GENERALIZED HOUGH TRANSFORM FOR CYTOLOGICAL IMAGE SEGMENTATION

by

HYUK-JOON OH

Bachelor of Science
Kyunghee University
Seoul, Korea
1990

Master of Engineering
Kyunghee University
Seoul, Korea
1994

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements
for the Degree of
MASTER OF SCIENCE
December, 1997

OKLAHOMA STATE UNIVERSITY

GENERALIZED HOUGH TRANSFORM FOR
CYTOLOGICAL IMAGE SEGMENTATION

Thesis Approved:

W. Nick Street
Thesis Advisor

J. Chandler

Wayne B. Powell
Dean of the Graduate College

AKNOWLEDGMENTS

I would like to express my sincere appreciation to my thesis advisor, Dr. Nick Street for his inspiration, guidance, friendship and patience. I also be very grateful to Dr. John Chandler, Dr. K. M. George and Dr. Blayne Mayfield. I would like to thank to Dr. Ok-Sam Chae who has been my advisor in Korea for his encouragement and academic support.

I would like to show my special appreciation to my parent for their encouragement and support. My appreciation extends to my lovely sister and her husband for their encouragement and love.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1 : Introduction

The purpose of this research is to develop an automatic segmentation method to locate cell nuclei in a cytological image by using the generalized Hough transform [1] to minimize the level of user interaction. The segmentation problem, that of extracting only the interesting features or objects from the background in a digital image, is a very important one in the image processing field. More powerful segmentation algorithms lead to better accuracy, faster performance and more user independence of automatic image processing applications. Many image-processing applications depend on these segmentation algorithms.

The Xcyt system is a graphical computer program to diagnose breast cancer and was developed by Street, Mangasarian and Wolberg in 1994 [12]. The system performs analysis of cytological features in a digital image, diagnosis of the image as benign or malignant, and prediction of when the cancer is likely to recur. The Xcyt system using the snake algorithm [9], an adaptive spline curve fitting technique, isolates the cell nuclei very well, but it requires a very precise initial snake from the user.

The generalized Hough Transform is a well-known and very powerful template matching algorithm to detect an arbitrary shape in a digital image. The GHT is usually used to detect a known shape in an image. Since the shapes of cell nuclei are unknown, the GHT is performed with several types and sizes of templates that may match the cell nuclei. After many GHTs, it is determined which template most closely matches each cell nucleus. Even if the template is not perfectly matched with the cell nucleus, the template can be used as an initial snake for the Xcyt system.

For the GHT, some preprocessing algorithms are applied. The preprocessing steps include filtering and edge detection. Many different ellipses are applied for the GHT as templates, since it was observed that most of the cell nuclei are shaped like an ellipse. Every template determined by the GHT can be used as an initial snake for the Xcyt system, so the proposed method can be used as a preprocessing step of the system.

1.1 Medical Image Segmentation

The segmentation problem, that of extracting only the interesting features or objects from the background in a digital image, is a very important one in the image processing field. More powerful segmentation algorithms lead to better accuracy, faster performance and more user independence of automatic image processing applications. Many image processing applications depend on these segmentation algorithms.

Because of this importance, many efficient image segmentation algorithms (thresholding, region growing, contour following, etc.) have been and are being developed. This image segmentation is applied to many medical image applications. Usually the medical image segmentation techniques are used to count, to isolate, to identify and to diagnose any organic features in a human body. We can see some examples of the medical image segmentation from the work of Mussio [14], Mui, Fu and Bacus [13], and Jain [8].

The thresholding algorithm is one of most widely used segmentation techniques. This thresholding technique is a method to separate an object from the background image by using the difference of properties between those two features. This technique requires an assumption that the property of the object, usually intensity of image pixels, is distinctively different from that of the image background. For example, if an white object is on a dark background, we can isolate the object from the background by choosing image pixels that are brighter than a certain value. Mussio's research implementing an automatic cell count in 1991 and Mui, Fu and Bacus's research in 1977 showed examples of medical image segmentation using the thresholding technique. This technique is very simple and effective for an image that is clear and not noisy. However the technique is very sensitive to light

and noise. In fact, good segmentation cannot be achieved, when an image is noisy or poorly focused. Deciding the thresholding value is another difficulty. Even if an image is very clear and the object is very distinct from the background, the accuracy of the segmentation depends on the thresholding value. This decision remains as a significant difficulty of the thresholding technique. Because of these limitations, the technique is usually combined with some other image processing techniques.

The region growing technique is another well known segmentation technique. This technique isolates an object that has similar properties with that of a seed point. This technique groups image pixels into a region by searching for image pixels that have similar properties. When a seed point is given, if neighbors of the point have similar properties the neighbor pixels are appended to the seed point. The research of Jain in 1980 to classify muscle cells in an image shows an example of segmentation using region growing. In the first part of the segmentation system, he employed the region growing technique and some low-level image processing techniques for the segmentation. Even though this region growing technique is a useful algorithm, the technique is very sensitive to noise, the seed point and stop condition. The result of the technique depends on them.

Even though these segmentation techniques are very widely used and effective algorithms, because the images we are using are very noisy and captured in bad light conditions, those techniques were not appropriate for our situation. Thus, a segmentation technique that is less effected by noise and light condition was required and the segmentation technique should be able to provide outlines of an object. For these reasons, we chose the generalized Hough Transform. The generalized Hough Transform is a well-known template matching method

and the technique is less effected by noise and light condition than other techniques. Furthermore, the template can be used to provide the outlines of the objects found by the generalized Hough Transform.

Our objective is to develop a method to automatically provide an initial snake for the Xcyt system. The snake algorithm is an advanced contour following segmentation algorithm. The snake algorithm was designed by Kass in 1988 [9]. This snake is an energy-minimizing spline. The spline is a controlled continuity spline influenced by the internal, image and external energy forces. The internal energy imposes a smoothness constraint. This internal energy forces the snake to move like a membrane and a thin plate. The image energy guides the snake onto the outline of a feature in an image. Some different image factors like lines, edges and terminations can act as different image energies. The image energy can be represented as a weighted combination of different image energy factors. The external energy may be provided by external energy sources like user interaction. This energy puts the initial snake near the location a local minimum in the energy space. This initial snake is then guided by internal and image energy. Thus, providing proper external energy initiates correct snake movement.

This work concentrates on how to isolate cell nuclei and to provide approximate outlines of the cell nuclei in a medical image. In fact, the Xcyt system using the snake algorithm, isolates cell nuclei very well and the system has been tested in some medical institutions. This system has shown very high levels of accuracy in isolating and diagnosing breast cancer. However, the user must provide an initial snake for every cell nucleus to be isolated. For this reason, our research has been focused on reducing the user interaction of the Xcyt system by providing approximate outlines of cell nuclei automatically.

1.2 Xcyt System

The Xcyt system [12] is a graphical computer program for diagnosing breast cancer. The Xcyt system performs analysis of cytological features based on a digital scan of a breast fine-needle aspirate, diagnosis of the image as benign or malignant, along with an estimated probability of malignancy and prediction of when the cancer is likely to recur for cancerous samples.

The Xcyt system uses a curve-fitting program to determine the exact boundaries of the cell nuclei. The boundaries are initialized by an operator using a mouse pointer. Several different features such as area, radius, perimeter, symmetry, number and size of concavities, fractal dimension, compactness, smoothness and texture are computed for each cell nucleus. The mean value, extreme value and standard error of each of these cellular features are computed for each image. The Xcyt system has been used by oncologists at several institutions. Even though the Xcyt system isolates the cell nuclei very well, it requires a very precise initial snake from the user. The user must provide the initial snake for every cell nucleus to be isolated. If the image includes just a few cell nuclei, that user interaction may not be a problem. However many images include several hundred cell nuclei, therefore the user interaction may be a heavy overhead to the user. Our research shows a method to minimize the user interaction through providing a precise initial snake by using GHT with gradient direction information.

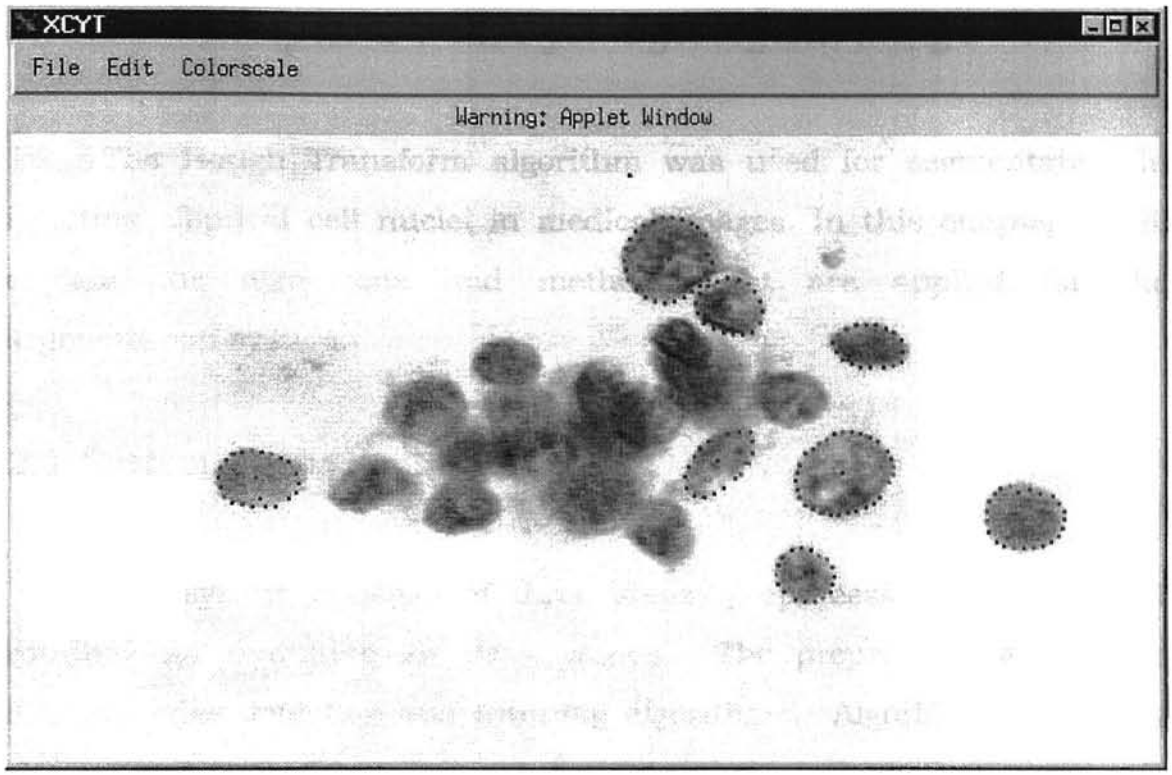


Figure 1-1: Xcyt System Cell Nuclei (isolated by the Snake)

Chapter 2 : GHT for Cytological Image

The Hough Transform algorithm was used for segmentation in isolating elliptical cell nuclei in medical images. In this chapter, I will explain the algorithms and methods that are applied for the segmentation system.

2.1 System Overview

The system consists of three steps: preprocessing, iterative HT routine and searching for peak values. The preprocessing includes filtering, edge detection and thinning algorithms. Algorithm 2-1 briefly shows every step of the proposed method.

```
Segmentation for cell nuclei
{
        Median filtering on the image;                  /* Preprocessing */
        Sobel edge detection on the image;
        Edge thinning;

        for i =Min_Size to Max_Size                     /* Iterative HT routine */
        {
                for j =Min_Size to Max_Size
                {
                        for θ = 0 to 360 step r
                        {
                                Hough transform( i, j, θ )
                        }
                }
        }

        /* Searching peak values */
        Find peak values in global accumulator
}
```

Algorithm 2-1

2.2 Preprocessing

2.2.1 Filtering

A filtering algorithm that is applied for the proposed method is an image smoothing algorithm. These algorithms are used for diminishing spurious effects that can be present in a digital image as a result of a poor sampling. Many different image filtering algorithms have been developed. The image smoothing filter algorithms include neighborhood averaging [11], lowpass filter [11] and median filter [11,13]. I tried to find an image smoothing algorithm that is simple but still minimizes white noise and rough edges in the original image.

The filtering algorithm in the proposed method is median filtering. The fast two-dimensional median filtering was proposed by Huang [14] in 1979. This algorithm replaces the gray level of each pixel by the median of gray levels in a neighborhood. In order to perform the filtering, the values of the pixel and neighbors are sorted and the median value is determined. Figure 2-1 shows two types of median filter templates.

P_8	P_1	P_2
P_7	P_0	P_3
P_6	P_5	P_4

$P_0' := Mid(P_i,\ i = 0,\ 1,\ 2,\ \ldots,\ 8)$

P_{24}	P_9	P_{10}	P_{11}	P_{12}
P_{23}	P_8	P_1	P_2	P_{13}
P_{22}	P_7	P_0	P_3	P_{14}
P_{21}	P_6	P_5	P_4	P_{15}
P_{20}	P_{19}	P_{18}	P_{17}	P_{16}

$P_0' := Mid(P_i,\ i = 0,\ 1,\ 2,\ \ldots,\ 24)$

Figure 2-1: 3×3 and 5×5 Median Filter

This algorithm is effective when noise consists of a strong, sharp component and where the characteristic to be preserved is edge sharpness. In this paper, a 5×5 median filter has been applied in order to reduce white noise and to make the edges more smooth. Since all templates for the HT are ellipses, a smooth edge is better for the matching. Figure 2-2 and Figure 2-3 show the effect of the 5×5 median filtering.

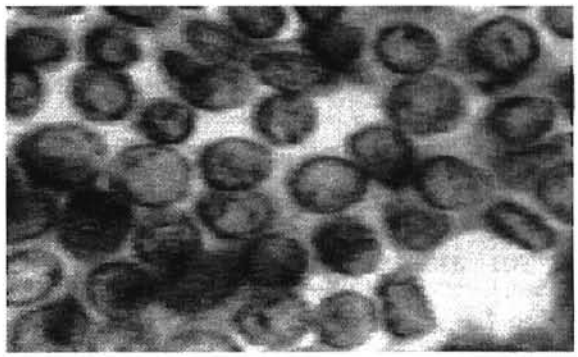


Figure 2-2: Before the 5×5 Median Filtering

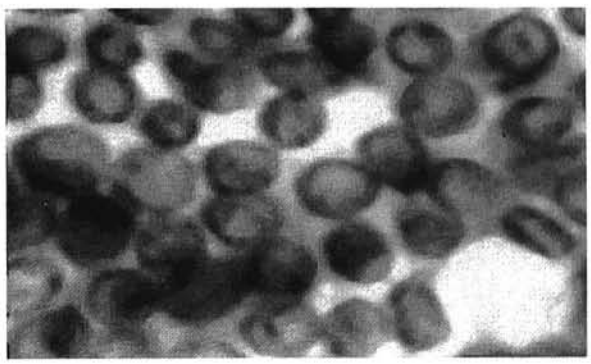


Figure 2-3: After the 5×5 Median Filtering

2.2.2 Edge Detection

An edge detection algorithm detects significant discontinuities in gray scale. This edge is a very important factor in the image processing field. Usually edges show the boundary of an object. This edge can represent the shape of an object much more compactly than an original image. For the reason, many edge detection algorithms have been designed. These include general gradient edge detector[4,11], Laplacian edge detector[8], Sobel edge detector[4,11] and Canny edge detector[6]. The GHT needs the edge and the appropriate gradient direction information. For this reason, the Sobel edge detection algorithm was chosen.

