$$m_{pq} = \sum\sum_{(x,y) \in R} x^p y^q dx dy \tag{8.11}$$

where R denotes the region of the object.

To generate the moments directly by using Equation (8.10) or (8.11) one has to perform a significant amount of computation, especially when the order is high. Fast methods for exact computation are available, and will be discussed in Chapter 10.

The standard two-dimensional discrete moments m_{pq} will vary for a given shape depending on the spatial position of the object. Translation invariance is obtained using the central moments [Hu62]

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q g(x, y) \tag{8.12}$$

where $(\bar{x}, \bar{y})$ is the centroid of the object

$$\bar{x} = \frac{m_{10}}{m_{00}}, \qquad \bar{y} = \frac{m_{01}}{m_{00}} \tag{8.13}$$

Many useful shape features are computed from the central moments. Scaling invariant central moments are obtained by the normalization [Hu62]

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\gamma}}, \quad \gamma = \frac{p+q}{2} + 1, \quad p+q \geq 2. \tag{8.14}$$

A set of seven combinations of the second and third order normalized central moments, invariant to translation, rotation and scale change, due to Hu [Hu62], are often cited:

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
\phi_6 &= (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]
\end{aligned}$$

These moment invariants are often used in recognition tasks, and also to evaluate the accuracy of moment computation [YA94d].

Hu [Hu62] also found four moments which were invariant under general linear transformations. However, errors in these four moments were pointed out and corrected by Reiss [Rei91]. Maitra [Mai79] presented a set of moment invariants which was additionally invariant to contrast change. Flusser and Suk [FS93] derived a set of moment invariants that was invariant under general affine transforms. Abo-Zaid *et al.* [AHH88] suggested a variant of Hu's moment invariants by defining a new moment normalization given by

$$\mu'_{pq} = \mu_{pq} \frac{1}{\mu_{00}} \left(\frac{\mu_{00}}{\mu_{20} + \mu_{02}} \right)^{(p+q)/2} \tag{8.15}$$

Abo-Zaid *et al.* claimed that the moments μ'_{pq} had decreased dynamic range when compared to μ_{pq}.

There are also other types of moments applied in shape analysis and pattern recognition. Many of them were surveyed by Prokop and Reeves [PR92].

Features Based on the Moments

Many shape features can be computed from the moments.

1. The **area** is the zeroth order moment of an object, i.e., $A = m_{00}$.

2. The **centroid** can be determined from the first order moments, as given by Equation (8.13).

3. The **radius of gyration** about the origin is the radius of a circle centered at the origin where all the mass of the object may be concentrated at the circumference of the circle without change to the moment of inertia about the origin [AF67, PR92]. Let I be the moment of inertia, we have [AF67] $R = \sqrt{I/A}$. In terms of the second order moments, this value is given by [PR92]

$$R = \sqrt{\frac{\mu_{20} + \mu_{02}}{\mu_{00}}} \tag{8.16}$$

   This feature is invariant to the orientation of the object.

4. The **orientation** of an object is defined [Jai89] as the angle of axis giving the least moment of inertia. Assume there is an axis passing through the centroid $(\bar{x}, \bar{y})$ of a planar object. The moment of inertia as a function of the angle θ of the axis is

$$I(\theta) = \sum_y \sum_x [(x - \bar{x}) \cos\theta - (y - \bar{y}) \sin\theta]^2 \tag{8.17}$$

   The orientation of the object can be obtained by minimizing Equation (8.17) with respect to θ. The result is

$$\Theta = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}} \right) \tag{8.18}$$

   where the μs are the central moments. Note that this breaks down for some symmetric objects which do not have unique orientations.

5. The **bounding rectangle** is the smallest rectangle enclosing the object that is also aligned with its orientation [Jai89]. Once the orientation Θ is known, we can use the transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} \begin{bmatrix} \cos\Theta & \sin\Theta \\ -\sin\Theta & \cos\Theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

   on the boundary points and search for x'_{min}, x'_{max}, y'_{min} and y'_{max}. These give the corner points of the rectangle with length $l_b = x'_{max} - x'_{min}$ and width $w_b = y'_{max} - y'_{min}$.

6. The **image ellipse** is defined as the ellipse whose least and greatest moments of inertia equal those of the object [Jai89]. The centroid of the ellipse is the centroid of the object $(\bar{x}, \bar{y})$. The orientation of the ellipse is the orientation of the object Θ. In terms of the central moments, the lengths of the semimajor and the semiminor axes of the image ellipse are [PR92]

$$(a, b) = \left(\frac{2\left[\mu_{20} + \mu_{02} \pm \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}\right]}{\mu_{00}} \right)^{1/2} \tag{8.19}$$

The image ellipse can have special importance for the cell shape analysis since a blood cell is often shaped like an ellipse.

8.2 Cell Motion Analysis

The motion of an object can generally be measured by using a sequence of shape features. Computer-assisted analyses of cell motion were studied by several authors [LYNB80, LNY83, NL86, Sol88, SS90]. As discussed in Chapter 2, two types of motion are of interest: locomotion and pseudopod. In this section, I present some methods to analyze these two types of motion.

8.2.1 Cell Locomotion Analysis

The cell locomotion can be studied by analyzing the trajectory of a cell, obtained by linking the centroids of the cell. Noble and Levine [NL86] presented two methods to characterize the trajectory. One is the "cross-section" method, in which the observations of the positions of the cell are made at prescribed time intervals. The other is the "continuous observation" method, where one has to find some critical points in the trajectory, called the turns. The motion behavior is then defined in terms of straight line segments connecting the turns. Noble and Levine [NL86] (see also the paper of Levine *et al.* [LNY83]) used the following parameters to characterize the cell trajectories:

1. The **displacement** is the distance between two successive time points.
2. The **direction** of the motion with respect to x-axis is a parameter.
3. The **velocity** is the ratio of displacement to time.
4. The **rotation**, also called the turn-angle, is the change in the direction of the cell.

Noble and Levine [NL86] proposed a method to quantify the chemotaxis, which is a directed locomotion related to chemical agents (see Section 2.1). The **degree**

of chemotactic motion is given by

$$D_c = \frac{L}{\sum_{i=1}^{t} L_i} \tag{8.20}$$

where L is the displacement occurred in t frames and L_i is the displacement between two succesive frames. Note that $0 < D_c < 1$. The greater the value of D_c, the more chemotactic is the motion. Let β be the angle between the direction of the motion and the direction of the stimulating substance, and normalize β by

$$\beta' = 1 - \frac{2}{\pi}\beta \tag{8.21}$$

so that its value is between -1 and 1. The sign of β' can thus be used to indicate positive $(0 < \beta' < 1)$ or negative $(-1 < \beta' < 0)$ chemotaxis. The **effectiveness of the external influence** to the chemotaxis is given by

$$\epsilon = \sqrt{|\beta'| D_c} \tag{8.22}$$

Note that $0 < \epsilon < 1$. The higher the value ϵ, the stronger the effectiveness of the chemotactic agent.

Two methods for the graphical representation of cell locomotion were proposed by Soll [Sol88].

8.2.2 Pseudopod Analysis

Many 2D shape features were used as parameters to analyze the pseudopod of the blood cells [LNY83, NL86, Sol88]. Each feature is a function of time which can be represented by a feature curve. Noble and Levine [NL86] proposed to smooth the feature curve by using the polygonization method due to Ramer [Ram72]. (See Section 8.1.3 for polygonization methods.) The resulting curve approximation consists of a set of vertices representing the key frames. Between any pair of subsequent key frames, the feature will exhibit a constant dynamic behavior, either increasing, decreasing, or stationary, depending on the slope of the line. Noble and Levine [NL86] used the following parameters to describe the change of a shape feature F between two subsequent key frames i and $i+1$:

1. The **time interval** $T(i)$ is a parameter.
2. The **amount of change** $|\Delta F(i)|$ is given by the magnitude of
$$\Delta F(i) = F(i+1) - F(i) \tag{8.23}$$
3. The **type of change** is $\text{sign}(\Delta F(i))$.
4. The **rate of change** $R_F(i)$ is given by
$$R_F(i) = \tan^{-1}\left(\frac{\Delta F(i)}{T(i)}\right) \tag{8.24}$$

Noble and Levine [NL86] proposed a set of symbolic qualifiers used instead of the numeric values of shape and motion parameters. They also proposed a cell decomposition method to study the shape of each subpart of a cell.

8.3 Discussion and Conclusion

I have presented a variety of shape features used to describe the geometric properties of a 2D connected object without holes. Many of the features are originally defined for a continuous object, while the blood cells are represented in digital images. Accurate estimation of the features from discrete data can therefore be an obstacle. The efficiency of the computation is also important since we are dealing with image sequences which may contain many frames of images. A method to estimate the curvature of a discrete contour is briefly described in this chapter. The bending energy of the contour is computed from the curvature. The fast and accurate estimation of the area, perimeter and geometric moments will be discussed in later chapters. The geometric moments are important in shape analysis since many shape features can be computed from them. In Chapter 11 shape features will be computed from the cell objects, and used in a supervised evaluation of segmentation results. Properties of shape features will be further discussed in Chapter 11.

Generally, the motion of an object can be measured by a temporal sequence of the shape features. However, methods to be used in the description are dependent on the phenomenon to be studied. In this chapter, I presented some description methods for two aspects of motion of the white blood cells: locomotion and pseudopod. Further study of the cell motion description should be carried out by combining the expertise in physiology and informatics.

Chapter 9

Computation of Area and Perimeter

Let A be the area, and P be the perimeter of a 2D object. The circularity C defined by $C = 4\pi A/P^2$ is 1 for a circle and between 0 and 1 for all other shapes. The area, perimeter and circularity are useful features to describe the shape of a blood cell, and again the motion of it [NL86]. This chapter deals with the estimation of the features from a discrete binary image.

The area can often be accurately estimated by counting the number of pixels inside an object. However, the estimation of the perimeter is a problem, since the length of the original contour might be considerably different from the length of the digital contour. It is impossible to reconstruct a general continuous contour from discrete data. This means that many possible contours, all having different lengths, correspond to a specific discrete realization. Thus, to develop a practical length estimator, some reasonable assumption about the original contour should be made. Many authors [Kul77, VS82, DS86, DS87] developed and evaluated length estimators for straight lines. One of these estimators was also found to be accurate for the boundaries of blob-like objects [Kul77, DS87].

I assume that the boundary of a cell consists of chains of circular arcs, and have evaluated the precision of several area and length estimators applied to circles. The circularity C is a scale, translation and rotation invariant shape feature. The precision of the circularity was used as one of the measures for the evaluation of the area and perimeter estimations. The problem of efficient and simultaneous computation of area and perimeter will also be discussed.

9.1 A Review of Methods

Digitization of a continuous object leads to a set of discrete points. Representation of a discrete object was discussed in literatures [Pav78, Mar89]. In this section, I describe some methods which estimate the area and the perimeter of a continuous object from its discrete representation.

9.1.1 Methods Based on Bit Quads

Gray [Gra71] proposed a systematic approach to computing the area and the perimeter. The method is also presented in the book of Pratt [Pra91]. Each small region in a binary image is matched with some pixel patterns. The number of matches for each pattern is counted. The area and the perimeter are then formulated as weighted sums of the counts.

Gray [Gra71] designed a set of 2×2 pixel patterns called bit quads:

$$
\begin{array}{lllllllll}
Q_0: & \begin{matrix}0&0\\0&0\end{matrix} & & & & & & & \\
Q_1: & \begin{matrix}1&0\\0&0\end{matrix} & & \begin{matrix}0&1\\0&0\end{matrix} & & \begin{matrix}0&0\\0&1\end{matrix} & & \begin{matrix}0&0\\1&0\end{matrix} & \\
Q_2: & \begin{matrix}1&1\\0&0\end{matrix} & & \begin{matrix}0&1\\0&1\end{matrix} & & \begin{matrix}0&0\\1&1\end{matrix} & & \begin{matrix}1&0\\1&0\end{matrix} & \\
Q_3: & \begin{matrix}1&1\\0&1\end{matrix} & & \begin{matrix}0&1\\1&1\end{matrix} & & \begin{matrix}1&0\\1&1\end{matrix} & & \begin{matrix}1&1\\1&0\end{matrix} & \\
Q_4: & \begin{matrix}1&1\\1&1\end{matrix} & & & & & & & \\
Q_D: & \begin{matrix}1&0\\0&1\end{matrix} & & \begin{matrix}0&1\\1&0\end{matrix} & & & & & &
\end{array}
$$

Gray's Method

Let $n\{Q\}$ be the number of matches between the image pixels and the pattern Q. Gray computed the area of the object as

$$A = \frac{1}{4}\left[n\{Q_1\} + 2n\{Q_2\} + 3n\{Q_3\} + 4n\{Q_4\} + 2n\{Q_D\}\right] \tag{9.1}$$

and the perimeter as

$$P = n\{Q_1\} + n\{Q_2\} + n\{Q_3\} + 2n\{Q_D\} \tag{9.2}$$

The area computed by Equation (9.1) is equal to the number of pixels of the object, which is known to be accurate. However, the perimeter formula of Gray is in considerable error for many types of objects [Pra91].

Duda's Method

Pratt [Pra91] presented more accurate formulas for the area and the perimeter, citing an unpublished note of Duda

$$A = \frac{1}{4}n\{Q_1\} + \frac{1}{2}n\{Q_2\} + \frac{7}{8}n\{Q_3\} + n\{Q_4\} + \frac{3}{4}n\{Q_D\} \tag{9.3}$$

and

$$P = n\{Q_2\} + \frac{1}{\sqrt{2}}\left[n\{Q_1\} + n\{Q_3\} + 2n\{Q_D\}\right] \tag{9.4}$$

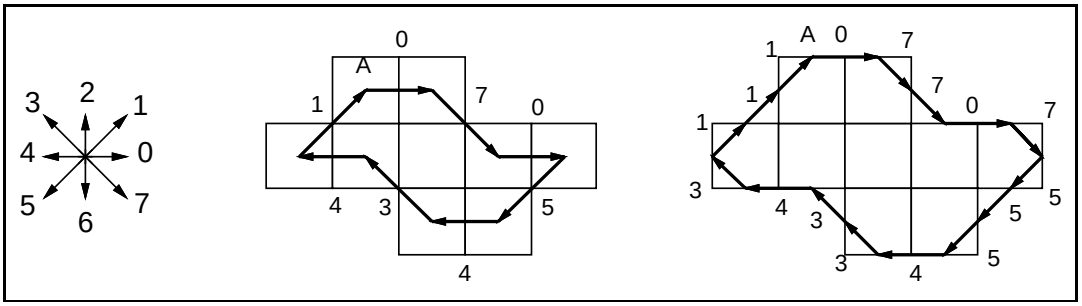


Figure 9.1: *Chain coding.* (left) *8 code words represent 8 directions.* (middle) *The Freeman chain code of the object is 07054341.* (right) *The mid-crack code of the object is 0770755543343111. "A" indicates a start point.*

9.1.2 Methods Based on Chain Codes

Chain Coding

Chain coding is a method to represent a binary object, and is useful in estimating the area and the perimeter.

The 8-connected Freeman chain coding [Fre70] uses a 3-bit code $0 \leq c \leq 7$ for each boundary point. The number c indicates the direction in which the next boundary point is located, as shown in Figure 9.1(left). Given a start point A for the object in Figure 9.1(middle), the chain codes of the object is then 07054341. There are some variations of the Freeman chain coding, for example, the 4-connected chain coding and the generalized chain coding [SF81].

The mid-crack chain coding [DM85] considers the mid-cracks instead of the centers of the boundary points. We assume that a pixel is a square with four sides. A mid-crack is then the mid-point of a pixel side. An example of the mid-crack codes is given in Figure 9.1(right). The mid-crack codes contain more redundancy than the Freeman codes, but possess some good properties in measuring shape features [DM85, SW92].

Contour Following

Boundary chain codes can be determined using a contour following [Pra91], which is a traversing process to identify the boundary of a binary object. The algorithm requires operations of $O(N)$.

Simple bug contour following should be used for 4-connected regions, and backtracking contour following should be used for 8-connected regions. During simple bug contour following, the bug makes a right turn when it is in the background, and makes a left turn when it is in the object, as illustrated in Figure 9.2(left). In backtracking contour following, the bug makes a right turn when it is in the background, and returns back to the previous point when it is in the object. This is illustrated in Figure 9.2(right).

