INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313 761-4700 800 521-0600

Order Number 9020777

Cytological image contour extraction and region segmentation

Li, Yuan, Ph.D.

City University of New York, 1990

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

CYTOLOGICAL IMAGE

CONTOUR EXTRACTION AND REGION SEGMENTATION

by

YUAN LI

Dissertation submitted to the graduate Faculty in Engineering in partial fulfillment of the requirements for the degree of Doctor of Philosophy. The City University of New York.

1990

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

1/30/90
Date

Haym Barba
Chair of Examining Committee

1/30/90
Date

Jacques E. Bonvaniste
Executive Officer

Professor J. Barba

Professor S. Ahmed

Professor N. Schienberg

Professor M. Colef

Professor S. Basu

Supervisory Committee

The City University of New York

ABSTRACT

CYTOLOGICAL IMAGE

CONTOUR EXTRACTION AND REGION SEGMENTATION

by

YUAN LI

Advisor: Professor Joseph Barba

In this doctoral dissertation, mult-ithresholding combined with binary morphological edge detection and region extraction approaches for cytological images are investigated. Specifically, techniques for detection of the cell and nuclear contours in cytological specimens are presented. Gray level images are viewed as stacked binary bit plane with each bit plane image representing the effect of thresholding the gray level image with multiple gray level windows. A binary morphological edge detector is employed to identify edges on selected bit planes. A simple contour tracking algorithm incorporating a priori information of the cell structure is used to identify the most likely cell and nuclear contours. Further refinery is made by matching each binary edge with the gradient of the original image. A figure of merit is defined and used for selecting specific binary edges (root edges)

which most likely contain the contours of the cell image with best extractability. A fast and effective technique is also presented for determining root edges automatically based on searching binary edges in the spatially shifted images. Additionally, a two-step hierarchical region and texture segmentation technique by using vector quantization is presented. The regions encoded by low rate vector quantizers are refined by searching the replacement of a much larger codebook containing all the edge information. Cell regions are also segmented into texton regions. Related texture features are then extracted for further analysis and classification.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

CHAPTER 1

INTRODUCTION

1.1 The Importance of Cytological Image Segmentaion

The main purpose of cytology is to differentiate between normal cells, reactive cells (those which have undergone a number of reversible changes as a result of inflammation or injury) and malignant cells; if malignant cells have been found, to determine the precise type of tumor and the prognoses of optimal treatment. Most of these goals are achieved by routinely examining the cytology specimens under microscopes. The diagnosis is based on the observation and interpretation of the morphological image features by a trained pathologist. For large amount of cell images, performing the examination is a tedious, fatiguing and time consuming process; in the United states alone, about one hundred million white blood cells are examined yearly. Various techniques have been devised to automate this process, unfortunately, many of these techniques fall short of expected performance.

The need for modern computers in cytological image analysis and classification have been well recognized for two decades. Computers can be used not only for the rapid identification of cells for purposes of diagnostic screening and classification, but also to aid the health professional in making difficult diagnostic decisions. In cyto-pathology, hematology and histo-pathology, there exist many

instances where the discriminatory powers of human vision and human visual assessment are limited in making a diagnostic decision. The computer may provide a quantitative tool which will be able to help solve these problems.

The majority of computer image processing in biomedical research is devoted to automated analysis of biological cells. The main reasons for placing emphasis upon cell analysis are: 1) cell analysis is a well-defined problem in the biomedical field for which modern computers may be successfully employed to produce useful results; and 2) cell analysis is a process of fundamental importance in biological and medical studies as exemplified by the analysis of blood smears and tissue sections in making diagnosis and prognosis.

Three benefits are expected from reliable computer cell image analysis and classification techniques:

1. To extend the benefit of an early detection of cancer to a large fraction of population;
2. To free the laboratories of tedious routine work by eliminating most of the negative cases;
3. To hopefully increase both sensitivity and specificity of the method.

Historically, analysis and classification of cells are done by the visual appearance of the cells under the microscope. The automation of this basic visual procedure will be more easily accepted by cytologists if the automatic technique imitates to some extent the

human recognition process. Not surprisingly, cell analysis by computer is performed in three successive steps (also true of image analysis in general):

(1) Segmentation, which means the decomposition of the cells into its components.
(2) Cell feature extraction, which is the step by which quantitative photometrical, geometrical, textural and morphological cell information is extracted from the segmented cell images.
(3) Classification, which consists of the assignment of class labels to the individual cells.

The most crucial part of automated cell image analysis is the segmentation of the cell scene into meaningful regions. The reason is that errors in the segmentation process may propagate to measurement extraction and to the classification of the objects in the scene and this may finally result in an erroneous scene interpretation.

Automated cell classification techniques have tremendous difficulties in correctly segmenting the vast varieties of cell images into meaningful segments in a reliable way. This is due to the large amount of imagery problems that must be addressed such as of possible low contrast between the different regions; the smooth texture variations observed inside the objects; as well as frequent contacts between the cells and artifacts, for which gray level, or even color characteristics, are often similar.

Numerous segmentation methods for digitized cell image scenes for peripheral blood smears or bone marrow smears have been proposed. A survey of the segmentation schemes developed in quantitative cytology is given by Fu and Mui[1]. Although the techniques were developed well over a decade ago most are still used in the laboratory experiment stage.

Motivated by the consideration that correct scene segmentation is a very important, difficult and error-prone process in automated cell analysis and classification and that no perfect scheme is available for this purpose, we have put much emphasis into the design of new techniques for the accurate segmentation of cell image.

1.2 General Goal of the Research

Current cytological image segmentation techniques can be categorized in the following three classes:

1. Thresholding and characteristic feature clustering
2. Edge detection
3. Region extraction

Approaches to cell image segmentation by thresholding or clustering are fast and simple but suffer because no spatial information is genrally used during the search for segmentation thresholds or cluster boundaries. This means that there is no guarantee that either the pixels of individual segments will be connected nor that segments shape fulfill any well-defined geometric conditions. Edge

detection will perform well when the boundaries of the cell image are discernible. But due to the frequent appearances of varying and/or low contrast boundaries and surrounding artifacts accurate boundary extraction is difficult. Region extraction by region growing is a promising approach, but due to a lack of reliable rules to control when to stop, regions usually grow too large. Additionally, current algorithms are usually very complex and time consuming.

Experiments and experiences have shown that a good technique and processing strategy for cytological image segmentation should be guided by the search for a compromise between the three criterions of accuracy, efficacity and complexity. Processing accuracy results from the appropriate modeling and accurate processing of local and global information. It is possible to derive cooperative algorithms based on the association of thresholding and edge detection processes. Region growing processes may also be efficiently controlled by means of a contour based shape evaluation.

Processing efficacity clearly depends on the adequacy of the chosen segmentation method, considering the particular problem to solve, and the structure of the image under study[2]. Without fully investigating the specificities of different classes of cells and using suitable method or strategy which correspondingly introduced a priori knowledge of photometrcal, geometrical, morphological, and topological information (such as intensity distribution, size, shape, closeness and convexity) a segmentation method can not be robust and reliable.

Careful analysis of possible processing strategies should also be conducted to reduce, as much as possible, the processing complexity. Recent advances in processing hardware technology allows us to take advantage of the parallel structure of many algorithms; increases in processing speeds allows more complex algorithms to be implemented for routine use in a hospital. In our research, simple processing strategies can be defined based on a two-step approach: a rough segmentation is performed in a first step by using a low complexity algorithm to approximately delimitate regions of interest. More complex algorithms may then be invoked on a reduced input database.

Based on the above analysis, the aim of the research is to develop accurate, effective and simple segmentation methods specifically suitable for isolated cell images. To reach this goal, the following important points are considered in the design and strategy selection:

1. Image modeling and handling are based on the analysis of the particular cell structure. Multiple resolution cell images either modeled by a stack of binary bit planes or the clusters approximately vector coded by VQ not only generate the heuristic and simple structures closely representing the original images, but also provide suitable image formations, such as binary edge representatives and simplified coarse segmented regions, for further hierarchical refinery processing.

2. Fully investigate the a priori information that can be employed in each stage of the segmentation, such as preprocessing, edge

detection, vector coding, tracing, matching, filtering and post processing. Such information will include the number of clusters; the size predicates of the cell and nucleus; the shape and convexity of the cell boundary; the connective features of the different regions and spatial intensity information if possible. Appropriately inserting these information in each of the processing stages is not only of great importance in correct segmentation, but also makes the method more reliable.

3. Combine global thresholding and local gradient techniques, region and edge information, aim to develop new cooperative techniques to take advantage of the specialties of the different approaches and all possible information.

4. To reduce the complexity of the algorithm for suitable implementation in a PC based system for routine interactive use in a hospital environment. Binary morphological processing techniques are adopted for fast processing and suitable for parallel processing.

The dissertation is organized as follows: In chapter 2, the previous work on cytological images are reviewed; the background is composed of the image structure feature analysis and a brief introduction of techniques in morphological image processing. In chapter 3, a bit-plane multithresholding combined with binary morphological edge detection techniques is presented. In chapter 4, the method in chapter 3 is extended by introducing the gradient the gradient information, a figure of merit is derived and used on the presented techniques for

spatially shifted images. In chapter 5, a region segmentation technique is presented based on spatial vector quantization, each component is the segmented into texton regions, and texture features are derived heuristically. Results of all the techniques are presented and expected future research is also proposed in chapter 6.

Finally, I wish to thank my mentor, Professor Joseph Barba, for his brilliant guidance, encouragement, discussions and full support through out the entire research period, and his final review and modification of this dissertation.

CHAPTER 2

REVIEW OF PREVIOUS WORK AND BACKGROUND

2.1 Introduction

Numerous segmentation methods for digitized cell images from peripheral blood smears, bone marrow smears or tissue sections have been proposed. A survey of the segmentation schemes developed in quantitative cytology is given in section 2 in this chapter. Background materials introduced in section 3 consist of morphological image processing techniques and Vector Quantization techniques, which will be used directly in the following chapters. Finally, the structural features of cytological images are listed and analyzed in section 4, since taking full advantage of these features will greatly improve the performances of segmentation techniques.

2.2 Survey of Methods in Cell Image Segmentation

Image segmentation is usually accomplished by one or more of the following approaches:

1. Thresholding.
2. Edge detection and boundary following algorithms.
3. Feature clustering.
4. Region extraction methods.

Applications of these segmentation approaches on cytological images are non-uniformly distributed due to different cell categories and qualities of the processed images, and the available hardware and speed requirements.

2.2-1 Thresholding techniques

The simplest technique of region segmentation is thresholding which can be classified into the following categories:

1) Global threshold selection.

This method is suited for scenes containing homogeneous objects against high-contrast uniform backgrounds. The threshold can be selected either by experience or by the histogram of intensity values, assuming that the histogram has clear peaks. Often the histogram is bimodal or multimodal.

Such histogram based techniques assume the thresholds values and resulting segmentation will cluster pixels with similar gray levels corresponding to continuous regions or blobs. If the image gray level histogram has well defined maxima and minima then a plausible thresholding process would select the threshold values at the minima between the histogram modes.

Garbay[3] and Gauvain[4] reported poor cell segmentation by multilevel thresholding the gray level histogram cytological images

to define four regions corresponding to the background, erythrocytes, cytoplasm and nucleus of cytological images. Better results were reported, but still not acceptable, by selecting threshold values derived from a multivariant analysis of the luminance, saturation and hue histograms. Similar results in the ineffectiveness of global thresholding of histograms in cytological applications have been reported by others[5,6].

2) Local or Adaptive Thresholding

Local or adaptive threshold selection techniques are surprising lacking in cytological applications. Komitowski and Zinser[7] used the local histogram to identify chromatin particle which appear a locally high gray level. Taylor[8] et al used a variation of adaptive thresholding by using threshold from extinction value histogram, the histogram of the logarithm of the extinction values or from information in the transition probability matrix of the image.

Local methods to compute thresholds can also depend on both the gray level at a point and some properties in the neighborhood of the point. Some of these methods rely on linear threshold operators[9] where a threshold is computed based on the average contrast over previously analyzed sections of the image. Similarly generated thresholds can be applied locally or to large regions. Ullman[10] described a technique to compute thresholds based on the gray levels of the neighborhood of

a point. In this scheme certain points of a 5*5 neighborhood of a point along with two rules are used to select a threshold. The whitest gray level decides which rule will be selected. Morrin(1974) used two-dimensional plots of gray level versus gradient value to select threshold. Panda[11] used a scheme where the threshold applied to a point depended on the gray level as well as the edge value of the point. Tri-modal distributions obtained from the plots of frequency as a function of gray level and edge value gave rise to several segmentation procedures. One such method[12] uses the mean gray level of those points having high edge values as a threshold. Another scheme searched for valleys in the gray level histogram of points having low edge values. The best schemes were those which used a combination of the valley search and mean gray level methods.

Dynamic threshold selection techniques are those in which threshold values depend on the gray level, some properties of the neighborhood of a point and on the location of the point in the image. Such a method was used by Chow and Kaneko[13] to detect boundaries in radiographic images. Images were subdivided into overlapping windows of size 7*7 for which the histogram was computed along with the variance of the gray level. The histogram were modeled as one or two normal distribution depending on whether the histogram was unimodal or bimodal.

Minimum error thresholds were selected for those windows whose mixture distribution satisfied a bimodality test. This is because these

windows contained both background and objects and should therefore be close to the boundaries of objects. Once the thresholds had been computed for each window a linear interpolation scheme was used to select the threshold to be applied to each pixel of the windows. Interpolation was also used to find thresholds in windows which had bimodal distributions. This made the scheme truly dynamic since a pixel's threshold depended on its gray level, local statistics and its proximity to boundary points.

Other techniques for threshold selection can take into consideration the size of the blobs[14] above threshold that are to be detected. The desired threshold would be selected to give the blob size which can be measured by their width or cross section. Such a technique was employed by Kulkarni[15] in order to detect white cells in venous blood diluted with an equal volume of isotonic saline. A fixed threshold is used to create a thresholded image contains dark blobs above a certain size which correspond to the white blood cells. Kulkarni reported that more accurate threshold values could be determine if a yellow image (using a yellow filter) was used. Color was also found to be one of the best features to discriminate between monocytes from myelocytes when processing by machine but was not considered that powerful by technicians. Another technique that can be used would select a threshold so as to minimize some measure of the homogenuity of the two segmented regions and/or maximize some measure of the difference between the regions.

1.2.2 Other segmentation methods used for cytological images

Edge detection (contour extraction) and region segmentation are essentially two approaches to solve a common problem. In edge detection pixels which have a property corresponding to edges are identified. The contours of image regions having different characteristics are extracted by joining the appropriate edge pixels. In segmentation one locates the different regions (based on some property) which form an image. Once these regions have been located, contours are simply defined as the set of pixels where different regions meet.

Edge detection or Gradient (convolutional) techniques have been employed to detect cell contours. A segmentation method for neutrophil cells based on edge detection has been proposed by Prewitt[16] and Brenner et al.[17] They have used edge following techniques which correlates the boundary with the points of maximal local gradient in the density distribution. Identification of contour points requires thresholding and some means of determining if a point of maximum local gradient actually belongs to the contour to be extracted or to some other structure. The contour can then be approximated by straight line segments or arc fitting between these points. Zinser and Komitowski[18] use the a priori information that cells are generally circular and thus a polar transformation can be used to restrict the maximal gradient search.

Relaxation processes are based on an initial stochastic labelling and iterative application of label-updating rules are performed to converge toward a globally adequate result. An example is Garbay[19] who used the relaxation to label and segment the bone marrow cells. A gradient relaxation method based on maximizing a criterion is proposed by Bhanu[20] et al for segmenting the cell images with unimodal distributions.

A region extraction method for blood smear segmentation is presented by Haussmann and Lsedtke[21], local properties of the scene, global information about neighborhood relations and shape of the scene components are used for the improvement of the segmentation. Texture and color features are also used for segmenting the blood smears using the above methods proposed in [22] and [23].

2.2.3 Discussion on current segmentation methods

The sturctures of cytological images usually have the following characteristics:

1. The boundaries of different regions are sometimes poorly defined or the background does not appear as a well-contrasted and homogeneous region.
2. The existences of the region textures and outside artifacts of the cell images.

The above structure characteristics often generate inadequate histogram mode or gradient information available for segmentation operation. Simple mode thresholding, although the simplest approach, is not suitable for the images having structure feature 1. Adaptive multiple threshold requires optimal thresholds finding techniques, there is still no reliable way of selecting such thresholds automatically and the thresholds for the same pictures using different optimal schemes have very large discrepancies. Thresholding operations in general suffer from their sensitivity to noise. However, the operation is simple and isolated cell can indeed be obtained by a straightforward thresholding provided that the background appears as a rather well-contrasted and homogeneous region. These techniques are still widely used for the cell images having well-defined boundaries.

Edge-based methods also suffer from the first difficulty. The extracted gradient map reflects the facts that most boundary pixels have locally maximum contrast values and can be thresholded to eliminate texture and artifacts. However, the boundary obtained is fragmented and threshold finding and thinning procedures may add to the complexity to the approach. Because edge detection is sensitive to the contrast variations it is suitable for discriminating the regions closely located.

Region-based approach have the good feature of not being very sensitive to noise or textural variation on the edges. This is due to the operation being implemented on regions instead of on the

individual pixels. However the major drawback is that usually complicated split-and-merge operations are required with a well-defined rule to determine how far to go with the algorithm. Sometimes different regions with evident intensity differences are still wrongly merged and the algorithms have problems segmenting two similar regions. Although region techniques are reportedly to have the best segmentation performances, because of their time consuming feature, they can not be employed in application requiring rapid results.

Feature clustering and relaxation based on an initial stochastic labeling appears as a more flexible approaches. However, the segmentation efficacity greatly depends on the initial labeling. Global errors, resulting from the initial potential for confusion between the different classes, may not be solved by such approach.

Each of the above approach have positive features but also disadvantage for segmentation of ill-posed cell images. No one approach can solve the above questions alone. Pixel based approaches, such as edge detection, have larger region discrimination power due to its sensitivity, but is also noise sensitive; region growing techniques have the inverse features; thresholding does well in handling the global information, but without combining the local information, error rates will be increased. Cooperative approaches for designing new algorithms to reduce the negative effects and expanding their good features seem to be one solution. Fully

investigating the particular characteristics of cell images, introducing the possible a priori information into the programs effectively, may provide more powerful criterions to each of the listed approaches above for solving the common difficulties.

2.3 Morphological Image Processing

Mathematical morphology is an approach to image processing based on set theoretic concepts of shape. It was formalize at the Ecole de Mine in Paris in the mid 1970's by G.Matheron[23] and extended by J. Serra[24]. Basically, morphological image processing is the analysis of the geometrical relationship between the image and a smaller image called a structuring element. The advantages of morphological image processing is that important geometrical image features can be preserved, the algorithms are simple and may lead to parallel implementation. Disadvantages are that the analysis is nonlinear and there is lacking an analytical criteria for choosing the structuring element.

Binary morphological filtering of binary images has been use for more than 10 years. Recently, Haralick,[25] et al used the top surface and umbra concept of an image to extend the properties of binary morphological processing to gray level morphological processing through their homomorphism theorem.

2.3.1 Basic binary morphological filters

1. Image translated by a structuring element(SE)

If g(m,n) is a SE described by a binary matrix containing only 1's and 0's, the translation of the binary image matrix X(u,v) by any matrix element g(i,j) ϵ g(m,n) is defined as:

$$[Tran(X;i,j)](u,v) = X(u-i,v-j) \quad \text{when } g(i,j)=1 \tag{2-1}$$

This means that for each of the activated element in the SE corresponding to a 1, X is translated to the position determined by the position of that activated element g(i,j). For each non-activated element in the SE, no translation is obtained. Thus, the number of translated pictures is exactly equal to the number of activated elements(1's) in the SE.

To determine the direction of the translation, an origin [(0,0) point] has to be specified with respect to the SE. The origin may be inside or outside the SE. Generally, the origin is chosen inside SE. For a symmetrical SE, the origin is usually at the center of the SE to simplify morphological operations.

2. Four basic basic morphological filters

a) Dilation or Minkowski addition is defined as:

$$d(X,g) = X \oplus g = \bigcup_{(i,j)\in g} Tran(X;i,j) \qquad (2\text{-}2)$$

The dilation, $d(X,g) = X \oplus g$, of X by g is the union image of all the translated original X images by g.

b) Erosion or Minkowski subtraction is defined as:

$$e(X,g) = X \ominus g = \bigcap_{(i,j)\in g} Tran(X;i,j) \qquad (2\text{-}3)$$

The erosion, $E(X,g) = X \ominus g$, of X by g, is the intersection of the translated original X images by g.