The Sobel edge detector is a kind of modified gradient edge detection algorithm. This algorithm detects an edge by using the gradient for image differentiation like the general gradient edge detector, but it is less affected by noise. Let $f(x,y)$ be the gray scale value of the image pixel (x,y). The gradient of an image $f(x,y)$, $G[f(x,y)]$, is usually defined as a vector of the partial derivatives df/dx and df/dy at every pixel location (Eq. 2-1).

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \\ \frac{df}{dy} \end{bmatrix} \qquad \textbf{(Eq. 2-1)}$$

For the edge detection, the magnitude M of this gradient vector is important.

$$M[f(x,y)] = (G_x^2 + G_y^2)^{1/2} \qquad \textbf{(Eq. 2-2)}$$

High magnitude values signify a big change of gray scale values. This generally occurs on the boundary of an object in an image. Practically, Sobel masks (a) and (b), shown in Figure 2-5 are used to measure the gradient of an image pixel. The direction of the gradient vector is also important. The direction is represented as

$$GD(x,y) = \tan^{-1}(G_x / G_y)\,. \qquad \text{(Eq. 2-3)}$$

In the proposed method, edge pixels are separated into eight different gradient directions and labeled with a number from 1 to 8. Figure 2-4 shows the relation between the edge and the directions.

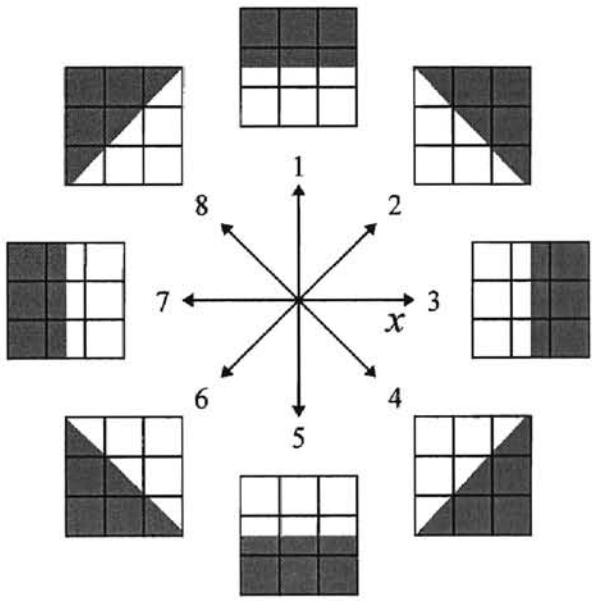


Figure 2-4: Eight Gradient Directions and Edge Types

The directions numbered 1, 3, 5 and 7 are the directions of the gradients along the x and y axes and 2, 4, 6 and 8 are the gradient directions along the lines $y = -x$ and $y = x$.

In the general Sobel algorithm, only the gradients along the x and y axis are directly measured. Gradients along the lines $y=-x$ and $y=x$ are indirectly computed from the gradients G_x and G_y by Eq. 2-1. More precise gradients on the lines $y=-x$ and $y=x$ can be obtained by direct measurement. These gradients can be directly measured by adding two more Sobel masks (c) and (d). The terms G_l and G_r represent the gradients that change along lines $y=-x$ and $y=x$.

x_0	x_1	x_2
x_3	x_4	x_5
x_6	x_7	x_8

$$G_x = (x_6 + 2x_7 + x_8) - (x_0 + 2x_1 + x_2)$$
$$G_y = (x_2 + 2x_5 + x_8) - (x_0 + 2x_3 + x_6)$$
$$G_l = (x_3 + 2x_6 + x_7) - (x_1 + 2x_2 + x_5)$$
$$G_r = (x_5 + 2x_8 + x_7) - (x_1 + 2x_2 + x_5)$$

-1	-2	-1
0	0	0
1	2	1

(a)

-1	0	1
-2	0	2
-1	0	1

(b)

-2	-1	0
-1	0	1
0	1	2

(c)

0	-1	-2
1	0	-1
2	1	0

(d)

Figure 2-5: Sobel Edge Detector Templates

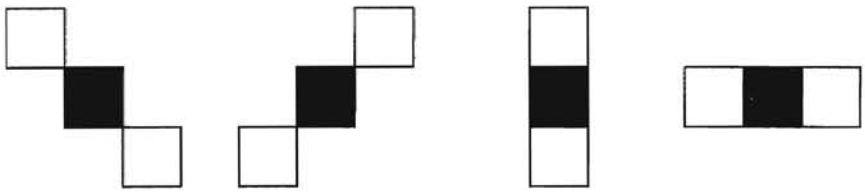
Since the gradients G_l and G_r are measured directly, Eq. 2-1, 2-2 and 2-3 are not necessary. Instead of those equations, a rule decides the gradient direction of every edge. As shown in Eq. 2-4, the largest of the four gradient factors of the edge is chosen as the gradient of the pixel.

$$G[f(x,y)] = Max\left\{G_x,\ G_y,\ G_l,\ G_r\right\} \qquad \textbf{(Eq. 2-4)}$$

$$M[f(x,y)] = Max\left\{\left|G_x\right|,\ \left|G_y\right|,\ \left|G_l\right|,\ \left|G_r\right|\right\} \qquad \textbf{(Eq. 2-5)}$$

$$GD[f(x,y)] = \begin{cases} 1: & M[f(x,y)] = |G_y|, & G_y > 0 \\ 2: & M[f(x,y)] = |G_l|, & G_l > 0 \\ 3: & M[f(x,y)] = |G_x|, & G_x < 0 \\ 4: & M[f(x,y)] = |G_r|, & G_r > 0 \\ 5: & M[f(x,y)] = |G_y|, & G_y < 0 \\ 6: & M[f(x,y)] = |G_l|, & G_l < 0 \\ 7: & M[f(x,y)] = |G_x|, & G_x > 0 \\ 8: & M[f(x,y)] = |G_r|, & G_r < 0 \end{cases} \quad \text{(Eq. 2-6)}$$

The edge image generated by this algorithm is a set of gradient values and gradient directions of the image pixels. The edge image is thinned by finding the maximum of the gradient values. For this operation, four different masks are used. Which mask is applied depends on the gradient direction of the reference pixel. Every mask includes a reference pixel and a neighborhood of two adjacent pixels. When the value of the reference pixel is greater than that of neighborhood pixels or greater than one and equal to the other pixel, the reference pixel is selected as an edge pixel, and the other two are eliminated. Figure 2-6 shows the thinning masks.



a) Gradient direction 4 or 8. b) Gradient direction 2 or 6.
c) Gradient direction 1 or 5. d) Gradient direction 3 or 7.

Figure 2-6: Edge Thinning Templates

2.3 Hough Transform for Elliptic Shapes

2.3.1 Hough Transform

The HT algorithm was introduced by Hough in 1962 [6]. The original algorithm detected a line in an image by using the relation between image space and a parameter space. Suppose a line in an image space passes through two points (x',y') and (x'',y''). When only those two points are known, the line can be defined.

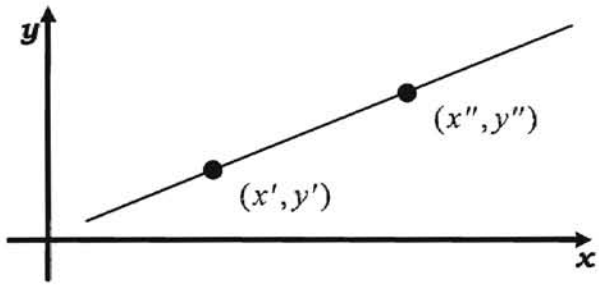


Figure 2-7 : A line defined by two points

The equation for a straight line that passes through a point (x,y) is $y = mx + c$. Now we can easily define those lines that pass through the point (x',y'). All lines with every m and c that satisfy the equation $y' = mx' + c$ pass through the point. Similarly, all lines that satisfy the equation, $y'' = mx'' + c$, pass the point (x'',y''). These two line equations can be rewritten as line equations of m-c parameter space :

$$c = -mx' + y' \text{ and } c = -mx'' + y''. \qquad \textbf{(Eq. 2-7)}$$

These lines can be represented in the parameter space and we know these two lines intersect at the point (m',c').

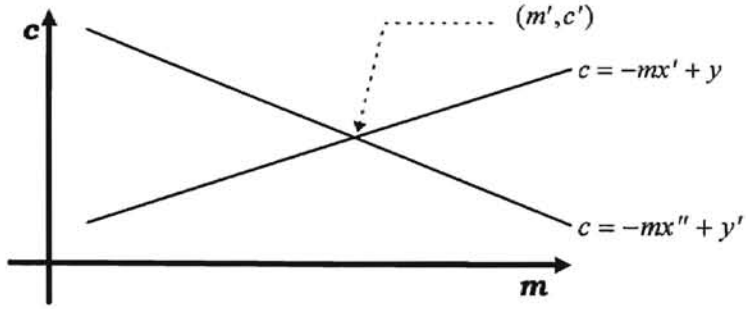


Figure 2-8 : A Parameter Space

Thus the line passing through the points (x',y') and (x'',y'') is $y = m'x + c'$. This is the main idea of the Hough transform.

This HT algorithm was modified by Duda and Hart in 1972 [4]. The original HT uses the slope-intercept *(m-c)* parameter space in two dimensions. The space is not bounded and this makes the applying HT complicated. Duda and Hart suggested alternative interpretations to eliminate that problem.

A straight line in the image space could be represented as a single point in the parameter space by changing the interpretation of the parameters. A single line could be represented as the angle, θ, of its normal and its distance from the origin. The equation of a line could be shown as

$$x\cos\theta + y\sin\theta = \rho. \qquad \textbf{(Eq. 2-8)}$$

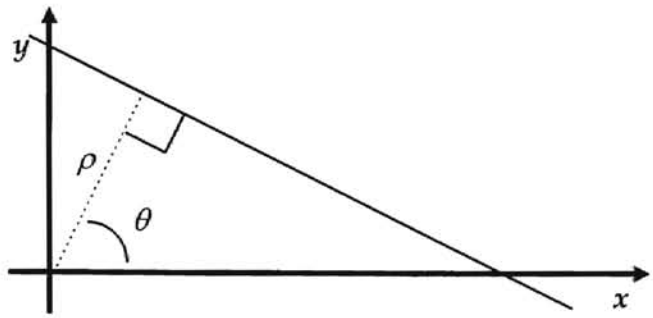


Figure 2-9 : A line represented by parameter ρ and θ.

Here, θ is restricted to the interval $[0,\pi]$, so the normal parameter for a line is unique. Every line in the image space corresponds to a point in the θ-ρ parameter space. Now, suppose that an image includes a set of points and we want to define a line that includes those points. The line passing through the point(x_i, y_i) is defined as

$$x_i \cos\theta + y_i \sin\theta = \rho \qquad (i = 0,\ 1,\ 2,\ \ldots,\ n) \qquad \textbf{(Eq. 2-9)}$$

When this equation is applied to the other points, a point where the θ-ρ curves intersect is going to be shown. This intersection point of the parameter space defines a line that includes the points. This interpretation can be extended to detect a circle. A circle can be represented as a set of parameters, the center, (x,y) and the radius, r. Now the circle is represented as a single point in the three dimensional x-y-r parameter space and HT can be used to detect the circle. If radius is already known, the location of the circle can be defined with one accumulator array.

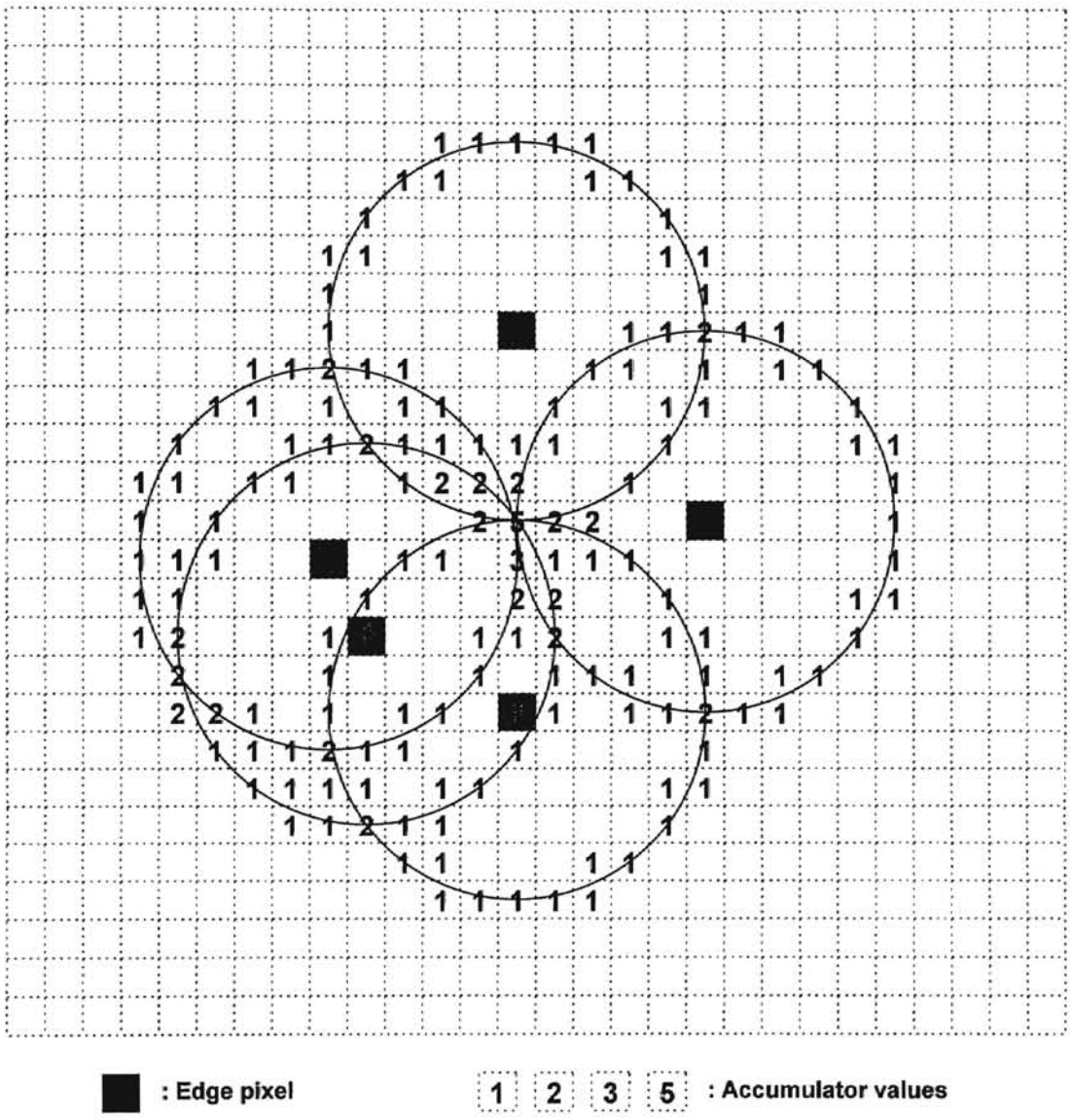


Figure 2-10 : An accumulator array (Radius = 5)

2.3.2 Generalized Hough Transform.

The generalized HT (GHT) algorithm may be used to detect any shape. This GHT algorithm was proposed by Ballard in 1981[3]. In a computer application for image processing, an image is the image space and a memory buffer, an accumulator, that has same size as the image is the parameter space. First, a template T that contains information about the shape to be detected is created. The template includes the shape and a reference point R of the shape. The shape of the template is same as the shape to be detected. The only difference is the shape of the template is rotated 180 degrees from the original shape. In higher-dimensional spaces, the template is constructed by reflecting the original shape through the reference point. In two dimensions, this is the same as a 180° rotation.

