ASSESSMENT OF CORNEAL ULCER SIZE PROGRESSION USING DIGITAL IMAGE PROCESSING.

By

NURUL HANANI BINTI HUSSEIN

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

ASSESSMENT OF CORNEAL ULCER SIZE PROGRESSION USING DIGITAL IMAGE PROCESSING.

by

Nurul Hanani Binti Hussein

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Mrs Zazilah May
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2005

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Nurul Hanani Binti Hussein

ABSTRACT

This project aims to perform assessment of corneal ulcer progression using digital image processing. Corneal ulcer is a type of eye disease that presently approaches 1.5 to 2 millions patients per year at developing world, estimated by World Health Organization. Current methods are mainly clinical assessment and bacteriological identification. Several problems are identified, which are parameters measured are inaccurate, difficulties in handling patients and only same ophthalmologist has to examine the progression of a patient. The study focuses on enhances and analyze eye images. Several steps of image enhancement have been performed by MATLAB to increase quality of the images. The methods used are grayscale conversion, intensity vale adjustment, image subtraction and thresholding. Analysis of corneal ulcer progression is established by using parameters measured; area, height and width. Finally, graphical user interface (GUI) is developed. Its aim is to ease ophthalmologist in monitoring corneal ulcer progression.

ACKNOWLEDGEMENTS

In The Name Of Allah, Most Merciful and Compassionate

The author would like to extent special acknowledge to Supervisor, Mrs Zazilah May for her helpful supervision towards completing this project. Huge appreciation goes to Prof P.A Venkatachalam for his professional guidance.

Big thanks to my collaboration partner, Dr Ezanee, as well as the Supervisor, Dr Elias. Exclusive appreciations go to prior examiners for their motivation comments to this project.

Special thanks to Ms. Siti Hawa, assistant coordinator for Final Year Project. Finally, my acknowledgement is extended to supportive husband, Mr Mohd Yunus Ismail and all may friends for their encouragement and help.

Thank you to all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Using the World Health Organization (WHO) definition of blindness as a visual acuity of 3/60 or less , it is estimated that currently there are 45 million individuals worldwide who are bilaterally blind and another 135 million that have severely impaired vision in both eyes [1]. Corneal disease, including corneal ulcer remains second only to cataract as a major cause of blindness of world today.

Current assessments of corneal ulcer, clinical and bacteriological identification perform complicated procedure in monitoring corneal ulcer progression. In easing ophthalmologist to detect and monitor progression or deterioration of corneal ulcer for normal case, this study proposes a new method of using digital image processing.

1.2 Problem Statement

Current assessment of monitoring corneal ulcer progression is mainly clinical, by using apparatus slit lamp. A few difficulties faced, one of them is difficulties in handling patients. Patients will be asked to open their eyes when ophthalmologist takes measurement through slit lamp. Patient's eye normally cannot open longer because of pain. Thus, it results in inaccuracy measurement.

1.3 Objectives and Scope of Study

The objectives and scope of study of the project can be summarized as follows:-

a) To obtain eye images

   Eye images should be obtained from hospitals that show progression of corneal ulcer.

b) To enhance eye images and extract features of corneal ulcer.

   Eye images should be enhanced by using any application programs such as MATLAB, C and C++. The feature of corneal ulcer, which is denoted in circle of green, should be extract clearly and smoothly.

c) To obtain required parameters; area, height and width of corneal ulcer

   Area of green circle should be determined as well as the longest height and the longest width.

d) To do analysis on corneal ulcer

   Corneal ulcer progression should be analyzed based on the area, height and width obtained previously. Several levels of disease seriousness should be determined.

e) To create graphical user interface

   Graphical user interface should be built in such a way to be presentable and user friendly. Its aim is to ease ophthalmologist in monitoring corneal ulcer progression.

CHAPTER 2
LITERATURE REVIEW

2.1 Corneal

Corneal is a transparency outer layer of eyes. Its sizes is 11.7 mm horizontally and 10.6 mm vertically.[2].

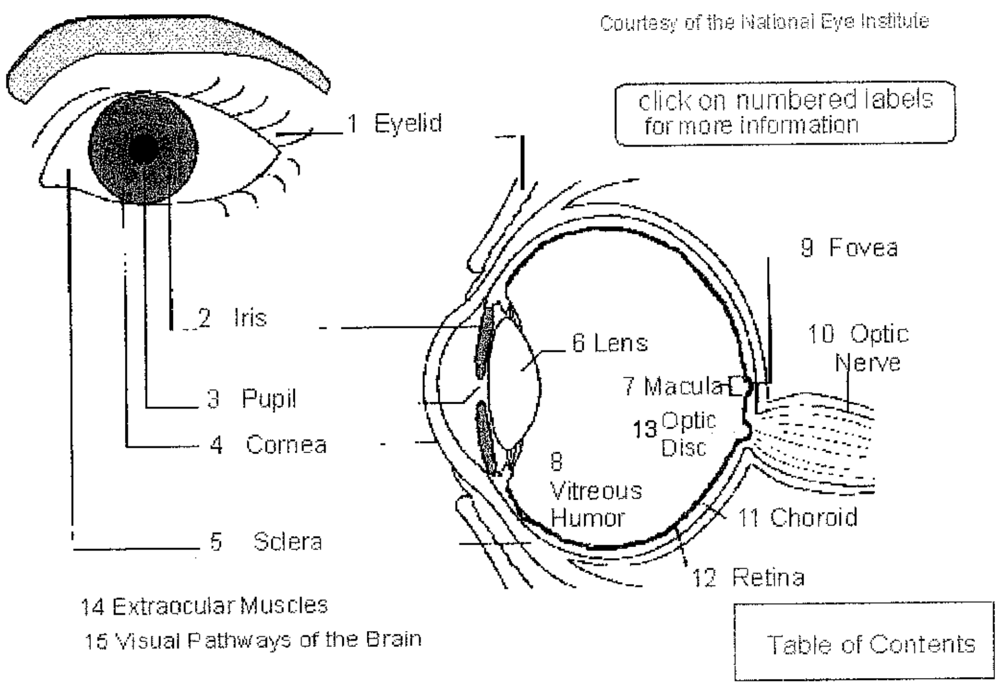


Figure 1 : Eye Anatomy

When looking to human eyes, several structures easily observed:-

a) **Pupil** - A black-looking aperture that allows light to enter the eye .It appears dark because of the absorbing pigments in the retina.

b) **Iris** - A colored circular muscle which is beautifully pigmented giving us our eye's color (the central aperture of the iris is the pupil). This circular muscle controls the size of the pupil so that more or less light, depending on conditions, is allowed to enter the eye. Eye color, or more correctly, iris color is due to variable amounts of eumelanin (brown/black melanins) and pheomelanin (red/yellow melanins) produced by melanocytes. More of the former is in brown eyed people and of the latter in blue and green-eyed people.

c) **Corneal** - A transparent external surface that covers both the pupil and the iris. This is the first and most powerful lens of the optical system of the eye and allows, together with **the crystalline lens** the production of a sharp image at the retinal photoreceptor level.

d) **Sclera** - The "white of the eye" which forms part of the supporting wall of the eyeball. The sclera is continuous with the cornea. Furthermore this external covering of the eye is in continuity with the dura of the central nervous system.

When the eye is removed from the orbit, it can be observed that the eye is a slightly asymmetrical sphere with an approximate sagittal diameter or length of 24 to 25 mm. and a transverse diameter of 24 mm. It has a volume of about 6.5 cc.

2.2 Corneal Ulcer

A corneal ulcer is an open sore on the cornea, the thin clear structure overlying the iris, which is the colored part of the eye. The symptoms of corneal ulcer is red eye, severe pain, feeling that something in the eye, tears, thick discharge draining from the eye, blurry vision, pain when looking at bright lights, swollen eyelids and a white round spot on the corneal that is visible with the naked eye if the ulcer is very large.

Ophthalmologist will be able to detect ulcer by using a special eye microscope, known as a slit lamp. To make the ulcer easier to see, a drop containing the dye fluorescein is put into your eye. Blinking spreads the dye around and coats the "tear film" covering the surface of the cornea. A blue light is then directed at the eye. Any abnormalities in the surface of the cornea will be stained by the dye and appear green under the blue light. The ophthalmologist can determine the location and probable cause of the cornea problem depending on the size, location, and shape of the staining. If the ophthalmologist thinks that an infection is responsible for the ulcer, he or she may then get samples of the ulcer to send to the laboratory for identification.

A corneal ulcer is a true emergency. Without treatment, the ulcer can spread to the rest of the eyeball, and can become partially or completely blind in a very short period of time. The cornea may also perforate, or could develop scarring, cataracts, or glaucoma. With the proper treatment, corneal ulcers should improve within 2-3 weeks. If scars from previous corneal ulcers impair vision, a corneal transplant may be needed to restore normal vision.

2.3 Digital Image Processing

By definition, Digital Image Processing refers to processing digital images by means of a digital computer [4]. For this project, the images processing are being performed with MATLAB version 6.1, with Image Processing Toolbox.

2.3.1 Grayscale Intensity Image Conversion

An intensity image is a data matrix, I, whose values represent intensities within some range. MATLAB stores an intensity image as a single matrix, with each element of the matrix corresponding to one image pixel. The matrix can be of class double, uint8, or uint16.While intensity images are rarely saved with a colormap, MATLAB uses a colormap to display them.

The elements in the intensity matrix represent various intensities, or gray levels, where the intensity 0 usually represents black and the intensity 1, 255, or 65535 usually represents full intensity, or white.

Example of grayscale conversion is as seen in figure below.