Performing the union (dilation) or intersection (erosion) at point (i,j) for n original translated images by SE g is equivalent to performing dilation or erosion with the original image and translating the results to point (i,j) and its n neighbors.

The effect of dilation is to expand an image, i.e. elements are added after a dilation process, whereas erosion results in the shrinking of the images, i.e. if a portion of an image is too small to contain the structuring element then that portion will disappear in the process of erosion.

c) Opening and Closing

By combining the operations of dilation and erosion we can define the opening and closing of an image by a SE. The opening of X by g is

defined as:

$$o(X,g) = (X \ominus g) \oplus g \tag{2-4}$$

while the closing of X by g is defined as :

$$c(X,g) = (X \oplus g) \ominus g \tag{2-5}$$

Opening can be used to suppress sharp capers and cuts narrow isthmuses. Closing can be used to fill in thin gulfs and small lobes.

2.3.1 Gray level morphological filters

a) Dilation and Erosion

If f and g are both multilevel images the dilation of f by g can be defined as:

$$D(f,g) = f \oplus g = \max_{(i,j)\in g} [Tran(f;i,j) + g(i,j)] \tag{2-6}$$

Similarly, erosion of f by g is defined as:

$$E(f,g) = f \ominus g = \min_{(i,j)\in g} [Tran(f;i,j) - g(i,j)] \tag{2-7}$$

If we consider only binary SE, where a value of 1 indicates activating a translation, then dilation and erosion become

$$D(f,g) = \max_{(i,j)\in g} [Tran(f;i,j)] \qquad (2\text{-}8)$$

$$E(f,g) = \min_{(i,j)\in g} [Tran(f;i,j)] \qquad (2\text{-}9)$$

By interchanging union to maximum, and intersect to minimum, the similarity of equations 2.8 and 2.9 to binary dilation and erosion can be seen.

b) Grayscale opening and closing

Gray scale opening and closing are defined as:

$$O(f,g) = D[E(f,g),g] \qquad (2\text{-}10)$$

$$C(f,g) = E[D(f,g),g] \qquad (2\text{-}11)$$

Dilation transforms all pixels with lower gray values to the highest value within the window g. Erosion changes all pixels with higher values to its lowest level within the window. These functions are often used to smooth the image or remove narrow peak features such as ridges, isolated peaks, etc. Opening and Closing are more frequently used, since except for noise suppression, they can also at

least partially recovered the useful information lost during the first step dilation or erosion. Binary and gray level morphological filters and their properties are listed in table 2.1.

2.3.3 About binary structuring element(SE)

Information about size, spatial distribution, shape, texture, connectivity, convexity, skeleton, smoothness and orientation can be obtained by transforming the image using different structuring elements. The SE can take a large variety of shapes and sizes; choosing the appropriate one depends on the kind of information to be extracted from the images. The position of the origin, the size and the shape will directly affect the output of the morphological filters. Structuring element have the folowinging features:

a) The position of the origin causes changes in the direction of the translated images, and generates different results.
b) The size of the SE determines the size of structures to be filtered out, and the distortion introduced. If the size is too small, noise with larger sizes may not be removed; while if the size is too large, region boundaries may be distorted by the opening and closing.
c) The shape of the SE is important in revealing the similar feature in an image. The shape selection is a difficult question and is generally determined by the application. For smoothing feature usually disk-like rhombus, or small square SE shapes are used. For

texture analysis, lines with different directions are often adopted. For recognition, the shape of the SE is chosen to be the template of the object to be recognized.

2.3.4 Applications

a) Some binary M-filters and their applications:

1) Edge detection:[26,27]

External edge:	Edge1 = d(X,g) - X	(2-12)
Internal edge:	Edge2 = X - e(X,g)	(2-13)
Saddle edge:	Edge3 = d(X,g) - e(X,g)	(2-14)

where g is a solid SE, and its size will control the edge thickness.

2) Noise suppression:[28]

O(x,g) suppresses the light noise peaks. C(X,g) and C[O(X,g)] can clean dark noise spikes.

3) Shape recognition:[29,30]

Hit-or-Miss operation:

$$X * (g1,g2) = (X \ominus g1) \cap (X \ominus g2)^{c} \qquad (2\text{-}15)$$

suppose g1 = g2 = g then X ⊖ g is the locus of all points where g fits inside X, and $(X \ominus g)^c$ is the locus of all points where g fit the background of X, and finally the intersection provides exactly those places where the shape of g with its required background occurs in the image X.

4) Region filling:[31]

The problem is to fill (paint) the inside of a boundary if a inside point p is known and the dilation begins at this point. The filling procedure is described by:

$$(X \oplus ig) \cap X^c \qquad (2\text{-}16)$$

where ig = g⊕ g⊕...⊕g. ig is obtained by dilating g by itself i-1 times, and g is a small symmetric and convex SE. Intersection is taken after each dilation. The procedure repeats j times until the region is completely filled.

5) Skeleton generation:[32]

The skeleton (or medial axis) can be obtained by using erosions and openings by a fixed SE g. Specifically, the skeleton of X is the finite union of disjoint skeleton subsets:

$$S_n = (X \ominus ng) - O[(X \ominus ng),ng] \qquad (2\text{-}17)$$

The original image X is the union of all dilated subsets, $S_n \oplus ng$, where ng denotes a structuring element of size n obtained by dilating g by itself n-1 times.

6) Thinning and Thickening:[33]

Thinning X by g is: $X \otimes g = X \cap (X * g)$ (2-18)

Thickening X by g is: $X \odot g = X \cup (X * g)$ (2-19)

where * represents hit-or-miss operation (see 1.15).

7) Conditional thinning and thickening:[34]

Given two binary sets X and Y and SE g, we can define conditional thinning as $Y \otimes g ; X$. This means Y is thinning by g with the restriction of given X.

Conditional thinning: $Y \otimes g ; X = (Y \otimes g) \cap X$ (2-20)

By duality:

Conditional thickening: $Y \odot g ; X = (Y \odot g) \cup X$ (2-21)

b) Some gray level M-filters and their application:

1) Edge Detection:[35]

$$\text{Edge1} = D(f,g)-f \tag{2-22}$$

$$\text{Edge2} = f-E(f,g) \tag{2-23}$$

$$\text{Edge3} = D(f,g)-E(f,g) \tag{2-24}$$

where f is a gray level image and g is a specific binary SE which is often solid.

2) Noise Suppression:[36]

O(f,g) suppresses the positive noise peaks. C[O(f,g),g] can clean the negative noise spikes.

3) Contrast Enhancement:[37]

If f_1 denotes the enhanced image, then

$$f_1 = f \oplus g \quad \text{if } (f \oplus g)-f < f-(f \ominus g) \tag{2-25}$$

$$f \ominus g \quad \text{otherwise} \tag{2-26}$$

2.4 Vector Quantization (VQ)

Vector quantization (VQ) has recently emerged as a powerful and effective approach for speech and image coding. It has also

demonstrated pronounced success at producing low-rate compressed speech and image. A general survey of vector quantization may be found in (39, 40).

2.4.1 Image Coding and Data Compression system

The transmission and storage of pictorial information is of practical importance in diverse application areas, including digital television, picturephones, space exploration, biomedical and industrial quality control. The central problem in image communication is channel capacity or achieving storage reduction while maintaining an acceptable fidelity or image quality. The possibility of a bandwidth reduction is indicated by two observations. First, there is a large amount of statistical redundancy or correlation in normal images. For example, two points that are spatially close together tend to have nearly the same brightness level. Second, there is a large amount of psychovisual redundancy in most images. That is, a certain amount of information is irrelevant and may be eliminated without causing a loss in subjective quality, and a large amount of redundancy may be removed without causing a complete loss of detail.

The general digital communication systems consist of a information source, encoder, channel, decoder, and an information user (see figure 2-2). It is assumed that the information source generates a sequence of discrete symbols taken from a finite set. That is, images mast be sampled and quantized for transmission over the channel. The encoder is

a device that transforms the source output into a form that may be transmitted over the channel. Errors occur during transmission over the channel. Then the decoder converts the channel output into a form suitable for interpretation by the user. The usual communication problems assumes that the source, channel, and user are given and cannot be altered, so the communication problem designer must construct the encoder and decoder to satisfy the rate and error criteria desired of the system.

Many image coding techniques and data compression systems have been developed in the past two decades such as pulse code modulation (PCM), Differential pulse code modulation (DPCM), predictive coding, transform coding and various hybrid coding techniques. Vector quantization (VQ) techniques have been developed since the late seventies. Conventional DPCM or predictive coding systems are based on scalar coding while the transform coding systems, although transformation is implemented on vectors, quantization is still operated on the single component. Thus they are eventually scalar coding systems.

The advantages of using VQ are as follows:

1) A fundamental result of Shannon's rate distortion theory[41] is that better performance can always be achieved by coding vectors instead of scalar. According to the theory, vector quantizer achieves optimal rate distortion performance subject to only a constraint on memory or vector length of the signal segment being coded.

2) In VQ, several components are quantized simultaneously instead of separately, thus, the intervector correlation can be exploited.

3) Vector Quantizer has a modular structure. Hence it can be mapped on to a systolic type architecture that is easy to implement using VLSI technology.

2.4.2 Vector Quantization

An image is partitioned into small blocks of p*q pixels, each cell is regarded as a vector of dimension p*q. Each ordered set of k=p*q samples (k dimension vector in Euclidean space) is mapped into one of a finite set of N representatives or "output vectors". A binary word can then be used to identify which of the standard set of output vector represents the original input vector. After digital storage or transmission of the codeword a table look-up or "codebook" can be used to reproduce the corresponding output vector which approximately reconstructs the original input vector.

General block diagram of a VQ system is shown in Figure 2-3.

1) Encoding scheme is described by:
   a) An input vector X of dimension k where $X = (x_1, x_2, \ldots, x_k)$.
   b) A rate of R bits/vector (or r = R/k bits/samples).
   c) A codebook $C = \{Y(i), i=1, 2, \ldots, N\}$, where Y are also k-d vectors and $N = 2^R$.
   d) An encoder maps an input source vector $X = (x_1, x_2, \ldots, x_k)$ into a binary R-tuple or a binary channel vector $U = (u_1, u_2, \ldots, u_R)$.
2) The decoder is a simple table lookup procedure, it maps or transforms the received the vector into a reproduction vector $Y = \beta(U) \in C$.
3) The optimal codebook C is stored into a large memory to hold an

accurate representation of $N = 2^R$ k dimensional codevectors.

4) The binary channel vector J is interpreted as the binary index of the reproduction vector to be output.
5) Define a nonnegative distortion measure d(X,Y) that measures the distortion or cost of reproducing an input vector Y and attempt to minimize the average distortion D(X,Y) = E[d(X,Y)].

From the analysis above, the key problems in VQ design are to develop effective codebook structure to allow high dimensionality vector coding and to find effective perceptually-based distortion measure for which optimal codebooks can be designed. These two problems are related because distortion measure may naturally give rise to a structure that facilitates accordingly, with little or no loss in optimality.

2.4.3 Optimal Vector Quantizer

A N-level vector quantizer is said to be optimal (or globally optimal) if it minimizes the expected distortion, that is, a special quantizer C^* is optimal if for all other vector quantizers C having N reproduction vectors, $D(C^*)<D(C)$, where D is defined as average distortion of the quantizer. A quantizer is said to be locally optimum if $D(C^*)$ is only a local minimum, that is, slight changes in C cause an increase in distortion. The goal of vector quantizer design is obtaining an optimal quantizer if possible and, if not, to obtain a locally optimal and hopefully "good" quantizer.

Developing and selecting proper distortion measures that coincide

subjectively with the human visual assessment is of extremely importance in vector quantizer design. Various distortion measures have been developed. If we assume the distortion caused by reproducing an input vector X by Y is given by a nonnegative distortion measure d(X,Y). Some of the important measures are as follows:

1) Square error distortion: $$d(X,Y) = \sum_{i=1}^{k} | x_i - y_i |^2 \qquad (2\text{-}27)$$

2) The weighted square distortion: $$d(X,Y) = \sum_{i=1}^{k} w_i | x_i - y_i |^2 \qquad (2\text{-}28)$$

3) Minkowski norm: $$d(X,Y) = \max_{1 \le i \le k} | x_i - y_i | \qquad (2\text{-}29)$$

where the square error or related mean square error (MSE) are the most frequently used criterion for image quality assessments. The major advantages of the MSE are the intuitive appeal (large errors are given more importance than small errors); the ease of computation; and the mathematical tractability. But this measure does not distinguish between different approximations having the same total square error and sometimes MSE is found to correlate poorly with subjective ratings.

Necessary conditions for an optimum vector quantizer may easily derived(43). In summary, associated with each quantizer output Y(i) is a (nearest neighbor) partition S(i), i=1,...,N, satisfying

$$S(i) = \{ X: d(X,Y(i)) < d(X,Y(j)), \text{ for } j \neq i \} \qquad (2\text{-}30)$$

For each partition S(i), Y(i) is the conditional mean,

$$Y(i) = E[X/X\epsilon S(i)] \qquad (2\text{-}31)$$

General conditions for optimaility are not known, and there may exist many local optimum quantizers that satisfy (2-31) and (2-32). These two properties are basic rules for optimal codebook design but they do not provide optimum values explicitly, since Y_i for a partition depends on the decision hyperplane of partition, and S(i) depends on the output level Y(i).

The basic concept for VQ technique is not new in fact it is the most general way to source coding given a constraint on the dimensionality of the blocks. No significant progress has been made until a systematical method of generating an optimal codebook is developed. The LBG algorithm[42] developed by Linde, Buzo and Gray is considered as the most general and effective method to generate good vector quantizers or codebooks. It is not only an independent image coding method but also frequently used as a powerful component in various vector quantization methods.

2.4.4 A General Optimum VQ Design Algorithm - LBG Algorithm

The necessary conditions (2-30) and (2-31) provide the basis of the iterative LBG quantizer design algorithm. The details of the algorithm are presented in (42). The basic idea of the algorithm is simple and can be summarized as follows:

If a codebook of vectors is defined as C = {Y(i); i=1,2,..., N}, and the each source vector from the original image are represented by X. A

distortion measure (square error and MSE is used in LGB algorithm) between X and Y is described as d(X,Y). Co is an initial guess for the codebook then LGB algorithm will iterate the following two steps:

1). Find the optimum representation (partition) for the source vectors by finding the nearest neighbor (with minimum square error) codevector. That is, partition each of the source vectors X into one of the N-th partitions S(i) based on (2-30).

2). Compute the best reproducing vector (new codevector) Y(i) for each partition S(i). Y(i) is the minimum distortion vector or centroid of S(i) under Mean square error measurement d(X,Y). If n_i represents the number of source vectors in the i-th partition, then:

$$Y(i) = \frac{\sum_{x \in S(i)} X}{n_i} \qquad i = 1,\ldots, N \qquad (2\text{-}32)$$

The distortion in step 2) must decrease or not change. Should it decrease, the partition found in step 1) will no longer be optimum and must be recomputed. Iterations between step 1) and 2) provides a nonincreasing distortion, and the algorithm eventually converges to a local optimum design(42).

The algorithm has overwhelming advantage of being the only general (local optimum) vector quantizer design algorithm and has several noteworthy features:

1) No source model need be postulated to determine how to allocate bits

within each block. Local optimal quantizer is achieved by training large amount of image data samples.

2) Distortion measures need not be restricted to only those which are tractable like MSE.

3) For finite alphabet distributions such as sample distributions, the algorithm always converges to a fixed-point quantizer in a finite number of steps(42).

One major disadvantage of the VQ used in higher rates is the exponential growth in the computational complexity when a large block length is employed, the numbers and size of the codewords should be carefully considered. Another disadvantage, when lower bit rates are used, is that the edges of the coded images are not well represented.

2.4.5 Various Vector Quantization Techniques

To reduce the effect of the above two major disadvantages, different versions of vector quantization structures or algorithms have been developed. The research has focused on design techniques for vector quantizers that have structures which yield a slower growth of encoder complexity with rate or dimension and also on developing perceptually-based techniques by using the human visual model in VQ design. Tree search structures; product codes and segmented codebooks are frequently employed in the VQ design to reduce the search complexity effectively. Even though, the generated vector quantizers are only suboptimum.

Differential VQ[44], multiple step VQ[45], Gain shape VQ[46], multiple

codebook VQ[47], predictive VQ[48], hierachical VQ[49], feedback VQ[50], lattice VQ[51], finite state VQ[52], transform VQ[53] and other VQ techniques [54,55,56] considering the human visual model have been developed. Most of these algorthms use LBG algorithm as a basic unit for the VQ design.

2.5 The Structure Characteristics of the Cytological Images

Four regions may be observed in a cytological image; the background, erythrocytes (unnucleated cells), and nucleus and cytoplasm of the leukocytes. The background represents the lighter portions of the image (highest gray-level values), nuclei appear as the darkest portions of the image (lowest gray-level values), while erythrocytes display an intermediate gray. Other attributes such as cytoplasm color, nuclear shape and texture may differ from one image to another, depending on the particular type of cell. For isolated cell images, the photometrical, geometrical, topological and morphological information can be formulated and successively inserted in each of the processing stages. In the preprocessing stage, different smoothing algorithms can be selected based upon the relatively smaller and darker artifacts outside the cell images, and the brighter and smoothed texture regions inside the cell images.

In the processing stage the following should be taken into design considerations:

1) The cell images usually constitute intensity clusters. From a global point of view, the gray level decreases gradually and monotonically from the background to the center (nuclear region) of a cell. Such an image is very suitable for hierarchical extraction by successively excluding the outside background of the meaningful regions.

2) Fixed number of clusters which represent the background, nucleus and cytoplasm is very useful not only for initial setting and labeling for clustering and relaxation, but also for splitting and merging control in the region growing.

3) The nucleus is located inside the cytoplasm and has a smaller perimeter than the cytoplasm region. These geometrical features will help in discriminating the two meaningful regions, such as in boundary tracking.

In the postprocessing stage:

1) The topological relations of the nuclear and cytoplasm regions can be used for region and contour extraction. Proper region or pixel filling can be made to fulfill this requirement.

2) The cell boundary has good convex feature generally. This shape factor is very important for testing the protrusions caused by all of the segmentation approaches above. Various shape analysis methods can be used to locate the erroneous protrusions and reconstruct the regions and boundaries so as to eliminate or reduce the segmentation errors.

Properly inserting these criterions in each of the processing stages will help create algorithms with more discriminatory power and robustness. The resulted large reduction of the errors will make the automatic cytological image segmentation acceptable and applicable.

CHAPTER 3

CYTOLOGICAL IMAGE CONTOUR EXTRACTION USING BIT-PLANE THRESHOLDING

3.1 Introduction

In cytology, where large amounts of cell images are processed, accuracy and simplicity are two most important features for automated image segmentation. Thresholding, due to its fast and effective performance, is widely used in cytological image segmentation. Many isolated cells can be segmented by a straight forward thresholding, provided that the background generally appears rather well contrasted and homogeneous, or that the cell images have bimodal or trimodal frequency distribution. Thresholding techniques prove inadequate when the cell boundaries are poorly defined; the gray level histogram is unimodal; or the cells are very close together. In such cases, lack of available spatial information and reliable threshold finding techniques result in more segmentation errors generated than region-based or edge-based segmentation techniques.

The major initiative of the presenting bit-plane threshold technique is not aimed at finding the optimal thresholds, nor at locating adaptive local thresholds analytically, but rather attempting to find a midway solution that inherits the fast feature of simple thresholding and still be able to deal with the various problem mentioned above. To improve the effectiveness of the technique

attention is focused on the accurate modeling and handling of the image structures. From a global perspective, incorporating a priori knowledge about the image, the intensity values of an isolated cell images decrease as one goes from the background through the intermediate cytoplasm region and into the inside nuclear region. Additionally, the cytoplasm region and the inside nuclear region usually exist as two separate aggregates. Artifacts should not be recognized as meaningful regions due to their much smaller sizes. These hierarchical structures and a priori information provide good conditions for spatial image decomposition and selecting proper processing strategy.