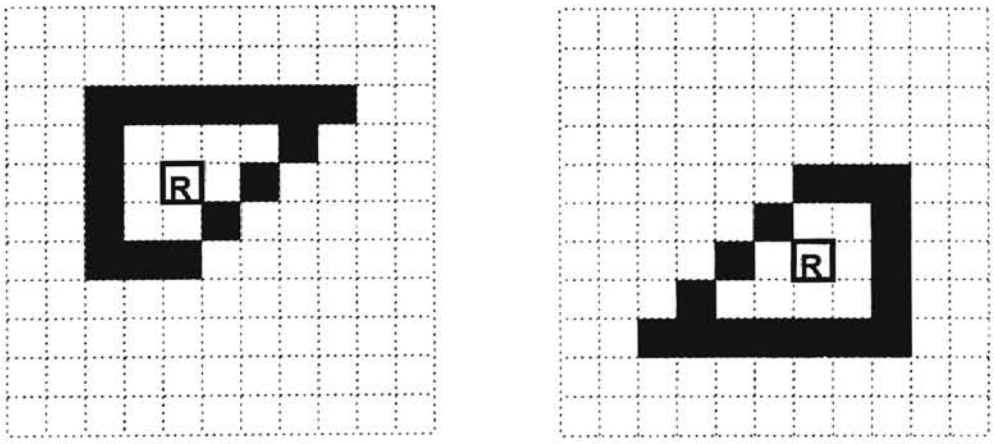


Figure 2-11 : Original Image and Template for GHT

The image can be transformed into an edge image by an edge detection algorithm (see Section 2.1.2). To detect the shape, assume that

every edge pixel is a candidate that the reference point of the template is possibly on. For the HT, an edge pixel E_i is chosen. Locations of boundary pixels of the template are defined assuming the edge pixel is the reference point. The values of corresponding locations in the accumulator are incremented. The process is repeated for every edge pixel in the image. After the iteration, a high peak point is going to be shown in the accumulator. This peak point defines the real reference point and the location of the shape.

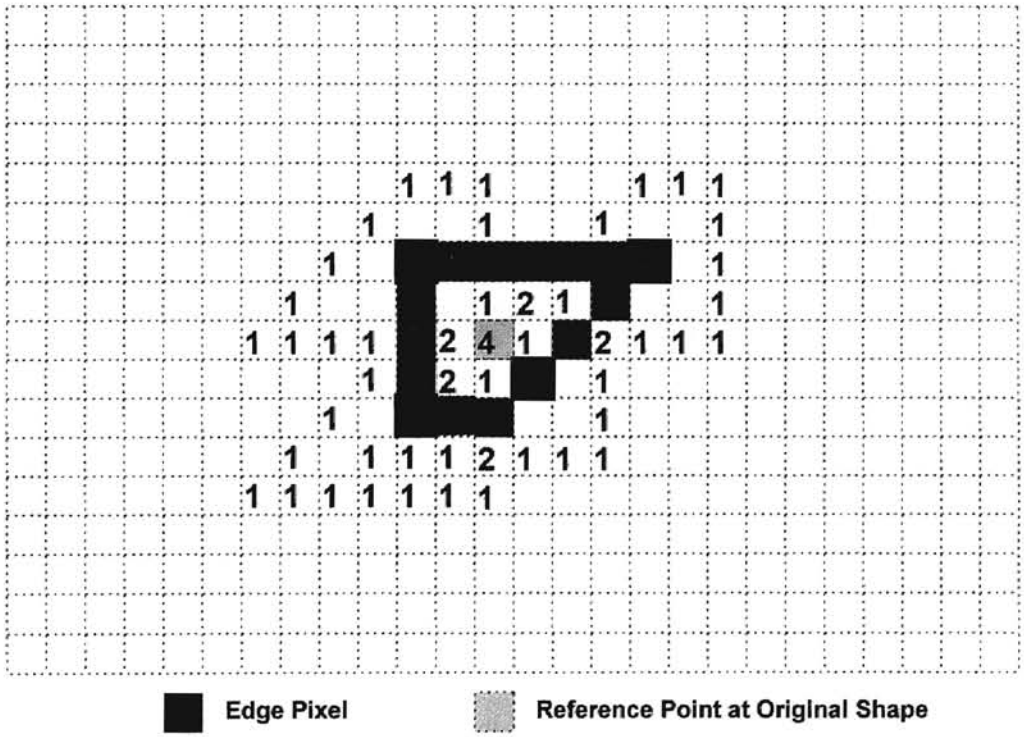


Figure 2-12: Accumulator

2.3.3 GHT with Gradient Direction.

Using gradient direction reduces the computation necessary for the GHT. The GHT without gradient direction is simple to implement, however the algorithm requires a great deal of computation. Some parts of the template increase unnecessary pixels in the accumulator, increasing the computation time. Figure 2-13 shows the unnecessary increments in the GHT accumulator. In the figure, only pixels in the edge boundary, not boxed, are effective pixels for the finding the object.

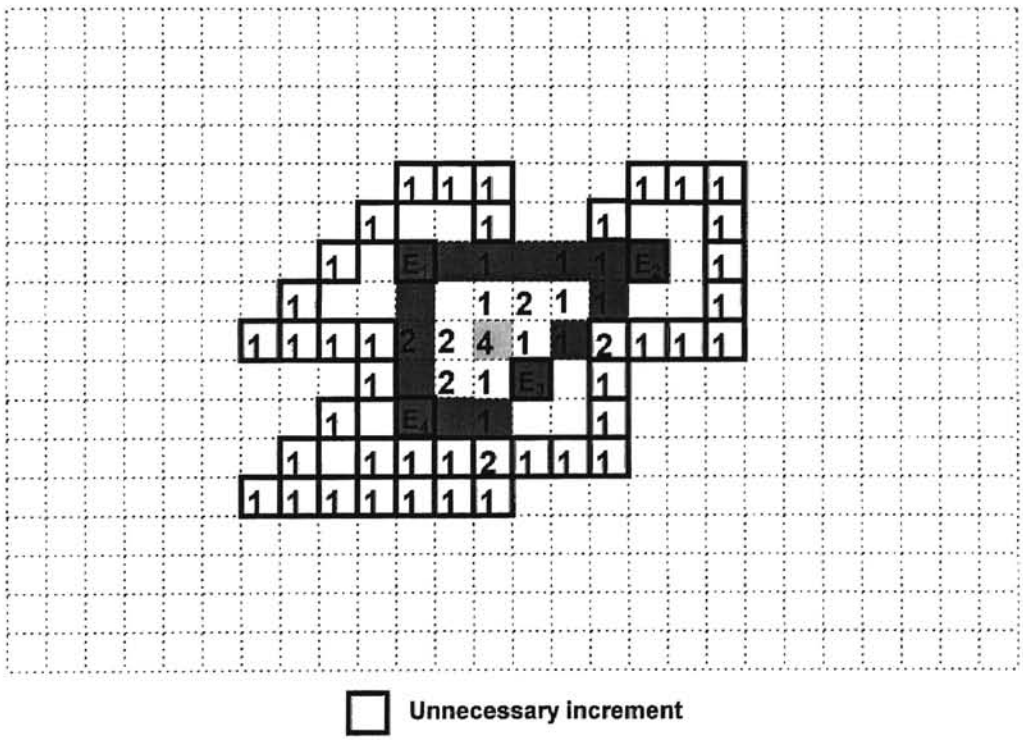


Figure 2-13: Example of unnecessary increments of GHT

In this algorithm, only the part of the template that is consistent with the gradient direction is applied to increase the value of the accumulator.

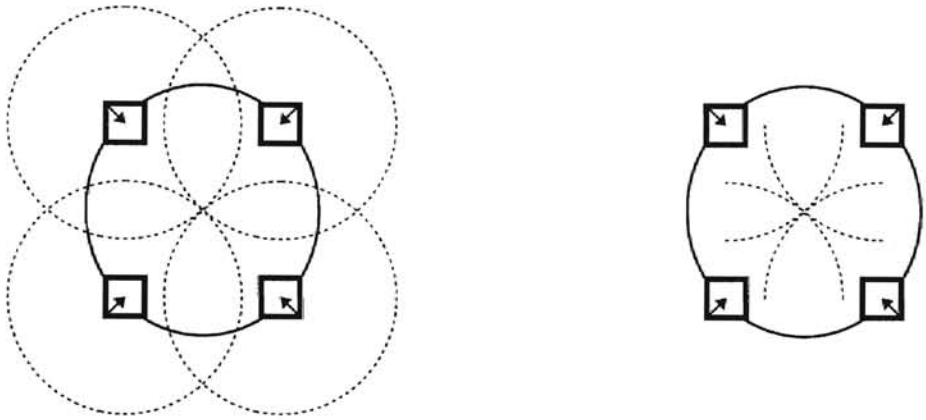
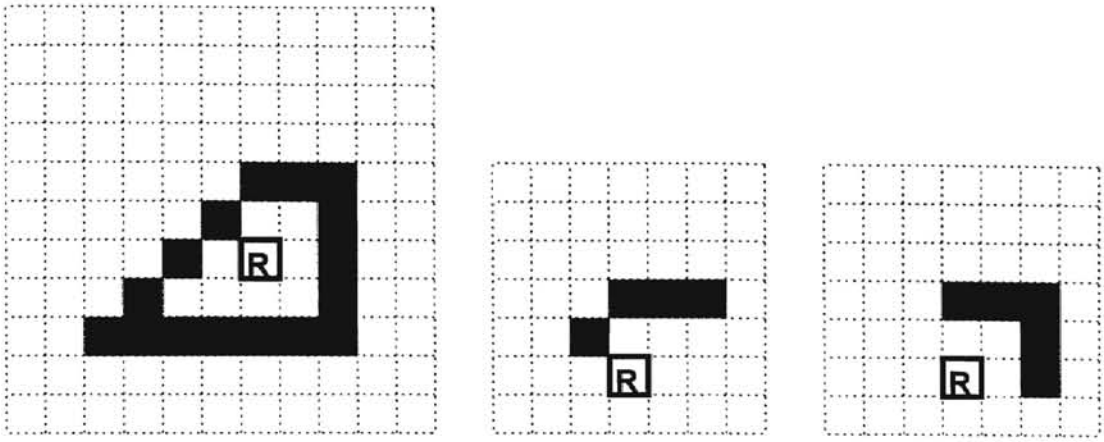


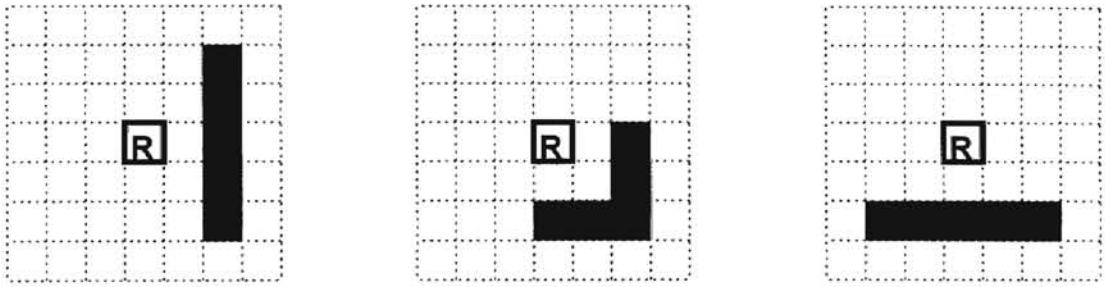
Figure 2-14: GHT without GD and GHT with GD

For instance, the proposed method detects eight different gradient directions, numbered from 1 to 8. Every edge pixel is labeled with the appropriate number during edge detection (see Section 2.1.2 and Figure 2-4) and a template is separated into eight small templates (see Section 2.3.4). Every small template is labeled with an appropriate number that represents a gradient direction. When the GHT algorithm processes an edge pixel, only a small template that corresponds to the gradient direction of the edge pixel is applied instead of the whole template. For instance, Figure 2-15 shows small templates separated from the template in Figure 2-19, and Figure 2-16 shows an example GHT with gradient direction using the same image as Figure 2-16.

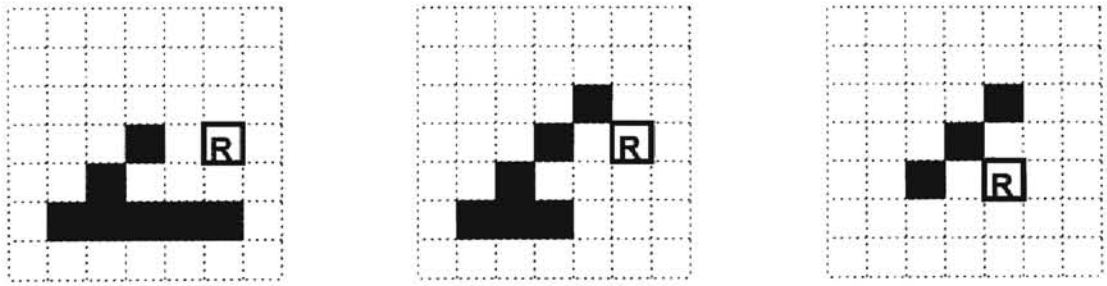


Whole Template

Small Templates for Direction 1 and 2



Small Templates for Direction 3, 4 and 5



Small Templates for Direction 6, 7 and 8

Figure 2-15: Example of Templates for GHT with Gradient Direction

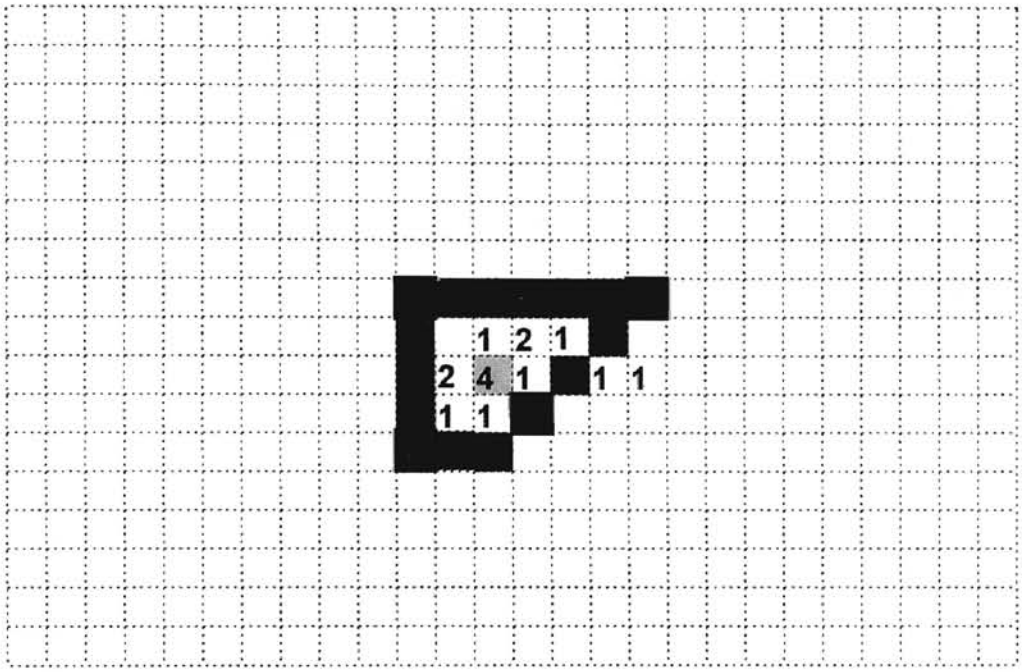


Figure 2-16: Example of GHT with Gradient Direction

2.3.4 Elliptic Template Generation

The template generation algorithm for the GHT locates pixels that accurately approximate the real boundary of an ellipse. Since the image that is used is a digital image, it is impossible to generate a perfect elliptic template. In fact, the template has an optimal shape that approximates the ellipse.