Figure 2 : Example of Gray Scale Conversion

2.3.2 Intensity Value Adjustment

Intensity adjustment is a technique for mapping an image's intensity values to a new range. A few methods exist, which are histogram equalization, contrast stretching, gamma correction, gray level slicing and bit-plane slicing. Low contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition. [4]. One of the intensity adjustment applications is by contrast stretching. The idea behind contrast stretching is that to increase the dynamic range of the gray levels in the image being processed.

In illustrating the concept, considers an image with location of point (r1.s1) and (r2.s2). These points control the shape of the transformation function. If r1 = s1 and r2 = s2, the transformation is a linear function that produces no changes in gray levels.

If r1 = r2, s1 = 0, and s2 = L – 1, the transformation becomes a thresholding function that creates a binary images. Intermediate value of (r1.s1) and (r2.s2) produce various degrees of spread in the grey levels of the output image, thus affecting its contrast.

In general, r1<r2, s1<s2, is assumed so that the function is single valued and monotonically increasing.

Figure 3 : Example of Intensity Adjustment

2.3.3 Enhancement using Image Subtraction

Image subtraction is the enhancement of differences between images. Processing is based on two images, a marker and a mask, rather than one image and a structuring element.

The difference between two images f (x,y) and h (x,y), expressed as,

$$g(x,y) = f(x,y) - h(x,y)$$

is obtained by computing the difference between all pairs of corresponding pixels from f and h.

In illustrating the concept, an image is taken for example, where higher-order bit planes of the image carry a significant amount of visually relevant detail, while the lower planes contributes more to fine (often imperceptible) detail. When discarding the four least significant bit planes of the original image, the resultant image is identical with original image, except for very slight drop in overall contrast due to less variability of the graylevel values in the image.

With the image subtraction application, marker and mask concept is introduced. Morphological reconstruction processes one image, called the marker, based on the characteristics of another image, called the mask. The high-points, or peaks, in the marker image specify where processing begins. The processing continues until the image values stop changing.

Morphological reconstruction can be thought of conceptually as repeated dilations of the marker image until the contour of the marker image fits under the mask image. In this way, the peaks in the marker image "spread out", or dilate. This figure illustrates this processing in 1-D. Each successive dilation is constrained to lie underneath the mask. When further dilation ceases to change the image, processing stops. The final dilation is the reconstructed image.

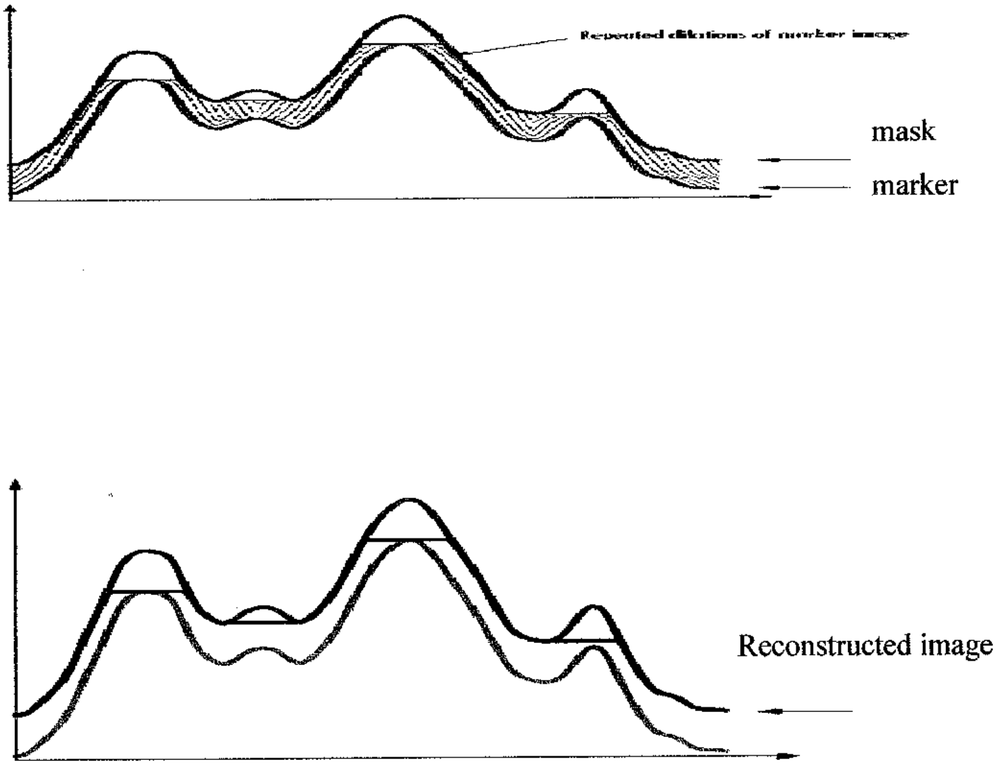


Figure 4 : Image Reconstruction Principle.

2.3.4 Thresholding

Thresholding is a method of image segmentation. Because of its intuitive properties and simplicity of implementation, image thresholding enjoys a central position in applications of image segmentation. [4]

Thresholding in simple words means extracts objects from background. For an image, f (x,y) composed of light objects on a dark background, a threshold T can separates these two dominant modes. Then, any point (x,y) for which f(x,y) > T is called an object point; otherwise the point is called a background point.

The most common form of image thresholding makes use of pixel grey level. Grey level thresholding applies to each pixel, the rules are,

$$g(x,y) = 0 \qquad f(x,y) < T$$
$$g(x,y) = 1 \qquad f(x,y)) > T$$

where T is the threshold value. [ ]

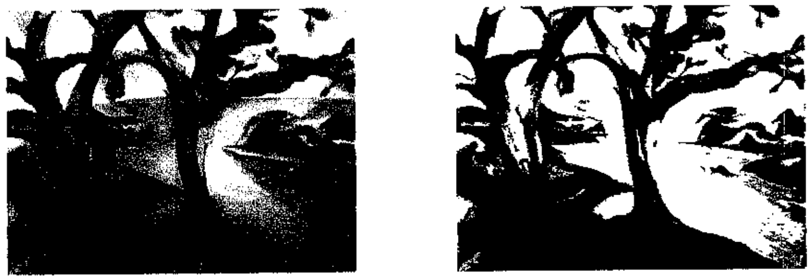


Figure 5 : Example of Image Thresholding

CHAPTER 3
METHODOLOGY

Five stages involve in this project as seen in figure below. The processing is done by using MATLAB version 6.1.

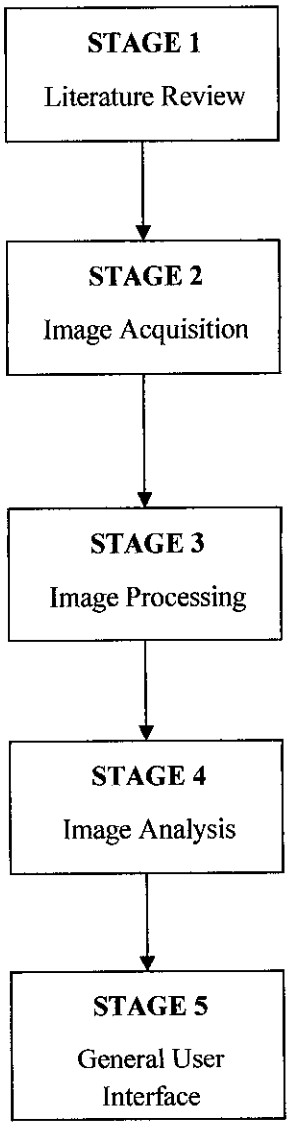


Figure 6 : Methodology Stages

3.1 Methodology Process

3.1.1 Stage 1: Literature Review

Literature review done concerning on the corneal ulcer, digital image processing and software used in this project. The source for literature reviews varies from books, knowledgeable people, published paper and web sites

3.1.2 Stage 2: Image Acquisition

This project is done in collaboration with Dr Ezanee Mokhtar (Universiti Sains Malaysia) in fulfilling his thesis master, with topic, *A comparative study on the Agreement of using digital imaging with analysis software for assessment of corneal ulcer size progression.* Mainly his project approach is on comparing existed method of monitoring corneal ulcer progression with digital imaging method. In fulfilling the thesis, a lot of images that show corneal ulcer progression have been obtained.

The sources of eye images for the project are obtained as mentions above. The images obtained are already digitized, thus digitization and quantization stage can be skipped.

3.1.3 Stage 3: Image Processing

This stage is parted into four parts:-

i. Grayscale intensity image conversion
ii. Intensity value adjustment
iii. Enhancement using image subtraction
iv. Thresholding

3.1.3.1 Grayscale Intensity Image Conversion

In converting RGB or colormap image to grayscale, function used is RGB2GRAY. It converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance. If the input is an RGB image, it can be of class uint8, uint16 or double; the output image I is of the same class as the input image. If the input is a colormap, the input and output colormaps are both of class double. The general syntax used is

Image2 = RGB2GRAY (Image);

3.1.3.2 Intensity Value Adjustment

In adjusting intensity value automatically, function used is IMADJUST and STRETCHLIM. The general syntax used is

Image2 = IMADJUST (Image , STRETCHLIM (Image), [ ]);

IMADJUST adjust image intensity values or colormap while STRETCHLIM find limits to contrast stretch an image automatically.

LOW_HIGH = STRETCHLIM(I,TOL) returns a pair of intensities that can be used by IMADJUST to increase the contrast of an image.

TOL = [LOW_FRACT HIGH_FRACT] specifies the fraction of the image to saturate at low and high intensities.