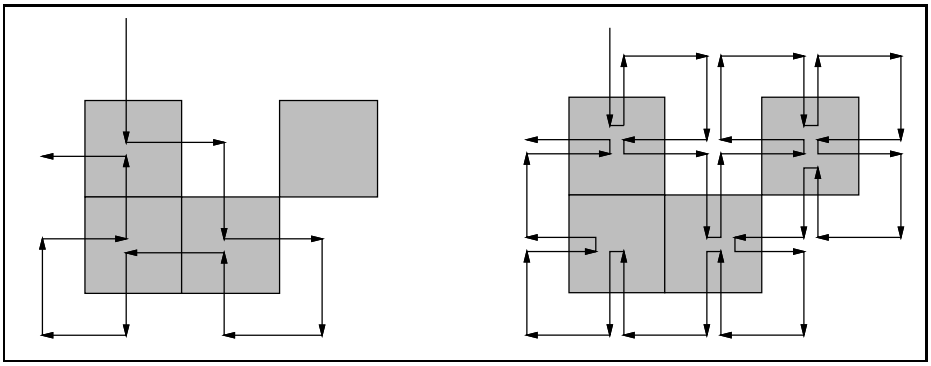


Figure 9.2: *Contour following strategies.* (left) *Simple bug contour following.* (right) *Backtracking contour following.*

Using Freeman's Chain Codes

Freeman [Fre70] computed the area enclosed by the contour of the Freeman chain codes $c_1 c_2 \cdots c_n$

$$A = \sum_{i=1}^{n} c_{ix}(y_{i-1} + c_{iy}/2) \tag{9.5}$$

where n is the length of the chain, c_{ix} and c_{iy} are the x and y components of the ith chain element c_i (c_{ix}, $c_{iy} \in \{1, 0, -1\}$ indicating the change of the x- and y-coordinates), and y_{i-1} is the y-coordinate of the start point of the chain element c_i in an arbitrary coordinate system. y_{i-1} can be computed under the contour following.

Equation (9.5) can be easily obtained from Green's theorem which gives

$$A = \oint y\, dx \tag{9.6}$$

Let A_i be the contribution of the chain element c_i to the integral, we have

$$A = \sum_{i=1}^{n} A_i, \quad A_i = \oint_{c_i} y\, dx \tag{9.7}$$

If $c_{iy} \neq 0$ we have

$$A_i = d_i \int_{y_{i-1}}^{y_i} y\, dy \tag{9.8}$$

in which $d_i = c_{ix}/c_{iy}$ is the x-slope of the chain element c_i. If $c_{iy} = 0$, then

$$A_i = \int_{x_{i-1}}^{x_i} y_{i-1}\, dx \tag{9.9}$$

Equation (9.5) is then obtained from the equations (9.7), (9.8) and (9.9).

Freeman [Fre70] computed the perimeter as the length of the chain. The formula for the perimeter is

$$P = n_e + \sqrt{2} n_o \tag{9.10}$$

where n_e is the number of even chain elements and n_o the number of odd chain elements. Referring to Figure 9.1, an even chain element indicates a vertical or horizontal connection between two boundary pixels, having length 1, while an odd chain element indicates a diagonal connection, which has length $\sqrt{2}$.

Vossepoel and Smeulders [VS82] improved Freeman's method in estimating lengths of straight lines by using a corner count n_c, defined as the number of occurrences of consecutive unequal chain elements in the Freeman chain code string. The corner count implies the direction of a straight lines. The length is given by

$$P = 0.980n_e + 1.406n_o - 0.091n_c \tag{9.11}$$

where the weights were found by a least-square fitting for all straight lines with $n_e + n_o = 1000$.

Although the methods are related to the chain coding, they can in fact determine the area and the perimeter without generating any chain codes. The values A, n_e, n_o and n_c can be computed by accumulation during the contour following.

Using Mid-Crack Chain Codes

Equation (9.5) can still be used to estimate the area when the mid-cracks are used. In this case, the computation of c_{ix} and c_{iy} is more complex since there are more possible values involved, i.e., c_{ix}, $c_{iy} \in \{-1, -1/2, 0, 1/2, 1\}$. During the contour following, a sequence of background-to-object transitions can be detected, c_{ix} and c_{iy} can then be determined according to the types of two subsequent transitions. There are totally four types of the transitions: north-south, east-west, south-north and west-east. More about the transitions is given in Chapter 10.

To estimate the perimeter, Equation (9.10) becomes

$$P = n_e + \frac{\sqrt{2}}{2}n_o \tag{9.12}$$

9.1.3 Area Estimation Using Discrete Green's Theorem

Freeman's method evaluates the area by an integration over a continuous region of a polygon enclosed by the chain elements. The algorithm is $O(N)$ since Green's theorem is used. The result is different from that of Gray's method, which equals the area to the number of pixels of a discrete region. Gray used an $O(N^2)$ algorithm to count the number of pixels. However, the counting can be done in the time of $O(N)$ by using a discrete Green's theorem [YA94a], which computes a sum of a two-dimensional function over a discrete region by a summation along its discrete boundary. The discrete Green's theorem gives exact result of a double sum, and has been used for fast and exact computation of geometric moments [YA94d]. The area is the zeroth order moment of a homogeneous region. More about the discrete Green's theorem and the computation of the moments of a discrete region is given in Chapter 10.

9.1.4 A Compensation Factor

The methods based on chain coding compute the perimeter as the length of the chain, and often give an overestimated result. Kulpa [Kul77] derived a compensation factor for computing the length of straight lines. With this factor, Equation (9.10) becomes

$$P = \frac{\pi}{8}(1+\sqrt{2})(n_e + \sqrt{2}n_o) \tag{9.13}$$

where the factor is approximately 0.948. Kulpa [Kul77] found that this compensation gave good results for most of the blob-like objects met in practice. Dorst and Smeulders [DS87] proved that Equation (9.13) gave a consistent estimate for the length of a circular arc of $\pi/4$.

9.2 Experiments and Results

The methods to be tested are the bit quad methods of Gray and Duda, Freeman's method and its analogue using the mid-crack chain codes, and Kulpa's method given by Equation (9.13). I tested the precision of these methods in estimating the areas and the perimeters of circles of radius R having integer values from 5 to 70 pixels, increasing in a step of one pixel.

Binary test images of the circles were generated by giving intensity value

$$g(x,y) = \begin{cases} 1 & \text{if } (x-x_0)^2 + (y-y_0)^2 \le R^2 \\ 0 & \text{otherwise} \end{cases}$$

where (x_0, y_0) is the coordinate of the centroid.

Using the above methods, I estimated the areas and the perimeters $\hat{A}$ and $\hat{P}$ of the circles, and computed relative errors defined by "relative error = $(x - \hat{x})/x$" where x is a true value. True values of the area and the perimeter are evaluated as $A = \pi R^2$ and $P = 2\pi R$.

The relative errors in area given by the Gray and the Duda method are shown in Figure 9.3(left). We see that the area estimations of Gray and Duda are both good. The result of the Duda method is slightly better, closer to the line of zero relative error. The mid-crack method gave a result very similar to that of Gray. The Freeman method underestimated the area, giving a relative error similar to that of the Duda method if we assume the radius is $R - 0.5$.

From Figure 9.3(right) we see that the perimeters estimated by using Kulpa's compensation factor is good, and more accurate than those estimated by the method of Freeman, which gave an overestimation. Gray's method was very inaccurate, giving a relative error of about 0.3 (overestimated). The methods of Duda and mid-crack all overestimated the perimeters. The relative errors of these two methods are similar to that of the Freeman method if we assume the radius is $R + 0.5$.

Combining the estimators of different methods, we computed the circularities shown in Figure 9.4. We see that using Kulpa's perimeter estimator together with

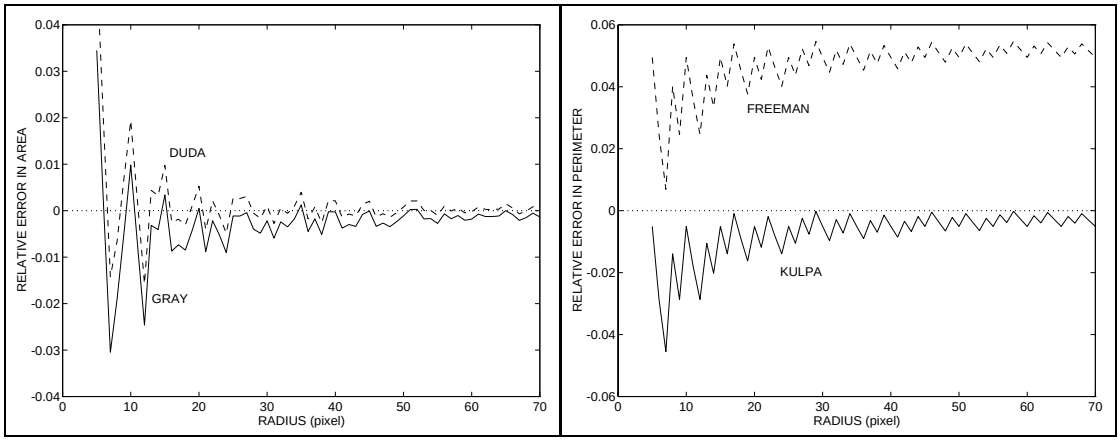


Figure 9.3: (left) *The relative errors in the areas estimated by the method of Gray, and Duda.* (right) *The relative errors in the perimeters estimated by the method of Freeman, and Kulpa. The radius is from 5 to 70 pixels, with a step of one pixel.*

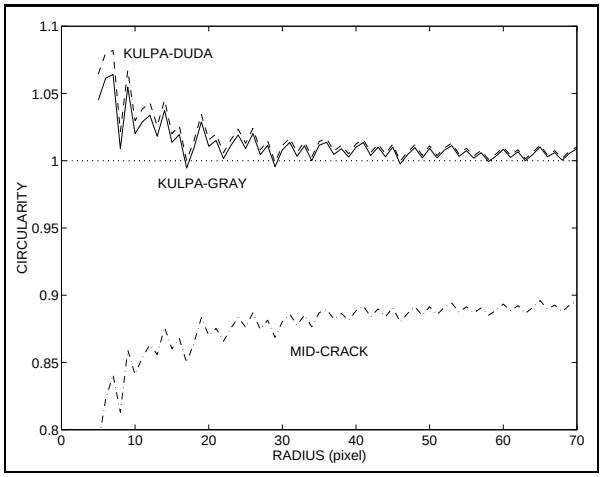


Figure 9.4: *The circularities estimated by combining different area and perimeter estimators, i.e. Kulpa's perimeters with Gray's areas, Kulpa's perimeters with Duda's areas, and the mid-crack perimeters with the mid-crack areas. The radius is from 5 to 70 pixels, with a step of one pixel.*

Gray's area estimator gives the best result, which is close to but often slightly larger than the true value of 1. It is better than combining Kulpa's perimeter with Duda's area although Duda's area is better than Gray's area. This is because Kulpa's perimeter and Gray's area are both slightly underestimated. Other combinations do not give good results. As an example we show the results when the areas and the perimeters are both computed by the mid-crack method.

9.3 Discussion

Different methods have different computational complexity. Using the bit quads, the order of the computation is N^2. (We assume an image has N^2 pixels.) It can be reduced to N by using the contour following algorithm. The order is reduced to N by using a contour following.

The area can be formulated as the count of the pixels in the object, or an integration over an approximated continuous region. The perimeter can be formulated as the length of the boundary of a polygon, approximating the original object. This length can be multiplied by a compensation factor, giving a better estimation. All the method presented above can be generalized as a linear model

$$[\hat{A} \quad \hat{P}]^T = \mathbf{W}\mathbf{n} \tag{9.14}$$

where the object is characterized by a set of counts $\mathbf{n}$, such as the counts of the pixel patterns, and the counts of the even and odd chain elements. The area and the perimeter are computed as linear combinations of the counts, using $\mathbf{W}$ as a set of weights.

I tested the accuracy of several area and perimeter estimators for circles, assuming that the boundary of a cell is a chain of circular arcs. From the above experiment, we see that Gray's bit quad method gives a good estimation of the area, but a bad estimation of the perimeter. Gray's method was improved by Duda in both the area and the perimeter estimation. But there is still a bias in the perimeter which causes a relative error of about 5 percent, and Duda's method overestimates the perimeter compared to the area. Freeman's method and the mid-crack method give results which are similar to that of Duda's method, but improve the computational performance by reducing the order from N^2 to N. The mid-crack codes were designed for a better estimation of shape features [DM85, SW92]. However, in computing the area and the perimeter of the circles, the mid-crack method does not give much better results than Freeman's method. The perimeters computed by Kulpa's method are much better than all the other methods, giving overestimation for a circle.

Different methods may have different assumptions of the location of the object boundary. The mid-crack method assumes that the boundary goes through the mid-cracks, and Freeman's method assumes that the boundary goes through the centers of the boundary pixels. The two boundaries are located in a distance of a half pixel. That means the area and the perimeter estimated by the mid-crack method are always larger than those estimated by the Freeman method.

To compute the circularity, the best result is obtained by using Gray's estimator of the area and Kulpa's estimator of the perimeter. However, they can not be computed simultaneously. Gray's area is equal to the number of the pixels in the region, which can be computed by using a discrete Green's theorem. This suggests the use of the discrete Green's theorem [YA94a, YA94d] instead of Gray's algorithm. Then the two estimators can be computed simultaneously by a contour following. Analogous to Green's theorem, the discrete Green's theorem evaluates a double sum over a discrete region by a single summation along the discrete boundary of the region, and thus gives computational advantages. It can also be extended to estimate the volume of a 3D object. More about the discrete Green's theorem will be given in the next chapter.

The results of the test, using test images of circles, should be useful for other blob-like objects. However, different shapes may require different perimeter estimators. It is therefore interesting to see how a good estimator can be found for a given type of shape. If one desires an optimal estimator, a faithful characterization (a set of counts) should be made. Dorst and Smeulders [DS87] believed that it was very difficult, and was even impossible for circular arcs. But, as a method to analyze a given characterization, they divided the parameter space of the continuous data (one dimensional R-space for the case of a circle) into regions each corresponding to one value of the discrete characterization $\mathbf{n}$. The region imply the spread of the ground truth for a given value of $\mathbf{n}$. Vossepoel and Smeulders [VS82] used three counts (see Equation 9.11) as a characterization to estimate the length of straight lines. They found the optimal weights by a least-square fitting. This method suggests a way to design the linear estimator, given by Equation 9.14. Using the linear model, the problem of finding a good estimator is equivalent to finding a set of counts (also known as the characterization of the discrete data [DS87]), and then to determine the optimal weights. In addition to the least-square fitting method used by Vossepoel and Smeulders, other learning methods, such as statistical regression [DS66] and neural network [HKP91], can also be possibly used for the determination of the weights for a linear or nonlinear model.

9.4 Conclusion

The area, perimeter and circularity are features used in cell shape analysis [LNY83, NL86]. An accurate estimation of the circularity depends on accurate estimations of the area and the perimeter. The area of a binary region can be accurately estimated by counting the number of the pixels inside the region. However, to estimate the perimeter is more difficult.

To find a good perimeter estimator, it is necessary to make some assumptions about the boundary of the object. I assume that the boundary of a cell is a chain of circular arcs. This assumption should be useful for many other blob-like objects met in practice.

In this chapter, I gave a review of several area and perimeter estimation tech-

niques. The computational performance was discussed, and the accuracy of the estimation was tested for circles of different sizes. I found that all the methods gave good estimations of the area, and the method of Kulpa gave a good estimation of the perimeter. To compute the circularity, the best result was obtained by using Kulpa's perimeter, and Gray's area, which was the number of the pixels of the region. The Gray's area can be computed by a discrete Green theorem. Then the area and the perimeter can be computed simultaneously and efficiently, based on a contour following algorithm.

Chapter 10

Moment Computation

We have seen that the Cartesian geometric moments (for short moments) play an important role in shape description, since the moments are related to the definition and the computation of many shape features, such as area, centroid, orientation, and image ellipse.

The $(p+q)$'th order moment of a function $g(x,y)$ is defined as

$$m_{pq} = \int_{y=-\infty}^{\infty} \int_{x=-\infty}^{\infty} g(x,y) x^p y^q dx dy \tag{10.1}$$

In discrete gray level images, the moment can be computed by

$$m_{pq} = \sum_{y} \sum_{x} g(x,y) x^p y^q \tag{10.2}$$

In binary images, it becomes

$$m_{pq} = \sum\sum_{(x,y)\in R} x^p y^q dx dy \tag{10.3}$$

where R denotes the region of the object.

To generate the moments directly by using Equations (10.2) or (10.3) one has to perform a significant amount of computation. It requires additions and multiplications both of the order N^2 ($O(N^2)$), where N is the vertical or horizontal size of an image. Since the value of the moments are large, integer words do not give enough range, and therefore long integers or float numbers have to be used. The upper bound for a $(p+q)$'th order moment is given by Sadjadi and Hall [SH78],

$$m_{pq} \leq g_{max} \left[\frac{N^{p+1}-1}{p+1} \right] \left[\frac{N^{q+1}-1}{q+1} \right] \tag{10.4}$$

where g_{max} is the maximal gray level. Using long integer or float calculations further slows down the computation of moments. Thus, the computational cost limits the use of moments in on-line, and even in off-line applications.

Many algorithms [Hat86, ZVZvK87, LS91, DBN92, YA94d] have been developed to speed up the computation of moments by reducing the computational redundancy.

Some of them work for gray level images, some of them for binary images, and some of them for parallel computation or optical implementation. In this chapter, I give a review of earlier methods, and then present a new method for fast computation of moments in binary images by using a discrete version of Green's theorem. The precision of the moment computation is also discussed. The exact computation is referred to the computation which gives the same results as if the moments were computed directly by Equation (10.2) or (10.3). I will demonstrate that the exact computation is important to obtain good results in computing shape features.