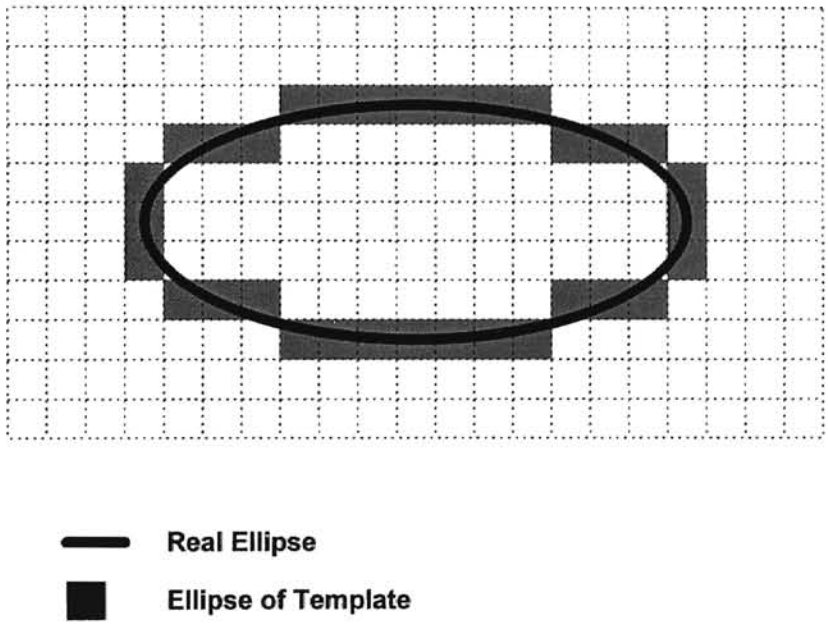


Figure 2-17: HT Template Generation

The template is a set of X-Y values of pixels under a logical coordinate system with the origin on the center of the ellipse: $T = \{P_i = (x_i, y_i) | i = 0,\ 1,\ 2,...,\ n\}$. In the template generation algorithm, P_i is defined from P_{i-1} by the algorithm. Every P_{i-1} has eight neighbors. The one of those neighbors that is the closest to the real ellipse is chosen as P_i. Initially $P_0 = (x_0, y_0)$ has to be defined to start the process. Since

the equation of an ellipse is $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$, P_0 could be $(0,\pm b)$ or $(\pm a,0)$. In this algorithm $(-a,0)$ is chosen as the starting point.

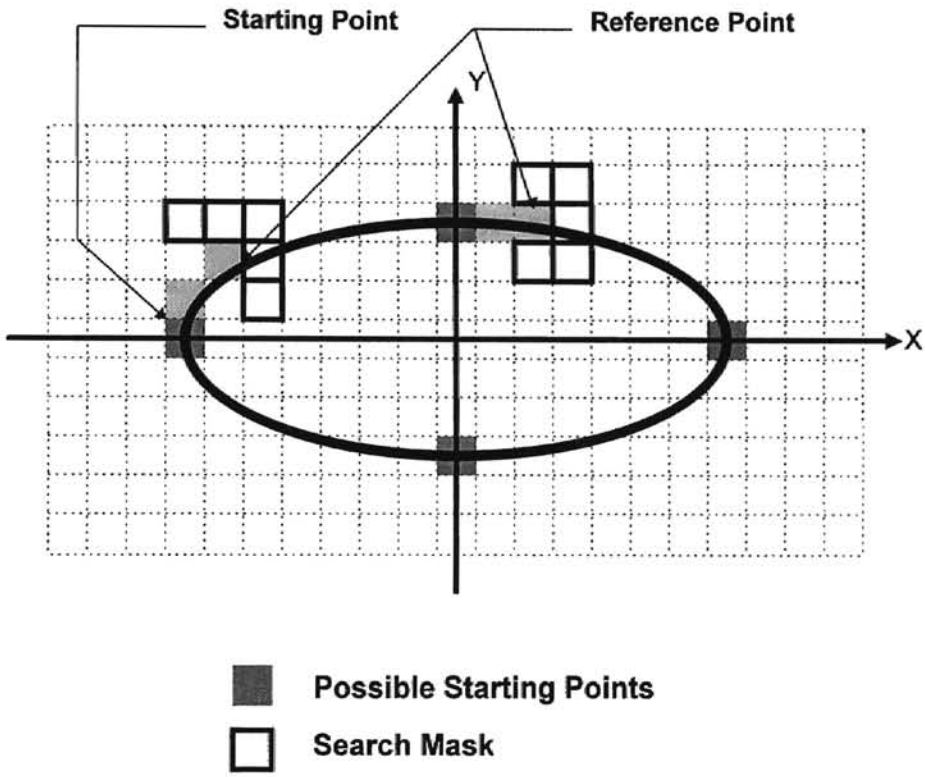


Figure 2-18: Finding Next Optimal Point

After determining P_0, P_1 is chosen from among its eight neighbors. P_1 has five neighbors that may possibly be P_2. Even though P_1 has eight neighbors, two neighbors that are adjacent to P_0 are eliminated because there is no angle that sharp in an ellipse. Every P_{i+1} is chosen from P_i by minimizing D. D can be described as below without rotation of the ellipse. D is zero when the (x,y) point is exactly on the boundary of an ellipse.

$$D = \left| \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 \right| \qquad \text{(Eq. 2-10)}$$

This process iterates until P_i is adjacent to P_0. Algorithm 2-2 shows the overall algorithm for the template generation.

The generated template is for the HT without gradient direction. For the HT with a gradient direction, this template has to be separated into small parts to be applied for each gradient direction. In the proposed method eight different gradient directions are used (see section 2.2.2). Only the appropriate part of the template is applied for the gradient directions of the edge pixel. Figure 2-19 shows which part of the big template is applied for each gradient direction and an example is shown in Figure 2-15.

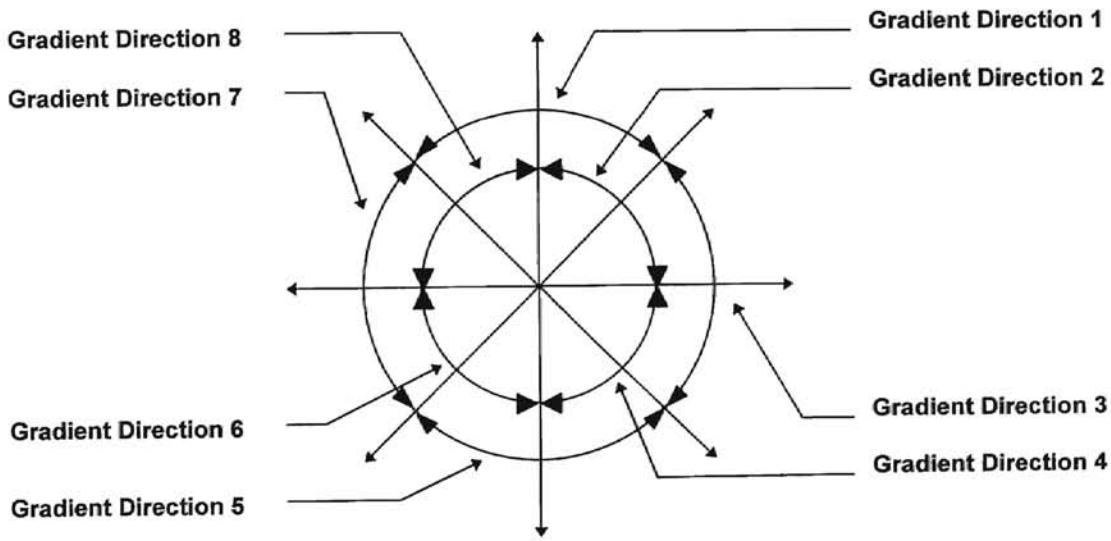


Figure 2-19 : Separated Template for GHT with GD

```
Template_generation()
{
        P0 ← (-a,0);
        Search eight neighbors of P0;
        Select one neighbor that minimizes D as P1;
        i ← 1;
        while(Pi != P0)
        {
                Search eight neighbors of Pi;
                Eliminate two neighbors adjacent to Pi-1;
                Select one neighbor that minimizes D as Pi+1;
                i ← i + 1;
        }
        Separate the template into eight parts;
}
```

Algorithm 2- 2

Template Rotation

Although this algorithm generates fine elliptic templates, the limitation of this algorithm is that only a vertical or horizontal elliptic templates can be generated by this algorithm. Since the directions of cell nuclei in a medical image are not uniform, it is very difficult to expect fine matching with only vertical or horizontal elliptic templates. For this reason, many different directions of elliptic templates were applied for more precise matching to the real cell nuclei (See Rotating Templates in Section 2.3.5). For the matching, it is required to rotate the elliptic templates. The rotation is performed by applying a simple rotation matrix to Eq. 2-10. When θ is the rotation angle, the rotation matrix is

$$R_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \qquad \text{(Eq. 2-11)}$$

In Eq. 2-12, *(x,y)* is an original coordinate and *(x',y')* is the coordinate rotated by the angle θ.

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

$$\begin{aligned} x' &= x \cdot \cos\theta + y \cdot \sin\theta \\ y' &= -x \cdot \sin\theta + x \cdot \cos\theta \end{aligned} \qquad \textbf{(Eq. 2-12)}$$

The rotated ellipse is solved by substituting Eq. 2-12 in Eq. 2-10.

$$D_R = \left| \frac{(x\cos\theta + y\sin\theta)^2}{a^2} + \frac{(y\cos\theta - x\sin\theta)^2}{b^2} - 1 \right| \qquad \textbf{(Eq. 2-13)}$$

Now a rotated elliptic template can be generated by substituting D_R in Algorithm 2-2 instead of *D*.

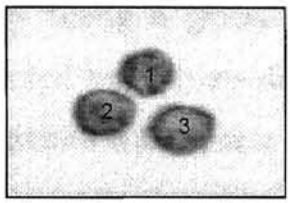
2.3.5 Iterative GHT

As mentioned in the previous section, usually the GHT is an algorithm to locate an already known shape from an image. However, the shapes of cell nuclei are not fixed in the proposed method and the image includes many different shapes and sizes of cell nuclei.. Although the shape of a cell nucleus is unknown, the region where the shape is expected to be could be defined and several different templates can be applied in the region. The purpose of the proposed method is to find the template that best approximates the shape of each nucleus.

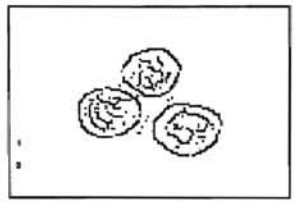
Assume that an image includes a cell nucleus, C, with unknown shape and that there are templates, $\{T_1, T_2, T_3, \ldots, T_n\}$, one of which closely approximates C. We can generate n accumulators, $\{A_1, A_2, A_3, \ldots, A_n\}$ by performing the GHT with the n templates. It is already known when the image includes a feature that closely matches the template, the accumulator shows the highest peak value. Although every accumulator includes its own peak value, the sizes of every value are different. If T_i is the most closely approximated template, the highest peak point should appear in A_i. In other accumulators, widely distributed and comparatively low values appear. We thereby determine that the image contains a nucleus with the shape at template T_i, centered on the peak point at accumulator A_i. Figure 2-20 (a) shows the example of this case. The cell nucleus 1 approximates the 11×14 ellipse. Thus when GHT is performed with the 11×14 elliptic template, the highest peak value appears as a darkest point in the accumulator (a).

Now consider an image that includes m different cell nuclei. When the GHT is performed with n different templates $\{T_1', T_2', T_3', \ldots, T_n'\}$, n accumulators $\{A_1', A_2', A_3', \ldots, A_n'\}$ are generated. If the image

includes cell nuclei that approximately match the shape of template T_i', the accumulator A_i' includes peak values on the locations of those nuclei and low values elsewhere. This situation occurs similarly on other accumulators. Thus it is possible to determine which template matches which cell nuclei. Figure 2-20 shows the examples.



Original Image



Edge Image

# of cell nucleus	Approximate ellipse size
1	11×14
2	12×15
3	12×17

(a) (b) (c)

Accumulator (a) : GHT with 11×14 elliptic template
Accumulator (b) : GHT with 12×15 elliptic template
Accumulator (c) : GHT with 12×17 elliptic template

Figure 2-20 : Example of GHT with 3 different templates

Global Accumulator

Even if it is possible to find the approximate shape and location of every cell nucleus, there is still one problem. For the iterative GHT, a huge amount of storage is necessary for the accumulators. If the image size is $m \times n$ and t templates are applied for the GHT, then the storage requirement for the accumulator is $\Theta(mnt)$. However this problem can be solved easily by using a global accumulator. The set of accumulators can be represented by a three-dimensional structure. Assume that a cell nucleus C_j' matches a template T_j', and the accumulator A_j' includes the highest peak point on the location (x,y). Now a line passing through all of the accumulators at the point (x,y) can be imagined. The line includes t points (the number of accumulators) and every point may have a different value. Since T_j' is the most closely approximated template to the cell nucleus, the jth point in the line is the highest. If a value of location (x',y') of an accumulator A_k' is higher than the value of the same location (x',y') of another accumulator, A_l', that means the template T_k' is a better match to the cell nucleus than template T_l'. Since we are interested in finding only the best match among the templates, only the highest value is important among all values on the line.

From the same idea, a global accumulator *GA* is designed. Every point of the global accumulator has the highest value of all accumulators at that location. Therefore the global accumulator can be written as,

$$GA(x,y) = Max\{A^1(x,y),\ A^2(x,y),\ A^3(x,y),\ \ldots,\ A^t(x,y)\}.$$

Now the collection of accumulators is not necessary, rather a local accumulator for each GHT step and a global accumulator are the only required accumulators. Finally, the global accumulator includes the highest values of each accumulator location. This global accumulator

has to include all necessary information. For instance, when the global accumulator was not applied, every different accumulator represented its own shape and direction of the template. In our system, the four-layer global accumulator was used to preserve the information. The first layer includes the accumulator values, the second and the third layers include the shape of the template by preserving two parameters of the ellipse, and the fourth layer includes the direction of the template.