3.1.3.3 Enhancement using Image Subtraction

Marker and mask concept is an extended application of enhancement using image subtraction. The function used in imsubtract and imreconstruct. The general syntax is,

IM = imreconstruct (MARKER, MASK);
MARKER = imsubtract (MASK, X);

In illustrating the working principle of marker and mark, an example is shown below. The image contains two primary regions, the blocks of pixels containing the value 14 and 18. The background is primarily all set to 10, with some pixels set to 11

$$A = \begin{bmatrix}
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 14 & 14 & 14 & 10 & 10 & 11 & 10 & 11 & 10 \\
10 & 14 & 14 & 14 & 10 & 10 & 10 & 11 & 10 & 10 \\
10 & 14 & 14 & 14 & 10 & 10 & 11 & 10 & 11 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 11 & 10 & 10 & 10 & 18 & 18 & 18 & 10 & 10 \\
10 & 10 & 10 & 11 & 10 & 18 & 18 & 18 & 10 & 10 \\
10 & 10 & 11 & 10 & 10 & 18 & 18 & 18 & 10 & 10 \\
10 & 11 & 10 & 11 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 11 & 10 & 10 & 10
\end{bmatrix}$$

A marker image is created. The peaks in the marker image should identify the location of objects in the mask image that want to be emphasized. One way to create a marker image is to subtract a constant (choose 2) from the mask image, using imsubtract.

$$marker = \begin{bmatrix}
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
8 & 12 & 12 & 12 & 8 & 8 & 9 & 8 & 9 & 8 \\
8 & 12 & 12 & 12 & 8 & 8 & 8 & 9 & 8 & 8 \\
8 & 12 & 12 & 12 & 8 & 8 & 9 & 8 & 9 & 8 \\
8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\
8 & 9 & 8 & 8 & 8 & 16 & 16 & 16 & 8 & 8 \\
8 & 8 & 8 & 9 & 8 & 16 & 16 & 16 & 8 & 8 \\
8 & 8 & 9 & 8 & 8 & 16 & 16 & 16 & 8 & 8 \\
8 & 9 & 8 & 9 & 8 & 8 & 8 & 8 & 8 & 8 \\
8 & 8 & 8 & 8 & 8 & 8 & 9 & 8 & 8 & 8
\end{bmatrix}$$

Then, morphological recontruction is applied, results in,

$$Recon = \begin{bmatrix}
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 12 & 12 & 12 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 12 & 12 & 12 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 12 & 12 & 12 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 16 & 16 & 16 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 16 & 16 & 16 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 16 & 16 & 16 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10
\end{bmatrix}$$

3.1.3.4 *Thresholding*

Image segmentation is done by using thresholding method. The function used is im2bw. Its purpose is to convert image to binary image by thresholding. This function produces binary images from indexed, intensity, or RGB images. To do this, it converts the input image to grayscale format and then converts this grayscale image to binary by thresholding. The output binary image BW has values of 0 (black) for all pixels in the input image with luminance less than LEVEL and 1 (white) for all other pixels. The general syntax is,

BW = im2bw (Image2, LEVEL);

LEVEL can be determined either automatically or manually. function *graythresh* determines LEVEL automatically. It computes a global threshold (LEVEL) that can be used to convert an intensity image to a binary image with IM2BW. LEVEL is a normalized intensity value that lies in the range [0, 1]. Function *graythresh* uses Otsu's method, which chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels. The general syntax is,

BW = im2bw (Image2, GRAYTHRESH);

LEVEL is determined manually by using function *impixel.* Normal button clicks is used to select pixels. After finishes selecting pixels, *impixel* returns an M-by-3 matrix of RGB values in the supplied output argument. The general syntax is,

P = impixel (Image);

3.1.4 *Stage 4: Image Analysis*

This stage is mainly aimed to analyze image that leads to monitoring the corneal ulcer size progressing. It is expected to calculate vertical, horizontal and area of the corneal ulcer. Refer to figure below.

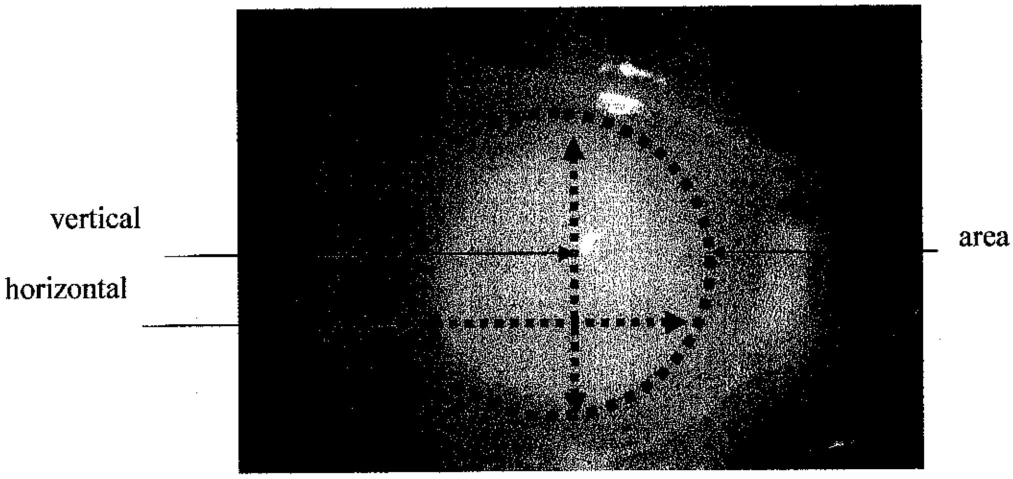


Figure 7 : Parameters of Monitoring Corneal Ulcer Progression

3.1.4.1 *Calculating vertical and horizontal*

The purpose of this stage is to determine the longest vertical and horizontal of the corneal ulcer. Calculations are being done by using software SCION, one of measured software.

3.1.4.2 *Calculating area*

The purpose of this stage is to estimate the percentage of the corneal ulcer area. Function used is **bwarea**. It computes the area of objects in binary image.

The general syntax is

TOTAL = BWAREA(BW)

where it estimates the area of the objects in binary image BW. TOTAL is a scalar whose value corresponds roughly to the total number of "on" pixels in the image, but may not be exactly the same because different patterns of pixels are weighted differently. For the class support, BW can be of class uint8 or double. TOTAL is of class double.

There are six different patterns, each representing a different area:-

i. Patterns with zero on pixels (area = 0)
ii. Patterns with one on pixels (area = 1/4)
iii. Patterns with two adjacent on pixels (area = 1/2)
iv. Patterns with two diagonal on pixels (area = 3/4)
v. Patterns with three on pixels (area = 7/8)
vi. Patterns with all four on pixels (area = 1)

3.2 Tools

Tools used for this project are:-

i. MATLAB software
ii. SCION software
iii. Adobe Photoshop software
iv. ACDSee software
v. Eye images
vi. Workstation
vii. Laser printer

CHAPTER 4
DISCUSSION AND FINDINGS

The project's result is divided into four sections:-

i. Image Acquisition
ii. Image Enhancement
iii. Image Analysis
iv. Graphical User Interface

4.1 Image Acquisition

11 samples of eye images are obtained from hospital. The samples are listed as below.

Patient	Image
Patient 1	Image 1
	Image 2
	Image 3
Patient 2	Image 1
	Image 2
	Image 3
	Image 4
Patient 3	Image 1
	Image 2
	Image 3
	Image 4

Table 1 : List of samples

The images obtained shows progression or deterioration of corneal ulcer in a patient. Each images for a patient is taken after two days continuously with given medicine. The detail of patient is confidential.

4.2 Image Enhancement

This stage is divided into four different sections, which are:-

i. Grayscale intensity image conversion
ii. Intensity value adjustment
iii. Enhancement using image subtraction
iv. Thresholding

4.2.1 Grayscale Intensity Image Conversion

Grayscale conversion is a conversion from RGB image to grayscale intensity mage. The images are converted to grayscale in order to eliminate the hue and saturation while maintaining luminance. The resultant images are shown in Figure 8, 9 and 10.

Referring to the figures below, it is observed that all hue and saturation is fully eliminated. In other words, the previous colored image changes to grey in color image. However, the intensity pixel is still constructed same as previous colored image. It is observed that some different grey level existed in the resultant image. The corneal ulcer feature still can be noticed clearly, even though it is blurring.

Patient 1	
Image 1	

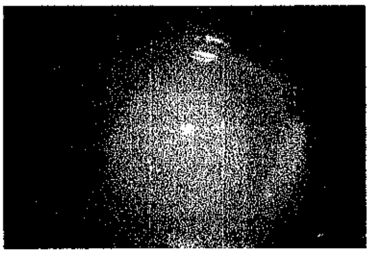
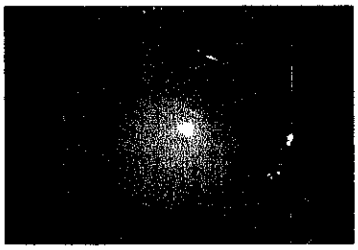

	

Image 2	

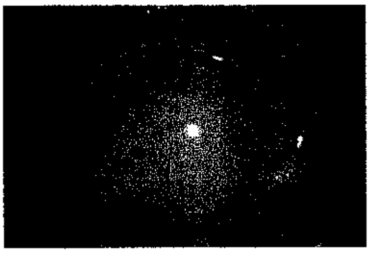
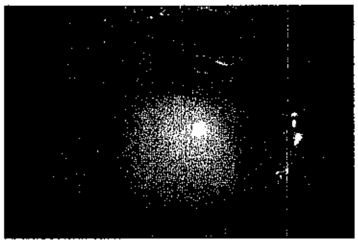

	

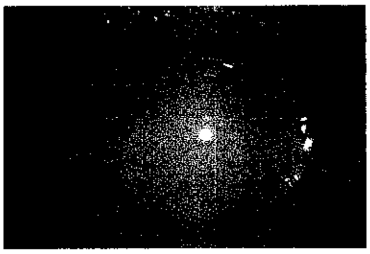
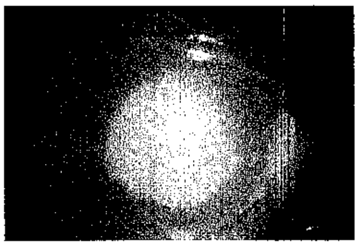
Image 3	
	