10.1 A Review of Methods

10.1.1 Image Filtering or Transform

Geometric moments of gray level images as well as of binary images can be calculated using image transforms. Hatamian [Hat86] computed the moments by a causal spatial filter. The filtering needs only additions of $O(N^2)$ for 2D images and $O(N)$ for 1D signals. Hatamian developed an algorithm for computing the moments of gray level images. Fu *et al.* [FYC93] found the relation between the moments and the coefficients of the Hadamard transform of an image. For a binary image, the 10 moments of orders up to three are totally determined by four projections of the image, and can be computed from the coefficients of the 1D Hadamard transform of the projections. The 1D Hadamard transform needs additions of $O(N \log_2 N)$. Fu *et al.* used a contour following [Pra91] to make the projections of a binary image.

10.1.2 Delta Method

The delta method is suitable for binary images represented by y-lines. A moment of an object is the sum of the moments of all the vertical or horizontal line segments of the object. δ is used to denote the length of a line segment, hence the name of the method. The moments of a line segment can be expressed in a closed form. The method was first proposed by Zakaria *et al.* [ZVZvK87], and then improved by Dai *et al.* [DBN92] and Li [Li93]. The method of Dai *et al.* is called an integral method since the moments of line segments are computed by integration, instead of summation used by the delta method. Given the y-line representation of an object, Li's algorithm requires about $6N$ multiplications and $17N$ additions to compute the 10 low order moments for a convex object.

10.1.3 Computation via Corner Points

This approach is applicable only to binary images. The geometric moments can be computed via corner points in the boundary of the object. The boundary between two corner points is a straight line. Strachan *et al.* [SNA90], Leu [Leu91], and Singer [Sin93] computed the double integral over the object by summing up the integrals

over some simple regions each containing one straight line boundary. The integral over such a simple region can be expressed in a closed form. Jiang and Bunke [JB91] used Green's theorem to transform the double integral to a single integral. The single integral along a straight line between two corner points can be expressed in a closed form. To compute the moments via corner points implies additions and multiplications of $O(C)$, where C is the amount of the corner points. Since a large amount of computation is required for each corner point, the method is efficient only for objects with simple shape.

10.1.4 Methods Based on Green's Theorem

This theorem has been used to compute the moments of binary images. We [YA94c] applied it also in gray level images. Green's theorem [EP82] evaluates a double integral over a region as a single integration along the boundary of the region. It is important for moment computation since the shape of a binary object is totally determined by its boundary. Li and Shen [LS91] proposed a fast moment computation method by using a contour following and Green's theorem. The moment kernel updating technique is used so that the algorithm needs only additions of $O(L)$, where L is the length of the boundary of the object. The method is efficient, but not accurate, since Li and Shen used an approximation of Green's theorem in a digital image lattice. There exist discrete versions of Green's theorem [Tan81, YA94a] for exact computation. By using a discrete version of Green's theorem, Philips [Phi93] proposed a method giving exact results. Unfortunately his method is not as efficient as the method of Li and Shen. We proposed a method [YA94d] which was as efficient as the method of Li and Shen, and gave exact results. In this chapter I will present our improved method [YA94c, YA94b], which is faster than previous methods, but still achieves exact computation of the moments.

We [YA94c, YA94b] also applied the discrete Green's theorem to gray level images, and presented a method for fast computation of moments of gray level regions. The method for gray level images is particularly efficient when the moments are to be computed in a number of regions of an image. In a recent paper [YA94b], we showed that the new method could be easily extended to compute the moments of three-dimensional objects represented by voxels.

10.2 Discrete Versions of Green's Theorem

Green's theorem [EP82] relates a line integral around a simple closed plane curve C to an ordinary double integral over the plane region R bounded by C. Suppose that the curve C is piecewise smooth, and functions $M(x,y)$ and $N(x,y)$ are continuous and have continuous first-order partial derivatives in R, then

$$\oint_C M(x,y)\,dx + N(x,y)\,dy = \int\int_R f(x,y)dA \tag{10.5}$$

where $\oint_C$ denotes a line integral along C in the counterclockwise direction, and

$$f(x,y) = \frac{\partial N(x,y)}{\partial x} - \frac{\partial M(x,y)}{\partial y} \tag{10.6}$$

Directly applying Green's theorem on a discrete region by changing the integrations in Equation (10.5) to summations results in an approximation, i.e.,

$$\sum_C (M\Delta x + N\Delta y) \approx \sum_R \sum f \Delta x \Delta y \tag{10.7}$$

Li and Shen [LS91] used this approximation to compute geometric moments. The results are poor when the object is small, or the shape of the object is complex. However, there are different versions of the discrete Green's theorem which exactly evaluate a sum over a region by a summation along the boundary.

10.2.1 Tang's Version

Tang [Tan81] proposed a discrete version of Green's theorem in which the boundary L of an object R is defined by

$$L = \{p | p \in R,\ \exists q \in N_4(p),\ q \notin R\} \tag{10.8}$$

where $N_4(p)$ denotes the set of the 4 neighbors of a pixel p. Tang's formula is

$$\sum_R \sum f(x,y) = \sum_L (F_x(x,y) D_Y(x,y) + f(x,y) C_Y(x,y)) \tag{10.9}$$

where

$$F_x(x,y) = \sum_{i=0}^{x} f(i,y) \tag{10.10}$$

In Tang's paper, $D_Y(x,y)$ and $C_Y(x,y)$ are defined by the Freeman chain code representation of the boundary. They can also be defined by the coordinate (x,y) of the boundary point, as we give below

$$D_Y(x,y) = \begin{cases} 1 & (x-1,y) \in R,\ (x+1,y) \notin R \\ -1 & (x-1,y) \notin R,\ (x+1,y) \in R \\ 0 & \text{otherwise} \end{cases}$$

$$C_Y(x,y) = \begin{cases} 1 & (x-1,y) \notin R \\ 0 & \text{otherwise} \end{cases}$$

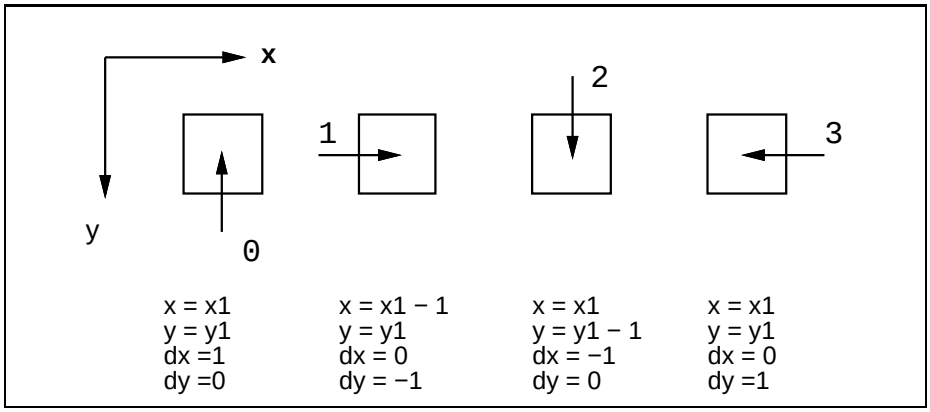


Figure 10.1: *The four possible directions of background to object transition. The squares represent the boundary points in the object, with coordinate* (x_1, y_1). $(x, y, \Delta x, \Delta y)$ *is computed based on* (x_1, y_1) *and the direction of the transition.*

10.2.2 Philips' Version

In Philips' version of discrete Green's theorem [Phi93], the boundary of an object R is defined in another way. Let $\partial R^+ = \{(x,y)|(x,y) \in R, (x+1,y) \notin R\}$ and $\partial R^- = \{(x,y)|(x,y) \notin R, (x+1,y) \in R\}$, the boundary ∂R is defined as $\partial R = \partial R^+ \cup \partial R^-$. Philips' formula is

$$\sum_R \sum \nabla_x f(x,y) = \sum_{\partial R^+} f(x,y) - \sum_{\partial R^-} f(x,y) \tag{10.11}$$

where $\nabla_x f(x,y) = f(x,y) - f(x-1,y)$. Note that ∂R is not a closed boundary of the region R. It is the west and east boundary. A dual formula can be obtained if the north and south boundary is given.

10.2.3 A New and Improved Version

We [YA94a] associate the discrete Green's theorem with a contour following algorithm (see Section 9.1.2).

During the contour following, the background to object transitions are detected. A transition can have four possible directions, 0, 1, 2, and 3, as illustrated in Figure 10.1. Thus, a transition is denoted by a quadruple $(x, y, \Delta x, \Delta y)$ which consists of a point coordinate (x, y) and $(\Delta x, \Delta y)$ giving directional information. The set of transitions is defined as

$$\begin{aligned}
T_0 &= \{(x,y,\Delta x,\Delta y)|(x,y) \in R, (x,y+1) \notin R, \Delta x = 1, \Delta y = 0\} \\
T_1 &= \{(x,y,\Delta x,\Delta y)|(x,y) \notin R, (x+1,y) \in R, \Delta x = 0, \Delta y = -1\} \\
T_2 &= \{(x,y,\Delta x,\Delta y)|(x,y) \notin R, (x,y+1) \in R, \Delta x = -1, \Delta y = 0\} \\
T_3 &= \{(x,y,\Delta x,\Delta y)|(x,y) \in R, (x+1,y) \notin R, \Delta x = 0, \Delta y = 1\} \\
T &= T_0 \cup T_1 \cup T_2 \cup T_3
\end{aligned}$$

Note that one boundary coordinate can correspond to many transitions. As we will see later, only a triplet is involved in the computation and is therefore registered under the contour following.

By using contour following and the transition concept, we present a new formula for the discrete Green's theorem

$$\sum_R\sum f(x,y) = \sum_T F_x(x,y)\Delta y \tag{10.12}$$

where $F_x(x,y)$ is defined by Equation (10.10). If we define $F_y(x,y) = \sum_{i=0}^{y} f(x,i)$, then we obtain a dual formula

$$\sum_R\sum f(x,y) = \sum_T F_y(x,y)\Delta x \tag{10.13}$$

Assuming that an object R consists of rows of points, and that the x-coordinates of the start point and the end point of the row r are $x_1(r)$ and $x_2(r)$, Equation (10.12) can be proved as follows:

$$\begin{aligned}
\sum_R\sum f(x,y) &= \sum_r \sum_{x=x_1(r)}^{x_2(r)} f(x,y) \\
&= \sum_r (F_x(x_2(r),y) - F_x(x_1(r)-1,y)) \\
&= \sum_{T_3} F_x(x,y) - \sum_{T_1} F_x(x,y) \\
&= \sum_T F_x(x,y)\Delta y
\end{aligned}$$

Equation (10.13) can be proved in a similar way.

10.3 A New Moment Computation Method

Using a contour following and the Green's theorem, Li and Shen [LS91] proposed a fast algorithm to compute the moments of binary objects, defined by Equation (8.17). Li and Shen's formula for moment computation is

$$m_{pq} = \frac{1}{p+1}\sum_L x^{p+1} y^q \Delta y \tag{10.14}$$

where L is the boundary defined in the same way as by Tang (see Equation (10.8)), and Δy is the difference of the y-coordinates of the point (x,y) and the point following (x,y) on the boundary L. Li and Shen used contour following to obtain the coordinates $(x,y) \in L$. The method is fast but not accurate, since Li and Shen's formula is obtained by a discrete approximation of Green's theorem.

Philips [Phi93] improved the accuracy by using the discrete version of Green's theorem given by Equation (10.11). This algorithm produces exact results. Substituting $f(x,y) = y^q \sum_{i=0}^{x} i^p$ into Equation (10.11), Philips gave a formula for moment computation.

$$m_{pq} = \sum_{\partial R^+} S_p(x) y^q - \sum_{\partial R^-} S_p(x) y^q \tag{10.15}$$

where

$$S_p(x) = (x+1) \sum_{j=0}^{p} \frac{[S^{-1}]_{pj}}{j+1} \sum_{v=0}^{j} [S]_{jv} x^v \tag{10.16}$$

in which S is the Stirling matrix of the first kind [Mor69].

Philips evaluated Equation (10.15) by accumulating $S_p(x)y^q$ at each boundary point. The method of Philips is less efficient than the method of Li and Shen, since more monomials $x^i y^j$ have to be computed in the Philips' method. To compute the moments of orders up to $(p+q)$, the method of Li and Shen uses the monomials $x^i y^j$ for $i = 1, \cdots, p+1$ and $j = 0, \cdots, q$, and the Philips' method uses the monomials for $i = 0, \cdots, p+1$ and $j = 0, \cdots, q$.

Our new method, as presented below, uses the same number of monomials as the method of Li and Shen. Substituting $f(x,y) = x^p y^q$ into Equation (10.12), we have

$$\begin{aligned} m_{pq} &= \sum_T \sum_{i=0}^{x} i^p y^q \Delta y \\ &= \sum_T \left(\frac{x^{p+1} y^q}{p+1} + \frac{x^p y^q}{2} + \sum_{j=2}^{p} \frac{1}{j} C_p^{j-1} B_j x^{p-j+1} y^q \right) \Delta y \end{aligned} \tag{10.17}$$

where C_p^{j-1} is a binomial coefficient, which can be computed as

$$C_p^k = \frac{p!}{k!(p-k)!} \tag{10.18}$$

and B_j is the j'th Bernoulli number. For $0 \le j \le 6$ the Bernoulli numbers are 1, $-1/2$, $1/6$, 0, $-1/30$, 0, and $1/42$. Let

$$u_{ij} = \sum_T x^{i+1} y^j \Delta y \tag{10.19}$$

Then Equation (10.17) becomes

$$m_{pq} = \frac{u_{pq}}{p+1} + \frac{u_{p-1,q}}{2} + \sum_{j=2}^{p} \frac{1}{j} C_p^{j-1} B_j u_{p-j,q} \tag{10.20}$$

This is our formula for moment computation. The moment m_{pq} is a linear combination of u_{jq} for $j = 0, \cdots, p$. The 10 low order moments, which are often used in

applications, can be computed as

$$\begin{bmatrix} m_{0q} \\ m_{1q} \\ m_{2q} \\ m_{3q} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/6 & 1/2 & 1/3 & 0 \\ 0 & 1/4 & 1/2 & 1/4 \end{bmatrix} \begin{bmatrix} u_{0q} \\ u_{1q} \\ u_{2q} \\ u_{3q} \end{bmatrix}$$

In our earlier algorithm [YA94d], we computed the monomials $x^i y^j$ for each transition registered during contour following, and u_{ij}, as defined by Equation (10.19), were computed by accumulation. The monomials $x^i y^j$ can be computed by updating, as done by Li and Shen [LS91], so that only additions are required.

Our earlier method is as efficient as the method of Li and Shen. We have found a more efficient method [YA94c, YA94b] in which u_{ij} are computed in another way. We treat the image line by line. For a given y-coordinate, we have a set of transitions $T(y)$ for this y-value.

$$T(y) = \{(x, y', \Delta x, \Delta y) | (x, y', \Delta x, \Delta y) \in T,\ y' = y\} \tag{10.21}$$

We define $v_i(y)$ as

$$v_i(y) = \sum_{T(y)} x^{i+1} \Delta y \tag{10.22}$$

We compute $v_i(y)$ for $i = 0, \cdots, p$ and all y-coordinates. This can be implemented by using $p + 1$ arrays of size N. During contour following, the array entries are updated.

Comparing Equation (10.19) with (10.22), we have

$$u_{ij} = \sum_{y} v_i(y) y^j \tag{10.23}$$

which means that u_{ij} is the j'th moment of a 1D signal $v_i(y)$. A fast algorithm for computing moments of a 1D signal was proposed by Hatamian [Hat86]. Letting $v_i^0(y)$ be the result of the Hatamian filtering of $v_i(y)$, we have

$$v_i^0(y) = \sum_{k=y}^{N} v_i(k) \tag{10.24}$$

Applying the Hatamian's filter recursively, we obtain

$$v_i^j(y) = \sum_{k=y}^{N} v_i^{j-1}(k) \tag{10.25}$$

Then, the 1D moments u_{ij} are linear combinations of $v_i^j(1)$. For $j \leq 3$ we have

$$\begin{bmatrix} u_{i0} \\ u_{i1} \\ u_{i2} \\ u_{i3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 1 & -6 & 6 \end{bmatrix} \begin{bmatrix} v_i^0(1) \\ v_i^1(1) \\ v_i^2(1) \\ v_i^3(1) \end{bmatrix}$$

The Hatamian filtering of a 1D signal requires N additions.

Implementation Our new method to compute the moments can be implemented in three steps:

1. The contour following is applied, and a triplet $(x, y, \Delta y)$ is recorded for each transition from the background to object. Then the array entries $v_i(y)$ are accumulated for $i = 0, \cdots, p$, see Equation (10.22)

2. Apply the Hatamian filtering to $v_i(y)$ and obtain $v_i^j(y)$ for $j = 0, \cdots, q$.

3. Compute the moment m_{pq} as a linear combination of $v_i^j(1)$.

For the computation of the 10 low order moments, the new method requires, in addition to the operations for contour following, 3 multiplications and 4 additions for each T_1 and T_3 transition in the first step. The second step needs $10N$ additions. The last step needs only a few multiplications and additions. So totally it needs about $10N + 8S$ additions and $6S$ multiplications in addition to the contour following, where S is the number of line segments of the object.