Accumulator Normalization

Comparing the results in different accumulators presents another problem. Suppose that there are two templates T_k' and T_l' with different sizes: T_k' contains n points, and T_l' contains $3n$ points. After the GHTs, it is possible that an accumulator A_k' shows the value n and A_l' shows the value $1.5n$ on the same position, (x,y). Even if A_l' shows a higher value than A_k' does, it cannot be said that the T_l' is better than T_k', even though the accumulator value is higher. The template T_l' showed 50% matching and T_k' showed 100% matching. To prevent this error, all values are normalized by dividing by the number of points of the template before updating the global accumulator. After the normalization, A_l' includes the value, 0.5 and A_k' includes the value, 1.

Rotating Templates

We used elliptic templates for iterative GHT. Even if most of the cell nuclei are shaped like ellipses, it is difficult to expect precise matching with just vertical and horizontal elliptic templates, since the directions of the cell nuclei have several different directions as in Figure 2-21. To minimize this difficulty, we applied many differently directed templates. In this research, templates are rotated every 25°.

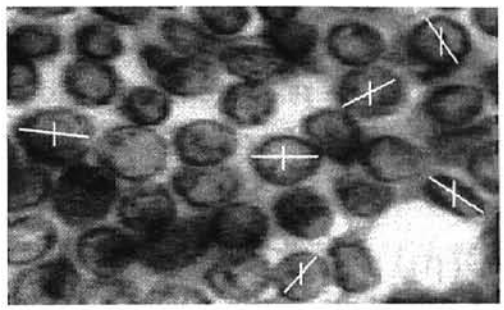


Figure 2- 21 : Directions of Cell Nuclei

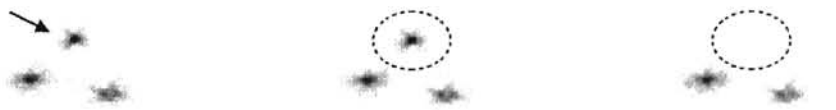
The result of iterative GHT was enhanced by applying the above additional steps. Algorithm 2-3 shows the iterative HT routine.

```
Iterative_HT( X, Y, GA )
Image Size : ( X, Y )
Global Accumulator : GA
{
        Local Accumulator : A
        for i = Min_size to Max_Size
        {
          for j = Min_Size to Max_Size
          {
            for θ = 0 to 180 step r      /* r=25 degree */
            {
                T = template with size i by j rotated by r;
                A = Hough Transform with T;
                for x = 0 to X
                {
                        for y = 0 to Y
                        {
                                A(x,y) ← ( A(x,y) / |T| );
                                if( A(x,y) > GA(x,y) )
                                        GA(x,y) ← A(x,y);
                        }
                }
        }
}
```

Algorithm 2- 3

2.3.6 Peak Finding

After the iterative GHT, the peak finding step is performed to define the location of a cell nucleus and the shape of the template that closely matches the cell nucleus. This process is performed by finding the highest value in the global accumulator. When the highest value is found, the location of the accumulator pixel that stored the value is defined as the location of the cell nucleus and the approximate shape is defined from the information of the pixel. After the defining, an area of the global accumulator which is included in the template is eliminated. When a high peak point is selected, the high peak point is the top of a plateau. Thus neighbors of the high peak point are also high values. These high value neighbors may be selected as high peak points in the next selection. This defines several template for one cell nucleus. The elimination is to prevent the situation. Figure 2-21 shows the process.



Searching Highest Value **Defining Shape** **Eliminating Area**

Figure 2- 22 : Eliminating an area of global accumulator

This process is iterated until there are no more values that satisfy the threshold value in the global accumulator. Since the accumulator values that are lower than the mean of the accumulator values are usually useless, we eliminated accumulator values that are lower than the mean of the accumulator values before the peak finding. The threshold value

may be in a range from 0 to 1. When the threshold value is 0 , all accumulator values are considered for the peak finding. If the value is 0.5, the accumulator values higher than the mean value are considered. The threshold value 1 means only the maximum value of the accumulator is considered.

Peak Sharpening

During this process, unexpected problems occur. The first problem is plateaus that occur in the global accumulator. Even though we are trying to match elliptic templates to cell nuclei, there is no cell nucleus that perfectly matches the elliptic templates. Therefore, plateaus sometimes occur near the center of a poorly matched nucleus. These plateaus cause difficulty in choosing the highest peak point. When a plateau appears, we choose the center of the plateau as the highest value. To achieve this, peak sharpening is performed as preprocessing for the peak finding step. This peak sharpening process increases the value at the center of the plateaus. Figure 2-22 shows the mask for the peak sharpening process.

x_8	x_1	x_2
x_7	x_0	x_3
x_6	x_5	x_4

$$x_0' = \frac{1}{2}x_0 + \frac{1}{16}(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8)$$

Figure 2-23 : Peak Sharpening Mask

Overlapped Template

The second problem is that of overlapped templates. Some cell nuclei match two or more different elliptic features. Even if a cell nucleus almost perfectly matches an elliptic template, if some edge data are lost, the cell nucleus may match several different elliptic templates. In that case, two or more high peaks appear inside the same cell nucleus and one of them has to be chosen. We gave priority to wider templates by multiplying a compactness factor to the accumulator values, since it was observed most cell nuclei are wide rather than narrow. Every newly selected high peak point is compared with the list of high peak points that have already been chosen. If one of them is included in the area of the template indicated by the new high peak point, both templates are considered to be overlapped templates. If overlapped templates are detected, the compactness factors of both templates are calculated. The compactness factor is *(Perimeter)²/Area*. Thus, when an ellipse is

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \qquad \textbf{(Eq. 2-14)}$$

the compactness factor of the ellipse is

$$4\pi\left(\frac{a^2+b^2}{2}\right)/\pi ab\,. \qquad \textbf{(Eq. 2-15)}$$

The compactness factor of the wider template should be bigger than the narrow one. Thus, a slight priority is given to the wider template.

Chapter 3 : Tests

This chapter shows the test samples, the output of every step and results. Two 256 gray scale medical images were tested to evaluate the performance of the system. The first image includes over 200 cell nuclei and some unfocused areas. Roughly 225 cell nuclei are recognizable by a human operator. The second image includes over 15 cell nuclei and 14 of those are recognizable. Figures 3-1 and 3-8 show the original sample images. Figure 3-2 and Figure 3-9 show the outputs of the median filtering. Figure 3-3 and Figure 3-10 show the edge images. These edge images are not thresholded. Figure 3-4 and Figure 3-11 show the contents of the global accumulators. The contents of the global accumulators were converted from double precision floating point values to integer values ranging from 0 to 255 to be shown as an image. In those images, the darker pixels indicate higher values.

After the iterative GHT, the peak finding routine is performed to define which elliptic template is most closely matched to which cell nucleus. The peak finding process was performed with several different thresholding values. The thresholding value 0.5 means accumulator values higher than $(Highest\_value - Average\_Value) \times 0.5$ are evaluated to find high peak values. Figure 3-5, 6 and 7 show the final results achieved from the test image 1. Figure 3-12, 13 and 14 show the final results achieved from the test image 2.