Figure 8 : Grayscale Conversion (Patient 1)

Patient 2	
Image 1	

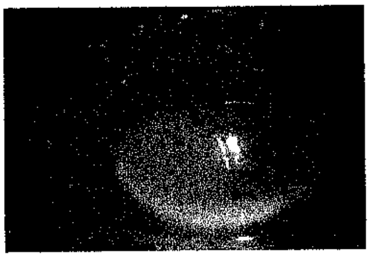
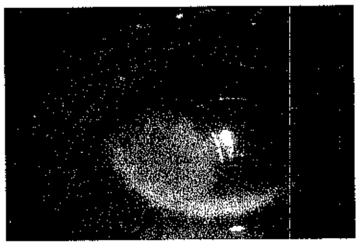

	

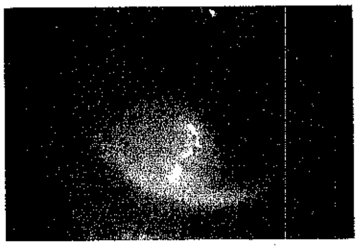
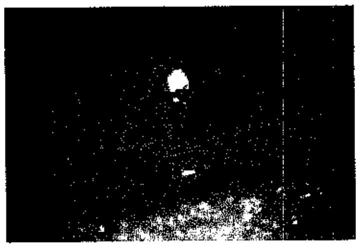
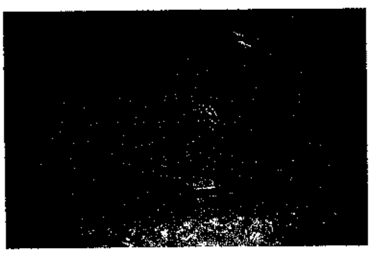
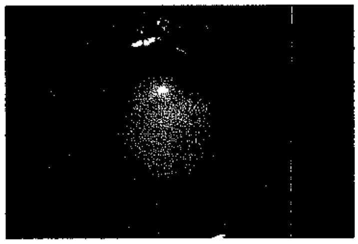
Image 2	
	
Image 3	
	
Image 4	
	

Figure 9 : Grayscale Conversion (Patient 2)

Patient 3	
Image 1	
	
Image 2	

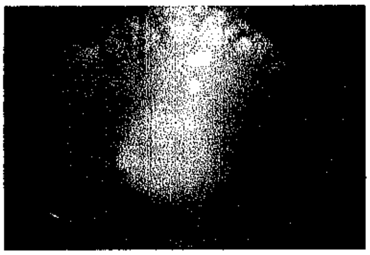
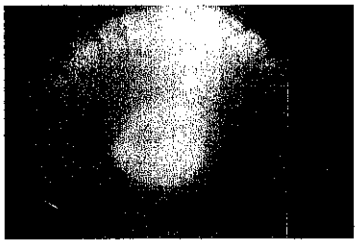

	

Image 3	

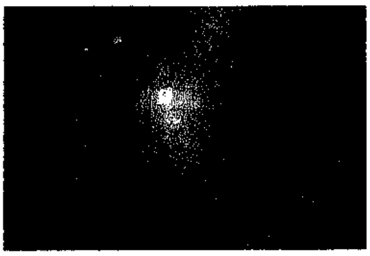
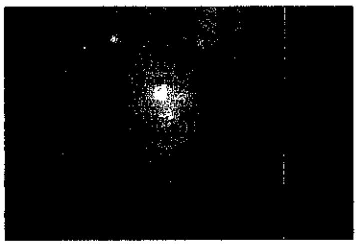

	

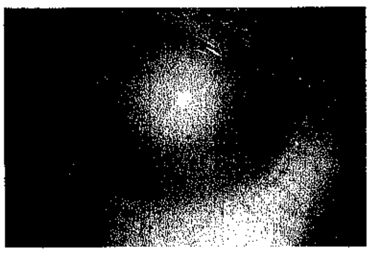
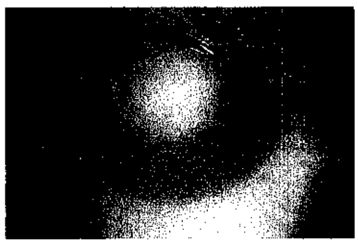
Image 4	
	

Figure 10 : Grayscale Conversion (Patient 3)

4.2.2 *Intensity Value Adjustment*

Contrast stretching is a method of intensity value adjustment. With the application of stretchlim function, the adjustment limit is being set up automatically. The resultant images and their histograms are shown in Figure 11, 12 and 13.

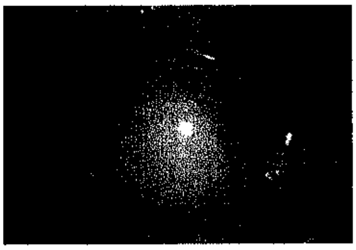
Referring to the figures below, it is observed that contrast stretching technique produces resultant images that lighten the corneal ulcer feature. The shape of corneal ulcer retains the same shape as the input images. Previously the corneal ulcer shape is too dark , as seen is the input images. After applying contrast stretching techniques, it brightens the image considerably, and also widens the dynamic range of the dark portions of the original image, making it much easier to see the corneal ulcer shape.

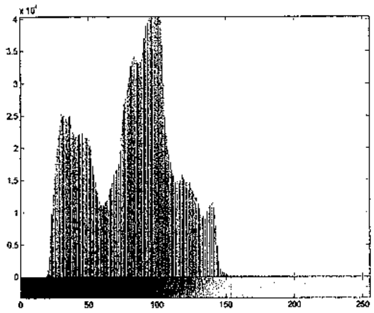
The input histograms does not cover the potential range of [0, 255], and are missing the high and low values that would result in good contrast. Comparing to the resultant images, the pixel values now extend across the full range of possible values. It is stretched out accordingly to fill in the dynamic range efficiently.

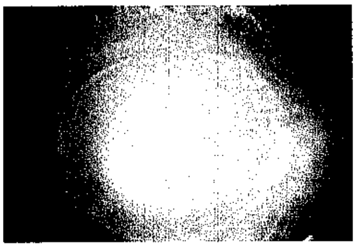
Contrast stretching technique can be applied to differentiate the corneal ulcer shape with the backgrounds, that easing much in later stage, segmentation.

Patient 1

Image 1







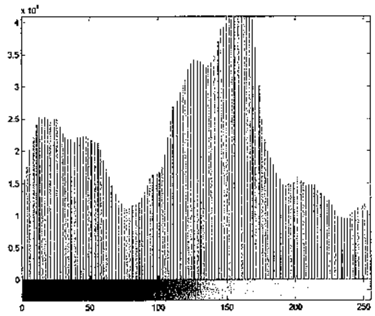
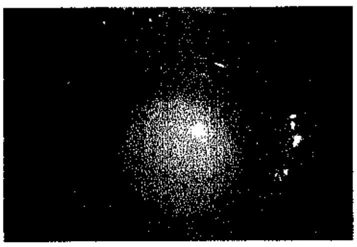
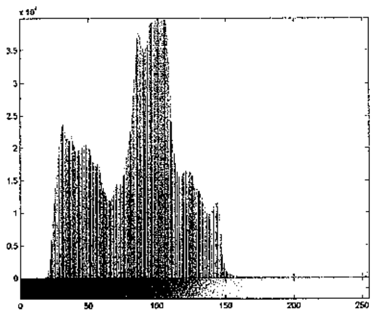
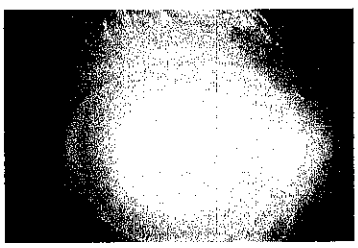