The 3 multiplications for each transition are used to compute x^i from x. In some computational environment, addition is much faster than multiplication. In this case we can use Li and Shen's updating technique [LS91], requiring only additions, to compute x^i, since x is changed at most by 1 from one transition to the next. Using the updating, 10 additions are used instead of 3 multiplications.

10.4 Experiments and Results

The new method was tested in computing Hu's moments and various moment-based shape features using test images which contained small perfect elliptical objects. The length of the semimajor and semiminor axes were 20 and 10 pixels respectively. The orientation of the ellipses were from 0 to 90 degrees, increasing in steps of 2 degrees. Both the new method and the method of Li and Shen were used in the computation so that the precisions of the computation could be compared.

Hu [Hu62] proposed a set of seven moments ϕ_1 to ϕ_7 which are invariant under translation, scaling and rotation of objects. The set of moments were discussed in Chapter 8. The invariant moments ϕ_1 and ϕ_2 computed by the two methods are shown in Figure 10.2. We see that the values computed by the new method were very stable during the orientational change, while the values computed by the method of Li and Shen were more varied.

For the other invariant moments of the ellipse, the results of the new method were also much better than the results of the method of Li and Shen. For these invariant moments, the values computed by the new method were very close to the true values of a continuous elliptical object, which were all zeros.

I used the two methods to estimate the orientation of the elliptical object. Let θ be the true orientation of an ellipse and $\hat{\theta}$ is the estimated orientation, computed as

$$\hat{\theta} = \frac{1}{2} \tan^{-1} \left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right] \tag{10.26}$$

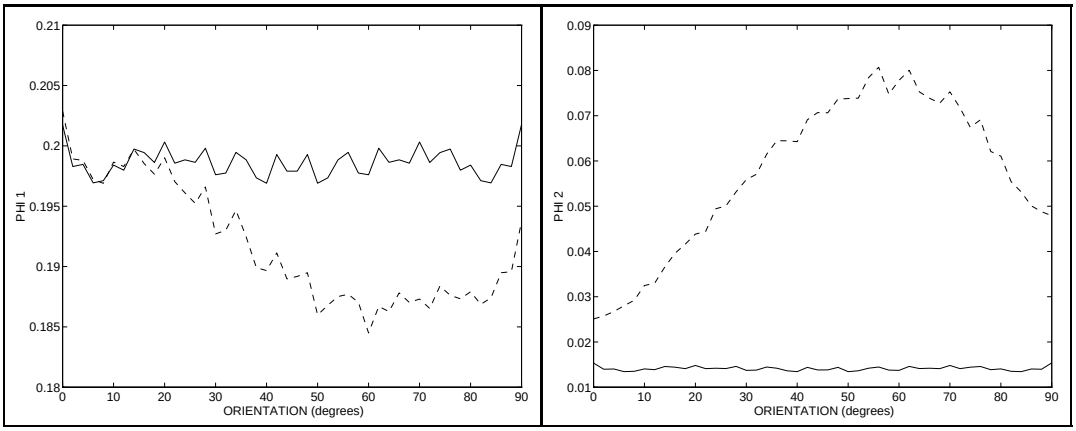


Figure 10.2: *Invariant moments* ϕ_1 (left) *and* ϕ_2 (right) *of an ellipse computed by the new method (solid line) and the method of Li and Shen (dashed line). The orientation is changing from 0 to 90 degrees.*

The error in the estimated orientation, given by

$$\Delta\theta = \hat{\theta} - \theta \tag{10.27}$$

is shown in Figure 10.3 as a function of true orientation. We see that the new method gave an insignificant orientation error compared to the error introduced by the method of Li and Shen.

The radius of gyration of an object is defined as the radius of a circle where we could concentrate all the mass of the object without altering the moment of inertia about its center of mass [PR92]. This feature is inherently invariant to image orientation, and is therefore a simple and useful rotationally invariant feature for shape analysis. In terms of second order central moments, it is given by

$$R = \sqrt{\frac{\mu_{20} + \mu_{02}}{\mu_{00}}} \tag{10.28}$$

The precision in the computation of this invariant feature for the two methods was examined. For the elliptical objects of different orientations, I computed the radii of gyration by the two methods, shown in Figure 10.4. We see that the results computed by our method were more stable during the rotation than that of the method of Li and Shen.

The image ellipse is defined as the ellipse whose least and greatest moments of inertia equal those of the object [Jai89]. There are five scale parameters of which the centroid can be determined by

$$\bar{x} = \frac{m_{10}}{m_{00}}, \qquad \bar{y} = \frac{m_{01}}{m_{00}} \tag{10.29}$$

the orientation is given by Equation (10.26), and the lengths of the semimajor and

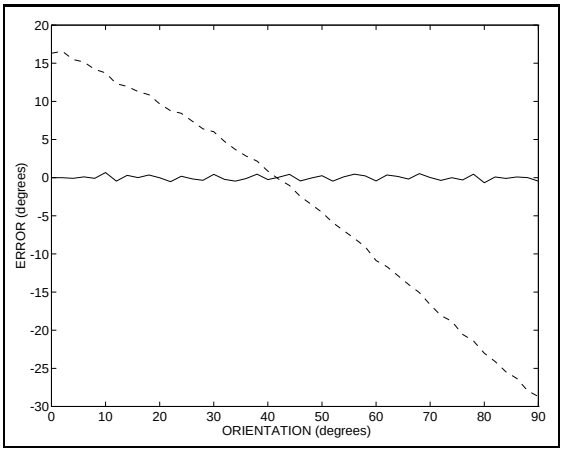


Figure 10.3: *Error in estimated orientation as a function of true orientation of the elliptical object for the new method (solid line) and the method of Li and Shen (dashed line).*

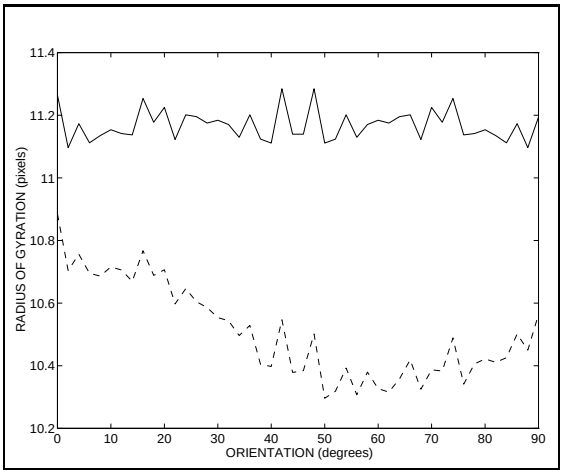


Figure 10.4: *The radius of gyration computed by the new method (solid line) and the method of Li and Shen (dashed line).*

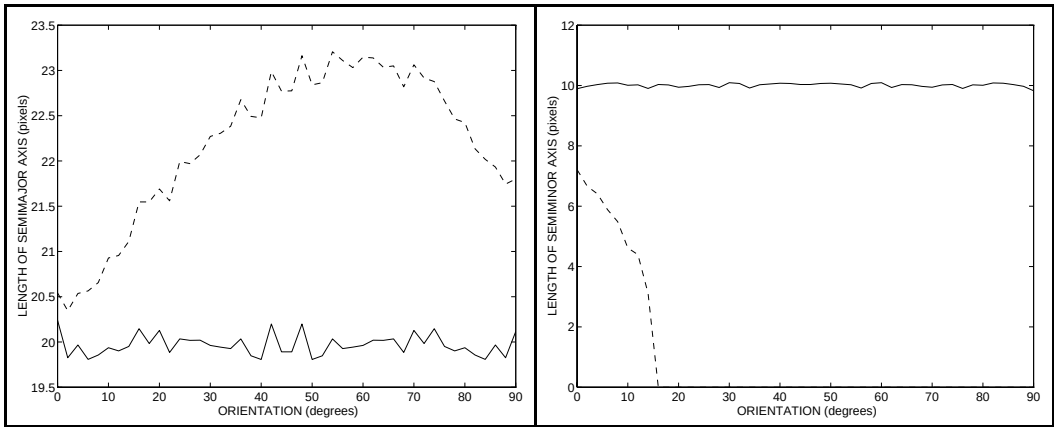


Figure 10.5: *The lengths of the semimajor axis* (left) *and the semiminor axis* (right) *of an ellipse estimated by the new method (solid line) and the method of Li and Shen (dashed line).*

the semiminor axes are [PR92]

$$(a,b) = \left(\frac{2\left[\mu_{20} + \mu_{02} \pm \sqrt{(\mu_{20}-\mu_{02})^2 + 4\mu_{11}^2}\right]}{\mu_{00}} \right)^{1/2} \tag{10.30}$$

a and b are invariant to the orientation of the object. For an elliptical object, the image ellipse should be the same as the ellipse object itself. I used the two method to estimate the lengths of the axes of the elliptical object, and compared them to the true values. From Figure 10.5 we see that the new method gave more rotationally stable results than the method of Li and Shen. In the case of the semiminor axis, the method of Li and Shen was not able to give positive real value results for all the orientations.

10.5 Discussion

10.5.1 The Efficiency of the Computation

Many different methods were proposed for the fast computation of the Cartesian geometric moments. The delta methods are suitable for binary objects represented by y-lines, the corner point methods works well for binary objects represented by the polygon approximation, and the methods based on Green's theorem are designed to compute the moments of objects represented by binary images.

Contour following [Pra91] is used in many fast computation methods including the method of Fu *et al.*, Li and Shen, and the new method. Contour following is an $O(N)$ algorithm to identify the boundary pixels of an object. To identify the boundary by scanning the image or by recursively including the neighbor points, starting from a seed point, one will need $O(N^2)$ operations.

Our early method [YA94d] is as efficient as the method of Li and Shen. This can be observed by comparing the formula of u_{ij} in Equation (10.19) and the formula for Li and Shen's computation in Equation (10.14). The moments are linear combinations of u_{ij}. To compute the 10 low order moments, the computation of the monomials of u_{ij} will require 9 multiplications for each of the T_1 and T_3 transitions. (An updating method can be applied so that more additions are used instead of the 9 multiplications when addition is faster than multiplication [YA94d].) To sum up the monomials requires 10 additions for each of the transitions. The new method does not compute the monomials directly. In the first step, it requires 3 multiplications and 4 additions for each of the T_1 and T_3 transitions. Then $10N$ additions are used for the Hatamian filtering. N is the size of the image, which will be much less than the number of T_1 and T_3 transitions if the object is large compared to the image. Thus the new method requires less computation than our early method.

10.5.2 The Precision of the Computation

Different methods give different precision, depending on the size, shape and complexity of the object. Exact computation means to obtain results as if the moments were computed by a double sum as in Equations (10.2) and (10.3).

A good precision is important to achieve invariance for Hu's moments [Hu62], which again leads to better results in shape analysis and recognition. Strict invariance is achieved by assuming a continuous image function and a double integral over the region. Due to the digital nature of the image, the moment invariants computed digitally are not strictly invariant. The precision of the computation of the seven invariant moments of Hu are often quoted, as discussed and tested by Gonzalez and Woods [GW92]. It can be shown [TC86] that the set of moment combinations is still invariant under image translation if they are computed by the exact computation. But under digital processing, the invariant moments are expected not to be strictly invariant under image rotation and scale changes.

Dai *et al.* [DBN92] evaluated the moment invariants of Hu computed by the exact evaluation of moments and by an integral method. They found that the results of the integral method were more sensitive to the sampling. The integral method is better for shapes with only vertical and horizontal edges. In this case, the integral method gives the true value of the moments. For the 10 low order moments, the differences between the results of the integral method and the exact computation have been formulated by Dai *et al.* [DBN92].

In our early papers [YA94d, YA94c, YA94b], we compared the precision of the new method and the method of Li and Shen. Test images of triangle, square, helicopter and aircraft objects were used for the evaluation. We found that the method of Li and Shen, using an approximation of Green's theorem, was not accurate for small objects or objects with a complex shape. By using the new exact method, the precision was largely improved.

In this thesis, I tested the precision of the two methods using a test image of a perfect small elliptical object, since a cell is often shaped like an ellipse. Hu's

moments and several other rotational invariant features were examined. I found that the new method provided a better rotational invariance.

An alternative way of achieving rotation invariance is to estimate the object orientation angle, given by Equation (10.26), and compute the normalized central moments in a rotated coordinate system. This *principal axes* approach was suggested by Hu [Hu62], and was utilized in early character recognition experiments by Giuliano *et al.* [GJK+61]. The method obviously breaks down for objects with no unique set of principal axes. Also, the method is subject to error, introduced by the choice of method when computing the moments. From the experimental results, we can see that the approximation to Green's theorem used by Li and Shen may introduce a significant error in the estimated orientation compared to our new method.

The new method was much less sensitive to sampling than the method of Li and Shen. This was demonstrated in our early papers [YA94d, YA94c, YA94b] by comparing the Hu's invariant moments of an object of different sizes. We found that the moments computed by the new method were much more stable under scale changes, and thus may be more useful in scale space applications of invariant geometric moments.

The moments play an important role in shape analysis since many shape features are based on the moments. The zeroth order moment is the total mass of an object or the area of the silhouette of a segmented object. The first order moments are used to locate the center of mass. The second order moments are related to several shape features including the orientation, the image ellipse and the radius of gyration. Computation of these features for an elliptical object was tested for the new method and the method of Li and Shen. I found that the new method gave better results.

10.6 Conclusion

The Cartesian geometric moments are an important tool in many image analysis tasks. A number of features used in shape analysis may be computed from the moments. The speed of computation then becomes a possible obstacle. Many algorithms have been proposed to speed up the computation of moments [YA94c]. Using Green's theorem we may evaluate a double integral over a region as a single integral along the boundary of the region. The fast method of Li and Shen uses an approximation of Green's theorem. Thus, as shown above, the method does not produce correct results for small and complex objects. However, there exists discrete Green's theorem for exact computation. By using a discrete version of Green's theorem, Philips [Phi93] proposed a method that gives exact results, but which is not as efficient as the method of Li and Shen. We [YA94d] proposed an exact method that is as fast as the Li and Shen method. In a recent paper, we [YA94c] introduced an improved method, which is faster than previous methods, but still achieves exact computation of the moments. Exact computation is important to achieve invariance of Hu's moments, and obtain precise results of shape features, especially for small

and complex objects.

The moments will be applied on the cell images to compute shape features. The results of the application of the moment-based shape features will be discussed in Chapter 11.

Chapter 11

System Evaluation

The evaluation of the image analysis system is based on the shape features computed from the segmented cell objects.

To evaluate the results of a segmentation system is an interesting research topic which is discussed in many papers [YMB77, WR78, LN82, LN85, ZG92]. In this chapter, I first give a review of segmentation evaluation methods, and then use a supervised method to evaluate the segmentation results of the cell image sequences. A supervised evaluation utilizes a reference segmentation and computes the difference between the reference segmentation and the results of the segmentation system to be evaluated. Different segmentation techniques are compared, and the effect of the noise reduction is also discussed.

In addition to the evaluation of the segmentation results, I also discuss the properties of some shape features.

11.1 A Review of Evaluation Methods

Usually, a human being is the best judge for evaluating the output of a segmentation algorithm. Subjective evaluation is therefore used in many applications. However, when the results can not be easily evaluated visually, a quantitative evaluation will be necessary. Some attempts have already been made on quantitative evaluation of segmentation results [YMB77, WR78, LN82, LN85, ZG92].

I divide the evaluation methods into two groups: supervised and unsupervised evaluation, depending on whether the method utilizes a priori knowledge of the correct segmentation.

11.1.1 Unsupervised Methods

Unsupervised evaluation does not depend on a correct segmentation. Haralick and Shapiro [HS85] established the following qualitative guideline for a good image segmentation: "Regions of an image segmentation should be uniform and homogeneous with respect to some characteristic such as gray tone or texture. Region interiors

should be simple and without many small holes. Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple, not ragged."

Quantitative segmentation performance measures were developed by several authors. Weszka and Rosenfeld [WR78] used busyness and classification error as performance criteria, where the busyness of the segmented image was an unsupervised measure. The idea behind this measure is that the segmented image should look smooth rather than busy. Its calculation is based on the gray level co-occurrence matrix whose entries are estimates for the joint probabilities of gray levels for adjacent pixels. Levine and Nazif [LN85] defined a set of parameters for unsupervised evaluation including region uniformity, region contrast, line contrast and line connectivity. They assumed that some features should be uniform inside each region, and distinct between adjacent regions. Sahoo *et al.* [SSWC88] used two measures for the evaluation of thresholding techniques. One is the uniformity criterion of Levine and Nazif. The other is called shape measure, computed from the gradient values and the selected threshold value.

11.1.2 Supervised Methods

A supervised evaluation utilizes a reference (correct) segmentation and measures the difference between the reference segmentation and the output of the segmentation system to be evaluated.