Ideally, three windows defined by two thresholds are enough for extraction of the cytoplasm and nuclear regions from the background. Due to inadequate thresholding two windows may not be enough to properly detect the meaningful regions. Three situations may occur: if no window is properly located the segmentation fails; if one window contains more than one object, the thresholds are not suffice for segmentation; if an object overlap several windows there may be no adequate threshold consistent representation of an object. In the present boundary extraction algorithm we address the first two situations above. The design strategy is to set more thresholds to increase the probability of fitting different meaningful regions into different windows. Solution to the third problem can be addressed by determining which contour representation most likely meets a series of a priori requirements, such as geometrical (closeness and size), morphological (shape) and photometrical (contrast).

3.2 Bit-plane Morphological Edge Detector

Gray level cytological images are viewed as stacked binary bit plane images representing the effect of thresholding the gray level images with multiple gray level windows. Meaningful object segmentations are searched hierarchically while reducing the size of the windows. A new Morphological filter structure is designed to detect all the binary edges of the above multi-thresholded image into one image. It preserves the advantages of binary detection methods which produces more connected and thin edges and obtains very simple implementation.

3.2.1 Structure of the Edge Detector

To inherit the merit of binary edge detection method having accurate and thin edge, we decompose the 256 level cell image f to generate a few bit planes X_i with weight 2^{8-i}, where i=1,2,...,j, maxj=8, and a gray level image R which is a difference picture of f with weighted sum pictures of X_i. Thus, the gray level image f can be represented by:

$$f = \sum_{i=1}^{8} X_i * 2^{8-i} \qquad (3\text{-}1)$$

$$f = \sum_{i=1}^{j} X_i * 2^{8-i} + R \qquad j < 8 \qquad (3\text{-}2)$$

1) Use binary dilation to filter each binary picture X_i, i=1,...,j respectively with a small size SE, g, to produce the dilated binary

images X_{di}, i.e.

$$X_{di} = \text{Dilate}(X_i, g) \qquad \text{where } i = 1,2,\ldots,j. \tag{3-3}$$

After dilation, in each weighted binary image plane X_i, the light area has been extended (gray level 1) and the dark region has been shrunk (level 0) by the same number of pixels. Suppose in each i^{th} plane, the image consisting of newly generated "1" pixels is X_{ei}. Then we have that

$$X_{di} = X_{ei} + X_i \tag{3-4}$$

2) Sum up all the dilated pictures and the residual gray level picture R (when j=8, R will be a 0 level image) substituting all 1's in X_i with the correspondent weights. The summed gray level image f_s will be:

$$f_s = \sum_{i=1}^{j} X_{ei} * 2^{8-i} + \sum_{i=1}^{j} X_i * 2^{8-i} + R \tag{3-5}$$

3) To extract the edge information we subtract the original image from the summed image. Using(3-2) and (3-5) we obtain:

$$f_e = f_s - f$$

$$= \left[\sum_{i=1}^{j} X_{ei} * 2^{8-i} + \sum_{i=1}^{j} X_i * 2^{8-i} + R \right] - \left[\sum_{i=1}^{j} X_i * 2^{8-i} + R \right]$$

$$= \sum_{i=1}^{J} X_{oi} * 2^{8-i} \qquad (3\text{-}6)$$

where the X_i represents the invariant images (root) before and after dilation at each i-th plane. The two R's cancel out and the remaining image becomes the linear combination of binary edges at different binary planes. The block diagram of this edge detector is shown in figure 3-1.

3.2.2 Features and Discussion

This edge detector has the following features:

1) Flexible and robust:

A cell image detected by the edge detector is transformed into one multi-thresholded edge map. The edge map provides multiple choice and easy extraction of different contours both for the viewer and computer analysis and recognition. Due to its structure a hierarchical coarse-fine search is possible by successively increasing the number of windows to reduce the probability of missing targets. The multiple threshold setting does not consider any spatial information as a presumption, so that this setting can be used in a more robust way for segmenting cell images with different spatial contrast varieties.

If a cell image has bimodal or trimodel histogram then some thresholds will most likely be close to the valleys. For single modal histogram some thresholds are expected to be close to the optimal thresholds values required for segmentation. Although the detected edge

map may not be the most accurate or optimal, it contains the necessary contour information with good accuracy thus making it flexible and robust.

2) High sensitivity and adjustable resolution

The bit-plane thresholds can be adjusted close enough to detect any gray level transitions across the threshold values. This property is useful for detecting cell images whose edges are not well defined due to low contrast and detecting texture variations within the cells.

Edge resolution is a measurement of how sensible the detector can recognize the gray level variation. From (3-6) we see that the resolution depends on the number of levels, j. The larger the value of j, the higher the detector resolution, and the lower the edge contrast that can be detected. Thus to process an image, j is chosen to be the number of binary planes ensuring grey level difference greater than 2^j-1 will be detected. Grey level difference less than 2^j-1 may be detected only if the grey level transition crosses the threshold values corresponding to the bit planes. If we define S as the edge detection resolution representing the maximum number of gray level diffenreces that can be detected, then

$$S = 2^j - 1 \qquad j = 1,...8. \tag{3-7}$$

This can be also evaluated from (3-6), if j levels are chosen by taking all the combinations from 1 to j. The maximum number of

combinations N is:

$$N = C_1^j + C_2^j + \ldots + C_j^j = 2^j - 1 \qquad (3\text{-}8)$$

Thus, if j=8 is chosen, then the edge gray level can be represented from 1 to a maximum of 255 levels. If j=1 then the detector becomes a binary edge detector with threshold at gray level 128. If we choose j=4 the maximum number of edge gray level is only 15. These gray levels are equally spaced segments appearing on the histogram, and represent threshold windows. By varying the value of j these windows can be adjusted close enough to detect the existence of the object contours. The objects required to be detected in the cytological images are usually two or three and 3 or 4 bit planes are commonly used. Using more bit planes will raise the sensibility, but at the same time noise and false edges (will be discussed later) will increase rapidly making the edge picture more complex beyond our needs. Local contrast variations of the original image, many of them generated by artifacts, are detected and frequently mapped into edges with different threshold levels. This separability feature partially reduces the degree of the noisyness of the binary bit plane edges and relieves the heavy burdens of noise filtering and the extraction.

3) Simple implementation

The image can be processed using easily implemented binary techniques. Here we use binary morphological edge detection which

greatly reduce the computations and is suitable for parallel processing. Simple binary tracking employed on the well connected binary edges makes much less contour errors by avoiding misleading due to the noise and the contrast variations on the contours.

3.2.3 Bit-plane edge detection analysis

The number of generated binary bit plane edges (3-6) is a monotonically increasing function of levels selected. The more levels chosen, the more edges that are obtained. If we expect single gray level edges at the j^{th} binary plane, X(j), then it is equal to the difference between the j level edge image f_j and the j-1 level edge image f_{j-1} :

$$X(j) = f_j - f_{j-1} = \sum_{i=1}^{j} X_i * 2^{8-i} - \sum_{i=1}^{j-1} X_i * 2^{8-i} = X_j * 2^j \qquad (3\text{-}9)$$

The relations between these different weighted edges can be classified according to the following cases:

1) When the image contains sharp step edges with large contrast edges from different bit planes may overlap.

2) When the image contains slope edges, the edges in different bit planes are not overlapped, multiple edges are generated in the resulting image. Suitable selection of a single response is necessary.

3) When the gray level contour has large intensity variations or

very low contrast corresponding binary edges may contain protrusions which are not part of the original contour. The reasons for these errors are due either to the thresholds setting or the existence of neighboring artifacts with similar intensity. Thus tracking single binary edges may not be able to extract the best representation of the cell contours. The first problem can be partially solved by locating better threshold values discussed in chapter 4. The second problem requires efficient noise suppression as a preprocessing step.

Thresholding operations in general suffer from their noise sensitive feature. It is very difficult to set thresholds so that there is a small probability of selecting noise edges while retaining its high sensitivity. Bit-plane multi-thresholding generates more noise due to the application of multiple thresholds. The bit-plane binary edge detector will recognize and mark those bit-plane transition pixels as edge points even though the corresponding local contrast is very low. Therefore false edges are generated; when there are two or more responses to the same edge, only one of them should be considered as the true edge and the others should be considered as false edges. The false edges are defined as pixels with non-maximum gradient with respect to their neighbors. The amount of the false edges increases as the gray level window size decreases. Although, decreasing the window size will generate more precise information of contours, the large amount of false edge contaminations make the boundary tracking difficult.

False edges can be minimized by prefiltering, prior to edge

detection, where edge pixel candidates are selected and labelled by comparing each pixel with its neighbors. If the gray level difference exceeds a prespecified threshold then the pixel is considered an edge candidate; only edge candidate pixels are subsequently processed. As a result a large portion of the false edges are eliminated, but at the same time segments of broken contours are obtained. Various attempts[57,58] have been made to gain better separation between true and false edge portions on the contours. Edge fragmentation is not only due to the thresholding, but is also due to the fact that the edge operator output fluctuates above and below the length of the contour[59]. Thus, when we consider the elimination of the contour noise we must assume that the contour is not just an aggregation of pixels with maximal contrast, but also a connective set of pixels mixing with false edges.

At this stage, the best binary representation for the cell and nuclear boundaries is probably the most important approach for eliminating the false responses, although limitation is due to the application of the fixed thresholds. However, using reasonable a priori knowledge, such as size, shape and relative positions of the objects are of great help for discriminating the different object boundaries.

3.2.4 Some considerations in the design process

On each binary plane, when a dark object (0 area) is within the light environment (1 area) dilation tends to shrink the dark object and

structure of cell images, darker artifacts are distributed in the lighter background, dilation, because it will eliminate or partially shrink the much smaller artifacts, is suitable for noise filtering. A small size structuring element (3 by 3) is selected to generate connected edges. Before edge detection, image enhancement is often necessary for ill-posed boundaries.

Artifacts may be distributed anywhere within the cell image. Two situations have to be considered:

1) artifacts overlap the cell boundary.

2) artifacts very close to the cell boundary.

These situations will create either distorted edges or gaps that might create tracing errors. To separate the these artifacts from the objects and reduce effects of light shading morphological closing can be used. Closing can fill the gaps by shrinking the darker, smaller sized-holes. Because relatively small structuring elements are employed the closing will not distort the shape of the cell which is much larger than the SE. The size of the SE is determined by the degree of contamination the artifacts has on cell boundary; the larger it is, the larger the required SE. Selecting a larger SE to implement the smoothing operation will distort the inside structure of the cell. This does not pose a real problem if the boundaries preserve the shapes. Unfortunately, often the processed contours become jagged and serious distortion may occur most likely in the low contrast contour regions.

3.3 Tracking the Contours of the Cell and Nucleus

Here we present a cell boundary tracking algorithm based on the result of our edge detector. The binary morphological edge detector identifies possible edge point which may correspond to the cell and nuclear contours. The function of the contour tracking algorithm is to extract possible cell contour candidates in each bit plane by removing those contours that are either not closed and/or do not meet a user specified minimum length.

3.3.1 Introduction

In the past, a number of contour extraction algorithms have been proposed in the literatures.[60,61] A general block diagram for most of the contour extraction algorithms is shown in Fig. 3.2. Essentially, they consists of two steps: edge detection and thresholding followed by a thinning and linking step. The images for tracking are either the gradient maps or the bi-level images obtained after thresholding the gradient images. The purpose of directly tracing the raw edge image is to preserve the connectivity property of the edge elements so as to make the extracted boundary as continuous as possible, but noise due to artifacts or texture in cytological images often causes various tracing errors. To avoid the errors, the tracing programs [62,63] are usually complicate and time consuming. When tracking the thresholded edge image the edges are fragmented due to inappropriate threshold setting and thinning, especially when the

cell image boundaries are not well defined.

The bit plane edges are generally connected and thin so that binary tracing can be employed in each bit plane. The tracking algorithm can be designed to be simple and effective.

3.3.2 Basic tracing strategy

Raster scanning is used to search the starting point (SP) of an edge segment so that trace can begin. Edge pixels have a prespecified gray level. Having finished the trace of one segment, all the traced pixels are changed to another specific gray level to avoid retracing the old paths in the successive scanning, and the scanning will resume starting from the next point of the SP. Based on the fact that the discontinuity is not serious in the extracted cell edges, a small size mask is used (7*7 square matrix, see fig.3-3) to connecting discontinuities. If the center point C is the pixel under consideration the first inner loop includes all its 8 neighboring pixels with distance 1 to C; the second loop contains all 16 pixels with distance 2 to C; and the third outside loop accommodates all 25 pixels with distance 3 to C.

The search proceeds from the inside loop of the mask to the outside loop. This implies that pixels closer to the center C have the priority to become part of the contour. Once the pixel is made part of the contour this pixel becomes the new center C.

Three possible situations are considered:

1) If only one pixel is found in the inner loop, it means that it

belongs to the contour. Nothing has to be done except move the center to that pixel. If only one pixel is found in the second or third loops, it implies a broken edges and one or two pixels have to be added using linear interpolation technique.

2) If no such pixel is found in the whole mask the coordinates of C together with the starting point will be registered as a break point pairs.

3) If more than one pixels are found in a loop, then we have to determine to which pixel the center C should be connected. Slope feature of connected line segments are selected based on the smoothness of the cell and the nuclear boundaries. The slope of the line segment connecting the current center pixel with the previous center pixel is calculated as K, and the n slopes of the segments between each candidate center pixels and the current center are also calculated as K_1, K_2, ...,K_n. Then one pixel j, $j \in n$, is connected to C if the slope satisfies the minimum slope difference rule:

$$K_j = \text{Min} [K_i - K] \qquad i=1,2,...n. \qquad (3\text{-}10)$$

Linear interpolation is applied for the edge filling. The flow chart of the distance slope search a nd connection algorithm are shown in fig. 3-3.

3.3.3 Tracing and filtering

The main program will repeatedly call the tracing sub-program which

provides data necessary for the main algorithm to determine the real contours as shown in figure 3. The major steps are as follows:

1) The main objective of the first scan and trace is to locate the longest segment with length MP after comparing all the closed or non-closed segments registered during this step. If the longest segment is closed and meet the minimum length requirement (we use 200 pixels), then the boundary candidate is generated and the program terminates. A segment is closed or not is judged by whether or not a tracing pixel entering the 3*3 square region with a SP as the center point.

2) During the second trace, each closed segment with length smaller than a prespecified value T is deleted. The size threshold T is empirical data, and is varied with MP (we choose T = MP/8 initially). The longer the MP is, the longer of the closed segments will be removed. The unclosed segments longer than T will not be filtered out because they may be located on the boundaries. The second longest segment (NP) is also found. Each unclosed segment (including MP) is stored with its two break points (BP).

3) In the third tracing, break points are connected based on the nearest neighbor rule using linear interpolation. To avoid mis-connection, the break point pairs of MP and NP are assigned higher priority for linking as MP and NP are most likely to be part of a boundary. Hopefully, the extension of MP and NP will generate a closed boundary. If MP increases, go back to step 2 for further filtering, and closeness test. If MP is closed, the extraction succeeds and terminates. If MP remains unchanged and and still not closed, it

means that the tracing leads to a dead end and small branches or loops have to be eliminated. After pruning, the program goes back to the first trace.

3.4 Contour Selection and Discrimination

Each bit plane is tracked in order to obtain the closed contour that meets the minimum length requirement; some bit planes will not generate such closed contour. If more than one such closed contours are found the cell contour and nuclear contours are discriminated by size and the second central moment. This is because the nucleus is located within the cell and although it may occasionally have longer perimeter than the cell contour, it will have a smaller second central moment. For more contours meeting the minimum perimeter requirement, selection is based on the circularity requirement, as the cell contours have convexity feature normally. Due to the threshold setting discrepancy, and cell image contrast condition, it is possible that no candidates or only one candidate satisfy the closeness and size conditions, further extraction process is necessary, which will be discussed in next chapter, or the user is prompted to manually trace the contour.

3.5 Region Segmentation and Feature Measurement

Edges of the cell, nucleus and chromatin structures can be extracted separately. To further investigate the internal structures of the cell

or nucleus, the cell and nuclear region recovery is necessary. For normal cell boundaries that are convex in shape, the region filling is easy. Since the nuclear contours may have non-convex shapes, parity check[63] or connectivity filling[65] methods can be used to detect the regions inside or outside the boundaries. Here we use 4 direction scanning and filling to paint the outside of the boundaries. Scanning and filling are performed downward, upward, towards the left and towards the right. The scanning along each column or row stops at the first edge points detected. To operate region filling, we can paint the regions outside and inside the boundaries to the two gray level extremities (255 and 0) to form a background picture f_b. By taking the maximum pixel gray levels of f_b and the original picture f, we will obtain the recovered cell images f_c by the following operations recovered nuclear can be obtained similarly):

$$f_c = \mathbf{Max}(f_b, f) \tag{3-11}$$

All geometrical measurement, such as perimeter, area, shape, centroid, medial axis, relative position for different portion in a cell image, etc. can be executed based on the extracted edges. After the segmentation, mathematical analysis can be implemented on the segmented image to extract the textures features and other high level feature generation or directly used for decision function generation in image recognition system.

In summary of this chapter, the structure of a new morphological edge detector for the extraction of cell and nuclear contours in cytological specimen is presented. Grey level images are converted to binary images by a bit plane multi-thresholding technique. A simple binary morphological edge detector which is coupled with a simple contour tracking algorithm to identify the most likely cell and nuclear contours. A priori knowledge about cell image is used in both tracking and contour selection. The method is applied in edge-based region segmentation on the cell images. The preliminary results show that extracted boundaries are well consistent with the original image.

CHAPTER 4

MULTI-THRESHOLDING BOUNDARY EXTRACTION USING EDGE MATCHING

4.1 Introduction

The bit-plane Multi-thresholding (BPM) algorithm and binary tracking discussed in chapter 2 are effective method for cell boundary extraction when the boundaries of the cell and nucleus are relatively homogeneous and the cell image has no serious texture variations inside. The homogeneous boundaries tend to be located in one bit plane or two adjacent planes. The generated contours are generally spatially close together, thus extracting either edge as the boundary from the multiple responses will not result in large contour errors. The assumption that there is no serious contacts between the contours and textures makes the extraction of the contours relatively easy and less prone to errors.

If the boundaries of the cells have large intensity variations, the multiple edges corresponding to these variations might be well separated. Even weak textures will generate multiple edges contacting the cell and nuclear boundaries on the various bit planes. Thus simply making selections based on the size, closeness and circularity lacks the necessary discrimination power for efficient automatic boundary selection and extraction. As the method stated in the previous chapter, only size and shape factors are used. The most important feature for contours, local contrast information, has not been fully applied. A

pixel can be a boundary pixel only when it satisfies local contrast requirements, otherwise it is just a false edge pixel. A bit plane binary edge is a good representation of the original cell image only if the two boundaries are approximately coincident. Thus it is reasonable that a bit plane edge candidate is accepted or rejected based on the coincidence of an edge map with the region boundary and only those candidates that best match the gradient map are used to describe the actual objects in the image. In addition, instead of blindly search every bit plane contours, we now can specify which contours should be searched.

Milgram[63]proposed an algorithm of evaluating edge/border coincidence on infrared (FLIR) images. A gradient map is generated and thinned, and the coincidence evaluation is made on all the borders of the thresholded image obtained at all gray levels above the mode of the histogram. Coincidence score, size and contrast are used for the final candidate selection where the contrast was measured by the absolute difference of average gray level between the border region of the component and its interior.

Loishertz and Schafer presented a multi-thresholding edge matching algorithm[64]. A blurred edge map of a thresholded image is matched with the thresholded gradient image, the difference of the two images is used as an indication of the number and location of the false edges. The program then enters into these areas for local processing by assigning new local thresholds to eliminate the false edges and add new edges.

These two methods, because of their time consuming exhaustive searching and complex operation, are not suitable for cytological image batch file processing.

In this chapter we propose a bit plane thresholding method that matches each of the thresholded bit plane borders (BE) with the thresholded gradient (GE) map by evaluating the coincidence of the two edges. Edge busyness is defined on each BE by counting the number of pixels included and is assumed to be an important factor describing noisyness preventing the correct contour extraction. The coincidence and the edge busyness are combined to generate a criterion for measuring the extractabilities of different BE. This criterion is then used for a suboptimal search for proper thresholds on the shifted cell images.

The matching images are generated by morphological operation to make the presented algorithm as simple as possible to suit cytological image segmentation. Experiment on the cell images, for which histogram and/or gradient information is ineffective for segmentation, show that segmentation by the proposed algorithm is good.