Image 2







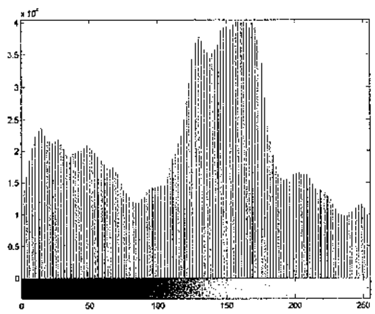


Image 3

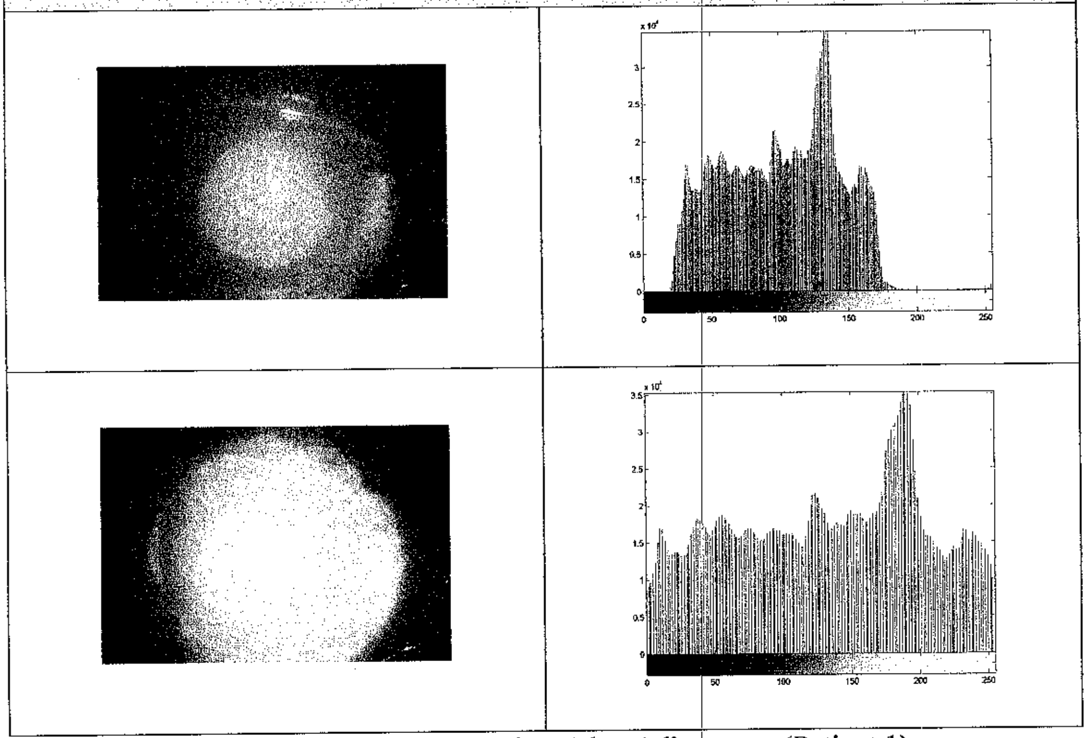
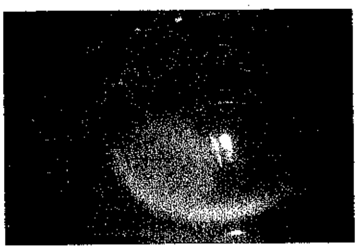
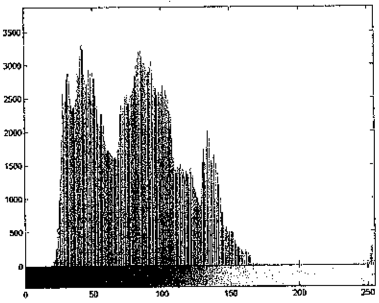
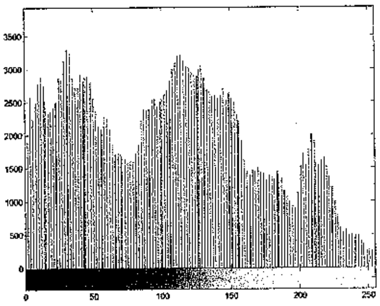
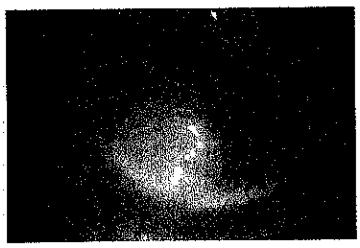
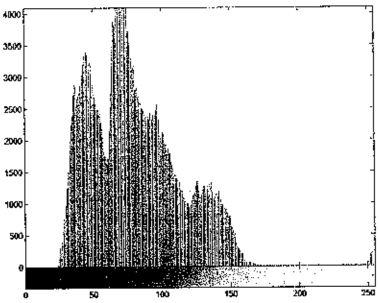
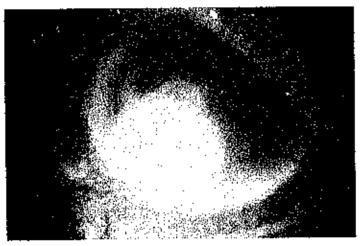
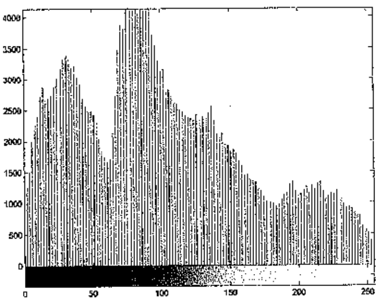


Figure 11 : Intensity Value Adjustment (Patient 1)

Patient 2	
Image 1	
	
	
Image 2	
	
	

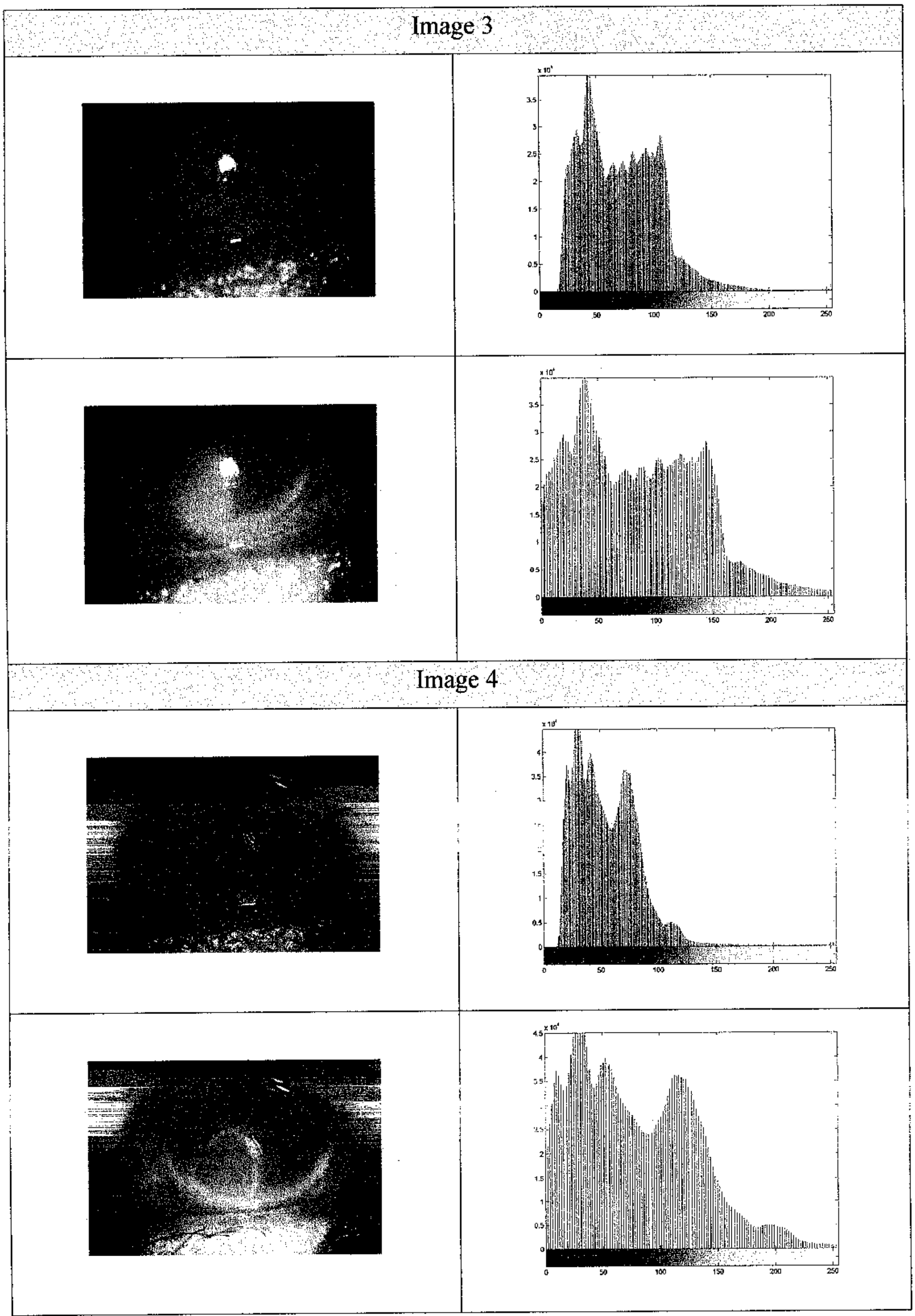
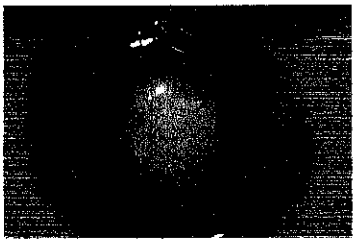
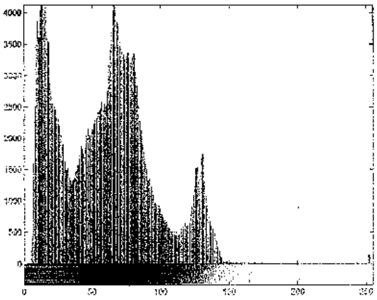


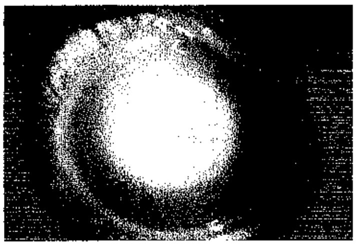
Figure 12 : Intensity Value Adjustment (Patient 2)

Patient 3

Image 1







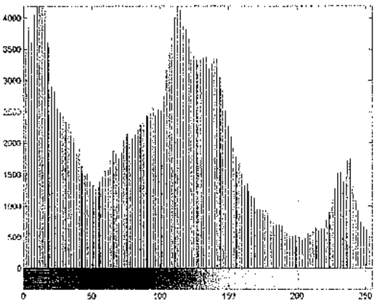
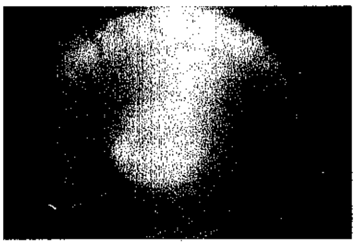
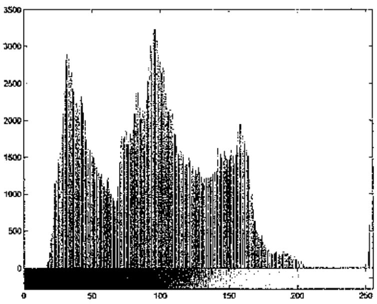
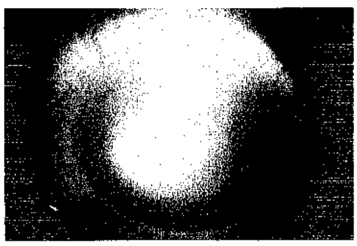
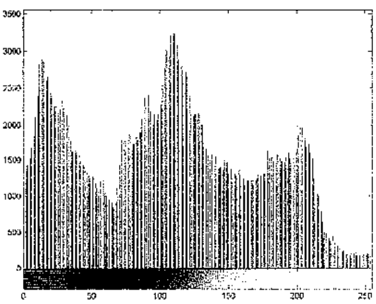