Measures of the difference between two segmentation results were developed by many authors. Levine and Nazif [LN82] proposed to use an under-merging error and an over-merging error for each region. The former is the number of pixels of the current region misclassified as other regions, and the latter is the number of pixels of other regions misclassified as the current region. The two measures are normalized and combined to produce a total difference measure. Similarly, Weszka and Rosenfeld [WR78] used a classification error, Lim and Lee [LL90] used a probability of error, de Graaf *et al.* [dGKVV92] used a correctness of segmentation, and Pal and Bhandari [PB93] used a divergence measure. All these measures are based on the number of the misclassified pixels. A more complex measure, in which the positions of the misclassified pixels were also taken into account, was developed by Yasnoff *et al.* [YMB77].

Goal-directed supervised evaluation was also proposed. In many image analysis tasks, the ultimate goal of segmentation is to obtain measurements of the object features. So, the accuracy of those ultimate measurements will indicate the performance of the segmentation. Zhang and Gerbrands [ZG92] proposed to use the ultimate measurement accuracy (UMA) to assess the performance of a segmentation algorithm. Let x be a feature computed from an object in a reference image, and $\hat{x}$ be that computed from the object in the automatic segmentation result. The UMA can be evaluated as

$$UMA = |x - \hat{x}| \tag{11.1}$$

Often the UMA has to be normalized. The difference between two segmentation results can be described by the accuracy of a set of ultimate measurements.

The goal-directed evaluation method was also used by Trier and Jain [TJ94] for the segmentation of document images. They used a correctness of recognition as a quantitative measure for the performance of the segmentation.

11.2 Experiments and Results

Since manual segmentation results for the blood cell images were obtained (see Chapter 3), I used a supervised evaluation method, in which the difference between the manual and the automatic segmentation was measured.

The method based on an under-merging error and an over-merging error, proposed by Levine and Nazif [LN82], was used to assess an overall segmentation accuracy. In addition, I computed the ultimate measurement accuracy (UMA) for several shape features. The temporal sequences of these features, i.e., the basis of the motion analysis, were also examined.

Three different segmentation techniques were compared. The first one was the Laplacian of Gaussian (LoG) filtering. The four-neighbor Laplacian mask was used. The positive response was labeled as the cell and the negative response was labeled as the background. The second one was the local thresholding method of Eikvil *et al.* where the size of the large window was 29×29 and the size of the small window was 9×9. The third one was the modified version of Bernsen's dynamic thresholding method. In this method I used a 21×21 window, a contrast threshold $T_C = 15$, and an additional parameter $\alpha = 0.65$. (See Chapter 5 and 6 for the descriptions of these methods and the discussion about the parameters.) These methods were quantitatively evaluated since their results were good by visual examination. The Gaussian filters of different standard deviations (σ) were applied for noise reduction before the segmentation.

The test data were the three image sequences which were manually segmented by the three different persons (See Chapter 3).

11.2.1 Shape Difference Measures

Since we have three different manual segmentation results for each cell, the method of Levine and Nazif [LN82] has to be slightly modified. We consider that a reference image has three regions: a background (B), an object (O) and an uncertain region. The background region consists of the pixels classified as the background by all three manual segmentations, the object region consists of the pixels classified as the object by all the manual segmentations, and the uncertain region consists of all the other pixels. The automatically segmented image has two regions: a background ($\hat{B}$) and an object ($\hat{O}$) region. The under-merging error (UM) is then defined by

$$UM = \text{area}(O \backslash \hat{O})/A \tag{11.2}$$

Segment. Method	Gauss. σ	UM		OM		DM	
		mean	std.	mean	std.	mean	std.
LoG	2.0	0.0274	0.0254	0.0914	0.1258	0.1188	0.1192
	3.0	0.0191	0.0221	0.0597	0.0715	**0.0788**	**0.0707**
	4.0	0.0076	0.0145	0.1024	0.0872	0.1150	0.0839
Eikvil	1.0	0.0681	0.0598	0.0088	0.0198	0.0769	0.0560
	2.0	0.0511	0.0466	0.0184	0.0323	**0.0695**	**0.0461**
	3.0	0.0389	0.0379	0.0390	0.0545	0.0779	0.0556
	4.0	0.0290	0.0302	0.0572	0.0789	0.0862	0.0766
Bernsen	1.0	0.0447	0.0389	0.0180	0.0202	0.0627	0.0499
	2.0	0.0242	0.0347	0.0384	0.0389	**0.0626**	**0.0405**
	3.0	0.0174	0.0266	0.0821	0.0759	0.0995	0.0729
	4.0	0.0098	0.0202	0.1009	0.0864	0.1107	0.0801

Table 11.1: *The means and standard deviations of the measures of differences between the correct segmentation and the automatic segmentation results.*

and the over-merging error (OM) is

$$OM = \text{area}(B \backslash \hat{B})/A \tag{11.3}$$

where the difference operation $R_1 \backslash R_2$ is defined by

$$R_1 \backslash R_2 = \{p | p \in R_1,\, p \notin R_2\} \tag{11.4}$$

area(R) is the area of region R, and A is the average of the areas obtained by the three manual segmentations. A total difference measure (DM) is

$$DM = UM + OM \tag{11.5}$$

In this evaluation method, we assume that the correct cell boundary is the uncertain region. When a cell boundary obtained by an automatic segmentation is inside the uncertain region, we say that the segmentation is correct, and UM, OM and DM are all zero. When the cell boundary is inside the background region B, then we obtain an over-merge error. When the cell boundary is inside the object region O, then we have an under-merge error.

I used this method to evaluate the three segmentation algorithms combined with the Gaussian smoothing of standard deviations $\sigma = 1.0$, 2.0, 3.0 and 4.0. The LoG method was very sensitive to the noise and broke down when $\sigma = 1.0$. The UM, OM and DM values were computed for each cell object. The means and the standard deviations of the UM, OM and DM values are shown in Table 11.1. The smallest mean and standard deviation of the DM values of each segmentation algorithm are indicated in boldface. According to the DM, the modified Bernsen's method with Gaussian smoothing $\sigma = 2.0$ gave the best results. More analysis of the results will be given in Section 11.3.

11.2.2 Accuracy of Shape Features

Further evaluation was done by examining the shape features. The area (A), circularity (C), orientation (Θ), elongation (EL) and radius of gyration (R) were computed from the results of the manual segmentation and the results of the automatic segmentation using different initial segmentation techniques and the Gaussian smoothing of different σs. (See Chapter 8 for descriptions of these shape features and Chapter 7 for the segmentation techniques.)

I used Equation (11.1) to compute the UMA values. Since there are three reference images for each cell, the correct feature value x is then computed by averaging the three feature values. For the area and the radius of gyration, the UMA was divided by the average of the feature values estimated from the three manual segmentation results, i.e.

$$UMA = \frac{|x - \hat{x}|}{x} \tag{11.6}$$

so that the error was represented as the percentage of the correct value. Table 11.2 shows the means and the standard deviations of the UMA values for the five features. In each column of the table, the smallest value obtained through an automatic segmentation is indicated in boldface. The UMA values were also computed for the manual segmentation results to show how a feature obtained from a manual segmentation differs from the average of the three. (Note that, as an alternative, one could use the *leave-one-out* technique, i.e., compute the average from the two other manual segmentation results.) They are given in the last row of the table for comparison. We see that the modified Bernsen's method gave the least errors for many features, and the least UMA means and standard deviations obtained from automatic segmentation are not much larger than the values from the manual segmentation.

To see how the automatic segmentation methods estimate the change of the features, I show the time sequences of the features in Figures 11.1 to 11.5. The five features of four cells were examined. The segmentation methods were the LoG method with $\sigma = 4.0$, the modified Bernsen's method with Gaussian filtering of $\sigma = 2.0$ and the method of Eikvil *et al.* with Gaussian filtering of $\sigma = 2.0$. I chose these σ values since they were good choices according to the UMA values. We see that the feature curves obtained from the automatic segmentation results generally have the same trends as those obtained from the manual segmentation.

In motion analysis, one may have to consider the correlation between two shape features. The correlation can be computed from the joint distribution of the features, which can be illustrated by a scatter plot. The scatter plot of R^2 and A is shown in Figure 11.6. We see that the values are distributed near and over a diagonal. The two features are therefore correlated. The diagonal is $R^2 = A/(2\pi)$ which is obtained for circles. For a given value of area A, a circle has the smallest moment of inertia I, and thus has the smallest radius of gyration $R = \sqrt{I/A}$.

Segment. Method	Gauss. σ	Area		Circularity		Orientation		Elongation		Radius of G.	
		mean	std.	mean	std.	mean	std.	mean	std.	mean	std.
LoG	2.0	0.1890	0.2094	0.1443	0.1135	8.8044	13.9664	0.2665	0.2448	0.1245	0.1273
	3.0	**0.1089**	0.1029	0.0554	0.0466	4.6270	5.1511	0.1976	0.1950	0.0747	0.0689
	4.0	0.1823	0.1220	**0.0410**	**0.0330**	2.4665	1.9379	0.1543	0.1583	0.1060	0.0704
Eikvil	1.0	0.1728	0.1128	0.0705	0.0556	6.3479	8.2841	0.1269	0.1313	0.0942	0.0534
	2.0	0.1442	0.0915	0.0838	0.0588	2.2054	1.8103	0.1383	0.1300	0.0784	0.0486
	3.0	0.1218	0.0983	0.1072	0.0575	6.6584	6.9033	0.1619	0.1420	0.0620	0.0465
	4.0	0.1688	0.1195	0.1215	0.0593	7.8542	7.6171	0.1917	0.1605	0.0801	0.0632
Bernsen	1.0	0.1202	0.0960	0.0649	0.0384	5.0791	5.6710	0.1076	0.1073	0.0579	0.0469
	2.0	0.1116	**0.0857**	0.0793	0.0548	**2.0972**	**1.6143**	**0.0745**	**0.0586**	**0.0547**	**0.0405**
	3.0	0.1544	0.1110	0.1055	0.0570	6.1646	6.2871	0.1619	0.1369	0.0619	0.0523
	4.0	0.1692	0.1201	0.1203	0.0584	7.1312	7.2541	0.1908	0.1597	0.0764	0.0601
Manual		0.0807	0.0642	0.0341	0.0266	1.9074	1.7093	0.0779	0.0703	0.0386	0.0312

Table 11.2: *The means and standard deviations of the UMA values.*

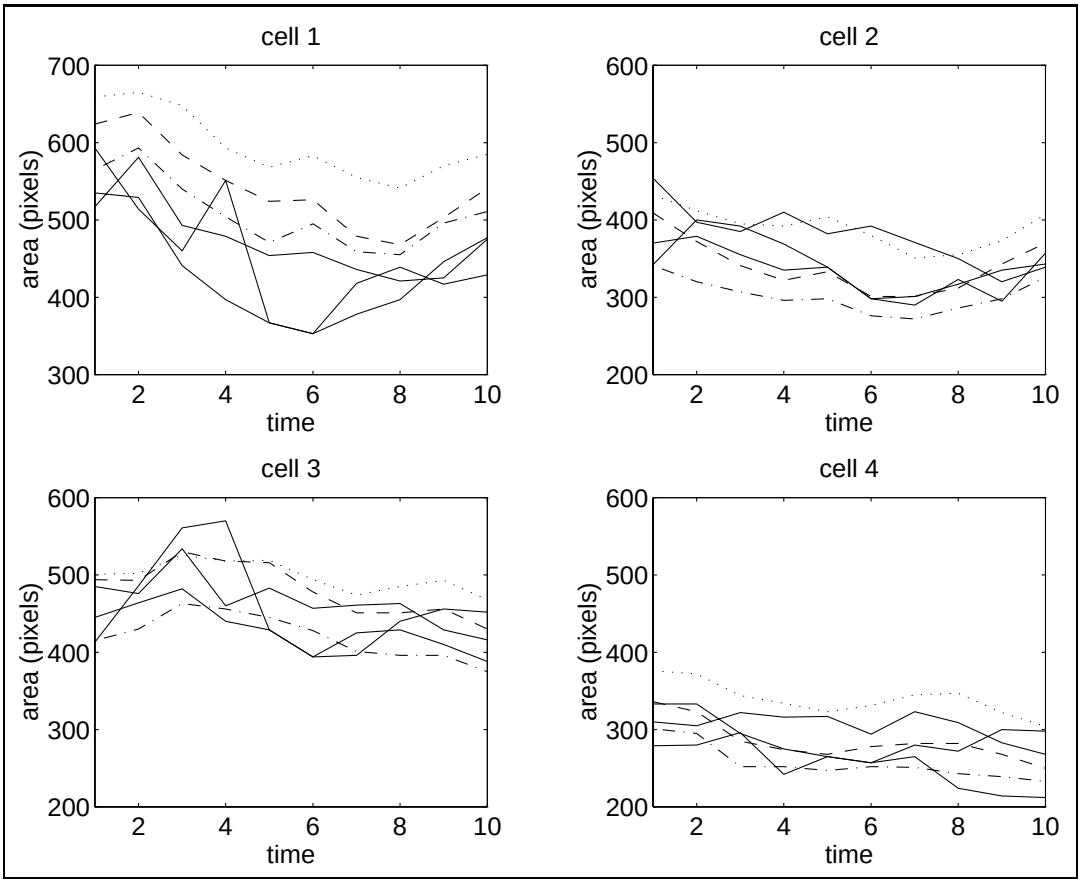


Figure 11.1: *Time sequences of area obtained from the manual segmentation results (solid line), automatic segmentation using the modified Bernsen's method (dashed line), the LoG method (dotted line), and the method of Eikvil* et al. *(dotted-dashed line).*

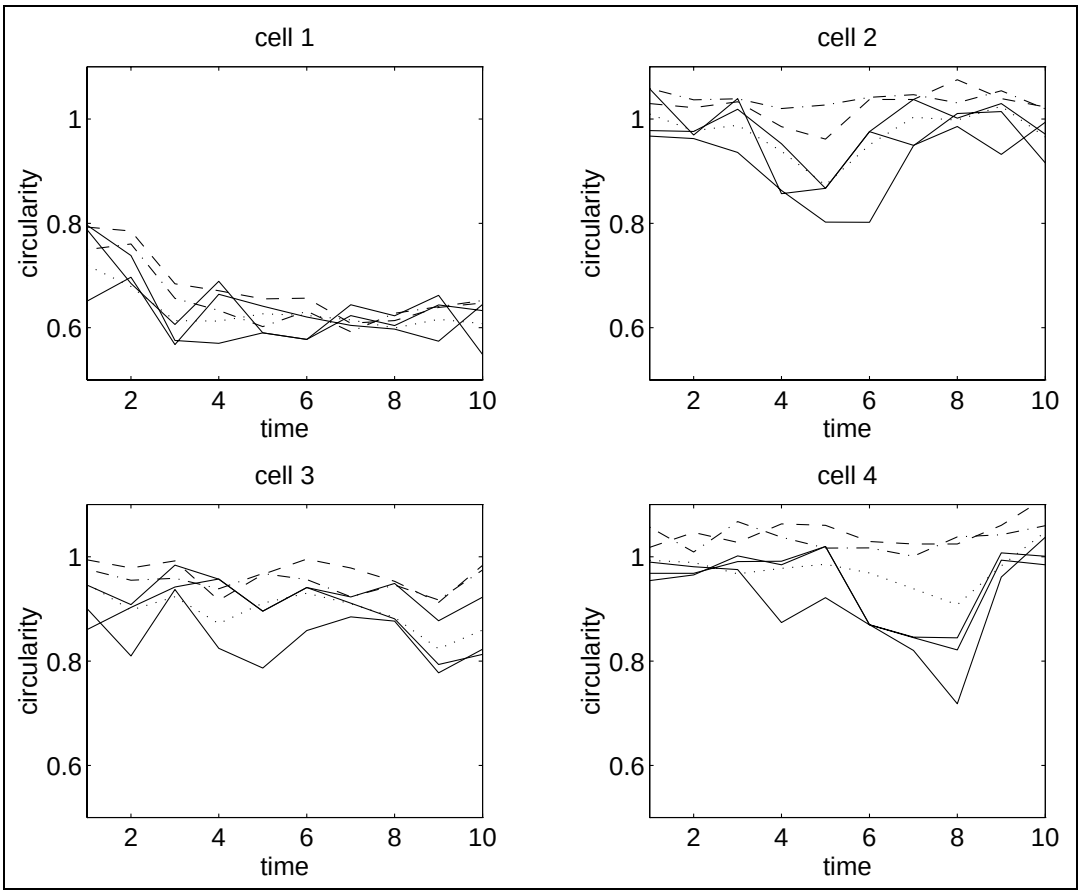


Figure 11.2: *Time sequences of circularity obtained from the manual segmentation results (solid line), automatic segmentation using the modified Bernsen's method (dashed line), the LoG method (dotted line), and the method of Eikvil* et al. *(dotted-dashed line).*