4.2 Bit Plane Multithresholding

A 256 gray level cell image f are viewed as stacked binary bit plane images with each bit plane image representing the effect of thresholding the gray level image with multiple gray level windows. If X_i is defined as the i-th bit plane with weight 2^{8-i}, and R is the residual image. Then:

$$f = \sum_{i=1}^{j} X_i * 2^{8-i} + R \qquad j = 1,2,...7 \qquad (4-1)$$

where j is the number of bit plane. If j is selected to be 3, only 3 most significant bit planes are used for the scene decomposition. The binary bit plane image X_i corresponds to the thresholding of the gray level image using 2^{j-1} gray level windows using uniformly spaced and of length 2^{8-i}. Thus, X_3 would corresponds to the third most significant bit plane of the gray level image with the following set of four gray level windows of length 2^5=32 gray levels (32-63,96-127,160-191,224-255), i.e., all pixels in the gray level image having values within the four ranges will have a value of one in X_3 otherwise it will be zero.

Each of the generated binary bit plane images X_i are then processed by a binary morphological filter for edge detection. If $d(X_i, S)$ represents the binary morphological dilation of X_i by a small structuring element S, then the binary edges(BE) on each bit plane can be obtained by the following operation :

$$BE(i) = d(X_i, S) - X_i \qquad i=1,2,...,j \qquad (4-2)$$

The contour tracking algorithm is used to remove contours that are either not closed and/or do not meet a user specified minimum length requirement. The most likely cell and nuclear contours are determined by the circularity test or directly presented to the user for selection.

This bit-plane multithresholding and contour extraction method is simple and performs very well when the contours of cell images are well defined.

Two questions remain:

1) When one region of cell image has very low contrast, such as the diffused contour of the nucleus or low contrast portion. It is possible that no bit plane edge will closely coincide with the boundary and results in extraction failure or the contour extracted has visible errors. One reason is that using thresholds spatially separated by 32 gray level are not adequate for detecting all the low contrast variations. Since adjusting the thresholds by increasing the number of bit plane thresholds will increase the sensitivity of the edge detector to the low contrast variations, smaller contour errors can be expected.
2) The related problem results when increasing the number of bit plane thresholds j directly; the detected bit plane edges will also be increased accordingly and this makes the edge pictures too noisy for contour extraction. A technique of increasing the number of bit planes without increasing the noisiness of the generated edge pictures should be taken into consideration. It is reasonable to set up an optimal or sub-optimal criterion analytically that can be practically used for the contour selection.

4.3 Matching each BE with the gradient image

Smoothing the original cell images by morphological closing or opening by a small structuring element is used as a preprocessing step

because these operation will reduce the contrast variations of the smaller artifacts and still preserve the edges of the objects. Large structuring elements have larger smoothing effect but possibly distorts the geometrical structure of the images. A thresholded image is a good representation of the original image if their corresponding edge images are approximately coincident. Local contrast variation is fully reflected by the gradient of the original cell images, and this information may be determined through a variety of techniques(11,12,13). We select a morphological edge detector with the same structuring element S as in the BE generation. Similar gray level dilation D(f,S) is implemented and is much faster than the conventional edge detectors. If GE represent the gradient image, then:

$$GE = D(f,S) - f \tag{4-3}$$

To evaluate the coincidences of each BE with the gradient image the gradient image has to be binarized by thresholding. To select the threshold the main consideration is that the binary edge, after thresholding GE, should be able to include all or the main portion of the original image boundary. Unfortunately, because the contrast intensities of the cell and nuclear boundaries often have large discrepancies for different classes of cell images, improper threshold setting will either miss one boundary when the threshold is too high or make the thresholded image contain too many gradient pixels when the threshold is too low. Thus, extracting the two boundaries simultaneously seems to be an error prone scheme and extracting the contour

hierarchically is preferable. Predicates of the cell perimeter are used for selecting the appropriate threshold T. If N_P represents the approximate perimeter of the cell, then N_T, the number of pixels above the threshold T can be considered as a function of N_P and the noisyness of the original image. In the image considered N_P generally ranges between 200 to 500 pixels and N_T can be considered as 2 to 3 times the value of N_P. Thresholding by T so that approximate N_T pixels are above T is close to gradient percentile thresholding at about 4-6 percent. Experiments show that the cell images either contain certain ill-posed boundaries or texture and artifact contaminations, the thresholded gradient images still contain the main portion of either the cell or nuclear boundary pixels. Thus, GE is thresholded by T which leave N_T pixels as 1 and the rest as 0. If the thresholded GE image is defined as TGE, then:

$$\begin{aligned} TGE(i,j) &= 1 \text{ when } GE(i,j) \geq T \\ &= 0 \text{ when } GE(i,j) < T \end{aligned} \qquad (4\text{-}4)$$

The N_P or N_T values only determine how many of the gradient pixels involved in the coincidence evaluation. Accurate prediction of these values is not necessary since we only compare the relative coincidences of each BE. As long as they are not specified too small which will violate the minimum size requirements or too large which makes the matching lose the discriminatory power.

Each of BE is matched with TGE by calculating the coincidence as matched pixel numbers N_C. We can expect that the BE with the highest

match score N_c may well coincide with the boundary of the original cell or nucleus. From the experiments of different cell images, if N_T is equal to $2N_P$, the highest scores will be around 50 to 70 percent of N_T.

4.4 An optimal criteria for searching root edges

The coincidence pixel numbers N_c (or match score) of each of the BE does not reflect information about the difficulty of the correct contour extraction. Here we define N_t as the total number of pixels in each of the BE representing the edge busyness and the difficulty in correct extraction. Normally, if different BE have similar N_c values, contour extraction will be easier in the BE with less pixels. In defining a figure of merit for measuring this feature the ratio of matched pixels N_c to total pixels N_t in the BE is used. Since the least significant bit plane contains the gray level variations crossing the smallest windows, the corresponding BE most likely has the best match score, but it contains much more pixels than other BE which prevents contour extraction. In some cases, BE of significant bit planes have scores close the best one, usually they are much less busier and very easy for extraction.

If we define Nc(i) as pixel number of coincidence and $N_t(i)$ as the total number of edge pixels in the i-th BE. We can define one heuristic and simple contour extractability measurement for compromising the match score and the edge busyness F(i).

$$F(i)= \frac{\bar{N}_c(i)}{1+ K*N_t(i)/N_c(i)} \qquad (4\text{-}5)$$

When $\bar{N}_c(i)$ is larger than N_p, $N_c(i)$ is set to N_p to weaken the priority of a low bit plane BE becoming the root edge candidate. K is a factor for adjusting the weight of the edge busyness for different images, in our experiment we use K=1.

A root edge (RE) is selected from BE with the largest F(i) and represent a compromise between coincidence and edge busyness. It can be used to find better boundary representatives from more bit plane edges.

4.5 Determination RE from bit plane edges of shifted image

The contours extracted from the cell images using the presented bit plane multithresholding scheme discussed in chapter 3 are in general good, but some noticeable errors still exist when compare with the original image. The reasons for the boundary extraction errors are either the large intensity variations appearing on the boundary or the threshold setting discrepancy; cell images that contain ill-posed boundary are common. The question arises whether a boundary with small contrast can still be detected correctly. We believe that in most cases the contour errors originate from the bit plane thresholds having not been fully adjusted to the right positions. More specifically, the current used bit plane thresholds separated by 32 or 16 levels might not

be adequate to discriminate two regions without causing noticeable segmentation errors. Increasing the number of bit plane can raise the sensitivity of the contour detection, so as to increase the discriminatory power of two regions separated by weak contrast boundaries. Simply increasing the bit planes might be inappropriate for contour extraction since as the window size decreases the edge busyness also increases drastically. Instead, gray level shifting a number of the most significant bit planes up or down will help to locate better thresholds without increasing the edge busyness. Shifting functions by finding the best multiple windows to fit in the meaningful image regions. Shifting a image by limited range is applicable since usually the highest and lowest intensity regions in the histogram contain no boundary contrast information. In the application, shifting is operated by subtracting the original image by a small gray level. Bit plane multithresholding is operating the same way as before. For example, if 3 most significant bit plane thresholds are selected, with lowest bit plane weight of 32, the original image is preprocessed by subtracting gray values 0, 8, 16 and 24, (subtracting 0 means to keep the original image), and then processing with the BPM respectively. Thus, instead of generating 3 bit plane edges, 12 bit plane edges are generated. In the non-shifted image bit plane thresholding technique we can create 32 windows of width 8, by using the 5-th most significant plane. However, the edge picture is too noisy to be used. Here, we create 8 windows of width 32 in each of width 32 in each of the 4 shifted images. The effective window width of the four shifted image is still 8 but edge busyness remains at the same

level as that of the 3 bit plane thresholding. This means the sensitivity is greatly increased without increasing the number of the windows. One of the BE with the largest F value is considered to be RE.

In summary of this cahpter, a refinery of bit plane multithreholding algorithm is presented. Bit plane edges (BE) that most coincide with the cell and nuclear boundaries are selected by calculating all the coincidences of the BE with the thresholded gradient map of the original cell image; a measurement reflecting the degree of difficulty in contour extraction is defined as edge busyness. A figure of merit is developed by combining the match score and edge busyness of BE and is used for contour selection; finaly, the bit plane thresholding method is extended to the shifted cell images and the selected root edges by the merit not only have good conformity with the real boundaries, but also easier extractable feature.

CHAPTER 5

CYTOLOGICAL IMAGE REGION SEGMENTATION USING LOW RATE VECTOR QUANTIZATION CLUSTERING

5.1 Introduction

Image segmentation techniques fall into two classes: edge detection approaches and region growing or clustering approaches. The major disadvantage of edge-based segmentation is its sensitivity to the noise, more error rate is generated comparing with region segmentation approach. But conventional region growing techniques suffer from large computation complexity which is unsuitable for large amount of cell image processing requirement.

In this chapter, a segmentation algorithm based upon sequential optimization which produces a hierarchical decomposition of the cell images is presented. A spatial low rate VQ is first employed to generate coarse segmentation. The generated codewords are then used to generate a fine codebook representing all the possible edge information. Fine segmented cell image are obtained by searching the nearest codewords and replacing the coarse code vectors. After region segmentation a higher rate VQ is used to segment texture regions inside the cell spatially. Different texture features are extracted in a heuristics way based on the codeword distribution and the

geometrical measure on the extracted textons. Preliminary results show that VQ technique used in the cell region segmentation inherits the computational simplicity of the edge-based approach and noise insensitive feature of the region-based techniques.

5.2 Segmenting Cell Image by Vector Quantization

Cell image is processed by spatial VQ using very low rate initial codebook with small dimension to create coarse segments. intensities of the codevectors are used to create larger codebook representing edge information for the vectors in the original image to find the replacements. Blob detection and small blob elimination is used for extracting the masks of the cell and nucleus.

5.2-1 Initial codebook selection

The initial codebook consists of N vectors of k dimension as an initial guess of final vector codes or codewords. Intuitively, we assume that the initial codebook can be easily generated and is close to the final codebook. Since prior knowledge about the final codebook is generally unavailable initial codevectors can be determined based on the image histogram. Correct assignment of the initial codewords will accelerate the speed of the convergence of the VQ algorithm and greatly reduce the computation. If an isolated cell image that consists of nucleus, cytoplasm and background regions has evident tri-modal feature

initial codevectors assigned to each of gray levels of the histogram modes should be a good guess since smaller interclass variances are expected between the valleys. If no model information can be used the initial codewords can be uniformly distributed within the dynamic intensity range of a cell image since the initial codeword assignment is relatively insensitive to the resulted final codebook. Three homogeneous codevectors are assigned as an initial codebook in this way and we assume each codevector will represent an initial guess of each meaningful region. Selecting the number of the codewords exactly pertaining to the meaningful regions avoids the complicated merge-and-split procedures.

A VQ operating at very low rate imposes a natural restriction on the dimension of each codevector. Small block length vectors are used since the lower dimension introduces less distortion in the reconstructed images for a fixed bit rates. Also, since the computation complexity increases exponentially as the dimension increases, much faster VQ operation can be expected with vector of low dimension. Although, using small dimension vectors will sacrifice the noise smoothing performance relative to using larger size vectors it can be compensated by smoothing operation in the preprocessing stage. We consider initial codebook consisting of three uniformly distributed, homogeneous code words with each codevector of dimension 4 (2 by 2).

5.2-2 Coarse Segmentation by Low Rate VQ

The VQ algorithm employed in this research is called the LBG algorithm by Linde et al (42). A full search VQ is used in the spatial domain where the mean square error (MSE) criterion is applied for partition optimization. The VQ evaluates the Euclidean distances of each 2 by 2 image blocks with all the initial codewords. The image vectors are stored into the nearest partition represented by the N initial vectors using the least square distance. After the first iteration, all the image vectors are clustered into maximally N partitions. For each partition, new codewords are generated by taking the average (centroid) of all the vectors in each of the vector group. The new codewords are used for distance calculation in the second iteration. The relative average MSE after each iterations is checked, if it is smaller than a prespecified value the program stops and each of the image vector is replaced by searching the nearest codevector in the final codebook. Otherwise, the iteration will continue.

In addition to regular VQ features (see section 2.4), the algorithm has the following advantages in cell image segmentation:

1) The program processes the cell image very fast due to the low rate and low dimension of the codevector selection. Except for blocky edges, simple and basic region structures are revealed with good approximation. Also it is suitable for processing large amount of the cell images of of the same categories stained and viewed under similar conditions because the final codebook of a single image (or

generated from several such codebooks) might be eligible for common use, or at least, for a good initial guess.

2) Due to a large amount of averaging, low rate VQ will generate nearly homogeneous codevectors. Partitioning the cell images with such well separated homogeneous vectors greatly reduces the negative effects of the noise, the weak textures, and the broken edges in the correct segmentation.

The disadvantages are as follows:

1) If adjacent regions have similar gray levels then their regions may be incorrectly connected and segmented as a single region. This may be corrected by further splitting the codevectors or by incorporating shape analysis.

2) The coded cell image has evident blockness on the edges due to the small size codebook selection and use of the MSE criterion. To reduce the block effect using a larger size codebook which represents more patterns of the edges can be considered as a solution.

5.2-3 Refine the Coarse Segments by Codebook Replacement

To more accurate represent the region boundaries of the segmented region the blockness has to be eliminated, and if possible, the intensity difference between adjacent regions that are enormously connected need to be revealed. Increasing the size of the final code book by adding all the possible edge patterns is expected to accomplish this goal.

The codebook with three codevectors is extended by adding 28 edge vector patterns. These patterns reflect all possible edge models in 2 by 2 squares constituted by 2 values from the neighboring codevectors. To preserve the simple segmented structure of the coded image the edge codevectors are heuristically determined based on the combination of the 3 near homogeneous levels of the coarse codebook.

Suppose Y_0 is the coarse codebook containing codevectors C_1, C_2 and C_3, where the value deviation between the elements in each vector is usually small. These three vectors can be considered as the three smooth area representatives in the extended codebook Y_1. The gray levels of the first elements of C_1, C_2, and C_3 denoted by C_{11}, C_{21}, and C_{31} respectively are used to construct a 28 edge codevectors added to Y_1. Considering that large gray level differences between the cell regions coded by C_1 and C_3 rarely appear as neighbors due to the structure of the regular cell images the possible edge code vectors between these two regions are not added in Y_1 to reduce the codebook size and computations.

Y_1 is not involved in iteratively determining the optimum partition, It is only a larger codebook heuristically generated to code the original image. It is generated right after the optimal code C_1, C_2 and C_3 are obtained but before replacing the original image. The coded image is obtained by looking up each of the 31 codevectors in Y_1, and each original image vector is replaced by the nearest codeword to reconstruct a fine segmented image.

The new segmented cell image effectively eliminates the block effects. Although the number of codevectors are increased to 31 the coded image only consists of the same small amount of gray levels as the coarse vector coded image but has the feature of being very suitable for region extraction. The structures of the coarse codebook and fine codebook are shown in Fig.5.2.

5.2-4 Blob Detection and Filtering

Three clusters with well separated levels in the fine coded images provide a convenient way of thresholding. Two thresholds T_{12} and T_{23} can be set in the broad valleys between the three clusters to generate the masks of the cytoplasm and nucleus associated with smaller blobs representing the vector coded artifacts regions. Blob detection techniques can be used for merging the included small blobs into their larger background. The segmentation errors caused by the merging of neighboring regions with similar gray levels can only be detected amd corrected by using a prior knowledge of the shape or other parameters.

Here the blobs are detected by the Multi-threshold morphological edge detector discussed in chapter 3. The difference is that the edge detection operates on the thresholds values T_{12} and T_{23} instead of on each bit planes. Again, only the two closed edges with largest perimeters are considered as the boundaries of the cells and nucleus, and all other small edges are eliminated.

As a final note of this section, either for the presented low rate VQ segmentation algorithm or for small blob elimination, prefiltering has proven advantageous. The gray level opening or closing are used for smoothing the original cell image based on the outside artifacts and inside texture conditions. Generally speaking, if the outside artifacts are the main source of contamination in a specific class of cell images (such as blood cells), closing operation is preferred. If cell images contain serious texture within the cells (such as the breast and lung cells images) opening is selected for smoothing. Because of the low sensitivity feature of the low rate VQ smoothing will not greatly affect the accuracy of the final segmentation since cell and nucleus are much larger clusters than the noise. However it might help to eliminate those artifacts that cause protrusions, or at least, separate those artifacts from the cell or nuclear boundaries.

5.3 Texture Extraction and Measurement

Cell classification is directly related to texture feature. Whether a cell is malignant or benign is often determined by its nuclear structure; whether the chromatin particles are of equal size; whether the chromatin particles are evenly distributed throughout the entire nucleus; whether each part of the nucleus resembles every other part, etc. Texture in the cytoplasm is also examined for the purpose of identifying the type of cell.

The texture in the cell images is normally distributed in a random fashion and rarely repeated texture structures be found even in one cell image. Additionally, all the texture variations aggregate in relatively small regions of the image. To extract accurate information texture analysis should be based on the correct segmentation of the texture regions. In the next section, a texture segmentation and analysis based on the VQ are presented.

5.3-1 Texture Segmentation

The segmented cell and nucleus are recovered for further texture segmentation and analysis. A texture element (texton) is assumed to be a region of homogeneous gray level; the problem of extracting elements in an image is equivalent to the problem of segmenting the image into homogeneous regions. VQ is recalled for further extraction of these regions.

In order to more accurately represent the texture regions, more codewords can be selected (up to eight 2*2 initial codevectors), and the MSE for each cluster is greatly reduced as the number of clusters increases. The number of codevectors is selected based on the dynamic range of gray levels occupied by the region being processed. More codevectors are seleicted for a cell image with larger dynamic range and less for a cell image with smaller dynamic range. Since normally texture within the cell or nucleus reprensets weak gray level variations, this a priori information can be used for designing a much smaller fine codebooks for codevector replacement. If C_1, C_2,...,C_j are

the codevectors obtained from the coarse codebook where maxj = 8; C_{11}, C_{21},..., C_{j1} are the first components of each of the codevectors. We construct the fine codebook by selecting every two neighboring gray levels C_i and C_{i+1} as the components of a new codevector, where i = 1, 2,..., j-1. Thus the number of the fine codebook will contain N code vectors which

$$N = (j-1) * 14 + j \qquad (5\text{-}1)$$

where j is the number of near homogeneous codevectors; 14 is the number of edge patterns of using two levels filling into a 4-d vector. The number of codevector is greatly reduced since if we use j level values to fit in a 4-d codevector nomally $N = j^4$ codevectors are required to obtaine all the patterns. Even if we only include the non-neighboring vector patterns constituted by 2 values $N = C_2^j * 14 + j$ are needed, where j≥2. Since no spatial information is assumed in the extraction. The initial codewords are assigned uniformly in the dynamic range. Using more codevectors will create more complex texton models, but the coded texton regions tend to be non-homogeneous.

Each near homogeneous texton region can be extracted by matching the codeword with the coded images which makes the structural measurement convenient. Although, measurement such as size, shape, perimeters of the texture boundaries and relative positions between different regions can be used for describing the texture features inside the cell images more heuristic features can be generated by considering the structures and distributions of the final codevectors.

Block diagram of region segmentation using VQ is shown in Fig.5.1.