Image 2









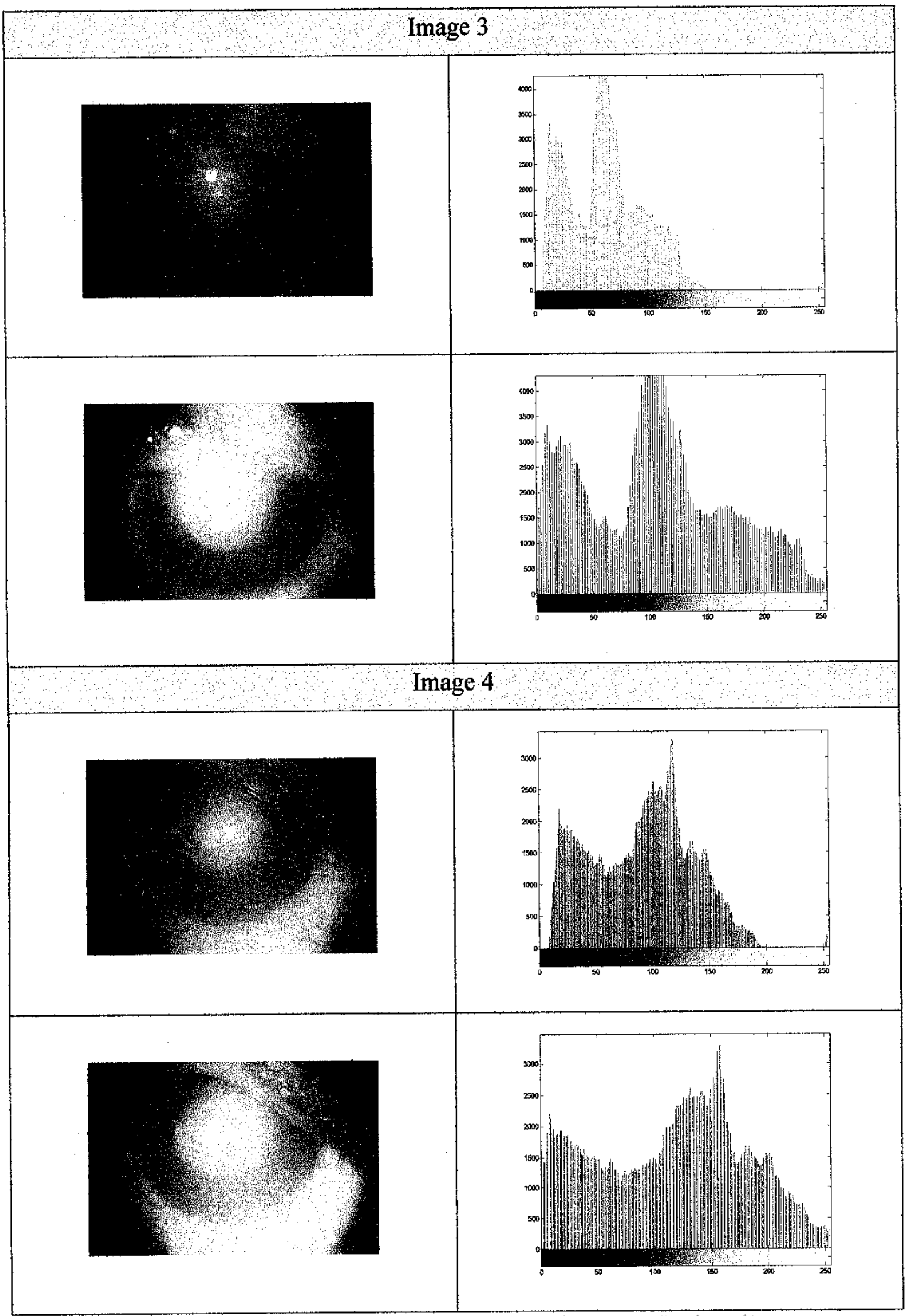


Figure 13 : Intensity Value Adjustment (Patient 3)

4.2.3 Enhancement using Image Subtraction

Image subtraction is a method of enhancement using image logic. Marker and mask method as an extended application of image subtraction is applied in this part. The resultant images are shown in Figure 14, 15 and 16.

Referring to the figures below, it is observed that marker and mask method produced clearly two dominant modes, background and objects. The background pixel is slightly darker from previous figure and has same value. It can be proved by same color at background. While for the objects, the corneal ulcer shape has darker image compared to the background.

By applying this step, it will ease in segmentation step later. This method eliminates different pixel values in the background that influence the resultant image.

4.2.4 Thresholding

Thresholding is a method of image segmentation. It differentiates object from background. The resultant images are shown in Figure 14, 15 and 16.

Referring to the figures below, it is observed that the corneal ulcer shape is segmented from its background. The object or the corneal ulcer shape is set as pixel 0 while the background as pixel 1. The images are threshold by setting the threshold level manually. The higher pixel value of the objects is taken and later being set as threshold value.

The resultant image leads to unwanted results. It is observed that the corneal ulcer shapes as well as the noise are being thresholded. This results in inefficiency in achieving this project's purpose, which is monitoring the corneal ulcer progression. This noise is misunderstood by the function as the objects too, since it has same pixel values with the corneal ulcer shape.

The noise sources from leakage of fluorescent die into eye tears. In other words, the green in color that supposes mark only the corneal ulcer shape has spread out to another part of eyes because of eye tears. That is how the corneal ulcer shape or object cannot be differentiated from the noise.

Instead of that, the noise also comes from the camera flash. This noise always exists in digital image, especially when using camera with higher resolution.

Patient 1	
Image 1	

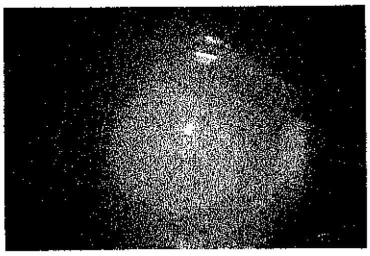
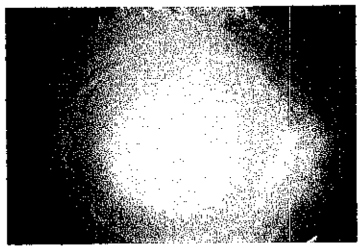
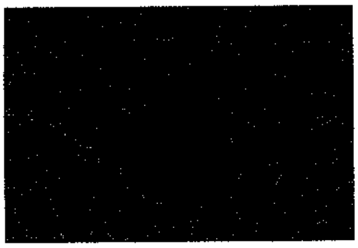

	


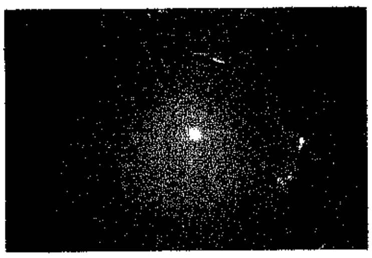
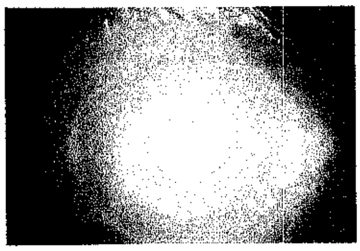
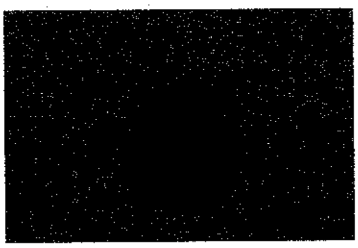
Image 2	
	
	

Image 3

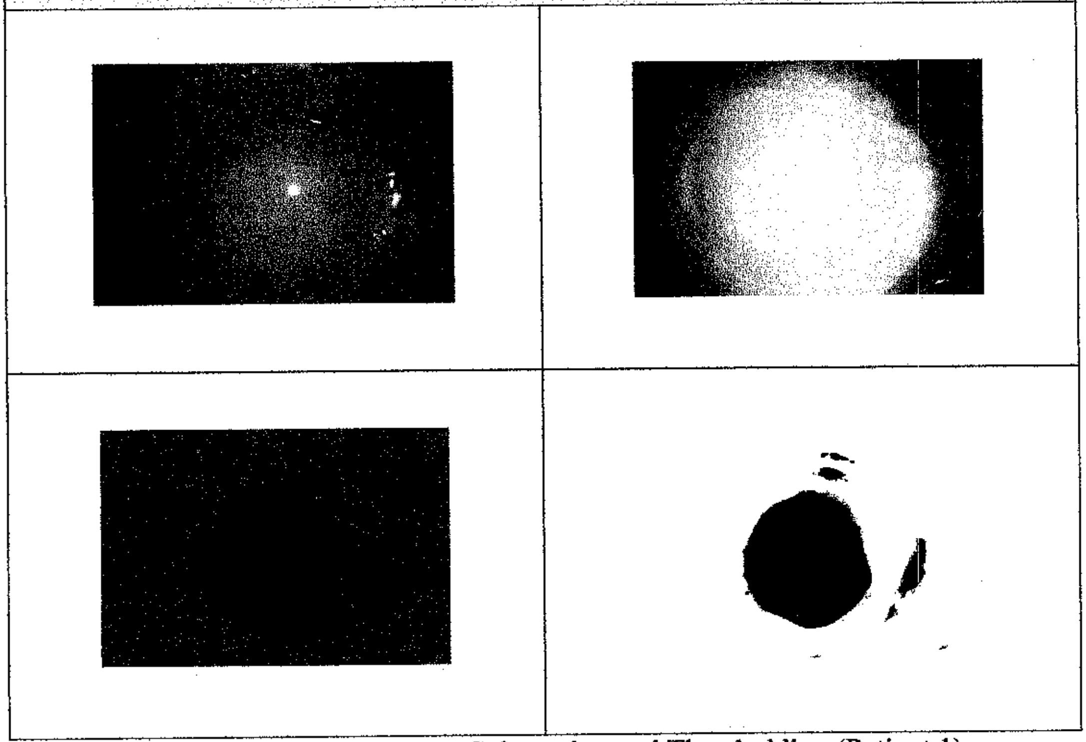


Figure 14 : Image Subtraction and Thresholding (Patient 1)