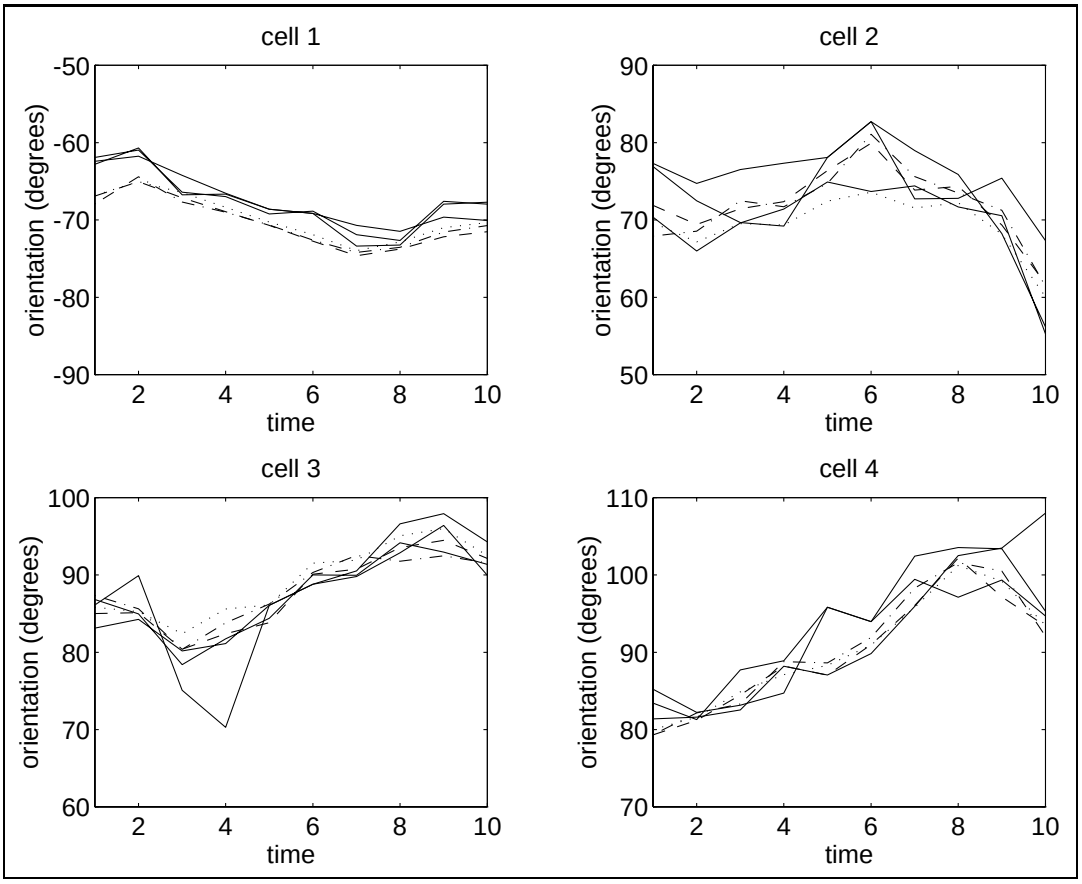


Figure 11.3: *Time sequences of orientation obtained from the manual segmentation results (solid line), automatic segmentation using the modified Bernsen's method (dashed line), the LoG method (dotted line), and the method of Eikvil* et al. *(dotted-dashed line).*

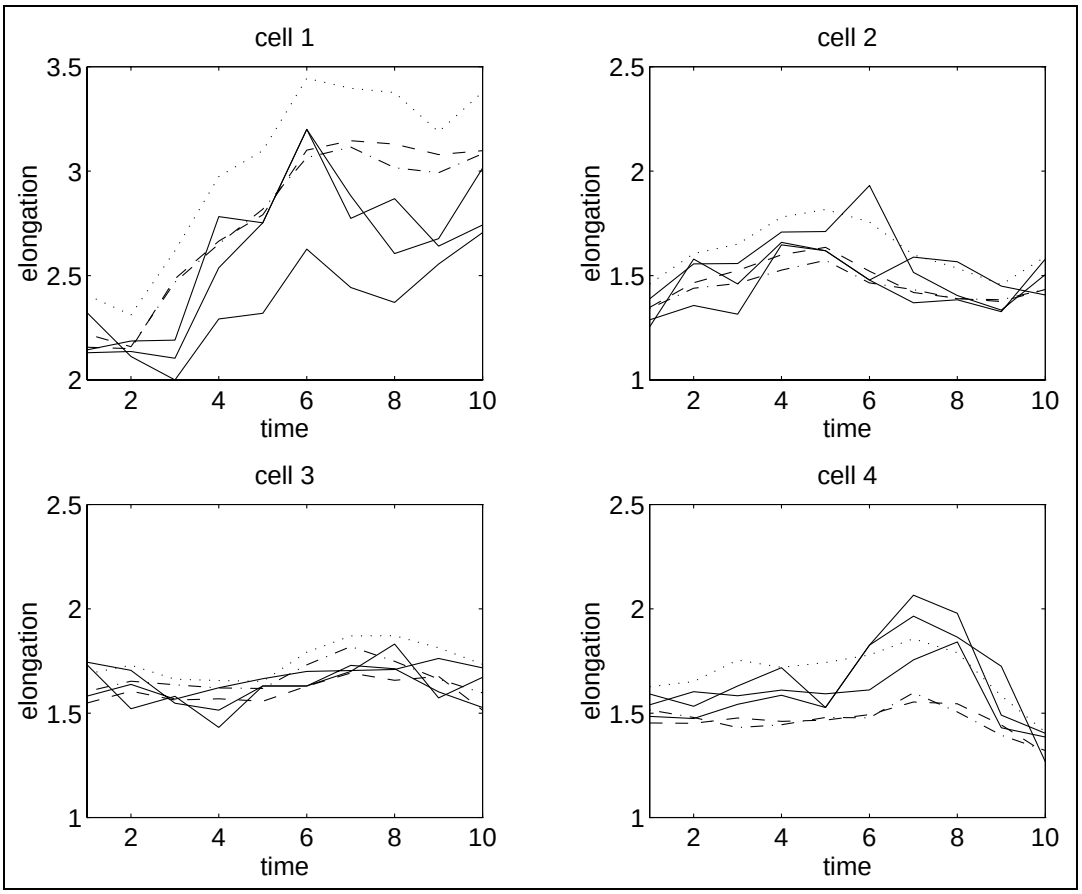


Figure 11.4: *Time sequences of elongation obtained from the manual segmentation results (solid line), automatic segmentation using the modified Bernsen's method (dashed line), the LoG method (dotted line), and the method of Eikvil* et al. *(dotted-dashed line).*

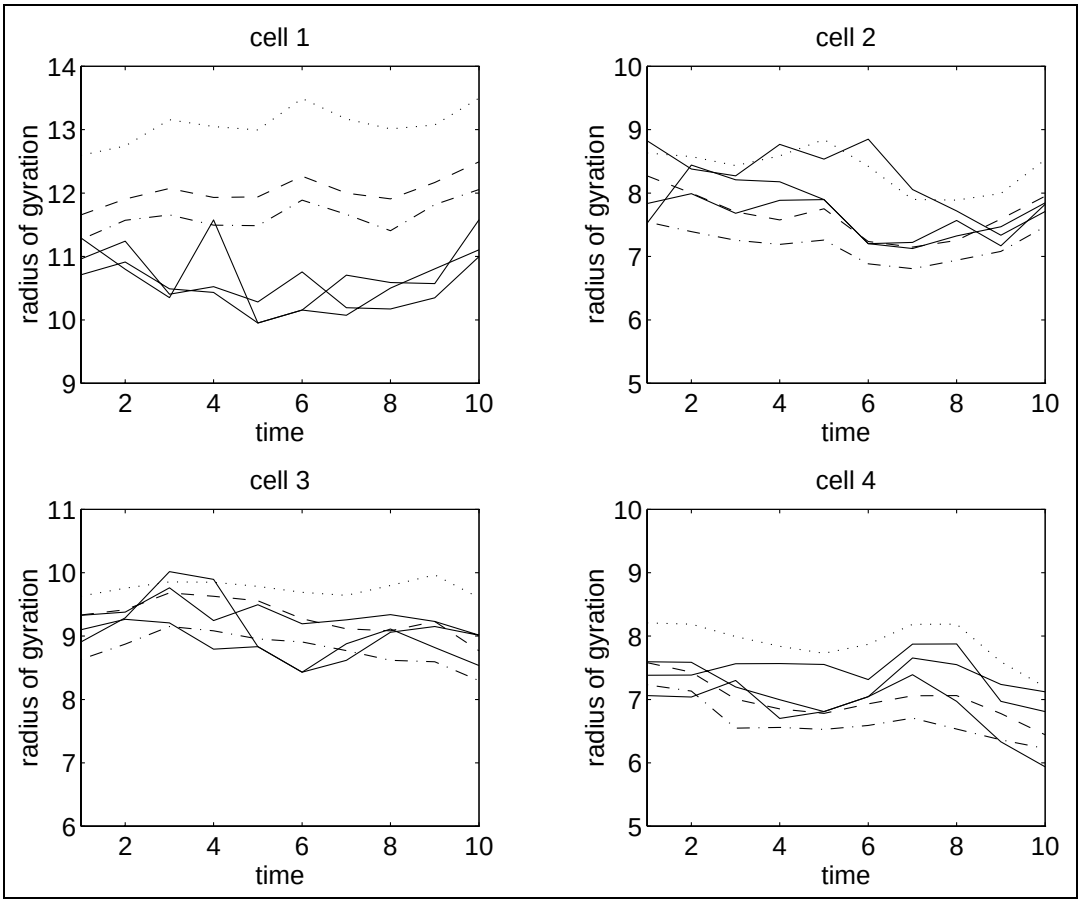


Figure 11.5: *Time sequences of radius of gyration obtained from the manual segmentation results (solid line), automatic segmentation using the modified Bernsen's method (dashed line), the LoG method (dotted line), and the method of Eikvil* et al. *(dotted-dashed line).*

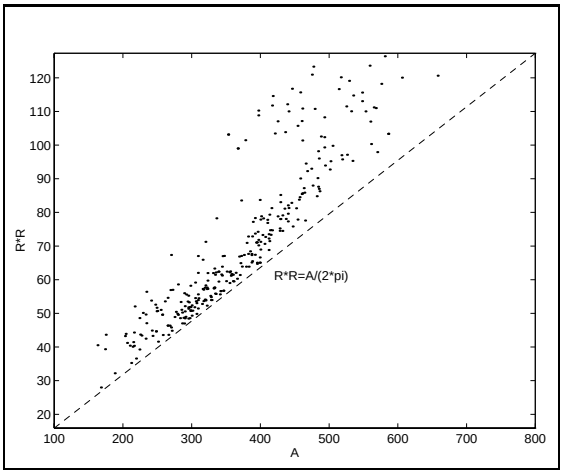


Figure 11.6: *Scatter plot of R^2 and A.*

11.3 Discussion

Segmentation is to divide an image into meaningful regions. A lot of segmentation methods have been proposed for different types of images [FM81, HS85, PP93]. One important fact is that no general theory about segmentation exists. As a result, segmentation has traditionally been an *ad hoc* process, and the selection of a segmentation method is often based on testing and evaluation.

I evaluated the segmentation by using a supervised method, where the manual segmentation results were used as reference segmentation and the differences between the reference segmentation and the automatic segmentation were measured.

We can see from the results that the noise reduction has a clear effect on the segmentation. The Gaussian smoothing can generally make the objects larger. This can be seen from Table 11.1 where the OM increases and the UM decreases when the degree of blur increases. Note that the modified Bernsen's method is able to compensate this effect since we can adjust the value of α. Different segmentation methods may require different degrees of smoothing. The Laplacian was more sensitive to the noise than the two thresholding methods, and could not work properly when the smoothing parameter σ was 1.0. From Table 11.1 and 11.2 we can see that the two thresholding methods worked well when $\sigma = 2.0$. The LoG method gave the best estimation of the object size when $\sigma = 3.0$ (according to the values of DM, area and radius of gyration). However, the other shape features were best estimated by the LoG method when $\sigma = 4.0$. A proper smoothing is therefore important to obtain good results.

Shape features may be correlated. I have illustrated the correlation between the area and the radius of gyration. Both of the features relate to the size of the object. The circularity and the elongation should also be correlated. Unlike the elongation, the circularity depends on the smoothness of the boundary. From Table 11.2 we see that the thresholding methods gave good estimations of the circularity when $\sigma = 1.0$. The error increases when the degree of blur increases. The reason might

be that the thresholding methods produced smooth boundaries. The LoG method, however, produced jagged boundaries and gave good estimation of the circularity when the degree of blur was large. From Figure 11.2 we can see that the values of the circularity are sometimes larger than 1. This problem was discussed in Chapter 9. The circularity can be over-estimated for small blob-like objects.

From the feature curves in Figure 11.1 to 11.5 we can see that the error in feature estimation varies from cell to cell. I therefore computed the standard deviations of the errors. An error with small standard deviation (e.g. systematic over-estimation or under-estimation) should be less serious than one with large standard deviation. We can see from Table 11.1 and 11.2 that a small error standard deviation usually associates with a small error value.

I illustrated the time sequences of the five shape features. From the feature curves we can see that the automatic segmentation can generally estimate the trends of the changes of the features. Quantitative evaluation of the feature curves will require the use of the motion features discussed in Chapter 8. This again requires a longer time sequence, since most motion features are dependent on the critical points in the feature curve.

The evaluation was based on the manual segmentation results. As discussed in Chapter 3, it is possible that the manual segmentation has a biased error due to a mistaken understanding. The most possible error can be in the size of the cells. As an advantage, the modified Bernsen's method can be adapted to a new set of manual segmentation by changing the parameter α. This parameter can even be optimally determined through a learning strategy.

To give an overall evaluation of the three segmentation techniques, I can say that the modified Bernsen's method with Gaussian filtering of $\sigma = 2.0$ is the best, since this method gave the least DM value, and the smallest errors for many shape features. Further more, this method is able to change the size of the object by adjusting the parameter α. The current value of α was determined according to visual examination of the results. Better results can be obtained if one determines the optimal value of α according to the manual segmentation.

There are some other segmentation methods which produced good results according to visual examination, e.g. the MGP method and the dynamic thresholding using a CRCGD. The purpose of the CRCGD is to use an adaptive region instead of a fixed window in Bernsen's method. The result of the CRCGD method will not much different from the result of Bernsen's method using an optimal window, but will be more robust when the spatial resolution is changing.

11.4 Conclusion

In this chapter, I gave a review of segmentation evaluation methods, and proposed a supervised method to evaluate the results of three segmentation techniques for the blood cell image sequences. The manual segmentation results were used as reference segmentation in the evaluation. From this evaluation I found that the

modified Bernsen's method with a Gaussian filtering of $\sigma = 2.0$ gave the best results according to many criteria. The difference between the results of this method and the reference segmentation is not much larger than the differences between the three different manual segmentation results.

I also examined the effect of noise reduction, and concluded that a proper noise reduction was also important. However, the choice of a smoothing method depended on the segmentation method to be used. Properties of the shape features were discussed. The correlation between the area and the radius of gyration was illustrated.

Chapter 12

Summary and Discussion

12.1 Summary

It has been found that the intracellular calcium signal plays an important role in many activities of white blood cells. The relation between the calcium concentration and the cell motion thus becomes a research interest in cell physiology. In the Department of Physiology, University of Oslo, a microscopy fluorescence cell imaging system has been developed. The imaging system produces fluorescence image sequences of living white blood cells. From the image sequences, the concentration of intracellular calcium ions can be measured. The objective of this work is to develop an image analysis system to measure the motion of white blood cells from the image sequences, so that the relation between the calcium concentration and the cell motion can be studied.

The cell motion analysis system should consist of three parts: image segmentation, feature extraction and feature analysis. The thesis mainly concentrates on the first two parts.

Image smoothing is used as a preprocess to the segmentation and its effect is examined for different segmentation algorithms. The choice of smoothing method depends on the segmentation method to be used. The Laplacian edge detection is more sensitive to noise than many thresholding methods.

Many basic 2D segmentation methods have been applied to the cell images. Some edge-based methods (MGP and LoG) are implemented in a pixel classification manner, which simplifies the segmentation process. Three types of gray level thresholding methods have been discussed. Global thresholding does not work properly for the cell images since one can not find a single threshold which is suitable for all the pixels. Some local and dynamic thresholding methods have given relatively good results. A modified version of Bernsen's method with an α-parameter gives better results than the original one. A new dynamic thresholding method, using a consistent region (CRCGD) instead of a window of fixed size, has been proposed. Compared to early methods, the new method is more robust when the size of the object is changing.

A spatial-temporal segmentation system (STSS) is proposed. This two-pass system first applies an initial segmentation in which the pixels are classified as cell and background pixels. Region labeling, correction and cell tracking are then done to give a final segmentation. The system is semi-automatic, requiring an interactive segmentation in the first frame of a sequence. Segmentation is automatically performed in the rest of the frames. Some 2D segmentation methods, including edge detection and gray level thresholding, have been used in the initial segmentation, and compared by a supervised evaluation. Noise reduction is also tested in the evaluation. According to the evaluation, the modified Bernsen's method after a Gaussian filtering with standard deviation 2.0 gives the best results for many criteria. This method becomes more robust when the CRCGD approach is incorporated.