5.3-2 Properties of the Texton Elements

The following properties of a texture element are measured:

1) MSE of each cluster.
2) Codevector homogeneity.
3) Inter-cluster variance.
4) Lengths of texton edges.
5) Edge per unit code area.
6) The ratio of the number of the edge codevectors to total number of codevectors.

Mean square error is the natural product of VQ. Larger MSE of a cluster is mainly caused by the abrupt intensity change of the texture edge areas. Most likely the more texture edges of the original image included in one cluster, the larger the MSE. This is so because the VQ encodes a region with more detail poorly than it does relatively uniform region. A cluster with larger MSE most likely corresponds to texture busier image region. To classify different textural encoded images, the average MSE over the number of N clusters can be used as a factor. If MSE(i) is the MSE of the i-th cluster, i = 1, 2,..., N, then

$$NMSE = 1/N \left[\sum_{i=1}^{N} MSE(i)\right] \qquad (5\text{-}2)$$

If the coded image maps $n_1, n_2, \ldots, n_N$ data vectors into each of the codewords $C_1, C_2, \ldots, C_N$ respectively, and the dimension of each codevector is k, the homogeneity of each codevector can be measured by

the variance of the k vector elements with respect to the element mean values. If c_i is the mean value of the k components in C_i, i = 1,..., N, and $C_i(j)$ is C_i's j-th component, j=1, 2,..., k, the variance e_i of the i-th code can be defined as :

$$e_i = (1/k)*\sum_{j=1}^{k} | C_i(j) - c_i |^2 \qquad i= 1, 2,..., N. \qquad (5-3)$$

The uniformity measurement of the codewords, E, will be:

$$E = 1/N \sum_{i=1}^{N} e_i \qquad (5-4)$$

This measurement describes all the micro-texture variations contained within the codevectors. If an image is coded by nearly uniform codewords, then E will be vary small. A normalized uniformity measurement of codevectors U can be defined as:

$$U = E/(1 + E) \qquad (5-5)$$

Inter-cluster relation can be described by the distributions of the N codevectors. Well separated codevectors tend to represent strong texton edges while close vectors weak texton edges. If a cell image is relatively uniform inside the number of total clusters may be less than N due to the elimination of one or more codevectors which represent no vectors in the original image (empty cells), or at least, one or more clusters only contain small number of codevectors. To discriminate such texture differences we define c_i as the component

mean of the i-th codevector, where i=1, 2,..., N, and actual number of final codevectors N_1 where $N_1 \leq N$, and D as the level difference between the largest code mean and the lowest code mean, $\bar{c}$ as the mean of the mean values of all codevectors. Then V can be used to describe distribution of the codevectors is defined as follows:

$$V = D * \frac{\sum_{i=1}^{N1} (c_i - \bar{c})^2}{N} \qquad (5\text{-}6)$$

The number of clusters represents the business of the textures in the original image. The more blobs that are coded by the codevectors the longer the boundaries encircle the textons. Since the coded blobs are easily extracted the perimeters of the blobs can be readily obtained. Suppose the total length of the boundaries is L and the total cell or nuclear area is S, then A can be defined as the mean boundary length over the total number of coded area S, representing the textural business.

$$S = \sum_{i=1}^{N1} n_i \qquad (5\text{-}7)$$

$$A = L/S \qquad (5\text{-}8)$$

If we define n_e as number of edge codevector, n as total number of the codevectors being used then a ratio R can be defined to describe the texture activities

$$R = n_e/n \qquad (5\text{-}9)$$

CHAPTER 6

EXPERIMENT RESULTS AND DISCUSSION

6.1 Introduction

The multiple bit plane thresholding with morphological edge detection method described in chapter 3; the sub-optimal edge detection on shifted images discussed in chapter 4; and the region and texton segmentation method using vector quantization presented in chapter 5 were applied to different classes of isolated cell images. These experiments were conducted in order to investigate:

1. The accuracy of different segmentation approaches for various cell images with spatial information.
2. The effect of morphological filtering on artifacts of different sizes and shapes.
3. The effect of using a priori information of reducing the segmentation errors and human interaction.
4. The gray level MSE values of the segmented texton regions with the original cell images at different rates.
5. The speed of implementation of the different algorithms.
6. Compare the advantages and disadvantages of different approaches.

The cell images were obtained from a personal computer based image processing system described by Jeanty et al(68). The operator selects

a region of interest containing the cell to be analyzed via a touch screen mounted on the face of a television monitor. The structures of interest are the cell and nucleus and their contours.

It is difficult to find reliable measures for the quality of image segmentation; little has been published on how to measure segmentation errors. We have used a direct subjective measure based on the visual judgment of observers to assess segmentation quality.

6.2 Results of Bit Plane Thresholding and Morphological Edge Detection

Figure 1 are four typical original cell images for segmentation. Figure 2, 3, 4 and 5 represent the procedure and the results of applying the bit plane multi-thresholding combined with the morphological edge detection method described in chapter 3. Figure a represents the original images; figure b represents the results of thresholding using four most significant bit planes and morphological edge detection with a 3*3 binary structuring element; for better visibility, some edges have been enhanced. Figure c represents the largest closed contour on the four bit planes meeting a minimum length requirement of 200 pixels and the best circularity and is presented as the most likely choice for the cell contour; figure d represents the extracted cell corresponding to the contour contained within the area defined by the contour in figure c; figure e represents the second longest closed contour extracted from the bit plane edges which is the most likely nuclear contour; figure f represents the extracted nuclear region.

The extracted contours superimposed on the original images, representing the quality of the presented segmentation method and the visible segmentation errors are shown in figure 6.

Figure 1 a, b, c and d. The four original cell images.

Figure 2 and 3 a, b, c, d, e, and f represent identical processing as described for figure 1. The original images are first smoothed by morphological closing using a 3*3 binary structuring element.

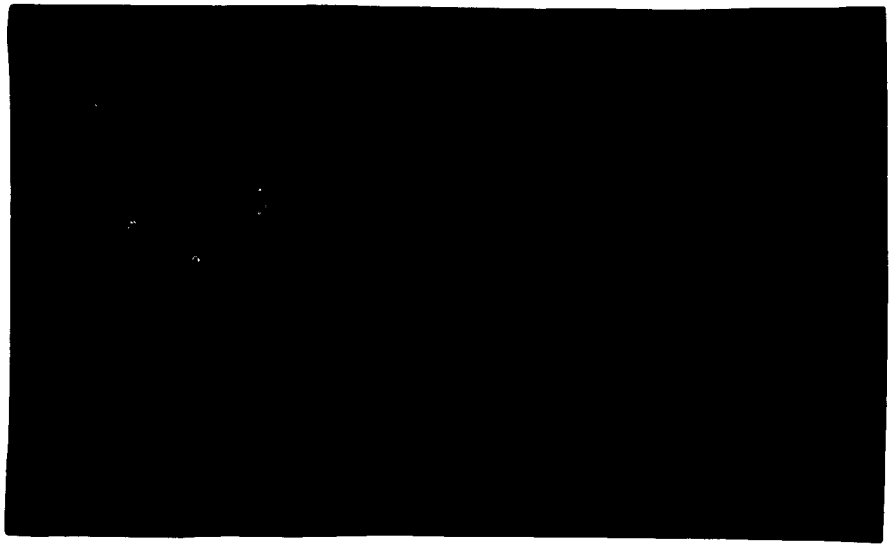


Figure 2. a) Original. b) Edges detected on the four most significant bit planes. c) First choice of cell contour by tracking algorithm. d) Extracted cell image. e) First choice of nuclear contour by tracking algorithm. f) Extracted nuclear image.

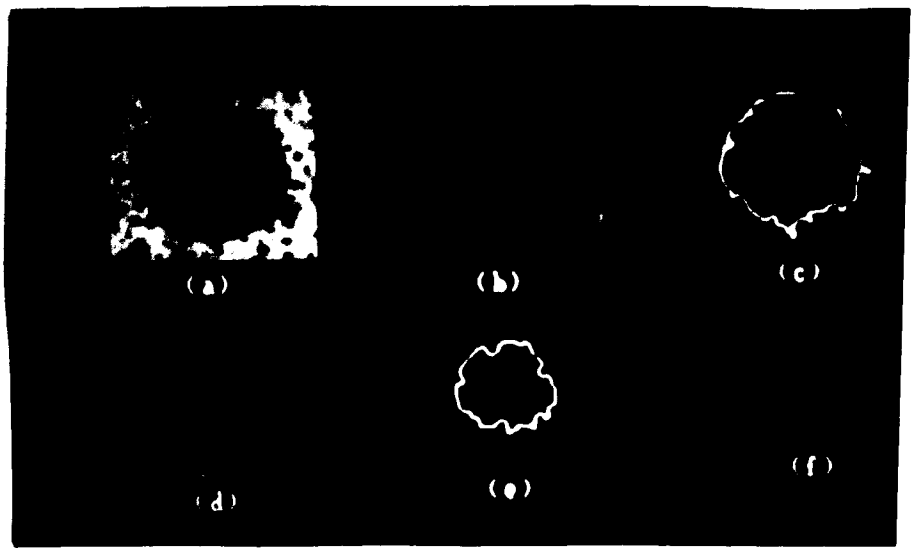


Figure 3. a) Smoothed image. b) Edges detected on the four most significant bit planes. c) Tracking algorithm first choice for cell contour. d) Extracted cell image. e) First choice of the nuclear contour by tracking algorithm. f) Extracted nuclear image.

Figure 4. a) Smoothed image. b) Edges detected on the four most significant bit planes. c) Tracking algorithm first choice for cell contour. d) Extracted cell image. e) First choice of the nuclear contour by tracking algorithm. f) Extracted nuclear image.

Figure 5b represents the result of the morphological edge detector on the four most significant bit planes of figure 5a, the varying contrast of the cell boundary and the effect of background artifacts prevent the contour tracking from extracting any closed contour meeting the minimum length pixels. Consequently, the gray level image is pre-

filtered, by gray level closing operation with 7*7 square binary structuring element. The result of pre-filtering is shown in figure 5c. Figure 5d-g represent the four significant bit planes as described in figure 2.

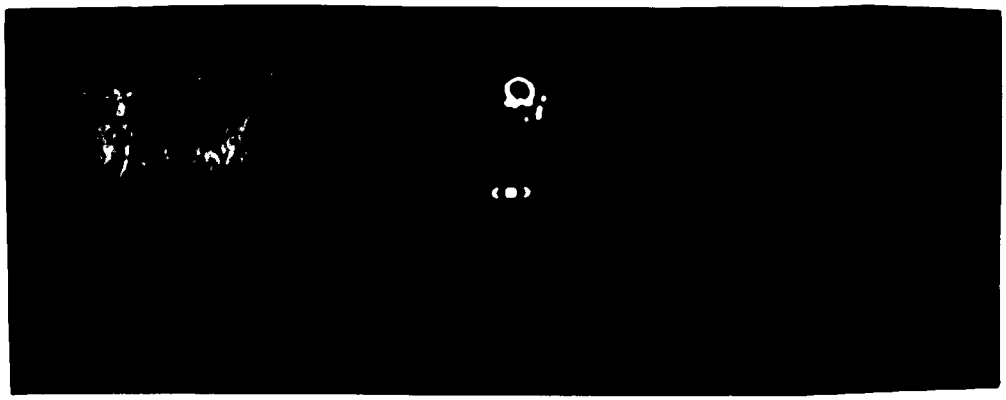


Figure 5. a) Original. b) Edges detected on the four most significant bit planes. c) Morphological closing of the original image with a 7*7 binary structuring element. d) Tracking algorithm first choice for the cell contour using the four most significant bit planes. e) Extracted cell image. f) Tracking algorithm first choice for the nuclear contour. g) Extracted nuclear image.

Inspection of the extracted contours reveal errors whenever an artifact is very close to the cell contour. An example of this can be seen in figure 2c where bottom of the cell contour contains a small protrusion which can be attributed to the dark artifacts in the original image. Such errors may be removed by further processing. However, one runs the risk of removing legitimate contour protrusions or identifications which may be indicative and result from a malignant cell.

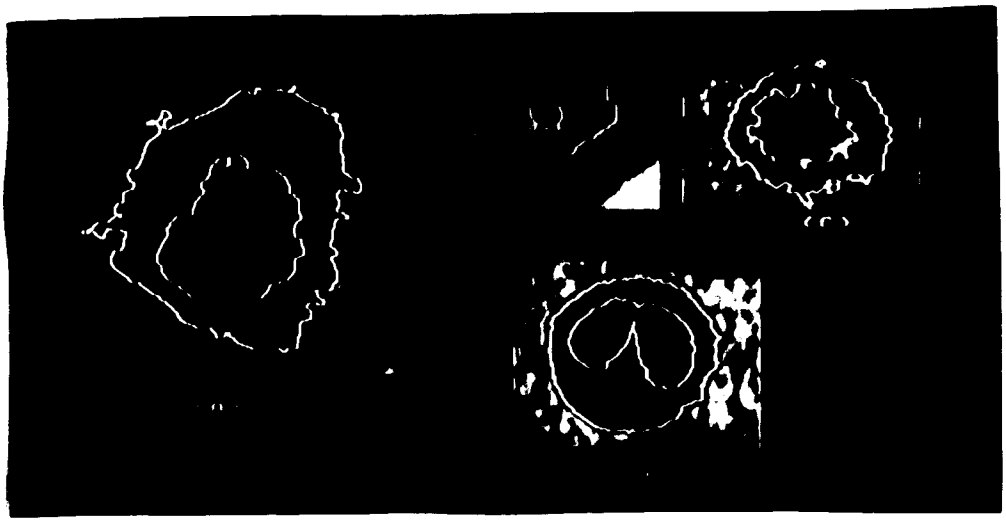


Figure 6a, 6b, 6c, 6d. The extracted contours superimposed on the original four cell images of figure 5.

6.3 Results of Suboptimal Thresholding on Shifted Cell Images

The figure of merit presented in chapter 4 is used for finding a better threshold of the original and three shifted cell images. The results show that the presented thresholding finding method not only improved the condition for the extraction of the contours, but also has the ability of correcting segmentation errors caused by the discrepancy of threshold setting by the bit plane thresholding method.

The gradient image is generated by the difference of the smoothed original image dilated by a 3 by 3 binary structuring element and the smoothed image. The gradient image is then thresholded by selecting

the top gray level pixels N_p ranging from 400 to 600. Figure 7 represents the two images and their 3 shifted images subtracted by gray values of 8, 16 and 24 respectively, and the selected edge image which is most likely to be the best candidate for the correct contour extraction.

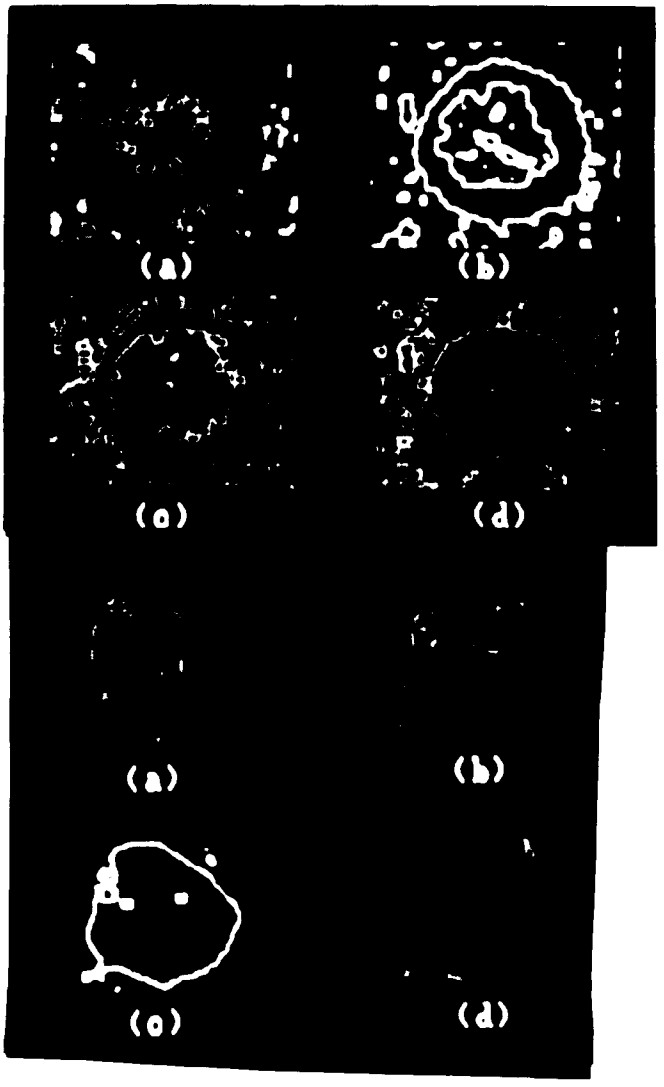


Figure 7 Edge detected by the BPM algorithm on four shifted bit-planes, the root edge is determined by the figure of merit presented in chapter 4. The number 0, 8, 16 and 24 represent the edge detection is operated on the 4 images obained by subtracting gray level of the original image by 0, 16, 24 and 24 respectively.

Figures 8, 9, 10, 11 and 12 show the procedure of generating the correct contours; a represents the original image; b represents the edges determined by the bit plane thresholding method presented in chapter 4; c represents the cell contours extracted from the binary edge images of b; d is the recovered region. These cwll images (breast and lung), due to their inside texture variations, all have unimodal histogram distributions. Figure 13 shows the result of the cell image 3c reprocessed by the new method. The result shows that the segmentation error is greatly reduced. This reduction of contour error results from the fine adjustment in the threshold determination.

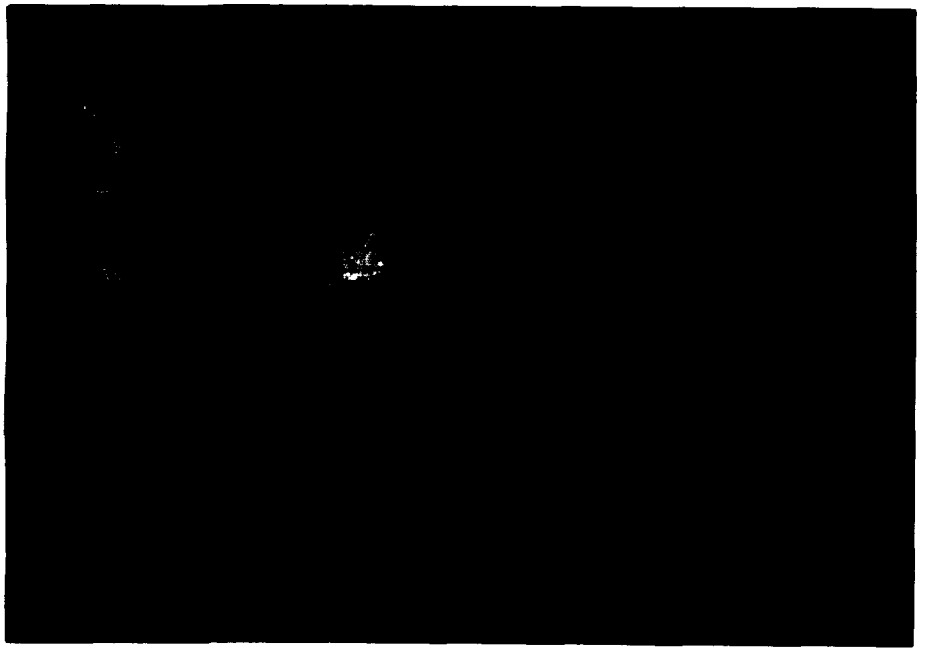


Figure 8 a) the original image; b) the one multiple bit plane edge image selected by maximizing the figure of merit presented in chapter 4; c) the extracted the cell contour; d) the segmented cell region.