Patient 2	
Image 1	

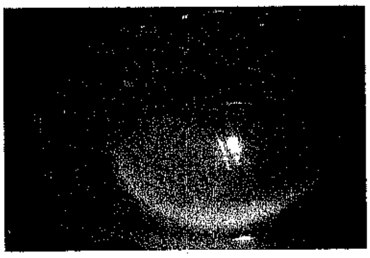
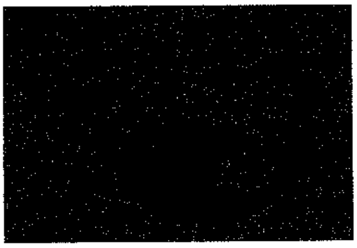
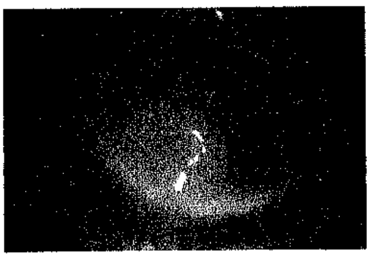
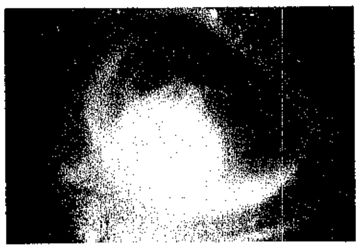
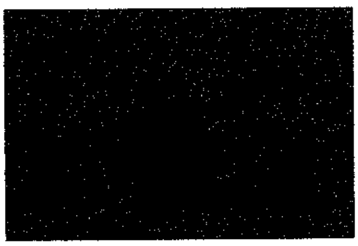

	


Image 2	
	
	

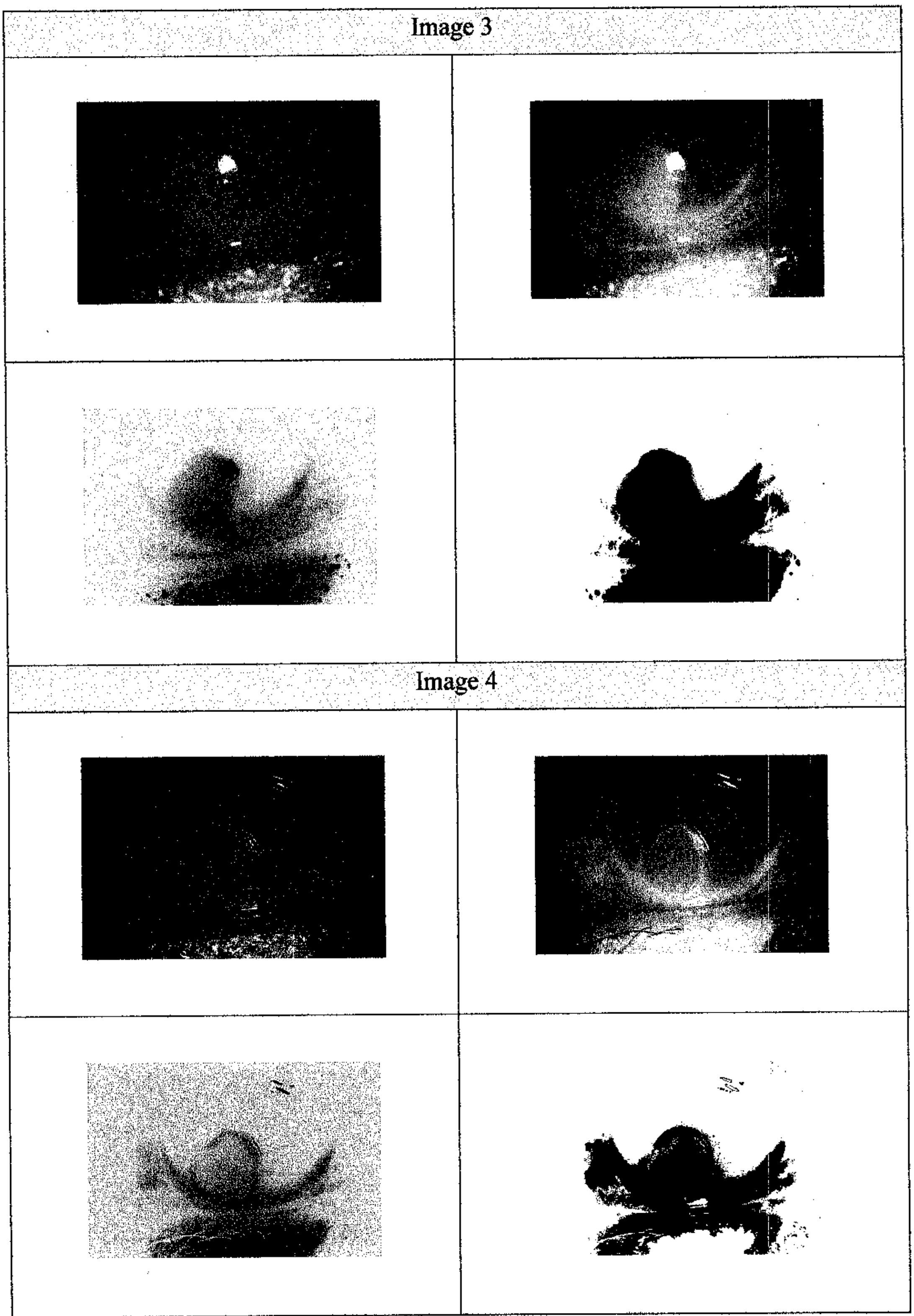
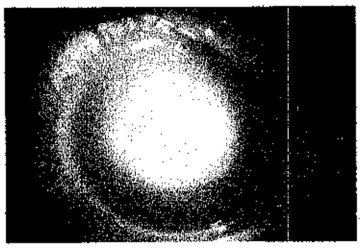


Figure 15 : Image Subtraction and Thresholding (Patient 2)

Patient 3

Image 1



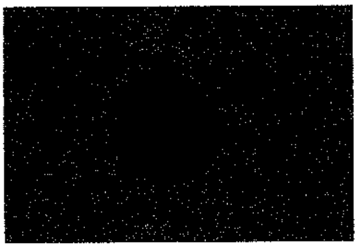
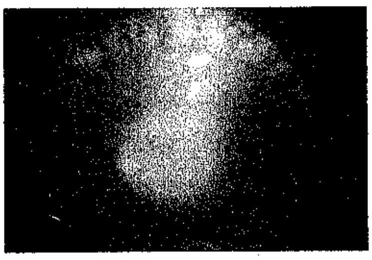
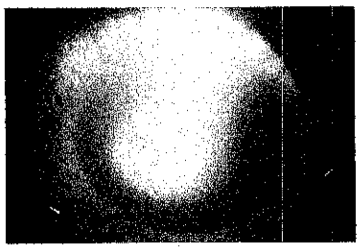
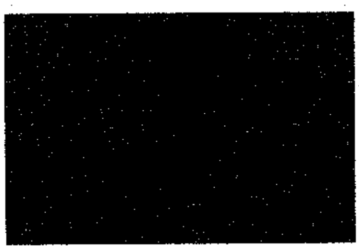