After the segmentation, shape features are estimated from discrete regions, and cell motion is then measured by using the shape features. A comprehensive list of shape features is provided. Properties of many shape features are discussed. Estimators of area and perimeter are evaluated. In this evaluation, the accuracy of circularity is used as one of the measure. A new and improved method is developed for fast computation of geometric moments, from which many shape features can be computed. A discrete version of Green's theorem is used to compute the area and the other moments.

Methods of evaluating the output of an image segmentation system are reviewed. A supervised method based on the shape features is used to evaluate the image analysis system. Manual segmentation is applied to obtain reference segmentation which is used in the evaluation.

12.2 Discussion

In this section, I discuss the relation between this work and the related works of others, and suggest some furture works.

The fluorescence imaging has been used as a method to measure the concentration of the intracellular calcium ions [SWB92]. However, automatic segmentation of fluorescence cell image sequences has not been reported. In the Department of Physiology, University of Oslo, an interactive segmentation system has been used in order to measure the calcium concentration. This interactive system first locates a rectangular region inside a cell, and then tracks this region from frame to frame. In this thesis, a novel semi-automatic system is proposed for the segmentation of fluorescence cell image sequences. Various basic techniques for accurate and efficient segmentation are also discussed, and the test results are provided.

Cell motion analysis has been studies by many researchers, especially by a group of researchers led by Levine [LYNB80, LNY83, NL86]. They studied how the motion can be described by using shape features. In contrast, I discuss the accurate and efficient estimation of the shape features. Many area and perimeter estimators are tested by applying to circles. The results should be useful to many blob-like objects met in practice. A new algorithm is proposed for fast and accurate computation of

geometric moments. This algorithm should be useful in many other image analysis and pattern recognition tasks.

An obvious future work is to apply various shape features to the cell objects, and to analysis the cell motion by using the shape features. This work is the third part of the image analysis system, and should be done by combining the expertise in physiology and informatics. The segmentation system can also be improved. To optimize parameters should be one approach to the improvement. According the evaluation, the modified Bernsen's method with a Gaussian filtering gives the best results for many criteria. Three important parameters in this method are the standard deviation σ of the Gaussian filtering, the window size and the α-parameter in the modified Bernsen's method. The window can be replaced by a CRCGD. How to determine σ adaptively according to image quality is an interesting topic. Note that in Chapter 4 we have discussed a possible approach using topographical structure features in image gray level surface. To determine the optimal value of the α-parameter is also interesting. This might be done by a supervised learning, as discussed in Chapter 6. I have tested three segmentation methods. However, there are a lot of other possibilities. Many of them are reviewed in this thesis. To test more methods should also be interesting.

Appendix A

XITE: an Image Processing Software Parkage

XITE stands for **X**-based **I**mage processing **T**ools and **E**nvironment, is a software package developed in the Image Processing Laboratory (BLAB), Department of Informatics, University of Oslo. It consists of:

- File format BIFF (BLAB Image File Format)
- X11-based BIFF image display program
- More than 140 C-programs for image processing and analysis
- More than 200 C-subroutines for image processing and analysis
- On-line documentation of all routines and programs
- Some images on BIFF format

The file format BIFF is a raster format for two-dimensional and three-dimensional images. In case of 3D, the image is viewed as a sequence of frames (bands). There are programs to convert between BIFF and many other image formats. More about the BIFF format and the XITE can be found in BLAB report [Lø90].

The XITE is a free software. The source codes and the documents can be obtained through anonymous `ftp` from `ftp.ifi.uio.no (129.240.82.2)` under directory `pub/blab/xite`.

Bibliography

[ADM81] J. K. Aggarwal, L. S. Davis, and W. N. Martin. Correspondence processes in dynamic scene analysis. *Proc. IEEE*, **69**(5):562–572, 1981.

[AF67] M. Alonso and E. J. Finn. *Fundamental university physics*, volume 1. Addison-Wesley, 1967.

[AGL83] F. Albregtsen, Ø. Graf, and T. Lønnestad. Adaptive noise filtering of solar photographic uv spectra: an evaluation. In *Proc. 6th Scandinavian Conf. Image Analysis*, pages 67–91, 1983.

[AHH88] A. Abo-Zaid, O. R. Hinton, and E. Horne. About moment normalization and complex moment descriptors. In *Proc. 4th Int. Conf. Pattern Recognition*, pages 399–407, 1988.

[Alb93] F. Albregtsen. Non-parametric histogram thresholding methods – error versus relative object area. In *Proc. 8th Scadinavian Conf. Image Analysis*, pages 273–280, Tromsø , Norway, 1993.

[Ber86] J. Bernsen. Dynamic thresholding of grey-level images. In *Proc. 8th Int. Conf. Pattern Recogn.*, pages 1251–1255, Paris, 1986.

[Ber87] F. Bergholm. Edge focusing. *IEEE Trans. Pattern Anal. Machine Intell.*, **9**(6):726–741, 1987.

[BHM87] A. C. Bovik, T. S. Huang, and D. C. Munson, Jr. The effect of median filtering on edge estimation and detection. *IEEE Trans. Pattern Anal. Machine Intell.*, **9**(2):181–194, 1987.

[BL93] R. M. Berne and M. N. Levy. *Physiology*. Mosby-Year Book, St. Louis, Missouri, 3 edition, 1993.

[Bri89] A. D. Brink. Grey-level thresholding of images using a correlation criterion. *Pattern Recognition Letters*, **9**:335–341, 1989.

[Bro94] M. Bro-Nielsen. Parallel implementation of active nets. Technical report, Institute of Mathematical Modelling, Technical Univ. of Denmark, Lyngby, Denmark, 1994.

[Bur83] P. J. Burt. Fast algorithms for estimating local image properties. *Comput. Vision Graph. Image Process.*, **21**:368–382, 1983.

[BWBD86] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**(1):26–33, 1986.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**(6):679–698, 1986.

[CdB90] T. G. Campbell and J. M. H. du Buf. A quantitative comparison of median-based filters. In *Visual Communications and Image Processing, Proc. SPIE, vol. 1360*, pages 176–187, 1990.

[CHM87] J. S. Chen, A. Huertas, and G. Medioni. Fast convolution with laplacian-of-gaussian masks. *IEEE Trans. Pattern Anal. Machine Intell.*, **9**(4):584–590, 1987.

[CK72] C. K. Chow and T. Kaneko. Automatic detection of the left ventricle from cincangiograms. *Computers and Biomedical Research*, **5**:388–410, 1972.

[Coh91] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, **53**(2):211–218, 1991.

[CRK90] A. C. F. Colchester, R. T. Ritchings, and N. D. Kodikara. Image segmentation using maximum gradient profiles orthogonal to edges. *Image and Vision Computing*, **8**(3):211–217, 1990.

[CY83] R. T. Chin and C.-L. Yeh. Quantitative evaluation of some edge-preserving noise-smoothing techniques. *Comput. Vision Graph. Image Process.*, **23**:67–91, 1983.

[Dan81] P.-E. Danielsson. Getting the median faster. *Comput. Graph. Image Process.*, **17**:71–78, 1981.

[Dav75] L. S. Davis. A survey of edge detection techniques. *Comput. Graph. Image Process.*, **4**:248–270, 1975.

[Dav87] E. R. Davies. Design of optimal gaussian operators in small neighbourhoods. *Image and Vision Computing*, **5**(3):199–205, 1987.

[dBC90] J. M. H. du Buf and T. G. Campbell. A quantitative comparison of edge-preserving smoothing techniques. *Signal Process.*, **21**:289–301, 1990.

[DBN92] M. Dai, P. Baylou, and M. Najim. An efficient algorithm for computation of shape moments from run-lenght codes or chain codes. *Pattern Recognition*, **25**(10):1119–1128, 1992.

[dGKVV92] C. N. de Graaf, A. S. E. Koster, K. L. Vincken, and M. A. Viergever. Task-directed evaluation of image segmentation methods. In *Proc. 11th Int. Conf. Pattern Recogn. C*, pages 219–222, 1992.

[DH72] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Com. ACM*, **15**(1):11–15, 1972.

[DH73] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis.* Wiley-Interscience, 1973.

[DM85] K. A. Dunkelberger and O. R. Mitchell. Contour tracking for precision measurement. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 22–27, St. Louis, 1985.

[DOSA91] J. S. Duncan, R. L. Owen, L. H. Staib, and P. Anandan. Measurement of non-rigid motion using contour shape descriptors. In *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, pages 318–324, 1991.

[DR79] C. R. Dyer and A. Rosenfeld. Thinning algorithms for gray-scale pictures. *IEEE Trans. Pattern Anal. Machine Intell.*, **1**(1):88–89, 1979.

[DS66] N. R. Draper and H. Smith. *Applied regression analysis.* Wiley, 2 edition, 1966.

[DS86] L. Dorst and A. W. M. Smeulders. Best linear unbiased estimators for properties of digitized straight lines. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**:276–282, 1986.

[DS87] L. Dorst and A. W. M. Smeulders. Length estimators for digitized contours. *Comput. Vision Graph. Image Process.*, **40**:311–333, 1987.

[DS90] P.-E. Danielsson and O. Seger. Generalized and separable sobel operators. In H. Freeman, editor, *Machine Vision for Three-Dimensional Scenes*, pages 347–379. Academic Press, 1990.

[EP82] C. H. Edwards, Jr. and D. E. Penny. *Calculus and analytic geometry.* Prentice Hall, 3 edition, 1982.

[ETM91] L. Eikvil, T. Taxt, and K. Moen. An adaptive method for binarization of gray level images. In *Proc. NOBIM Norwegian National Conf. Image Processing and Pattern Recognition*, pages 123–131, 1991.

[FM81] K. S. Fu and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, **13**:3–16, 1981.

[Fre70] H. Freeman. Boundary encoding and processing. In B. S. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics*, pages 241–266. Academic Press, 1970.

[Fre78] H. Freeman. Shape description via the use of critical points. *Pattern Recognition*, **10**(3):159–166, 1978.

[FS93] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, **26**(1):167–174, 1993.

[FvDFH90] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer graphics: principles and practice.* Addison-Wesley, 2 edition, 1990.

[FYC93] C.-W. Fu, J.-C. Yen, and S. Chang. Calculation of moment invariants via hadamard transform. *Pattern Recognition*, **26**(2):287–294, 1993.

[GAL88] Ø. Graf, F. Albregtsen, and T. Lønnestad. An evaluation of adaptive noise filters applied on solar photograghic uv-spectra. BLAB Report no. 30, Dept. of Informatics, Univ. of Oslo, 1988.

[GCR92] L. D. Griffin, A. C. F. Colchester, and G. P. Robinson. Scale and segmentation of grey-level images using maxmimum gradient paths. *Image and Vision Computing*, **10**(6):389–402, 1992.

[GD88] C. R. Giardina and E. R. Dougherty. *Morphological methods in image and signal processing.* Prentice Hall, 1988.

[GHN90] M. Gabbouj, P. Haavisto, and Y. Neuvo. Recent advances in median filtering. In *Proc. Bilkent Int. Conf. New Trends in Communication, Control and Signal Processing*, pages 1080–1094, 1990.

[GJK+61] V. E. Giuliano, P. E. Jones, G. E. Kimball, R. F. Meyer, and B. A. Stein. Automatic pattern recognition by a gestalt method. *Inform. Control*, **4**:332–345, 1961.

[GM93] S. Ghosal and R. Mehrotra. Orthogonal moment operators for subpixel edge detection. *Pattern Recognition*, **26**(2):295–306, 1993.

[GP93] J. M. Gauch and S. M. Pizer. Multiresolution analysis of ridges and valleys in grey-scale images. *IEEE Trans. Pattern Anal. Machine Intell.*, **15**(6):635–646, 1993.

[Gra71] S. B. Gray. Local properties of binary images in two dimensions. *IEEE Trans. Computers*, **20**(5):551–561, 1971.

[GW92] R. C. Gonzalez and R. E. Woods. *Digital image processing.* Addison-Wesley, 1992.

[Har84] R. M. Haralick. Digital step edges from zero crossing of second directioanl derivatives. *IEEE Trans. Pattern Anal. Machine Intell.*, **6**:58–68, 1984.

[Hat86] M. Hatamian. A real-time two-dimensional moment generating algorithm and its single chip implementation. *IEEE Trans. Acoust. Speech Signal Process.*, **34**(3):546–553, 1986.

[HGP87] C. H. Hayden, R. C. Gonzalez, and A. Ploysongsang. A temporal edge-based image segmentor. *Pattern Recognition*, **20**(3):281–290, 1987.

[HH81] T. S. Huang and Y. P. Hsu. Image sequence enhancement. In T. S. Huang, editor, *Image Sequence Analysis*, chapter 4. Springer-Verlag, 1981.

[HKP91] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the theory of neural computation.* Addison Wesley, 1991.

[HL89] D. W. Hosmer, Jr. and S. Lemeshow. *Applied logistic regression.* Wiley, 1989.

[HM86] A. Huertas and G. Medioni. Detection of intensity changes using laplacian-gaussian masks. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**(5):651–664, 1986.

[HR87] S. L. Hurt and A. Rosenfeld. Noise reduction in three-dimensional digital images. *Pattern Recognition*, **6**:155–162, 1987.

[HS85] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Comput. Vision Graph. Image Process.*, **29**:100–132, 1985.

[HS93] R. M. Haralick and L. G. Shapiro. *Computer and robot vision*, volume 1. Addison-Wesley, 1993.

[HSZ87] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Trans. Pattern Anal. Machine Intell.*, **9**(4):532–550, 1987.

[Hu62] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, **8**:179–187, 1962.

[Hue71] M. F. Hueckel. An operator which locates edges in digitized pictures. *J. ACM*, **18**:113–125, 1971.

[HYT79] T. S. Huang, G. J. Yang, and G. Y. Tang. A fast two-dimensional median filtering algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, **27**(1):13–18, 1979.

[IK88] J. Illingworth and J. Kittler. A survey of the hough transform. *Comput. Vision Graph. Image Process.*, **44**:87–116, 1988.

[Jai81] R. Jain. Dynamic scene analysis using pixel-based processes. *IEEE Comput.*, pages 12–18, 1981.

[Jai89] A. K. Jain. *Fundamentals of digital image processing.* Prentice-Hall, 1989.

[JB91] X. Y. Jiang and H. Bunke. Simple and fast computation of moments. *Pattern Recognition*, **24**(8):801–806, 1991.

[JB94] H. Jiang and E. Bølviken. A general parameter updating approach in image classification. to appear in Proc. 12th Int. Conf. Pattern Recogn., 1994.

[JR75] E. G. Johnston and A. Rosenfeld. Digital detection of pits, peaks, ridges, and ravines. *IEEE Trans. Sys. Man Cyb.*, **5**:472–480, 1975.

[JTS+91] M. E. E. Jaconi, J. M. Theler, W. Schlegel, R. D. Appel, S. D. Wright, and P. D. Lew. Multiple elevations of cytosolic-free ca^{2+} in human neutrophils: initiation by adherence receptors of the integrin family. *J. Cell Biol.*, **112**(6):1249–1257, 1991.

[Kai87] J. F. Kaiser. On the fast generation of equally spaced values of the gaussian function $a\exp(-at*t)$. *IEEE Trans. Acoust. Speech Signal Process.*, **35**:1480–1481, 1987.

[KAN91] L. Koskinen, J. Astola, and Y. Neuvo. Analysis of noise attenuation in morphological processing. In *Proc. SPIE, Vol. 1451, Nonlinear Image Processing II*, pages 102–113, 1991.

[KCWL90] K.-H. Krause, K. P. Campbell, M. J. Welsh, and D. P. Lew. The calcium signal and neutrophil activation. *Clinical Biochemistry*, **23**:159–166, 1990.

[KI86] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recognition*, **19**(1):41–47, 1986.

[Kim82] C. E. Kim. Digital convexity, straightness, and convex polygons. *IEEE Trans. Pattern Anal. Machine Intell.*, **4**(6):618–626, 1982.

[Koe84] J. J. Koenderink. The structure of images. *Biol. Cybern.*, **50**:363–370, 1984.

[KSW85] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Comput. Vision Graph. Image Process.*, **29**:273–285, 1985.

[Kul77] Z. Kulpa. Area and perimeter measurement of blobs in discrete binary pictures. *Comput. Graph. Image Process.*, **6**:434–451, 1977.

[KWA+80] H. U. Keller, P. C. Wilkinson, M. Abercrombie, E. L. Becker, J. G. Hirsch, M. E. Miller, W. S. Ramsey, and S. H. Zigmond. A proposal for the definition of terms related to locomotion of leukocytes and other cells. *Bulletin of the World Health Organization*, **58**(3):505–509, 1980.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *Int. J. of Computer Vision*, **2**:321–331, 1988.

[Leu91] J.-G. Leu. Computing a shape's moments from its boundary. *Pattern Recognition*, **24**(10):949–957, 1991.