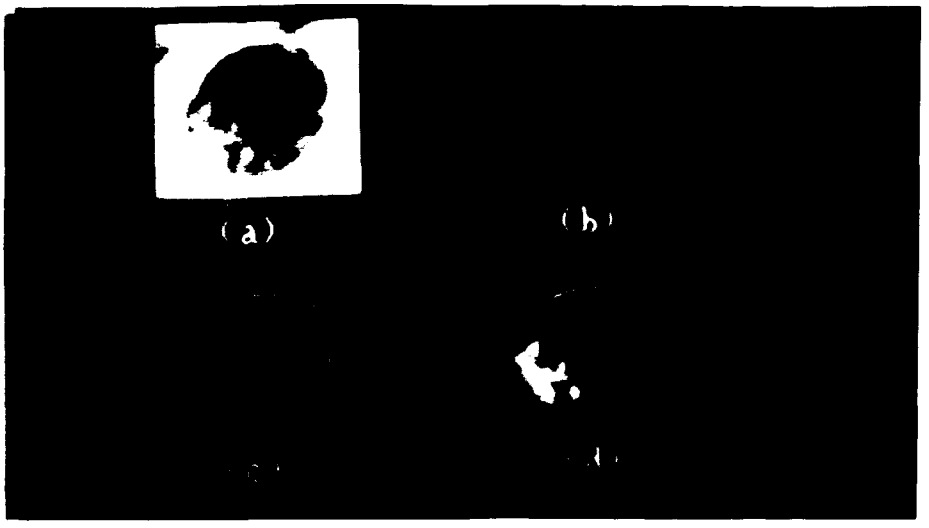


Figure 9 a) the original image; b) the one multiple bit plane edge image selected by maximizing the figure of merit presented in chapter 4; c) the extracted the cell contour; d) the segmented cell region.

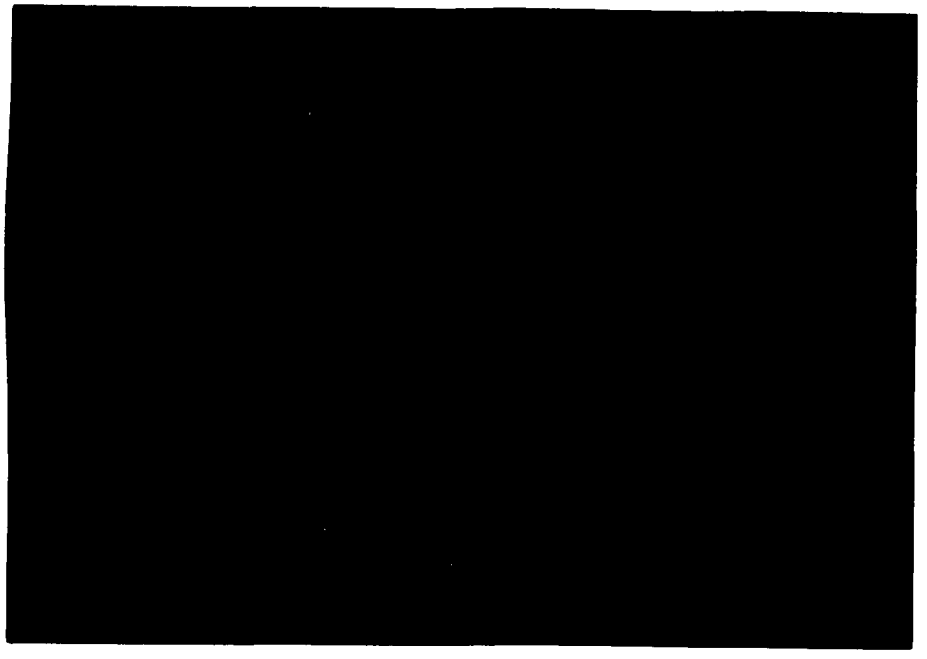


Figure 10 a) the original image; b) the one multiple bit plane edge image selected by maximizing the figure of merit presented in chapter 4; c) the extracted the cell contour; d) the segmented cell region.

Figure 11 a) the original image; b) the one multiple bit plane edge image selected by maximizing the figure of merit presented in chapter 4; c) the extracted the cell contour; d) the segmented cell region.

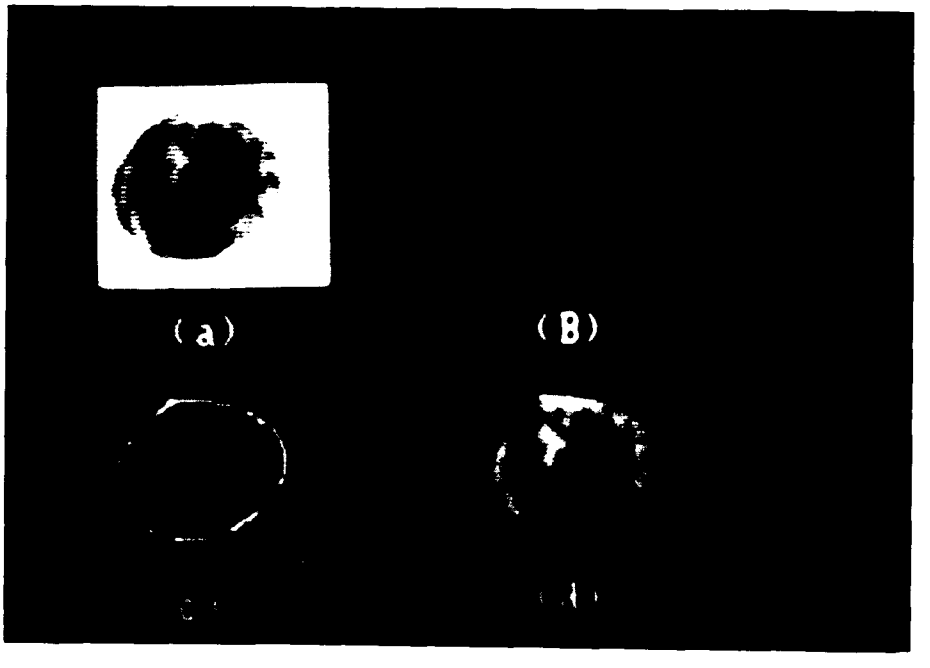


Figure 12 a) the original image; b) the one multiple bit plane edge image selected by maximizing the figure of merit presented in chapter 4; c) the extracted the cell contour; d) the segmented cell region.

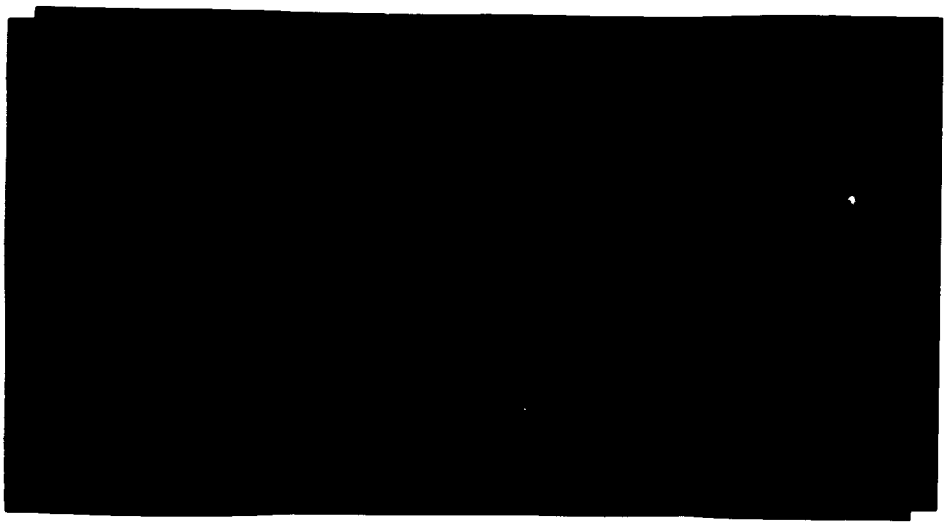


Figure 13 a) the original image; b) the one multiple bit plane edge image selected by maximizing the figure of merit presented in chapter 4; c) the extracted the cell contour; d) the segmented cell region. e) the extracted nuclear contour; f) the segmented nuclear region.

6.4 Region Segmentation Using Low Rate Vector Quantization

Three 2*2 homogeneous codevectors are assigned uniformly within the dynamic range of the smoothed original cell images as the initial code vectors. After the vector quantization, three homogeneous or near homogeneous codevectors are generated. A fine codebook containing all the patterns of 4 dimension vectors using 3 gray levels is then generated. Every 2 by 2 blocks of the original image search the nearest codewords replacement. When the relative mean square error of two consecutive iteration of the VQ is smaller than a prespecified value of 0.005, the program stops. Region segmentation can be achieved through simple thresholding and small blob elimination using edge detection and tracking discussed in chapter 3 and the only difference is that

the thresholds are not bit plane thresholds but determined by the gray levels of the final code vectors. Texture segmentation can be implemented by recalling the similar VQ operation on the segmented cell or nuclear regions.

Figure 14 shows the procedure of processing the cell image by the presented VQ segmentation method in chapter 5. Figure 14a is the original image; figure 14b is the vector coded image without fine codeword replacement, the jagged edges are shown in the picture; figure 14c is coarse and fine coded cell image; figure 14d is the extracted ccell region; figurec 14e isc the extracted nuclear region; figure 14f is the segmented texton regions inside the nucleus by 3 codevectors.

Figure 15, 16 and 17 are the three cells processed by the similar operation. Figure a are the original images; figure b are the results of coding a) by 3 by 3 codevectors; figure c are the recovered cell images; figure d are the segmented texton regions by using 3 codevectors; figure e are segmented regions are the segmented texton regions by using 5 codevectors. We noticed that in figure 17, at the very low contrast edge portion, the corresponding intrusion represents a minor segmentation error.

Figure 18 compares the texture segmentation of two cell regions using 3, 4 and 5 codevectors. Figure a represent the original cell images; figure b, c and d represent the texton region segmentation by using 3, 4 and 5 codevectors respectively. The mean square errors of the coded images with the original images are calculated.

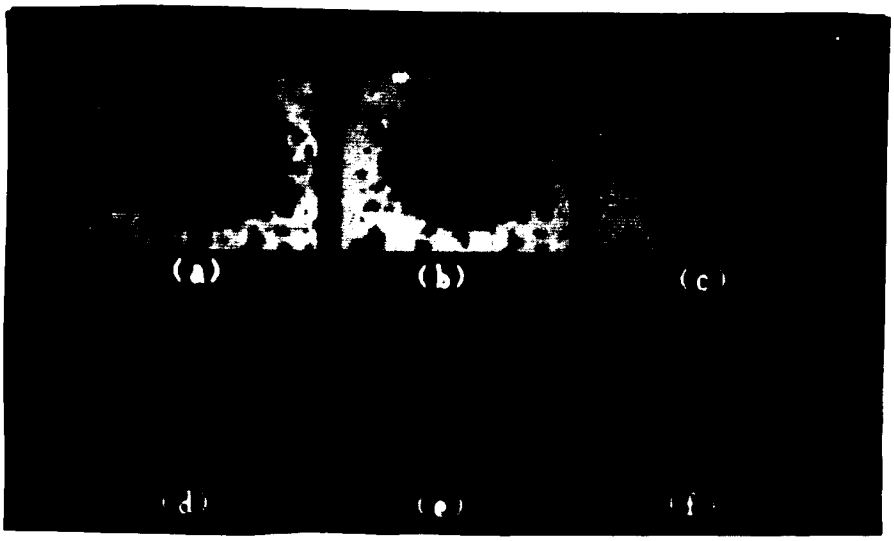


Figure 14 a) the original image; b) the VQ coded 3 image regions without fine code replacement; c) the VQ coded image with fine codewords replacement; d) the extracted cell region; e) the extracted nuclear region; f) the coded 3 texton regions inside the nucleus.

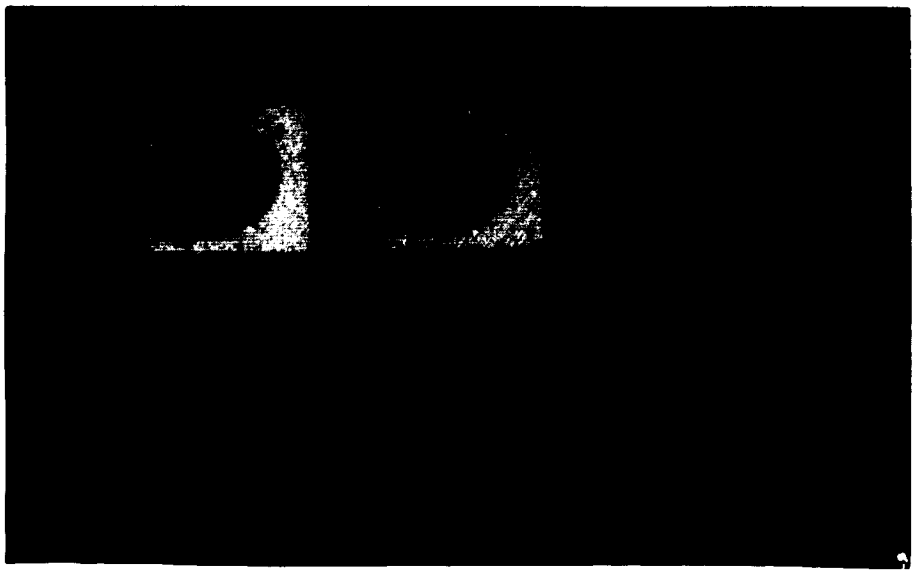


Figure 15 a) the original image; b) the VQ coded image with fine codewords replacement; c) the extracted cell region; fig d) and e) are the texton regions inside the cell coded by 3 and 5 codevectors.

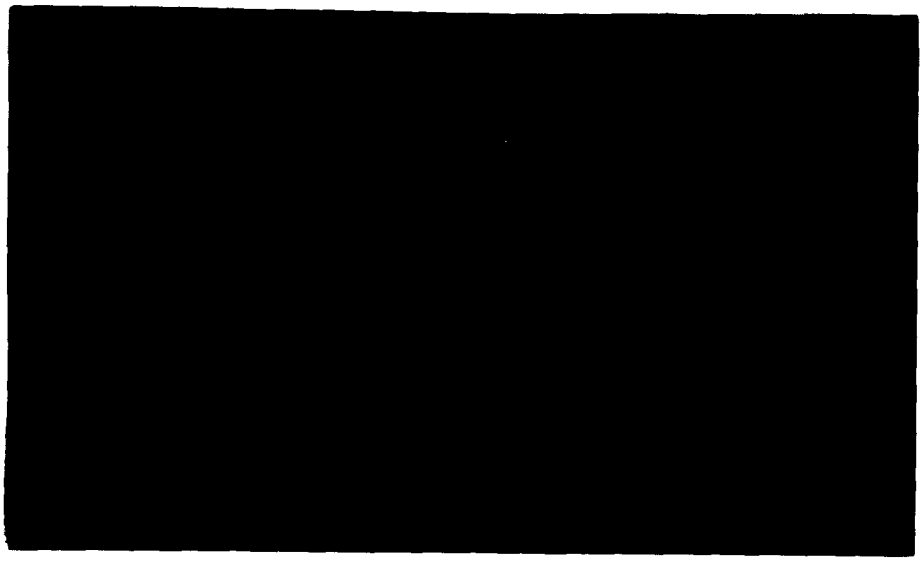


Figure 16 a) the original image; b) the VQ coded image with fine codewords replacement; c) the extracted cell region; fig d and e are the texton regions inside the cell coded by 3 and 5 codevectors.

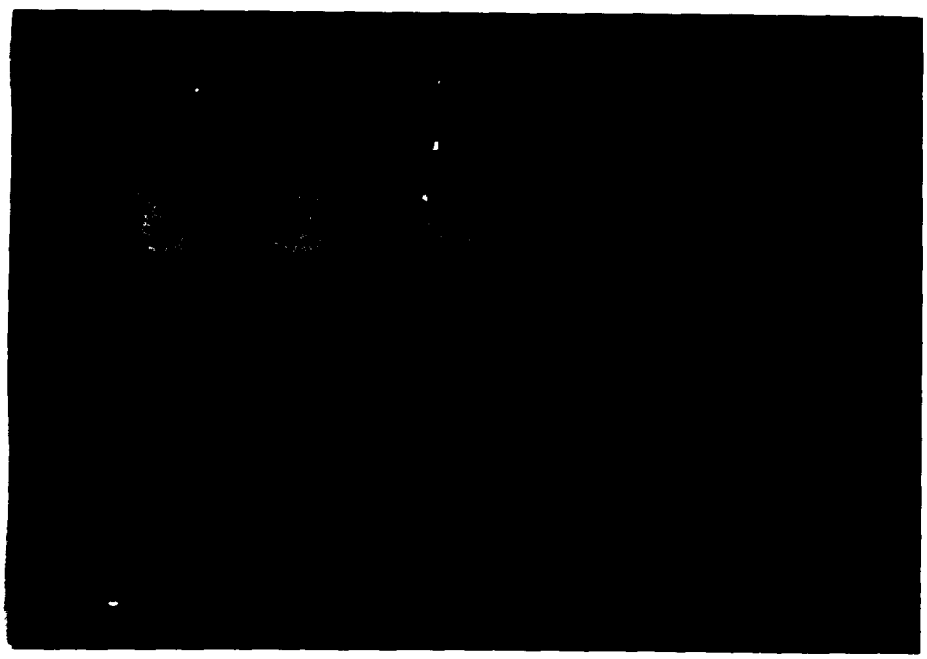


Figure 17 a) the original image; b) the VQ coded image with fine codewords replacement; c) the extracted cell region; fig.d and e are the texton regions inside the cell coded by 3 and 5 codevectors.

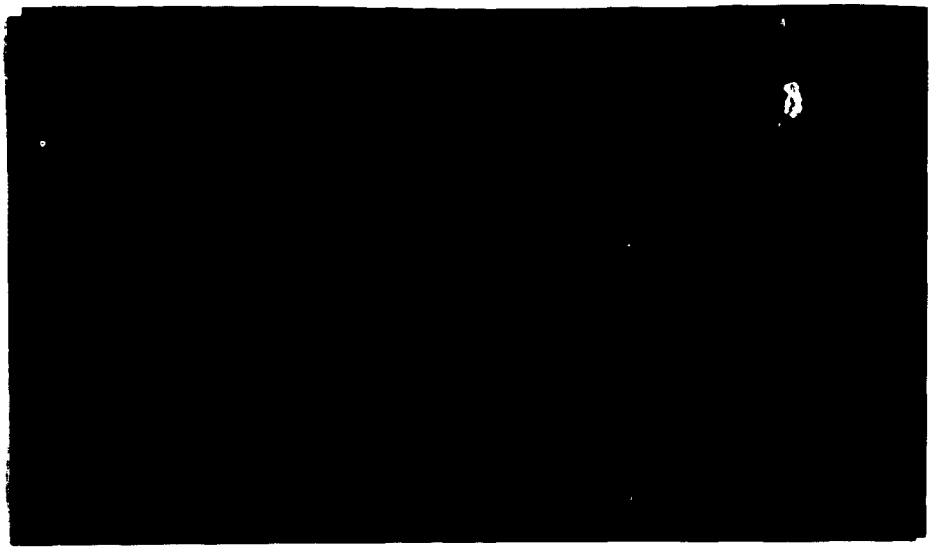


Figure 18(1)

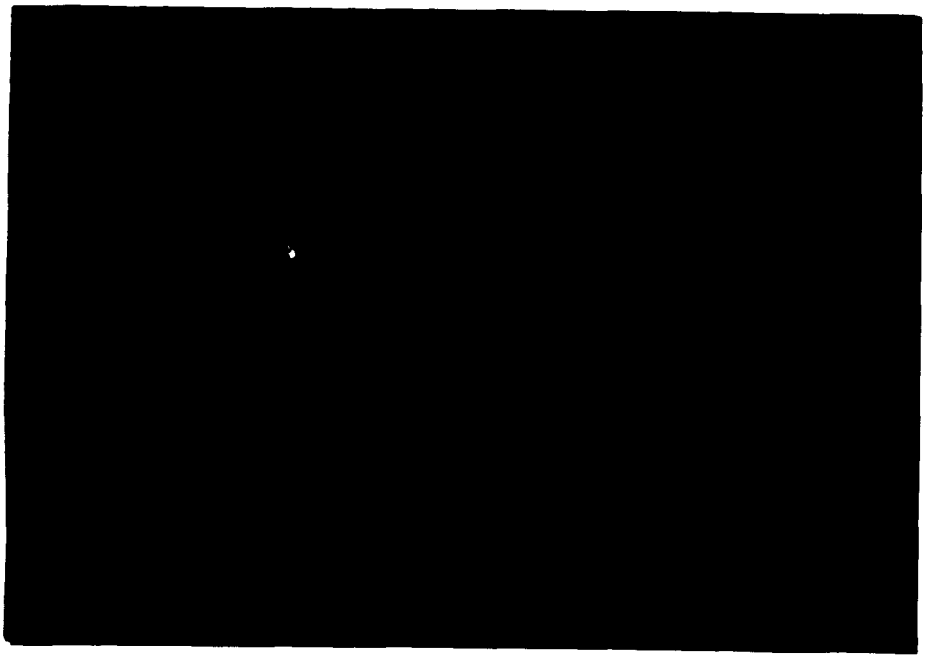


Figure 18(2)

Figure 18(1)a and 18(2)a are two segmented cell images; b, c and d are the images coded by Vector quantization using 3, 4 and 5 codevectors respectively.