3.1 Test 1

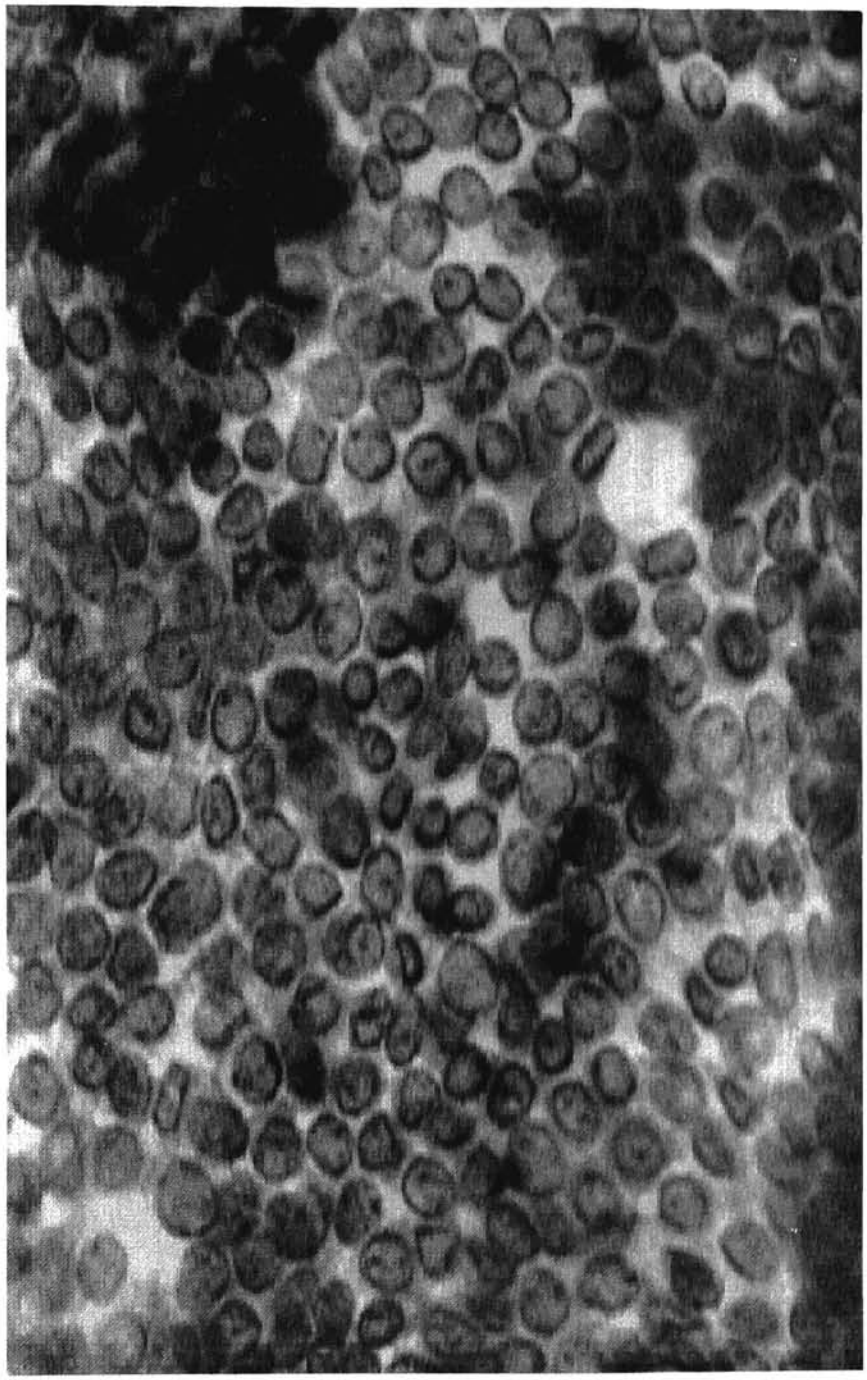


Figure 3-1 : Original Image

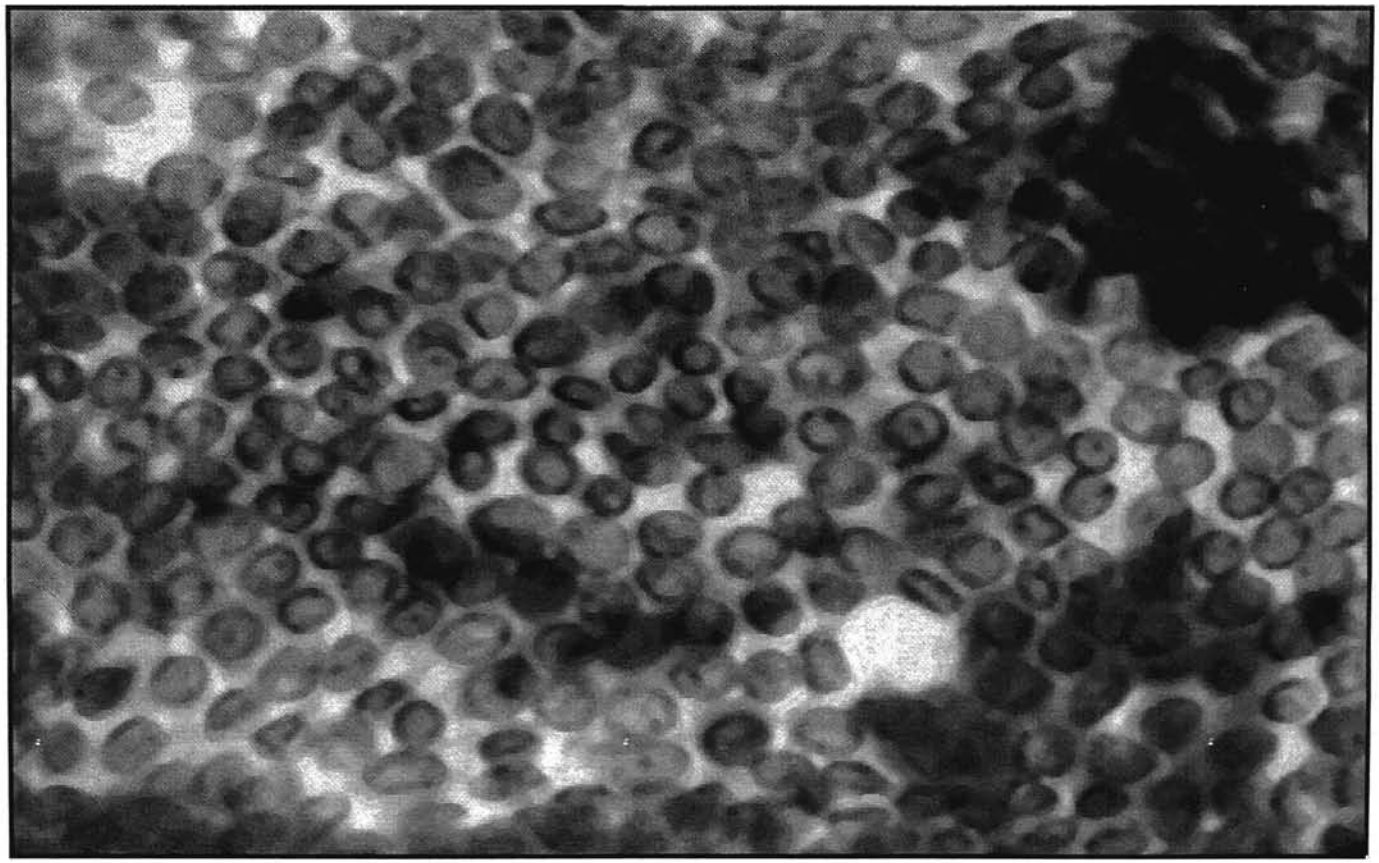


Figure 3-2 : Median Filtered Image

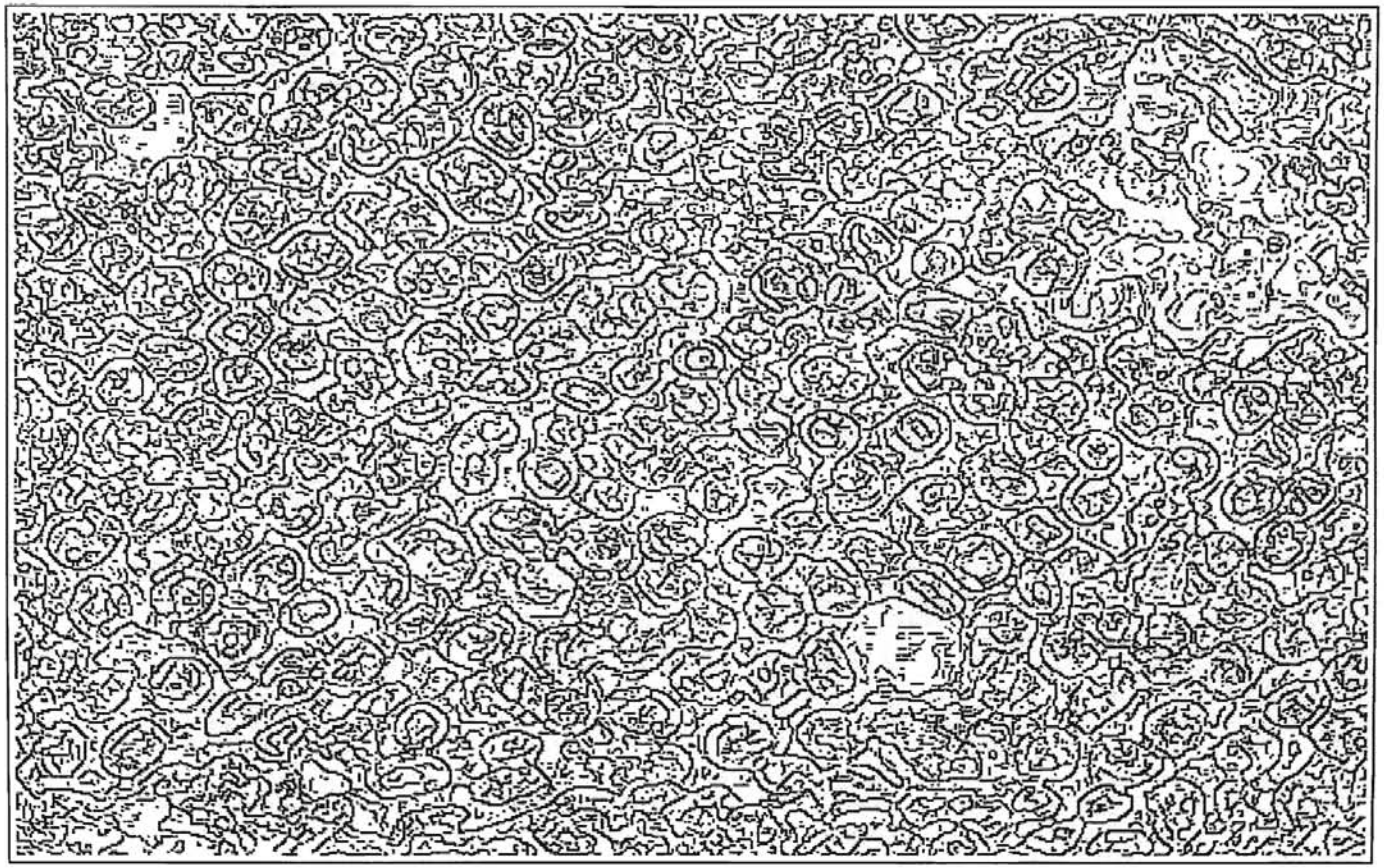


Figure 3-3 : Edge Image

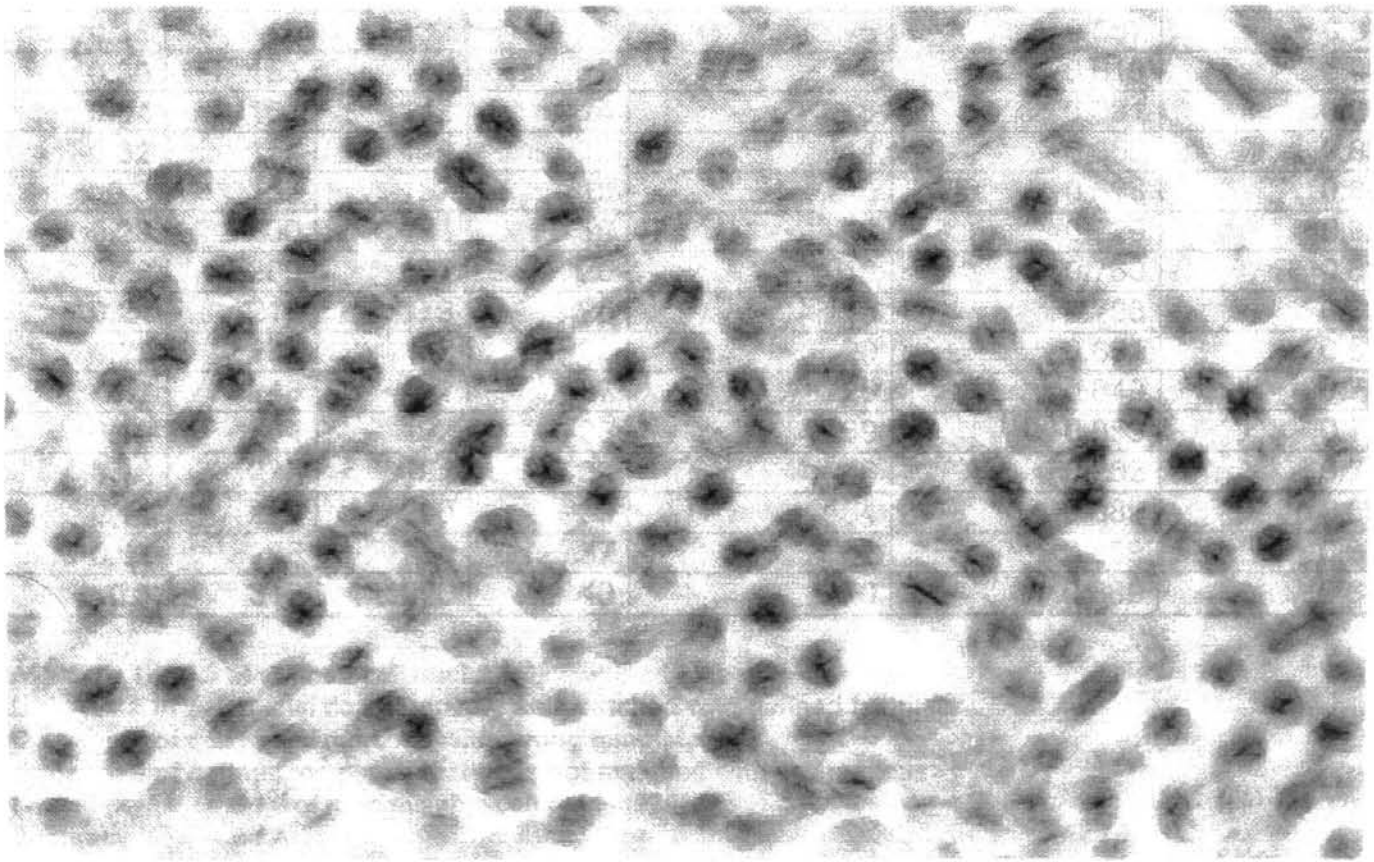


Figure 3-4 : Global Accumulator

	A	B	C	Error	Sensitivity	Accuracy
0.8	13	13	0	0	0.058	1.0
0.7	29	29	0	0	0.129	1.0
0.6	55	54	1	0	0.240	0.982
0.5	91	87	4	1	0.387	0.956
0.4	137	126	6	5	0.560	0.920
0.38	145	132	8	5	0.587	0.910
0.36	153	138	9	6	0.613	0.902
0.34	164	142	14	8	0.631	0.866
0.32	177	148	15	14	0.658	0.836
0.3	187	157	15	15	0.698	0.840
0.28	200	164	18	18	0.729	0.820
0.26	209	171	20	18	0.760	0.818
0.24	218	179	20	19	0.796	0.821
0.22	230	187	23	20	0.831	0.813
0.2	243	191	27	25	0.849	0.786
0.18	264	192	33	39	0.853	0.727
0.16	273	194	33	46	0.862	0.711
0.14	289	200	35	54	0.889	0.692
0.12	298	200	37	61	0.889	0.671
0.1	310	200	40	70	0.889	0.645

A = Total # of defined cell nuclei
B = # of cell nuclei comparatively well matched with defined template
C = # of cell nuclei matched with wrong template
Error = # of defined templates that does not matched with cell nucleus
Sensitivity = B / # of recognizable cell nuclei(225)
Accuracy = B / A

Figure 3-5 : Table of results of test one

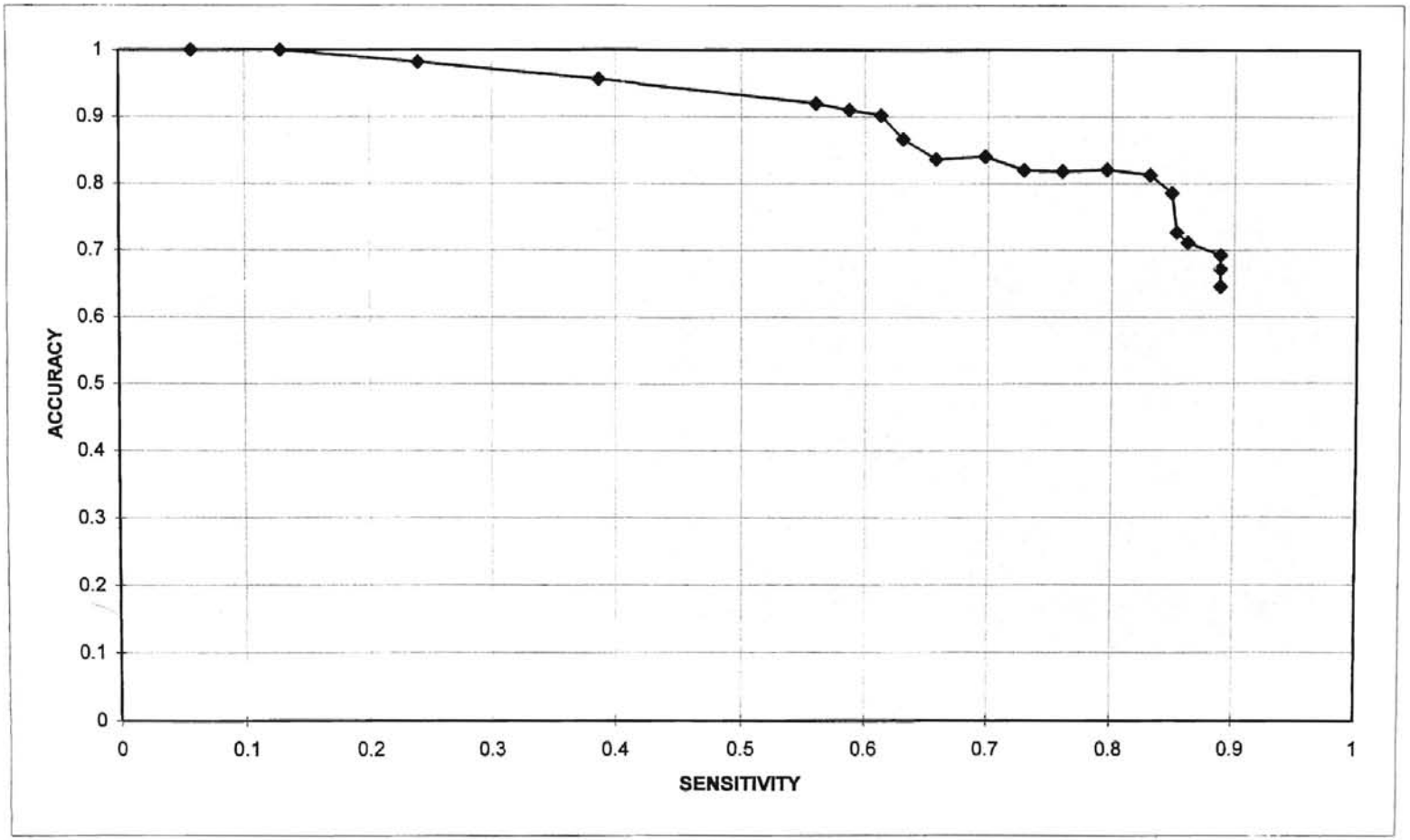


Figure 3-6 : Accuracy and Sensitivity

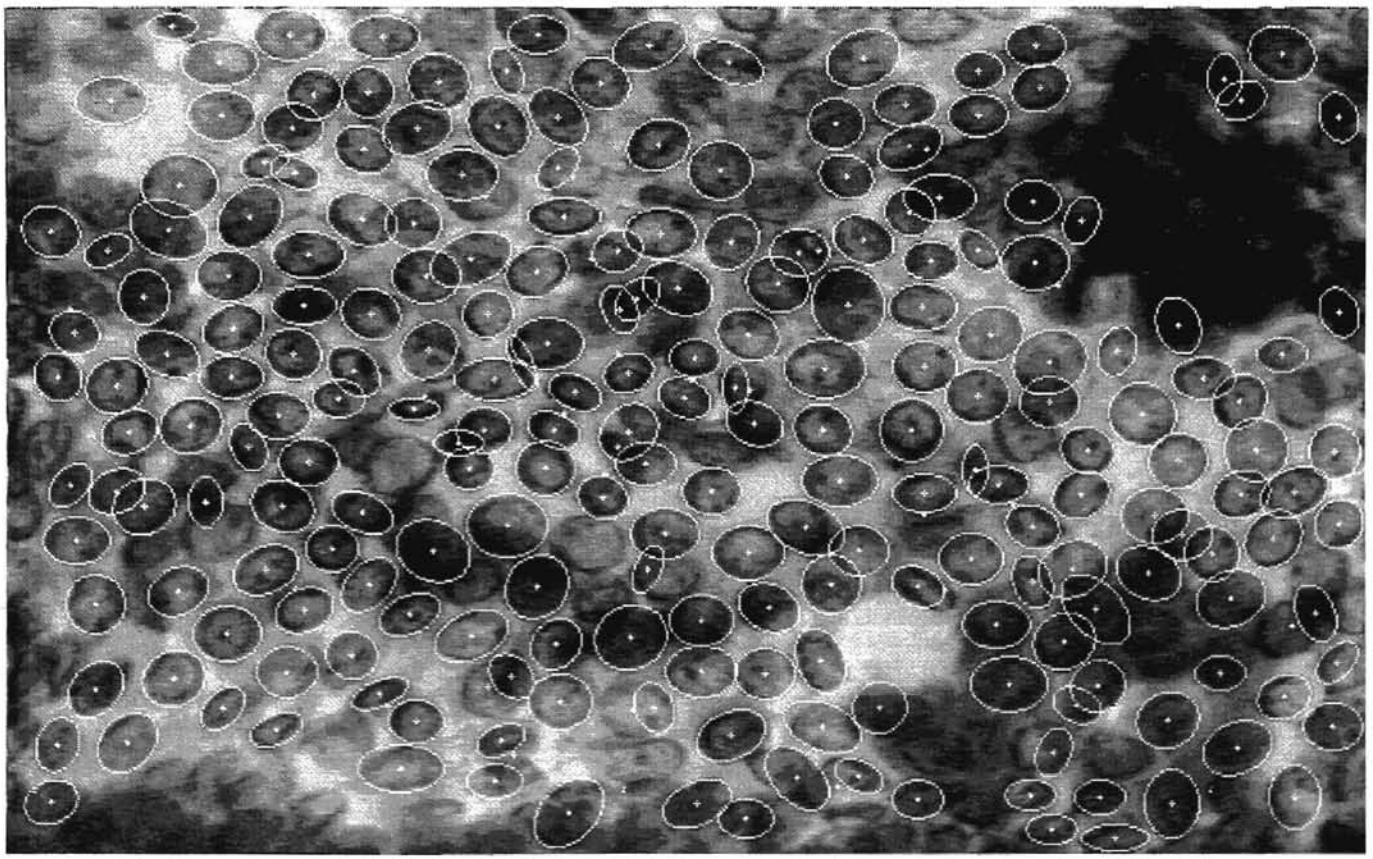


Figure 3-7 : Test Result (Thresholding Value = 0.24)