[Lev85] M. D. Levine. *Vision in man and machine*. McGraw-Hill, 1985.

[Lew89] D. P. Lew. Receptor signalling and intracellular calcium in neutrophil activation. *European Journal of Clinical Investigation*, **19**:338–346, 1989.

[Li93] B.-C. Li. A new computation of geometric moments. *Pattern Recognition*, **26**(1):109–113, 1993.

[LL89] F. Leymarie and M. D. Levine. Shape features using curvature morphology. In *Visual Communications and Image Processing IV, Proc. SPIE, vol. 1199*, pages 390–401, 1989.

[LL90] Y. W. Lim and S. U. Lee. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition*, **23**(9):935–952, 1990.

[LLS92] L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies – a comprehensive survey. *IEEE Trans. Pattern Anal. Machine Intell.*, **14**(9):869–885, 1992.

[LN82] M. D. Levine and A. M. Nazif. An experimental rule-based system for testing low level segmentation stragegies. In K. Preston Jr. and L. Uhr, editors, *Multicomputers and Image Processing Algorithms and Programs*, pages 149–160. Academic Press, 1982.

[LN85] M. D. Levine and A. M. Nazif. Dynamic measurement of computer generated image segmentations. *IEEE Trans. Pattern Anal. Machine Intell.*, **7**(2):155–164, 1985.

[LNY83] M. D. Levine, P. B. Noble, and Y. M. Youssef. Understanding blood cell motion. *Comput. Vision Graph. Image Process.*, **21**:58–84, 1983.

[Lø88] T. Lønnestad. Connected filters for noise removal. In *Proc. 9th Int. Conf. Pattern Recogn.*, pages 848–850, 1988.

[Lø90] T. Lønnestad. The biff image concept, file format, and routine library. BLAB Report No. 29, Dept. of Informatics, Univ. of Oslo, 1990.

[LS91] B.-C. Li and J. Shen. Fast computation of moment invariants. *Pattern Recognition*, **24**(8):807–813, 1991.

[LYNB80] M. D. Levine, Y. M. Youssef, P. B. Noble, and A. Boyarsky. The quantification of blood cell motion by a method of automatic digital picture processing. *IEEE Trans. Pattern Anal. Machine Intell.*, **2**(5):444–450, 1980.

[Mai79] S. Maitra. Moment invariants. *Proc. IEEE*, **67**:697–699, 1979.

[Mar82] D. Marr. *Vision.* Freeman, 1982.

[Mar89] S. Marshall. Review of shape coding techniques. *Image and Vision Computing*, **7**(4):281–294, 1989.

[Mas85] G. A. Mastin. Adaptive filters for digital image noise smoothing: an evaluation. *Comput. Vision Graph. Image Process.*, **31**:103–121, 1985.

[MFD92] P. Medina-Rodriguez, E. Fernandez-Garcia, and A. Diaz-Urrestarazu. Adaptive method for image segmentation based in local feature. *Cybernetics and Systems*, **23**:299–312, 1992.

[MH80] D. Marr and E. Hildrith. Theory of edge detection. *Proc. R. Soc. Lond.*, **B207**:187–217, 1980.

[Mor69] N. Morrison. *Introduction to sequential smoothing and prediction.* McGraw-Hill, 1969.

[Mus93] O. R. Musin. Topographic structure of image. In *Lecture Notes in Computer Science, vol. 719*, pages 24–30. Springer Verlag, 1993.

[NB80] R. Nevatia and K. R. Babu. Linear feature extraction and description. *Comput. Graph. Image Process.*, **13**:257–269, 1980.

[NB86] V. S. Nalwa and T. O. Binford. On detecting edges. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**(6):699–714, 1986.

[Nib86] W. Niblack. *Introduction to digital image processing.* Prentice-Hall, 1986.

[NL86] P. B. Noble and M. D. Levine. *Computer-assisted analyses of cell locomotion and chemotaxis.* CRC Press, Boca Raton, Florida, 1986.

[NR79] Y. Nakagawa and A. Rosenfeld. Some experiments on variable thresholding. *Pattern Recognition*, **11**:191–204, 1979.

[OC76] F. O'Gorman and M. B. Clowes. Finding picture edges through collinearity of feature points. *IEEE Trans. Computers*, **25**(4):449–456, 1976.

[Ots79] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys. Man Cyb.*, **9**(1):62–66, 1979.

[Pav78] T. Pavlidis. A review of algorithms for shape analysis. *Comput. Graph. Image Process.*, **7**:243–258, 1978.

[PB93] N. R. Pal and D. Bhandari. Image thresholding: some new techniques. *Signal Processing*, **33**:139–158, 1993.

[PCK84] W. K. Pratt, T. J. Cooper, and I. Kabir. Pseudomedian filter. In *Proc. SPIE Conf.*, Los Angeles, Calif., 1984.

[PD75] T. K. Peucker and D. H. Douglas. Detection of surface-specific points by local parallel processing of discrete terrain elevation data. *Comput. Graph. Image Process.*, **4**:375–387, 1975.

[Phi93] W. Philips. A new fast algorithm for moment computation. *Pattern Recognition*, **26**(11):1619–1621, 1993.

[PP93] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, **26**(9):1277–1293, 1993.

[PR92] R. J. Prokop and A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGPR: Graphical Models and Image Processing*, **54**(5):438–460, 1992.

[Pra91] W. K. Pratt. *Digital image processing.* Wiley-Interscience, 2 edition, 1991.

[Pre70] J. M. S. Prewitt. Object enhancement and extraction. In B. S. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psychopictorics.* Academic Press, 1970.

[Ram72] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Comput. Graph. Image Process.*, **1**:244–156, 1972.

[RAM83] A. P. Reeves, M. L. Akey, and O. R. Mitchell. A moment based two-dimensional edge operator. In *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, pages 312–317, 1983.

[RC78] T. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *IEEE Trans. Sys. Man Cyb.*, **8**:630–632, 1978.

[Rei91] T. H. Reiss. The revised fundamental theorem of moment invariants. *IEEE Trans. Pattern Anal. Machine Intell.*, **13**(8):830–834, 1991.

[RK76] A. Rosenfeld and A. C. Kak. *Digital image processing.* Academic Press, 1976.

[RLH90] M. W. Roe, J. J. Lemasters, and B. Herman. Assessment of fura-2 for measurements of cytosolic free calcium. *Cell Calcium*, **11**:63–73, 1990.

[Rø91] J.-A. Røttingen. Digital simulering av kalsiumsignalet ved stimulering av nøytrofile granulocytter. Work report (in norwegian), Dept. of Physiology, Univ. of Oslo, 1991.

[Rø94] J.-A. Røttingen. Personal communication. Dept. of Physiology, Univ. of Oslo, 1994.

[Rob77] G. S. Robinson. Edge detection by compass gradient masks. *Comput. Graph. Image Process.*, **6**(5):492–501, 1977.

[RT71] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Trans. Computers*, **20**(5):562–569, 1971.

[SB89] G. E. Sotak, Jr. and K. L. Boyer. The laplacian-of-gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output. *Comput. Vision Graph. Image Process.*, **48**:147–189, 1989.

[Ser82] J. Serra. *Image ananlysis and mathematical morphology.* Academic Press, 1982.

[SF81] J. A. Saghri and H. Freeman. Analysis of the precision of generalized chain codes for the representation of planar curves. *IEEE Trans. Pattern Anal. Machine Intell.*, **3**(5):533–539, 1981.

[SG91] D. Schonfeld and J. Goutsias. Optimal morphological pattern restoration from noisy binary images. *IEEE Trans. Pattern Anal. Machine Intell.*, **13**(1):14–29, 1991.

[SH78] F. A. Sadjadi and E. L. Hall. Numerical computations of moment invariants for scene analysis. In *Proc. IEEE Conf. Pattern Recognition and Image Processing*, pages 181–187, Chicago, Illinois, 1978.

[Sin93] M. H. Singer. A general approach to moment calculation for polygons and line segments. *Pattern Recognition*, **26**(7):1019–1028, 1993.

[SJ84] M. A. Shah and R. Jain. Detecting time-varying corners. In *Proc. 7th Int. Conf. Pattern Recognition*, volume 1, pages 2–5, 1984.

[SKI91] K. Sasakawa, S. Kuroda, and S. Ikebata. A method for threshold selection in binary images using mean adjacent-pixel number. *Systems and Computers in Japan*, **22**(3):66–73, 1991.

[SNA90] N. J. C. Strachan, P. Nesvadba, and A. R. Allen. A method for working out the moments of a polygon using an integration technique. *Pattern Recognition Lett.*, **11**:351–354, 1990.

[Sol88] D. R. Soll. Dms, a computer-assisted system for quantitating motility, the dynamics of cytoplasmic flow, and pseudopod formation: its application to dictyostelium chemotaxis. *Cell Motility and the Cytoskeleton*, **10**:91–106, 1988.

[SS90] S. I. Simon and G. W. Schmid-Schönbein. Kinematics of cytoplasmic deformation in neutrophils during active motion. *Journal of Biomechanical Engineering*, **112**:303–310, 1990.

[SSWC88] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen. A survey of thresholding techniques. *Comput. Vision Graph. Image Process.*, **41**:233–260, 1988.

[SW92] F. Y. Shin and W.-T. Wong. A new single-pass algorithm for extracting the mid-crack codes of multiple regions. *Journal of Visual Communication and Image Representation*, **3**(3):217–224, 1992.

[SWB92] R. A. Silver, M. Whitaker, and S. R. Bolsover. Intracellular ion imaging using fluorescent dyes: artefacts and limits to resolution. *European journal of Physiology*, **420**:595–602, 1992.

[SWF87] M. Scanlon, D. A. Williams, and F. Fay. A Ca^{2+}-insensitive form of fura-2 associated with polymorphonuclear leukocytes. *J. Biol. Chem.*, **262**:6308–6312, 1987.

[Tan81] G. Y. Tang. A discrete version of green's theorem. In *Proc. IEEE Conf. Pattern Recogn. Image Process.*, pages 144–149, 1981.

[TC86] C.-H. Teh and R. T. Chin. On digital approximation of moment invariants. *Comput. Vision Graph. Image Process.*, **33**:318–326, 1986.

[TC88] C.-H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Trans. Pattern Anal. Machine Intell.*, **10**(4):496–513, 1988.

[TFJ89] T. Taxt, P. J. Flynn, and A. K. Jain. Segmentation of document images. *IEEE Trans. Pattern Anal. Machine Intell.*, **11**(12):1322–1329, 1989.

[TH84] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. Pattern Anal. Machine Intell.*, **6**(1):13–27, 1984.

[TJ94] Ø. D. Trier and A. K. Jain. Goal-directed evaluation of binarization methods. In *Proc. NSF/ARPA Workshop on Performance versus Methodology in Computer Vision*, 1994.

[TP86] V. Torre and T. A. Poggio. On edge detection. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**(2):147–163, 1986.

[TR82] M. Tavakoli and A. Rosenfeld. Edge segment linking based on gray level and geometrical compatibilities. *Pattern Recognition*, **15**(5):369–377, 1982.

[Tsa85] W.-H. Tsai. Moment-preserving thresholding: a new approach. *Comput. Vision Graph. Image Process.*, **29**:377–393, 1985.

[TSM85] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical analysis of finite mixture distributions.* Wiley, 1985.

[TT93] Ø. D. Trier and T. Taxt. Evaluation of binarization methods for utility map images. Preprint 1993-6, Dept. of Informatics, Univ. of Oslo, Oslo, Norway, 1993.

[TYIW92] K. Tamura, S. Yoshida, T. Iwai, and I. Watanabe. Effects of isoprenaline and ouabain on cytosolic calcium and cell motion in single rat cardiomyocytes. *Cardiovascular Research*, **26**:179–185, 1992.

[Vei92] M. A. Veiss. *Data structure and algorithm analysis.* The Benjamin-Cummings Publishing Company, 1992.

[Vin91] L. Vincent. New trends in morphological algorithms. In *Nonlinear Image Processing II, Proc. SPIE, vol. 1451*, pages 158–170, 1991.

[Vok92] M. Vokáč. Compression of 2-d solar uv spectra by adaptive polygonization. BLAB Report No. 61, Dept. of Informatics, Univ. of Oslo, 1992.

[VS82] A. M. Vossepoel and A. W. M. Smeulders. Vector code probability and metrication error in the representation of straight lines of finite length. *Comput. Graph. Image Process.*, **20**:347–364, 1982.

[Wel86] W. A. Wells, III. Efficient synthesis of gaussian filters by cascaded uniform filters. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**(2):234–239, 1986.

[Wes78] J. S. Weszka. A survey of threshold selection techniques. *Comput. Graph. Image Process.*, **7**:259–265, 1978.

[Wes79] N. K. Wessells. Preface. In *The Cell Surface: Mediator of Development Process, Proc. 38th Symp. Soc. Dev. Biol.*, pages 25–27, 1979.

[WH88] P. C. Wilkinson and W. S. Haston. Chemotaxis: an overview. *Methods in Enzymology*, **162**:3–16, 1988.

[Wha91] R. J. Whatmough. Automatic threshold selection from a histogram using the "exponential hull". *Comput. Vision Graph. Image Process.*, **53**(6):592–600, 1991.

[Wit83] A. P. Witkin. Scale-space filtering. In *Proc. 8th Int. Joint Conf. Artif. Intell.*, pages 1019–1022, Los Altos, Calif., 1983.

[WR78] J. S. Weszka and A. Rosenfeld. Threshold evaluation techniques. *IEEE Trans. Sys. Man Cyb.*, **8**(8):622–629, 1978.

[WR83] J. M. White and G. D. Rohrer. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM J. Res. Develop.*, **27**(4):400–411, 1983.

[WS92] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Understanding*, **55**(1):14–26, 1992.

[WS93] M. Worring and A. W. M. Smeulders. Digital curvature estimation. *Comput. Vision Graph. Image Process.*, **58**(3):366–382, 1993.

[WT90] A. Waks and O. J. Tretiak. Robust detection of region boundaries in a sequence of images. In *Proc. 10th Int. Conf. Pattern Recogn.*, pages 947–952, Atlantic City, 1990.

[WWL92] W.-Y. Wu, M.-J. J. Wang, and C.-M. Liu. Performance evaluation of some noise reduction methods. *Comput. Vision Graph. Image Process.*, **54**(2):134–146, 1992.

[YA94a] L. Yang and F. Albregtsen. Discrete green's theorem and its application in moment computation. In *Proc. 1st Int. Conf. Electronics and Information Technology*, Beijing, China, 1994.

[YA94b] L. Yang and F. Albregtsen. Fast and exact computation of cartesian geometric moments using discrete green's theorem. Submitted to Pattern Recogn., 1994.

[YA94c] L. Yang and F. Albregtsen. Fast and exact computation of moments using discrete green's theorem. In *Proc. NOBIM Norwegian National Conf. Image Analysis and Pattern Recognition*, pages 82–90, 1994.

[YA94d] L. Yang and F. Albregtsen. Fast computation of invariant geometric moments: a new method giving correct results. to appear in Proc. 12th Int. Conf. Pattern Recognition, 1994.

[YALG94] L. Yang, F. Albregtsen, T. Lønnestad, and P. Grøttum. Dynamic thresholding using connected consistent regions. In *Proc. NOBIM Norwegian National Conf. Image Processing and Pattern Recognition*, pages 143–147, 1994.

[YB89] S. D. Yanowitz and A. M. Bruckstein. A new method for image segmentation. *Comput. Vision Graph. Image Process.*, **46**:82–95, 1989.

[YH81] G. J. Yang and T. S. Huang. The effect of median filtering on edge location estimation. *Comput. Graph. Image Process.*, **15**:224–245, 1981.

[YIK89] H. K. Yuen, J. Illingworth, and J. Kittler. Detecting partially occluded ellipses using the hough transform. *Image and Vision Computing*, **7**(1):31–37, 1989.

[YKA93] K. Yoshino, T. Kawashima, and Y. Aoki. Dynamic reconfiguration of active net structure. In *Proc. Asian Conf. Computer Vision*, pages 159–162, Osaka, Japan, 1993.

[YMB77] W. A. Yasnoff, J. K. Mui, and J. W. Bacus. Error measures for scence segmentation. *Pattern Recognition*, **9**:217–231, 1977.

[YP86] A. L. Yuille and T. A. Poggio. Scaling theorems for zero crossings. *IEEE Trans. Pattern Anal. Machine Intell.*, **8**(1):15–25, 1986.

[YWB74] I. T. Young, J. E. Walker, and J. E. Bowie. An analysis technique for biological shape. I. *Info. Control*, **25**:357–370, 1974.

[ZG92] Y. J. Zhang and J. J. Gerbrands. Segmentation evaluation using ultimate measurement accuracy. In *Proc. SPIE Vol. 1657, Image Processing Algorithms and Techniques III*, pages 449–460, 1992.

[ZVZvK87] M. F. Zakaria, L. J. Vroomen, P. J. A. Zsombor-Murray, and J. M. H. M. van Kessel. Fast algorithm for the computation of moment invariants. *Pattern Recognition*, **20**(6):639–643, 1987.