6.5 Discussion

A multiple thresholding combining with edge detection algorithm has been described which results in good segmentation for cell images having relatively homogeneous regions and boundaries; For cell images having single histogram peak and textured regions, a suboptimal thresholding criterion has been developed operating on binary edges of the shifted images considering both coincidence matching with a gradient map and the edge busyness. A region based segmentation technique using vector quantization is also presented to reduce the sensitivity to the noise of the previous methods. The segmented regions are further segmented by VQ into easily extractable texton regions for feature extraction.

Due to the various amounts of a priori knowledge introduced into the segmentation process the techniques provide fairly good results for various cell images of varying photometrical conditions. Errors occur mainly in the extraction of the cytoplasm because of two reasons. The first is that in many cases the cytoplasmic areas are more likely to be ill-posed and textured. The second reason is that gray level feature does not provide sufficient discriminatory power between some regions. Figure 19a and 19c are two original images; figure 19b is the edge image obtained by bit plane thresholding method which causes the contour error. This reason for causing this error is that the contrast of the contour is too weak whereas that of the edge of inside texture is too strong. Figure 19d represents the resulting image using VQ segmentation by 3 codevectors. The cytoplasm area can not be separated from the

closed artifacts.

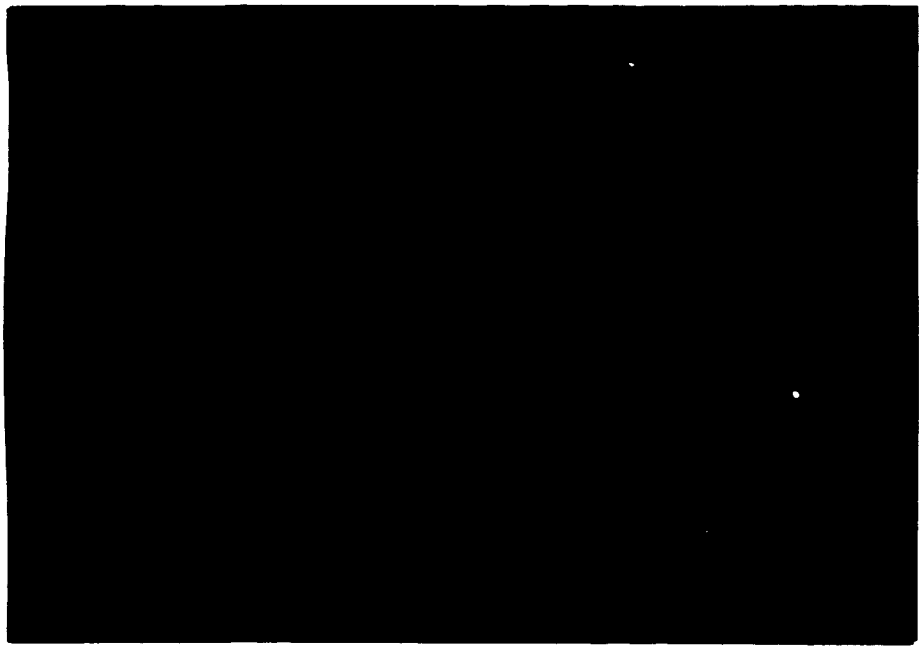


Figure 19. The missegmented cell images: a) The original image; b) the result of bit plane edge detection;c) the original image; e) the image coded by 3 codevectors; f) the image coded by 5 codevectors.

Multithresholding and edge detection methods are generally sensitive to the gray level variations near the contours, artifact edges and textures. When cell images are textured and contaminated by artifacts, or no histogram mode information can be applied the detected edge images may be very noisy. The edge maps may contain very accurate contours for some cell images, but extracting the cell and nuclear contours becomes very difficult and error-prone and very elaborate binary tracking algorithms are usually required. The presented thresholding based on matching provides a heuristic selection rule and more multiple edge options

for tracking. The algorithm greatly reduces the tracking error and difficulty of tracking algorithm design. The accuracy of the contour extraction has been greatly improved. Minor contour errors still exist when contrast between the regions is too low or some regions are seriously textured.

Region segmentation by VQ is much simpler than normal split and merging techniques and as sensitive to single pixel level variations, thus the VQ coded images have simple structure and contain much less artifacts and textures. Meanwhile the subsequent operations such as thresholding, tracking and filtering are greatly simplified. The accuracy and complexity of the segmented textons can be easily controlled by selecting the number of codewords for different application. A priori knowledge of shape can be more easily inserted into the operation at the primary level for early or successive error correction. The disadvantage is that simply by assigning 3 code vectors to three regions in VQ operation sometimes does not have enough discriminatory power for segmenting different regions correctly. Increasing the number of codevectors will raise a new question of developing a new rule for correct merging since for any region approach, to discriminate the different regions with similar gray levels requires the introduction of a priori knowledge of shape. We would rather select fast and simple region segmentation approach and leave the error correction to the successive shape analysis, such as convex hall generation.

Although the proposed techniques provided satisfying segmentation results, improvements still seem to be necessary and possible. The following problems will be the subjects for further studies:

(1) A priori knowledge of cell shape should be effectively inserted in the presented algorithms of segmentation for controlling the process or correcting the erroneous results.

(2) The figure of merit of selecting the root edge needs to be refined based on more experiment so that the potential of the multiple thresholding method can be fully developed.

(3) Binary tracking algorithm should be raised to knowledge-based level by combining the a priori information and feedback techniques.

(4) Combining more features, such as color and texture, in the algorithm to raise the discriminatory power for correct segmentation.

(5) To increase the level of automated segmentation, more approaches can be effectively organized to form a cooperative algorithm to reduce the rates of segmentation failure.

VII. TABLES AND FIGURES

TABLE 2.1

Properties of Binary Morphological Filters(BMF)
and Gray Level Mophological Filters(GMF)

The properties of BMF and GMF are listed in table 1 without proof. f and g are the gray level image and SE respectively. All the basic BMF and GMF are translation invariant(with respect to shifts of both the argument and the amplitude of signals), increasing, nonlinear, and generally noninvertable.

	BMF	GMF
Commutivity:	$X \oplus g = g \oplus X$	$f \oplus g = g \oplus f$
Associativity:	$(X \oplus g1) \oplus g2 = X \oplus (g1 \oplus g2)$	$(f \oplus g1) \oplus g2 = f \oplus (g1 \oplus g2)$
	$(X \ominus g1) \ominus g2 = f \ominus (g1 \oplus g2)$	$(f \ominus g1) \ominus g2 = f \ominus (g1 \oplus g2)$
D-E Duality:	$X \oplus g = (X \ominus g)^c$	$f \oplus g = (f \ominus g)^c$
Idempotency:	$O[O(X,g),g] = O(X,g)$	$O[O(f,g),g] = O(f,g]$
	$C[C(X,g),g] = C(X,g)$	$C[C(f,g),g] = C(f,g)$
Opening Antiextensivity:	$O(X,g) \subseteq X$	$O(f,g) \subseteq f$
Closing Extensivity:	$C(f,g) \supseteq f$	$C(f,g) \subseteq f$
Opening Increasing:	If $X1 \subseteq X2$ Then: $O(X1,g) \subseteq O(X2,g)$	If $f1 \subseteq f2$ Then: $O(f1,g) \subseteq O(f2,g)$
Closing Increasing	If $X1 \subseteq X2$ Then $C(X1,g) \subseteq (X2,g)$	If $f1 \subseteq f2$ Then $C(f1,g) \subseteq C(f2,g)$
Duality:	$O(X,g) = C(X^c, g)$	$O(f,g) = C(f^c, g)$

FIGURE 2.2 COMMUNICATION SYSTEM MODEL:

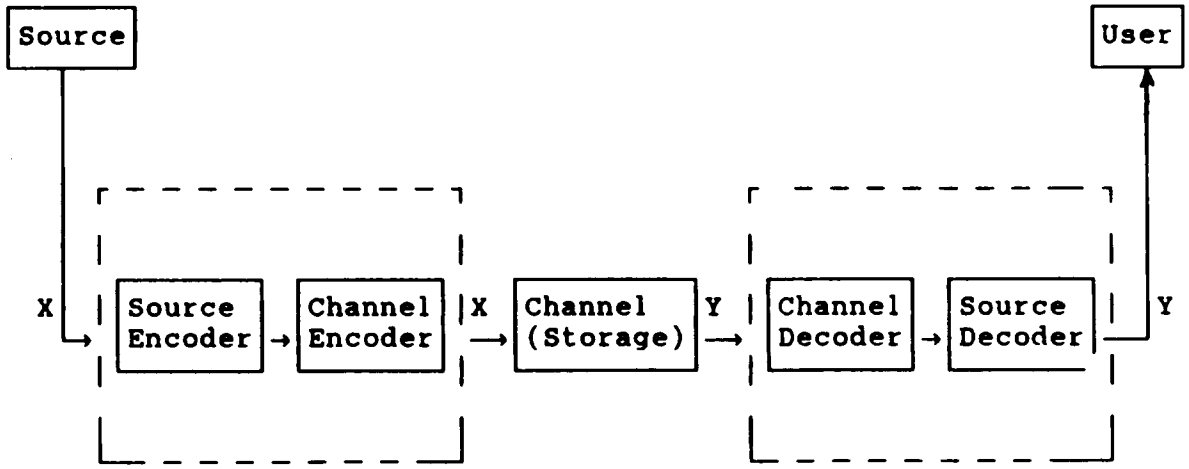


FIGURE 2.3 BLOCK DIAGRAM OF VECTOR QUANTIZATION

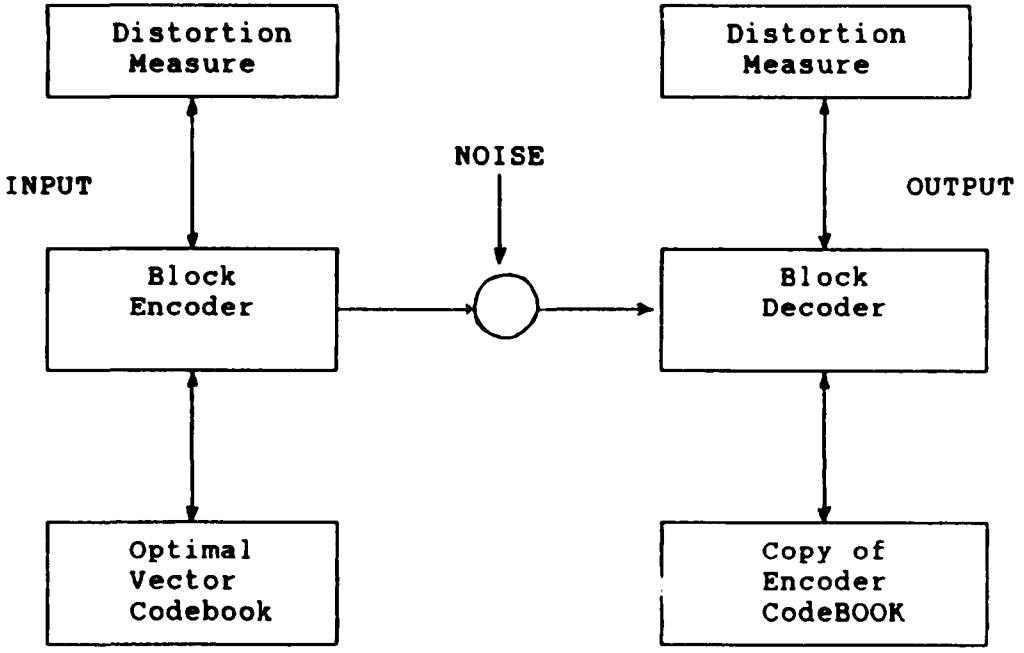


FIGURE 3.1 THE BLOCK DIAGRAM OF BIT-PLANE MULTITHREHOSLDING AND MORPHLOGICAL EDGE DETECTION

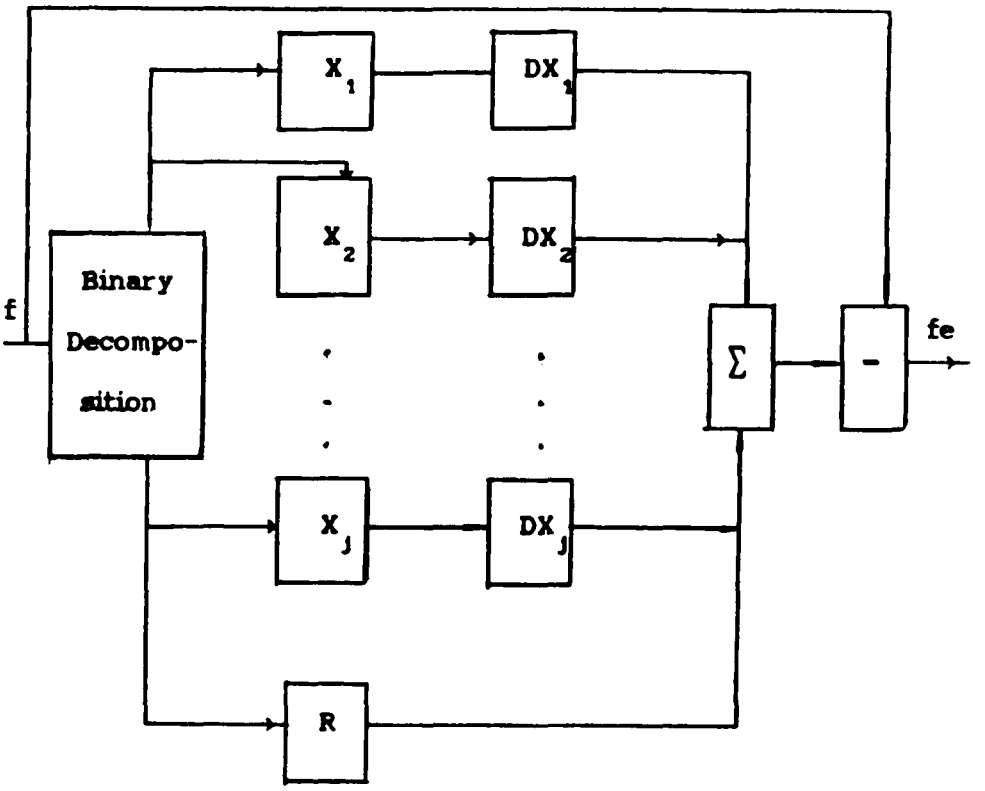


FIGURE 3.2 A BLOCK DIAGRAM OF CONVENTIONAL CONTOUR TRACKING

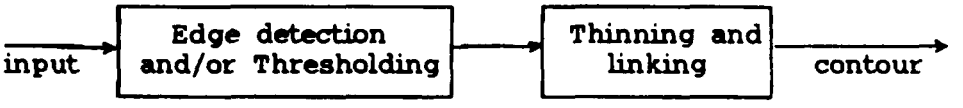


FIGURE 3.3A BLOCK DIAGRAM OF TRACKING SUB-PROGRAM

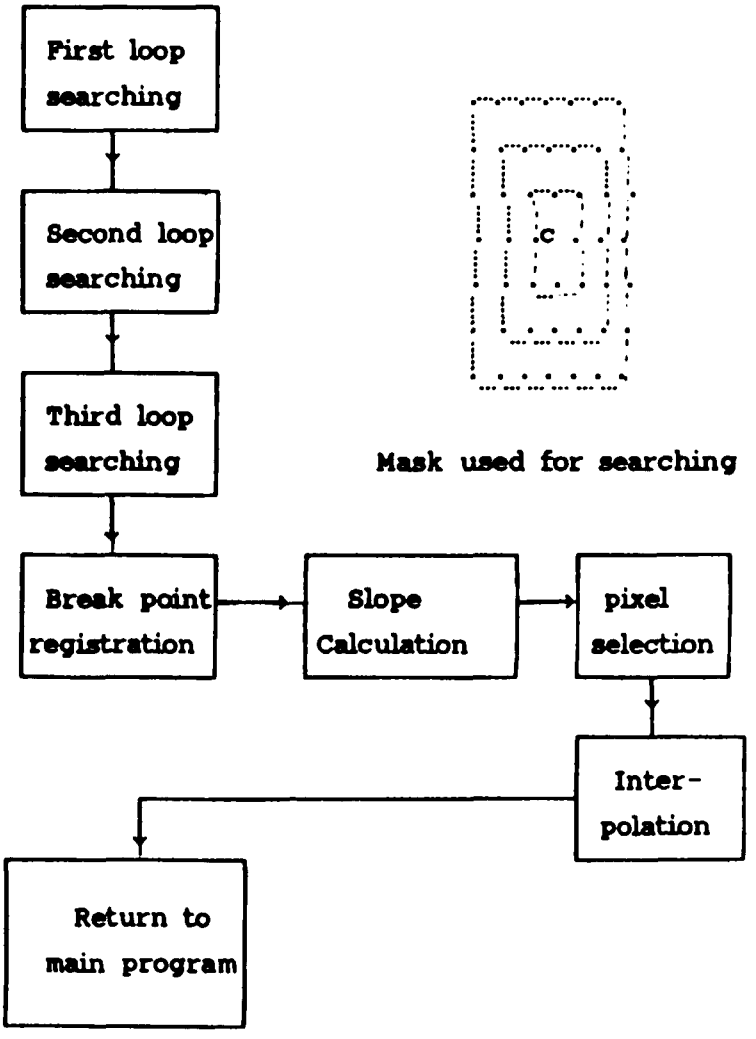


FIGURE 3.3B MAIN PROGRAM FOR TRACING AND FILTERING

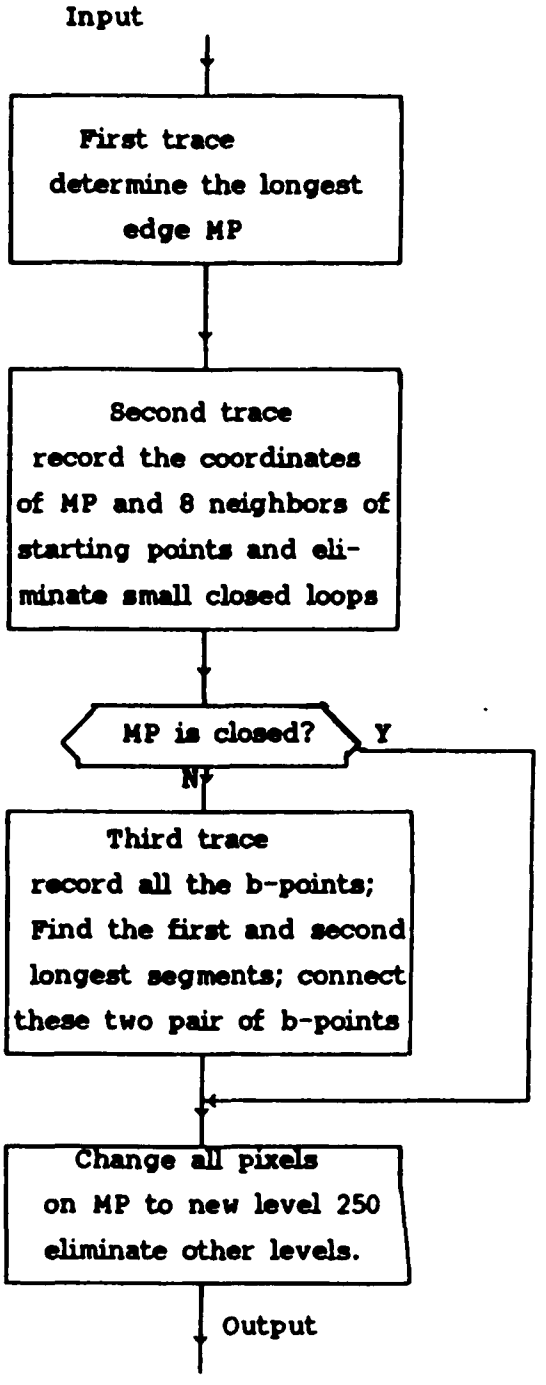
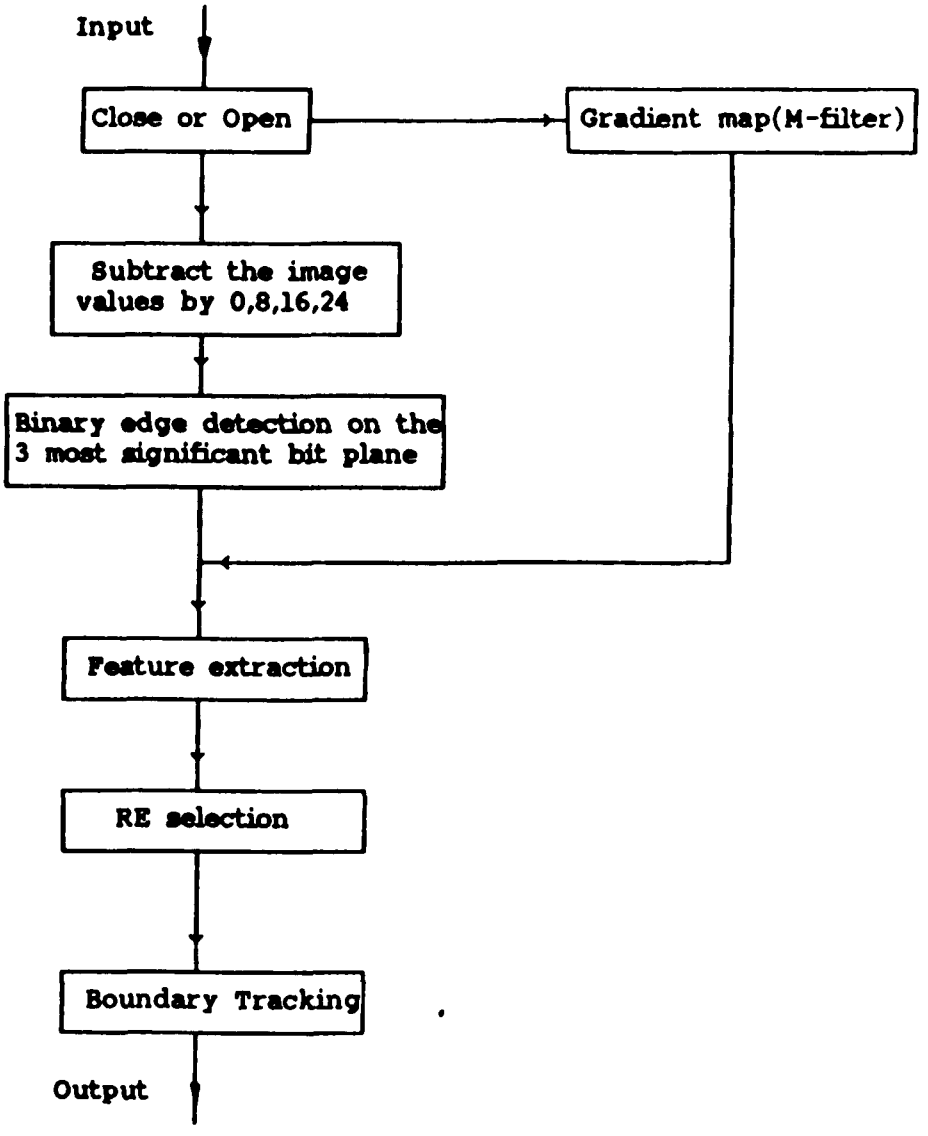
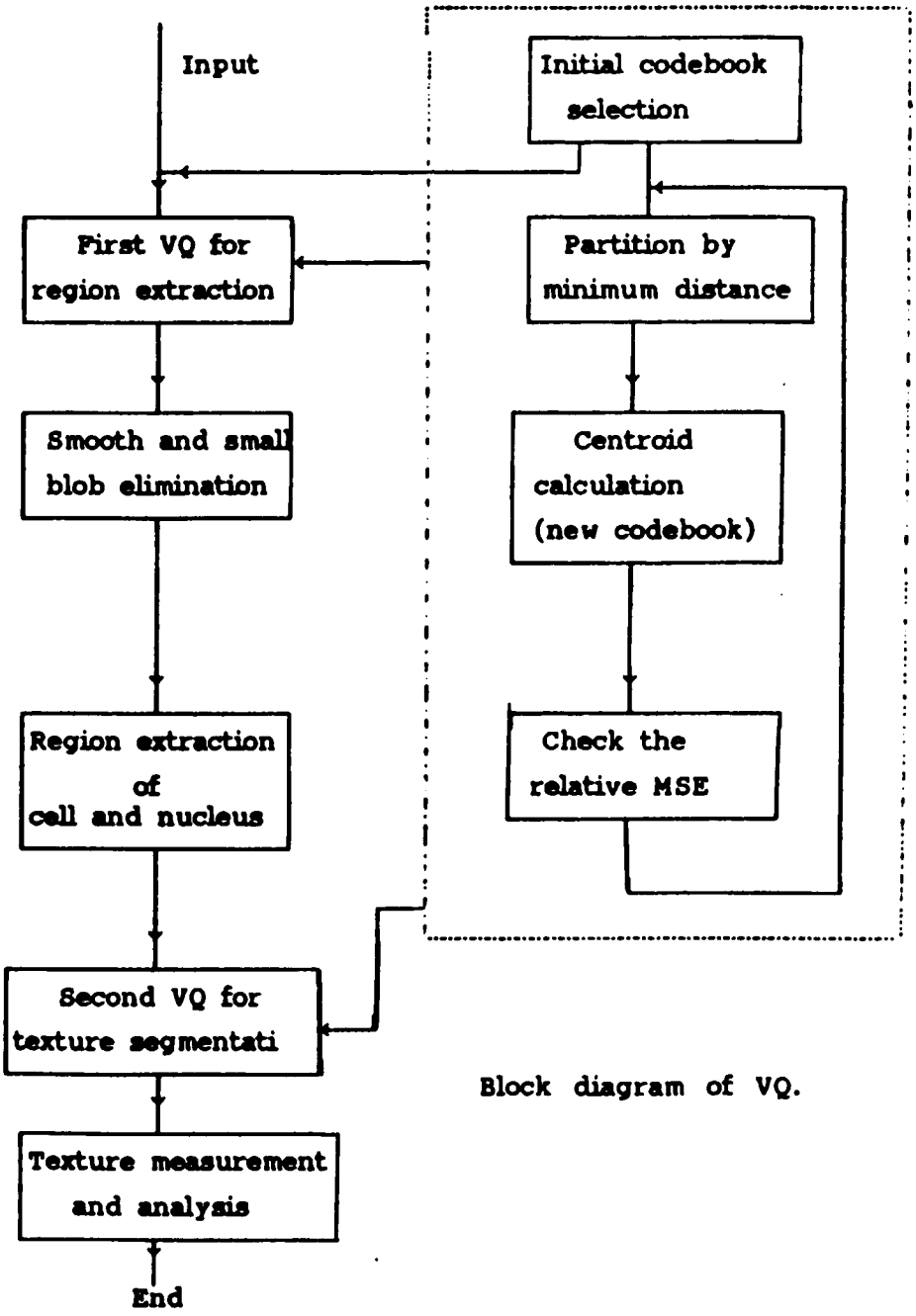


FIGURE 4.1 BIT PLANE MULTI-THRESHOLDING ON 4 SHIFTED IMAGES USING EDGE MATCHING



FIGUR 5.1 BLOCK DIAGRAM OF REGION SEGMETNATION USING VECTOR QUANTIZATION.



Block diagram of VQ.

TABLE 5.2 THE FINE CODEBOOK STRUCTURE FOR REGION BOUNDARY REFINEMENT

a) 3 near homogeneous codevectors:

C_{11} C_{12}	C_{21} C_{22}	C_{31} C_{32}
C_{13} C_{14}	C_{23} C_{24}	C_{33} C_{34}

b) 28 edge codevectors:

C_{11} C_{21}	C_{11} C_{11}	C_{21} C_{11}	C_{21} C_{31}	C_{21} C_{21}	C_{31} C_{21}
C_{21} C_{21}	C_{21} C_{21}	C_{11} C_{11}	C_{31} C_{31}	C_{31} C_{31}	C_{21} C_{21}
C_{21} C_{11}	C_{11} C_{21}	C_{11} C_{21}	C_{31} C_{21}	C_{21} C_{31}	C_{21} C_{31}
C_{21} C_{21}	C_{11} C_{21}	C_{11} C_{11}	C_{31} C_{31}	C_{21} C_{31}	C_{21} C_{21}
C_{21} C_{21}	C_{21} C_{21}	C_{11} C_{11}	C_{31} C_{31}	C_{31} C_{31}	C_{21} C_{21}
C_{11} C_{21}	C_{11} C_{11}	C_{21} C_{11}	C_{21} C_{31}	C_{21} C_{21}	C_{31} C_{21}
C_{21} C_{21}	C_{21} C_{11}	C_{11} C_{11}	C_{31} C_{31}	C_{31} C_{21}	C_{21} C_{21}
C_{21} C_{11}	C_{21} C_{11}	C_{11} C_{21}	C_{31} C_{21}	C_{31} C_{21}	C_{21} C_{31}
C_{21} C_{11}	C_{11} C_{21}		C_{31} C_{21}	C_{21} C_{31}	
C_{11} C_{21}	C_{21} C_{11}		C_{21} C_{31}	C_{31} C_{21}	

REFERENCES

1. Mui.J.S and Fu.J.K., " A survey on image segmentation," *IEEE. Transaction on "Pattern Analysis and Machine Intelligence,"* vol. PAMI-2, No.5, Sept. 1980,pp 429-443.

2. A.Rosenfeld & L.S.Davis,"Image segmentation and image models," *Proc. IEEE*,vol.67,no.5,pp 764-772,1979.

3. Garbay V.,"Image structure represention and processing: a discussion of some segmentation methods in cytology", *IEEE. Trans. Pattern. Anal. Mach. Intell.* PAMI-8(2), pp 140-146,1986.

4. Gauvain G, Seigneurin D, Brugal G, "A quantitative analysis of the human bone marrow granulocytic cell images using a SAMBA 200 cell image processor, part I: The normal maturation sequence," *Anal. Quant.Cytol.*, vol.6.no.3,pp. 168-178, 1984.

5. Bengtsson E, Eriksson O, Kolmquist J, Nordin B, Stenkvist B, *Jour. Histochem Cytochem.* vol.27.no.1, pp. 621-628,1979.

6. Young I.T, "The classification of white blood cells", *IEEE Trans. on Biomed. Eng*, BME-19.vol.4 pp 291-298,1972.

7. Komitowski D,Zinser G, *Patol. Res. Pract.*, vol.180,682-6,1985.

8. Talor J, Bakr G, Bartels PH, Bibbo M, Richards DL, Wied GL, *ACTA. Cytologica*, vol.19, No.3, 1975,289-298.

9. Bartz M.R., Proc. Internal.Joint.Conf.Artificial.Intell.Mitre Corp. bedford MA 79-90, 1969.

10. Ulman JR, *Pattern Recognition*,vol- 6 pp.127-135,1974.

11. Panda D.P., *Univ. of Maryland Computer Science Center* TR-508, Feb. 1977.

12. Komitowski D,Zinser G,*Pathol.Res.Pract.*,vol.180 pp.682-6,1985.

13. C. Chow, T. Kaneko, " Automatic boundary detection of the left ventricle from ciniangiographs", *Comput Biomed Res*,pp.338-410,1972.

14. Ingram M, Norgren P.E, Preston K.Jr,"Automatic analysis of blood cells," *Ann. N.Y. Acad. Sci.*, 157:275, 1969.

15. Kulkarani A.J, *Histochem Cytochem.*,27(1):210-216,1979.

16. Prewitt J.M.S & Mendelsohn ML, "The analysis of cell images," *Ann. N.Y. Acad. Sci.*, 128:1035-1053, 1977.

17. Brenner J.F., Gelsema E.S, Nechles T.F, Neurath PW, Sellers M.D, Vastola E, "Scene segmentation techniques for the analysis of routine bone marrow smears from acute lymphoblastic leukemia patients", *J. Histochem.*, vol.22,pp. 697-706,1974.

18. Zinser G., Komitowski D,*Jour. Histochem and Cytochem.*, vol.31, No.1, PP. 94-100, 1983.

19. C.garbay, "Image structure representaion and processing". *IEEE. Trans. Patt.Anal. & Mach. Intell.*, Vol. PAMI-8, No.2. March 1986.

20. Bir.B. and Olovier. D.F., " Segmentation of images having unimodal distributions," *IEEE, Tran. Patt.Anal. & Mach. Intell.*, vol. PAMI-4, No.4, July 1982.

21. H.Harms, U.Gunzer, and H.M.Aus." Combined Local color and texture analysis of stained cells," *Comp. Vision and Ima. Proc.*, 33. 364-376, 1986.

22. D.Wermser, G.Hausmann, and C.E.Liedtke, "Segmentation of blood by Hierarchical thresholding," *Comp. Vis.and Ima.Processing.*, 25, pp.151-168, 1984.

23. Matheron G, "Random Sets and Integral Geometry," *Welly*, New York, 1975.

24. Serra J., " Image analysis and mathematical morphology," *Academic Press*, 1982.

25. Haralick R.M., Sternberg S.S, Xinhua S," Grayscale morphology," in *Proc. IEEE. Conf. on Comp. Vision and Pattern Recognition*, June, 1986.

26. J.Serra, "Image analysis and mathematical morphology," *Academic press.* 1982.

27. P.Moragos and R.W. Schafer, "A unification of linear, median O-S and morphological filters under mathematical morphology," IEEE. *ICASSP. Proc.* Tempa, FL.March, 1985 ,pp.34.8.1-34.8.4.

28. V.Goetcherian, "From binary to greytone image processing using fuzzy logic concepts," *Pattern.Recognition*,.vol.12, pp 7-15,1980.

29. T.R.Crimiins and W.M.Brown, "Image algebra and automatic shape recognition," *IEEE Trans.AEROSP.ELECT.SYS.*,vol. AES-21. pp 60-69, Jan. 1985.

30. P. Moragos, "Pattern spectrum of images and morphological shape size complexity," *IEEE ICASSP Proc.*pp.241-244. 1987.

31. P.Moragos and RW.Schafer, "Applications of morphological filtering to image analysis and processing," *IEEE. Proc. ICASSP.*,1986, Tokyo.

32. P.Moragos & R.W Schafer. "Morphological skeleton representation and coding of binary images,"*IEEE. Tran.Acoust.Speech and Signal Processing.* vol. ASSP-34, pp 128-144, Oct.1986.

33. Klein.J.C. and Serra.J., " The texture analyzer," *Jour. Microscopy,* part 2, April,pp 349-356, 1973.

34. J.Serra, "Image Analysis and Mathematical Morphology," *New York Academic Press,*1982.

35. J.W.Klingler,JR. and C.L.Vaughan, "Segmentation of Echocardiographic Images using Mathematical morphological ," *IEEE Trans. Biom.Eng.*,vol.35, No,11 pp. 925-934. 1988.

36. S.R. Sternberg, " Biomedical image processing," *IEEE. Computer Magazine.* pp 22-34. Jan. 1983.

37. P.Maragos and RW. Schafer,"Morphological filters part 1", *IEEE. Tran. Acoust. Speech. Sig. Processing,* vol. ASSP-35, No. 8. Aug.1987.

39. R.M.Gray, "Vector Quantization," *IEEE ASSP MAG.*, vol.1, pp. 4-29, apr. 1984.

40.A. Gersho, "On the structure of vector quantization," *IEEE Trans. Info. Theory,* IT-28, No.2, march. 1982.

41. C.E.Shannon," A mathematical theory of communication," *Bell. system. Technical.Journal.* vol. 27, pp. 379-423,623-656,1948.

42. Y.Linde, A.Buzo and R.M.Gray," An algorithm for vector quantization design," *IEEE Trans. Commun.*vol.Com-28,pp.84-95, Jan.1980.

43. M.R.Gray,D.S.Johnson,and H.S. Witsenhausen," The complexity of the generalized Lloyd-Max problem," *IEEE Trans. Inform.Theory,*vol., IT-28, pp.255-256, Mar.1982.

44. A. buzo, A.H.Gray, Jr.,RM. Gray,J.D. Markel,"Speech coding based upon vector quantization," *IEEE Trans.Acoust.,Speech, Signal Processing,* vol.ASSP-28,no.5pp.562-574.

45. Juang.B.H. and gray A.H, "Multiple stage vector quantization for speech coding," *Proc.IEEE.Inter.Conf.on ASSP.* pp 597-600. April 1982.

46. W.H.Chen and C.H.Smith, "Adaptive coding of monochrome and color images," *IEEE. Trans.Commun.*,vol. Com-25,pp.1285-1292, Nov.1977.

47. A.Gersho and B.Ramamurthi, "Image coding using vector quantization," *Proc. Intl.Conf.Acoust., Speech, and Signal Proc.*, Paris, France,1982.

48. J.D.Gibson,S.K,Jones,and J.L.Melsa,"Sequentially adaptive prediction and coding of speech signals," *IEEE Trans. Commun.*,vol.COM-22,pp.1789-1797,Nov.1974.

49. P.A.Ramamoorthy," A hybrid coding involving ADM and VQ for Digital Image vedio data compression,: ICASSP 1986, Tokyo

50. V.Cuperman and A,Gersho,"Adaptive differential vector coding of speech," in *Conf.Rec. GLOBECOM 82,* Dec.1982,pp.1092-1096.

51. D.L.Newman,"The hexagon thorem,"*IEEE Trans.Info.Theory.*,vol.IT-28, pp.137-139, Mar. 1982.

52 J.Foster,R.M.Gray,"and M.O.Dunham,"Finite state vector quantization for waveform coding,"*IEEE Trans.Info. Theory.*,vol.IT-31pp.348-359, May 1985.

53. R.A.King et al."Image Coding using VQ in the Transform domain". *Pattern Recognition Letter*(1983). pp.323. July.1983.

54. MannosJ.L. and Sakrison D.J."The effects of a visual fidelity criterion on the encoding of images," *IEEE.Trans. Info.theory*,vol.IT-20, No.4 pp. 380-393, Aug.1979.

55. D.Faugeras,"Digital color image processing within the framework of a human visual model," *IEEE Tran.ASSP*,vol.ASSP,pp.380-393, Aug.1979.

56. K.S. Thyagarajan. " A matrix Quantizer incorporating the human visual model," *IEEE. Proc.* 1985.

57. A.P.Pentland, "Visual inference of shape computation from local features," *Ph.D dissertation*, Dep.Phychol, M.I.T. Cambridge.MA. 1982.

58. A.Rosenfeld and M.Thurston, " Edge and curve detection for visual scene analysis." *IEEE. Tran. Comput.* vol.c-20 no.5, pp.562-569, 1971.

59. J.Canny, "A conputational approach to edge detection," *IEEE.Tran. Patt. Analys.Mach. Intell.* vol.PAMI-8, no.6.Nov.1986. pp.679-698.

60. E.Johnston and A.Rosenfeld, "Angle detection on digital curves", *IEEE. Tran. Comput.*, vol.24.pp.1006-1010.1975.

61. B.Neuma, "Interpretation of imperfect object contours for identification and tracking," *Proc. Fourth Int. Joint Conf. Pattern. Recognition*, 1980,pp 691-693.

62. L. Davis, "Understanding angles and sides," *IEEE. Trans. Comp.* vol. C-26, pp. 236-242 Mar. 1977.

63. Y. T. Zhou, V. Venkateswar, and R. Chellappa "Edge Detection and Linear Feature Extraction Using a 2-D Random Field Model", *IEEE. Trans. Pattern Analysis and Machine Intelligence*, vol.No 1. Jan. 1989.

64. Dudani.S.A, "Region extraction using boundary following", *C.H. Chen Ed. Pattern and Artificial Intelligence New York, Press*, 1976, pp. 216-232

65. T. Pavlidis, "Filling algorithms for raster Graphics" *Computer Graphics and Image Processing*, 10. (1979) pp.126-141.

66. D.Milgram,"Region extraction using convergent evidence," *Comp.Graph. Ima. Proces.* 11, 1979,pp 1-12.

67. L.Herts and R.Schafer,"Multilevel thresholding using edge matching," *Comp.Vis. Graph. Ima.Process.* 44. pp. 279-295,1988.

68. Jeanty,H.H.,Barba,J.Gil,J.(1987),"On the components and specifications of personal computer based image processing systems.ACTA Sterol., 1987, 6/2, pp. 233-246.