3.2 Test 2

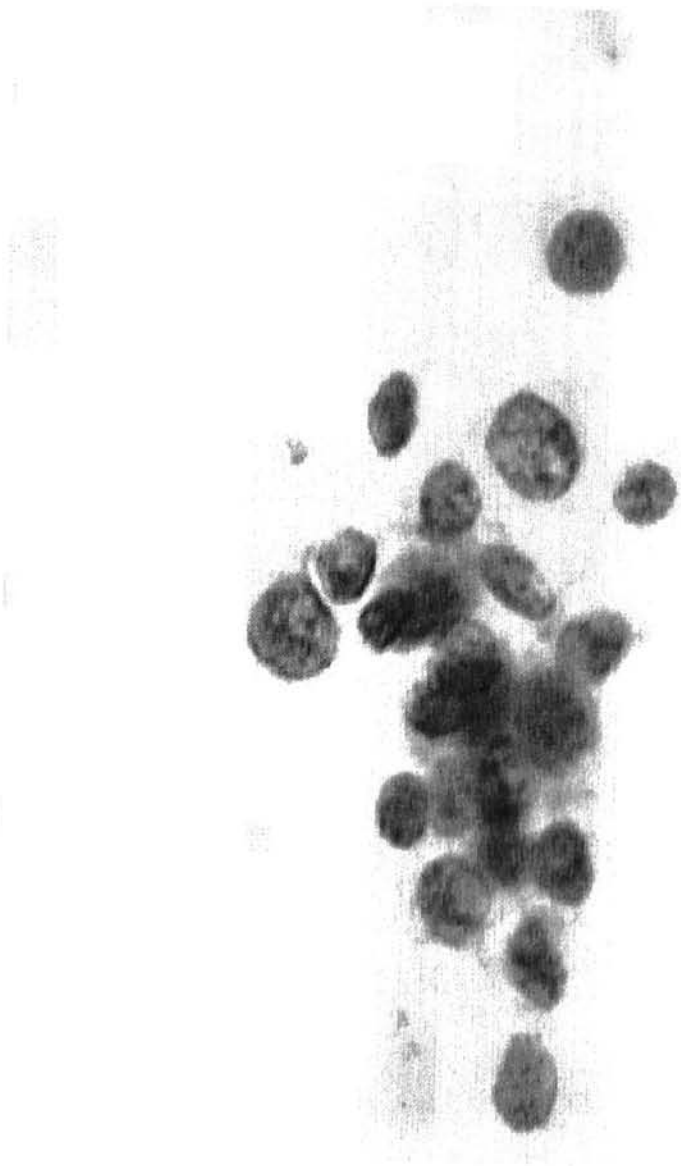


Figure 3- 8 : Original Image

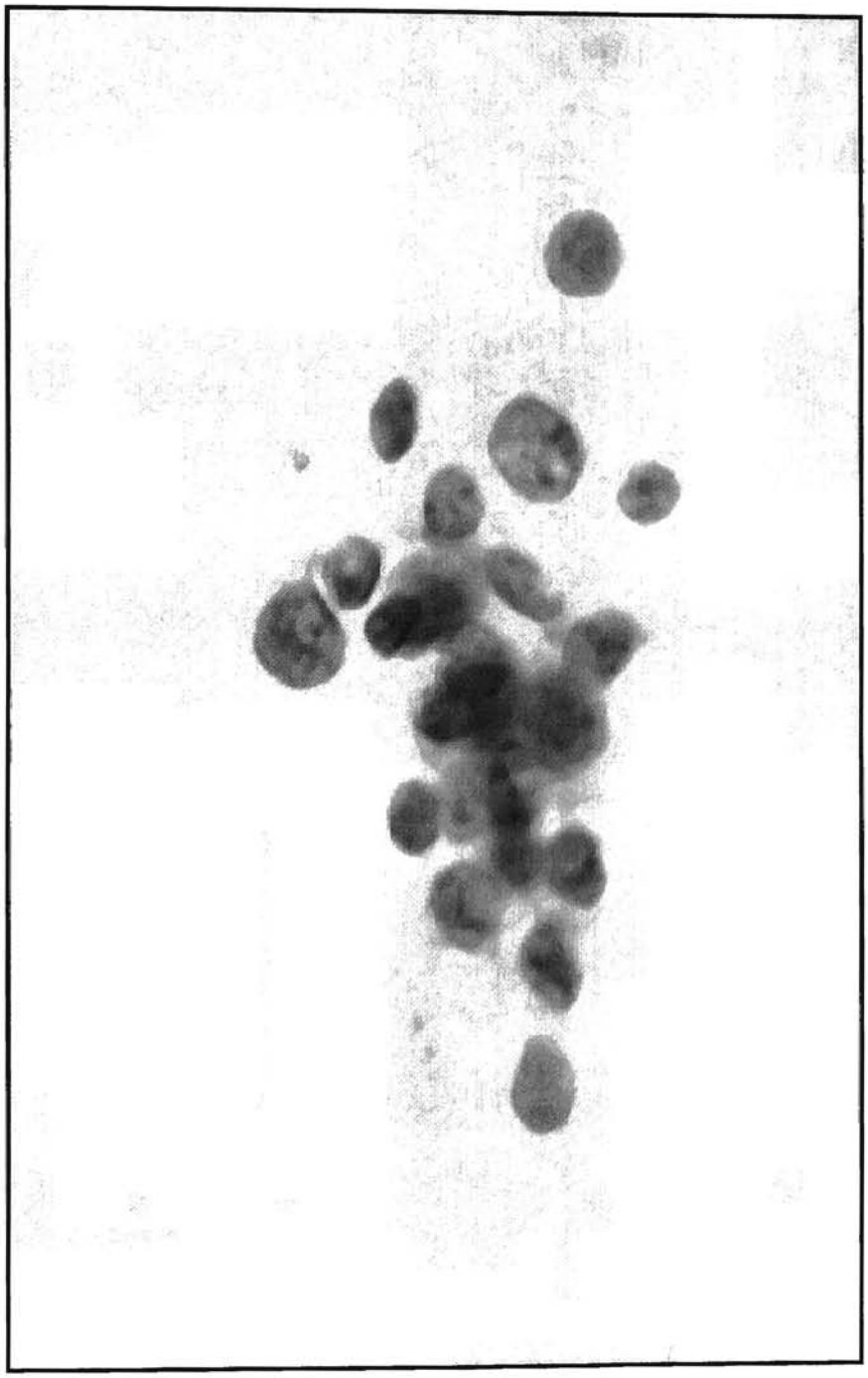


Figure 3-9 : Median Filtered Image

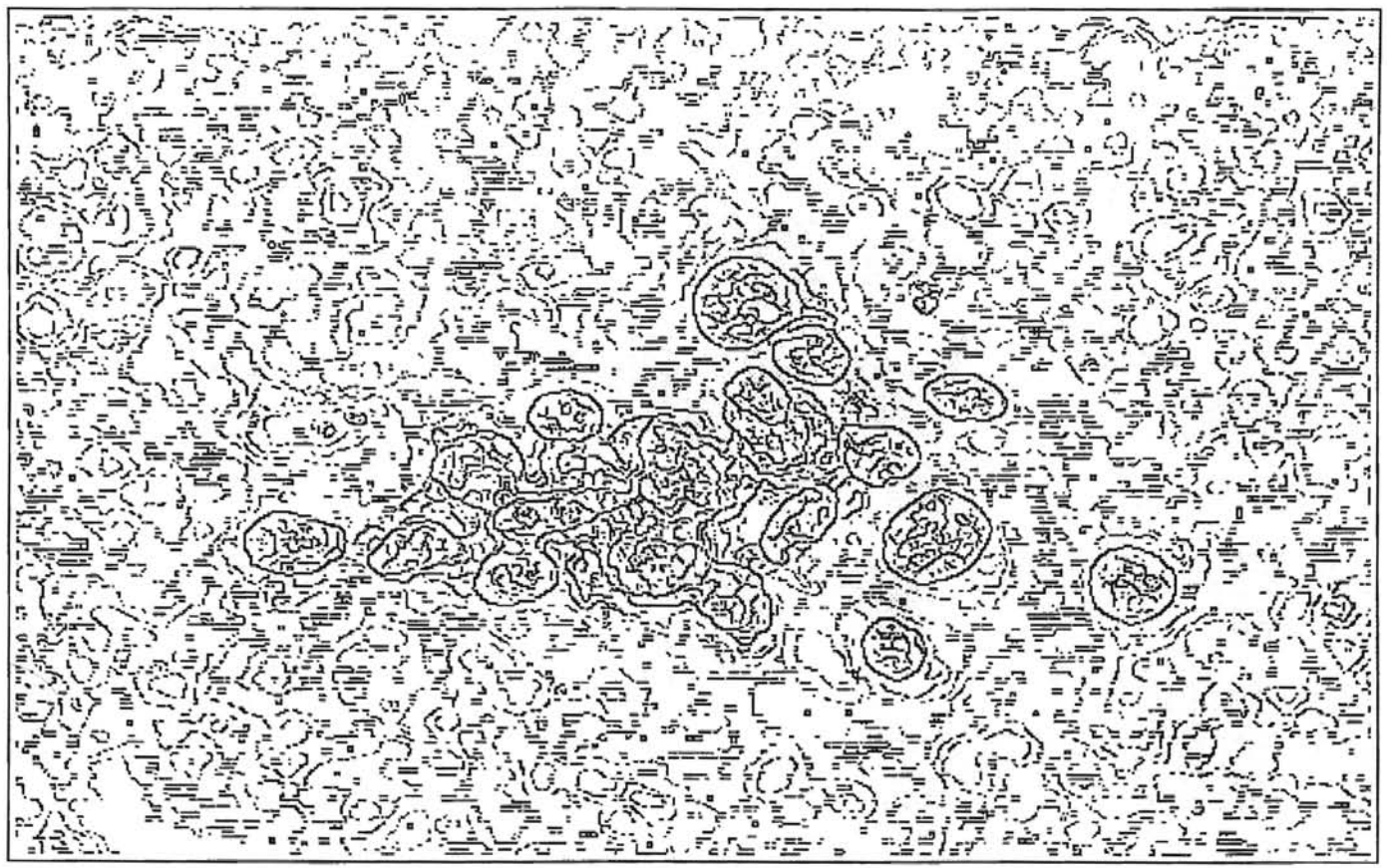


Figure 3-10 : Edge Image

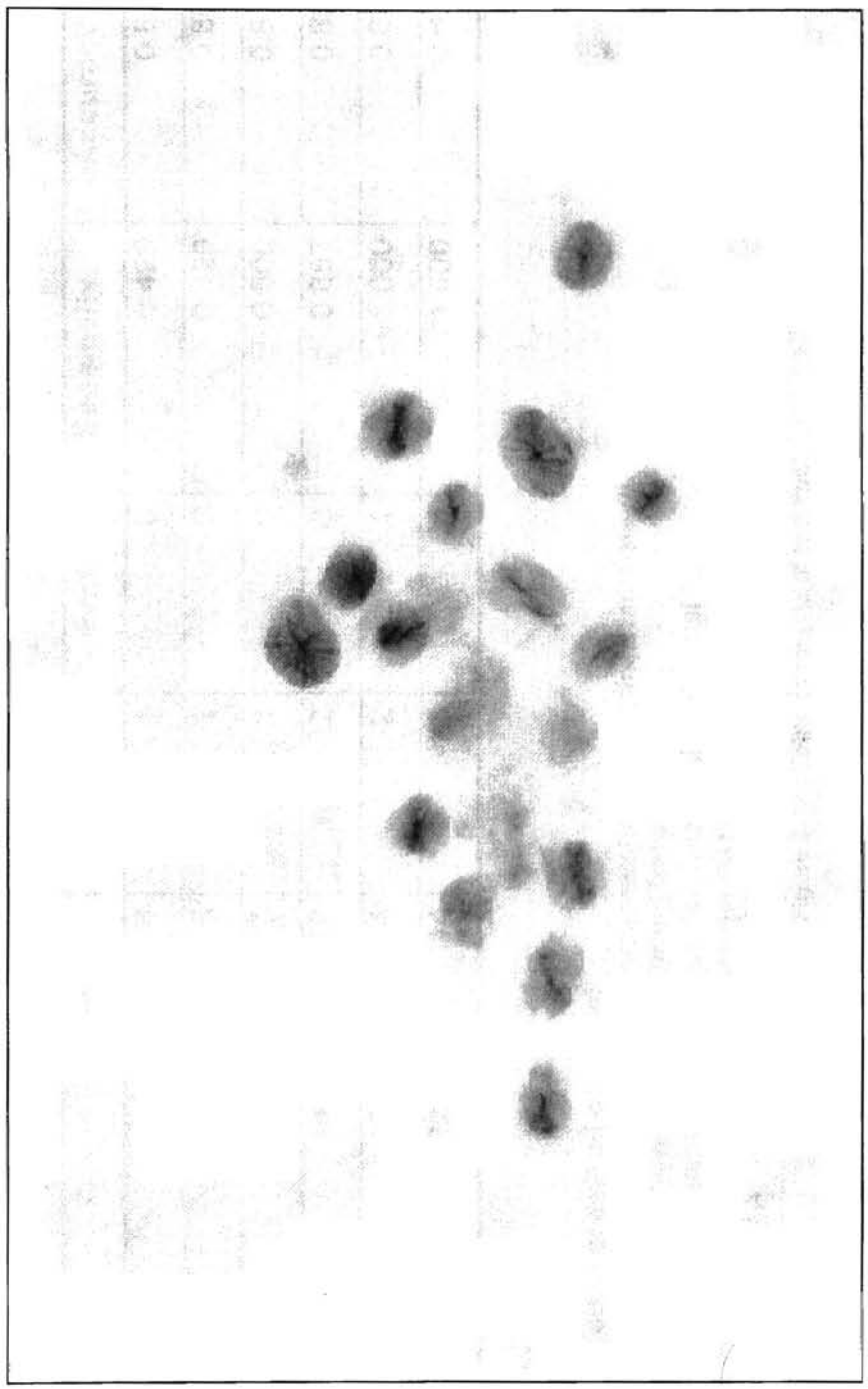


Figure 3-11 : Global Accumulator

	A	B	C	Error	Sensitivity	Accuracy
0.8	7	6	1	0	0.429	0.857
0.7	7	6	1	0	0.429	0.857
0.6	8	7	1	0	0.500	0.875
0.5	14	12	2	0	0.857	0.857
0.4	17	14	2	1	1.000	0.824
0.3	22	14	3	5	1.000	0.636

A = Total # of defined cell nuclei
B = # of cell nuclei comparatively well matched with defined template
C = # of cell nuclei matched with wrong template
Error = # of defined templates that dose not match with cell nuclei
Sensitivity = B / # of recognizable cell nuclei(14)
Accuracy = B / A

Figure 3-12 : Table of results of test two

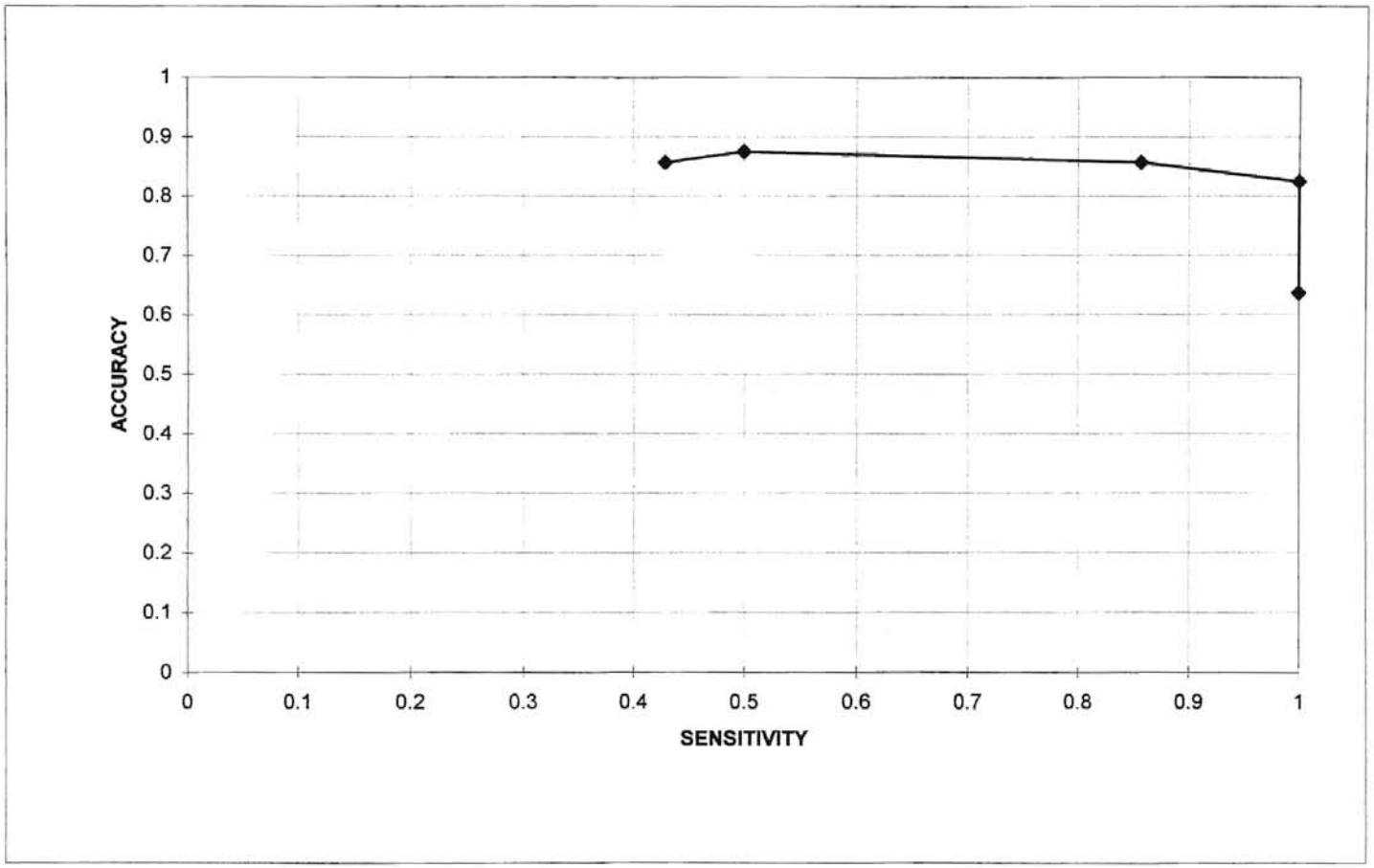


Figure 3-13 : Accuracy and Sensitivity

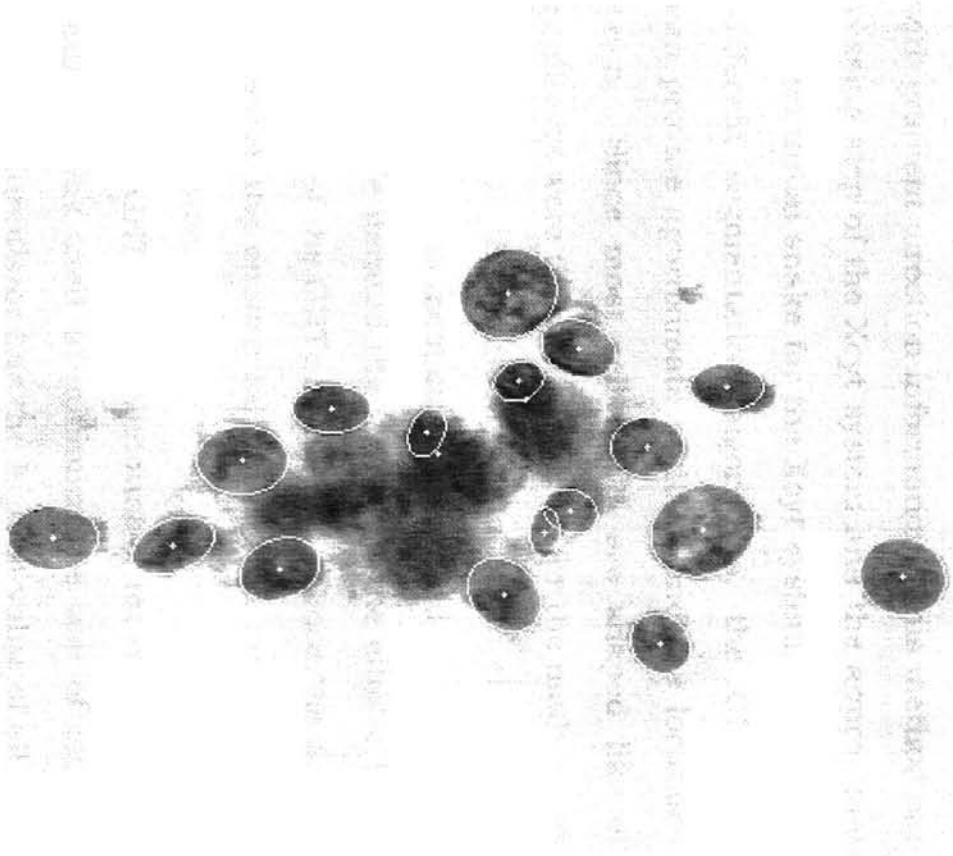


Figure 3-14 : Test Result (Thresholding Value = 0.5)

Chapter 4 : Conclusions and Future Work

4.1 Conclusions

In this thesis, a method to automatically provide an approximate outline and location of a cell nucleus in a medical image was designed by using GHT with gradient direction information. This method can be used as a preprocessing step of the Xcyt system and the approximate outline can be used as an initial snake for the Xcyt system. Even though there are many different segmentation algorithms, the GHT was chosen because it can provide approximate outlines and the locations with no user interaction. Since most cell nuclei are shaped like ellipses or circles, many ellipses were used as templates for the method. Even when there is no cell nucleus that perfectly matches an elliptic template, we can find the template that is most closely matched with each cell nucleus by using the GHT. We iterated the GHT with the elliptic templates and compared all outputs of the GHTs at the end. Since the more varied the templates, the better the chance to find a good match, we rotated the templates every 25° degree.

For the iterative GHT, a huge amount of memory is required. The global accumulator was used to minimize the use of memory. Every point of the global accumulator has the highest value of all accumulators on same location. This reduces the use of the memory from $\Theta(mnt)$ to $\Theta(mn)$ by using the global accumulator, when a image size is $n \times m$ and t templates are applied.

At the end of the iterative GHT, the peak finding step is performed to find which template is most closely matched with which cell nucleus. Because in general no cell nucleus perfectly matches any of the

templates, plateaus appear in the global accumulator. We applied the peak sharpening algorithm to sharpen the plateaus. Since the peak sharpening algorithm raised the center of the plateaus, the peak sharpening algorithm made it easier to choose high peak points, increasing the accuracy of the system.

Every high peak point was considered the location of a cell nucleus. The peak finding step repeatedly finds the current highest peak point that represents a location of a cell nucleus in the global accumulator and defines a template that most closely matches the cell nucleus. We performed the tests with two 640×400, 256-gray-scale images and all steps of the system worked successfully. We achieved over 80% sensitivity and accuracy with each test image. These results show at least 80% of user interaction of Xcyt system can be eliminated by using this method as a preprocessing step, making the system significantly more robust and user independent. Furthermore the operator of the Xcyt system can save a lot of time and effort that have been spent to provide initial snakes. We have therefore shown a method using the GHT for segmentation for cytological images to provide approximate outlines and locations of cell nuclei in this thesis.

4.2 Future Work

Although the system currently shows over 80% accuracy and sensitivity and demonstrates the possibility of using the GHT for segmentation of a medical image, there are some more things to do. Higher levels of accuracy and sensitivity may be achieved by employing some more advanced algorithms and concepts.

First, the accuracy of GHT depends on the templates. In this thesis, we rotated elliptic templates every 25°. However, if the major axis of a cell nucleus is between those degrees, the system does not achieve precise matching. Therefore if more rotations are employed, more precise matching may be achieved. However, applying more templates requires longer processing time. This trade-off could be solved by allowing user interaction to control the number of templates applied.

Second, we used unthresholded edge data to prevent losing valuable edges. However, this preserves not only valuable edge data but also the noisy edge data. For instance, because every cell nucleus has structures inside the cell nucleus, intensity values of the inside of the cell nucleus are not uniform. These differences of intensity values cause some unexpected edge lines. These edges generate some unnecessary increments in the accumulator. These increments cause longer processing time and some errors. Figure 4-1 shows the edges and the accumulator generated by an example nucleus. In Figure 4-1 (a), some unfocused area is present in the cell nucleus. In the part (b) many noisy edge segments are present in the inside of cell nucleus. As this example shows, the number of useless edge pixels is much more than valuable edge pixels. Thus, a significant amount of the processing time is spent for the unnecessary increments.

(a) (b)

Figure 4-1 : Unexpected Edge pixels

Therefore, if the useless edges could be eliminated without losing valuable edge pixels, the processing time would be significantly reduced and better accuracy may be achieved.

Third, even though we used the compactness factor and the peak sharpening step to minimize the number of overlapped templates, the problem still exists. Although most cell nuclei are wide, obviously many narrow cell nuclei exist in the images. Using the compactness factor may in some cases cause the loss of good matches for the narrow cell nuclei. Another way to minimize the problem is to investigate how many different directions of edges are used to increment the accumulator value during the GHT. Even if only a small part of the boundary of a cell nucleus has been used to increment an accumulator value, we might still prefer this match if those edge pixels came from different parts of the nucleus. We can determine this by checking how many different directions of edge pixels have been found. Solving the overlapped template problem may give meaningful improvement of the accuracy.

Bibliography

1. D. H. Ballard, *Generalizing the Hough Transform to Detect Arbitrary Shapes*, Pattern Recognition 13[2]: 111-122, 1981.

2. D. H. Ballard and Christopher M. Brown, *Computer Vision*, Rochester, New York: Prentice-Hall, 1982.

3. J. Canny, *A Computational Approach to Edge Detection*, IEEE Trans. on Pattern Analysis and Machine Intelligence 8[6]: 679-698, November 1986.

4. R. O. Duda and P. E. Hart, *Use of the Hough Transform To Detect Lines and Curves in Pictures*, Communications of the ACM 15(1): 11-15, 1972.

5. R. C. Gonzalez and P. Wintz., *Digital Image Processing*, Knoxville, Tennessee: Addison-Wesley, 1987.

6. P. C. Hough, *Method and Means for Recognising Complex Patterns*, U.S. Patent 3,069,654, Dec.18, 1962.

7. T. S. Huang, G. T. Yang and G. Y. Tang, *A Fast Two-Dimensional Median Filtering Algorithm*, IEEE Trans. Acoust., Speech, Sig. Proc. 27:13-18,1979.

8. A. K. Jain, S. P. Smith and E. Backer., *Segmentation of Muscle Cell Pictures: A Preliminary Study*, IEEE Trans. On Pattern Anal. and Machine Intelligence PAMI-2(3): 232-242, May 1980.

9. M. Kass, A. Witkin and D. Terzopoulos, *Snakes: Active Contour Models*, International Journal of Computer Vision: 321-331, 1988.

10. C. Kimme, D. Ballard, and J. Skalnsky, *Finding Circles by an Array of Accumulators*, IEEE Trans. on Computers 15(4): 120-122, April 1972.

11. K. F. Lai and R. T. Chin, *On Regularisation, Formulation and Initialisation of the Active Contour Models(Snakes)*, Madison: University of Wisconsin-Madison,Dec 23, 1992.

12. O. L. Mangasarian, W. N. Street and W. H. Wolberg, *Breast Cancer Diagnosis and Prognosis via Linear Programming*, Operations Research 43(4): 570-576, July-August 1995.

13. J. K. Mui, K. S. Fu and J. W. Bacus., *Automated Classification of Blood Cell Neutrophils*, The Journal of Histochemistry and Cytochemistry 25(7): 633-640, 1977.

14. P. Mussio, M. Pietrogrande, P. Bottoni, M. Dell'Oca, E. Arosio, E. Sartirana, M. R. Finazon and N. Dioguardi., *Automatic Cell Count in Digital Images of Liver Tissue Sections*, Proceeding of the Fourth Annual IEEE Symposium on Computer Based Medical Systems: 153-160, 1991.

VITA

Hyuk-Joon Oh

Candidate for the Degree of

Master of Science

Thesis : GENERALIZED HOUGH TRANSFORM FOR CYTOLOGICAL IMAGE SEGMENTATION

Major Field : Computer Science

Biographical :

Personal Data : Born in Seoul, Korea, February 25, 1967, the son of Byoung-Keun and Il-Sun Maeng Oh.

Education : Graduated from Daewon High School, Seoul, Korea in February, 1985; received Bachelor of Science in Mathematics from Kyunghee University, Seoul, Korea in February, 1990; received Master of Engineering in Computer Engineering from Kyunghee University in Seoul, Korea in February, 1994; completed requirements for the Master of Science in Computer Science at Oklahoma State University in December, 1997.

Professional Experience : Lecturer , Dept. of Computer Engineering, Seoil Junior College, Seoul, Korea, March, 1994 to August, 1995; Teaching Assistant, Dept. of Computer Science, Oklahoma State University, August, 1997 to December, 1997.