Image 2







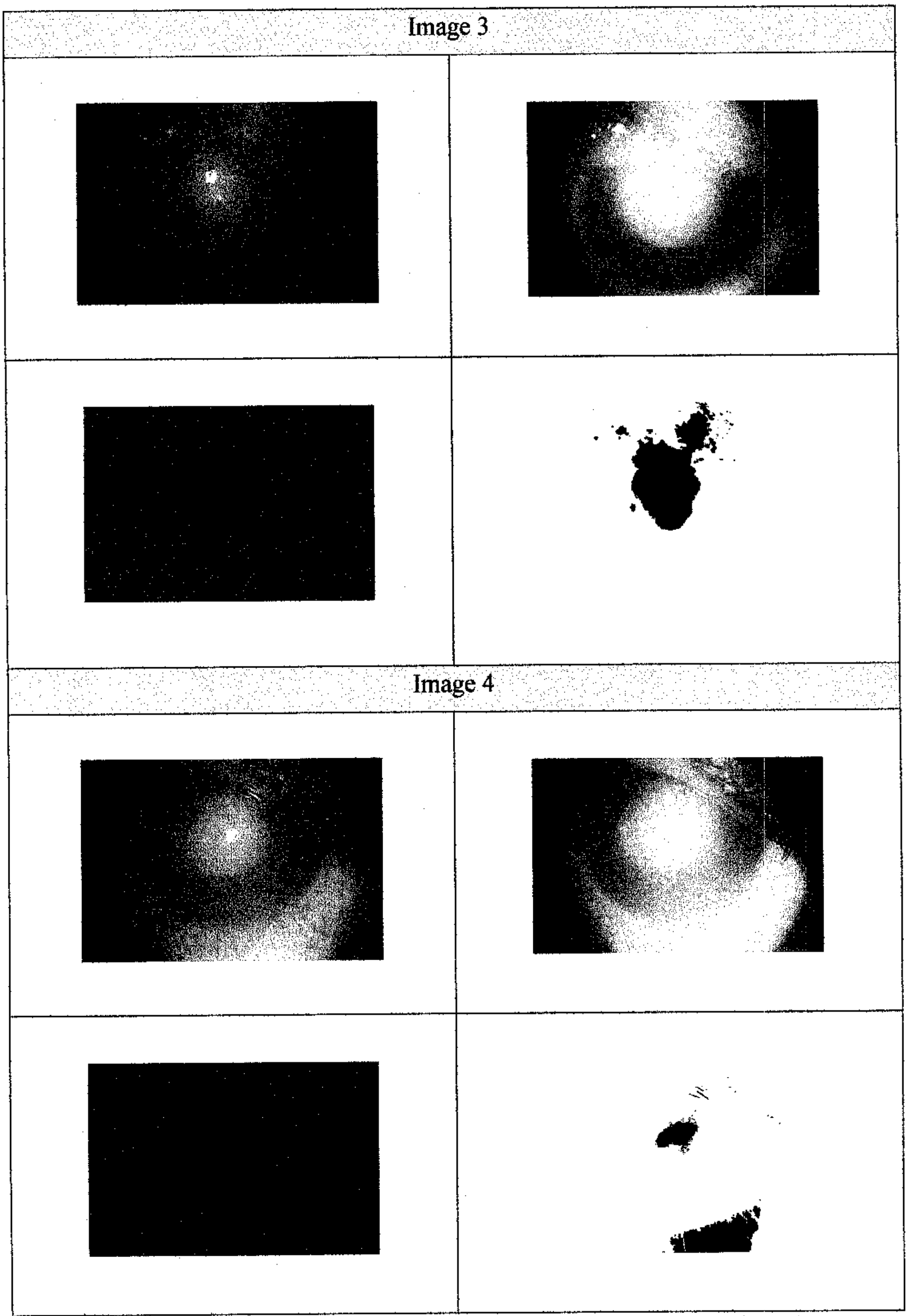


Figure 16 : Image Subtraction and Thresholding (Patient 3)

4.3 Image Analysis

This image analysis consists of two parts:-

i. Calculating vertical, horizontal and area
ii. Analysis on the progression of corneal ulcer

4.3.1 Calculating vertical, horizontal and area

One of the methods to monitor the corneal ulcer progression is by comparing the vertical, horizontal and area of corneal ulcer shape. From the figure, both the vertical and horizontal are determined on the longest point-to-point. The parameters are calculated using two software, MATLAB software and SCION software. The result of this process is shown in the Figure 11.

Referring to the figure, it can be observed that both software produce different results. MATLAB software results are quite not reasonable and accepted because of existed error in previous stage. SCION software establishes reasonable result and can be accepted compared to MATLAB software result. Thus, the study chooses to accept SCION software results in calculating horizontal, vertical and area.

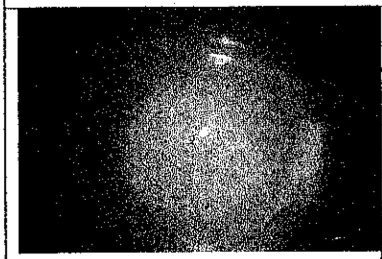
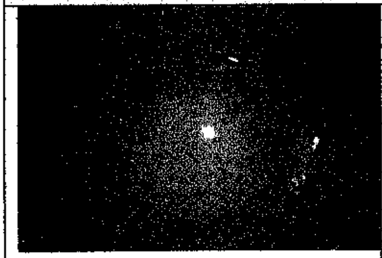
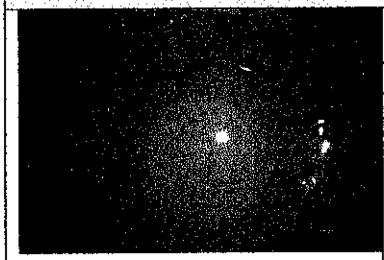
Patient 1			
Image 1			
		MATLAB	SCION
	Horizontal		0.81
	Vertical		0.76
	% Area	29	15.75
Image 2			
	Horizontal		0.73
	Vertical		0.69
	% Area	28.36	13.65
Image 3			
	Horizontal		0.71
	Vertical		0.69
	% Area	27.64	12.92

Figure 17 : Horizontal, Vertical and Area of Corneal Ulcer (Patient 1)

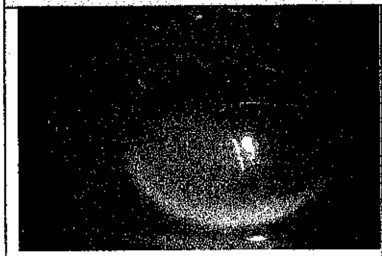
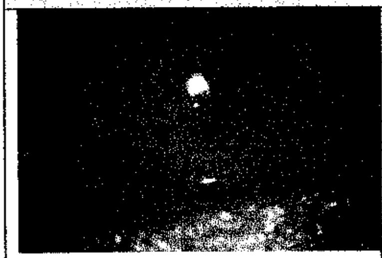
Patient 2			
Image 1			
	Horizontal		0.63
	Vertical		0.60
	% Area	2.83	9.42
Image 2			
	Horizontal		0.68
	Vertical		0.59
	% Area	2.97	9.79
Image 3			
	Horizontal		0.53
	Vertical		0.54
	% Area	22.1	8.51
Image 4			
	Horizontal		0.42
	Vertical		0.53
	% Area	24.9	8.35

Figure 18 : Horizontal, Vertical and Area of Corneal Ulcer (Patient 2)

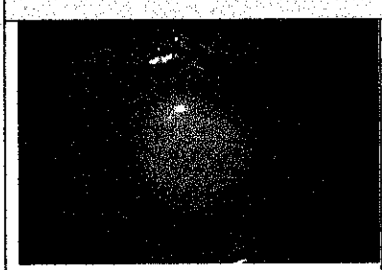
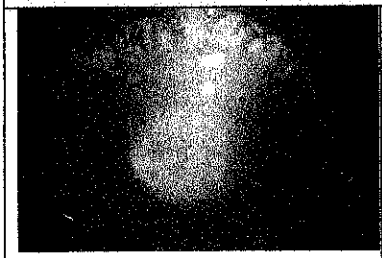
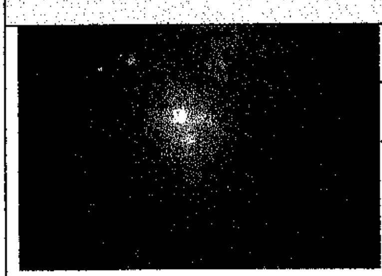
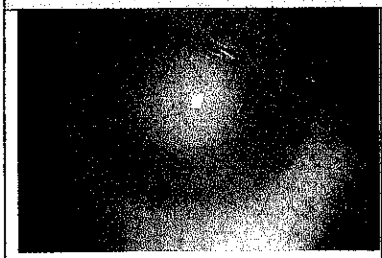
Patient 3			
Image 1			
	Horizontal		0.59
	Vertical		0.49
	% Area	3.06	7.77
Image 2			
	Horizontal		0.37
	Vertical		0.27
	% Area	3.06	3.96
Image 3			
	Horizontal		0.23
	Vertical		0.19
	Area	3.08	1.37
Image 4			
	Horizontal		0.12
	Vertical		0.12
	Area	3.23	0.34

Figure 19 : Horizontal, Vertical and Area of Corneal Ulcer (Patient 3)

4.3.2 Analysis on the progression of corneal ulcer

Three patients involve in the study, namely Patient 1, Patient 2 and Patient 3. The discussion on the results for each patient is as follows.

4.3.2.1 Analysis on Patient 1

Patient 1 has three images, namely image 1, image 2 and image 3. It is observed from the images that the corneal ulcer shape half covers corneal area. However, the images show that it is slightly decreasing from an image to an image.

The images show deterioration on corneal ulcer area, from 15.75 % to 13.65 % to and to 12.92 %. The decreasing percentages for the images are 2.1 % and 0.73 %, which are quite small and classified as slow healing. It is also observed that final image still consist of corneal ulcer. Thus, it is concluded that Patient 1 still not heals fully, but only deteriorates.

4.3.2.2 Analysis on Patient 2

Patient 2 has four images, namely image 1, image 2, image 3 and image 4. It is observed from the images that the corneal ulcer shape quite fully covers corneal area. However, the images show that it is slightly decreasing from an image to an image.

The images show both deterioration and progression on corneal ulcer area, from 9.42% to 9.79% to 8.51 % and to 8.35 %. Corneal ulcer area increases for image 2, with 0.37 %. While, the decreasing percentages for other images are 1.28% and 0.16 %, which are quite small and classified as slow healing. It is also observed that final image still consist of corneal ulcer. Thus, it is concluded that Patient 2 still not heals fully, but only deteriorates.

4.3.2.3 Analysis on Patient 3

Patient 3 has four images, namely image 1, image 2, image 3 and image 4. It is observed from the images that the corneal ulcer shape covers small portion of corneal area. However, the images show that it is slightly decreasing from an image to an image.

The images show deterioration on corneal ulcer area, from 7.77% to 3.96% to 1.37% and to 0.34%. The decreasing percentages for other images are 3.81%, 2.59% and 1.03% which are quite large and classified as fast healing. It is also observed that final image still consist very small portion of corneal ulcer. Thus, it is concluded that Patient 2 still not heals fully, but only deteriorates.

4.4 Graphical User Interface

Finally, graphical user interface is built to ease ophthalmologist. The interface is using the GUI offered in the MATLAB software. Figures below show the graphical user interface for this project.

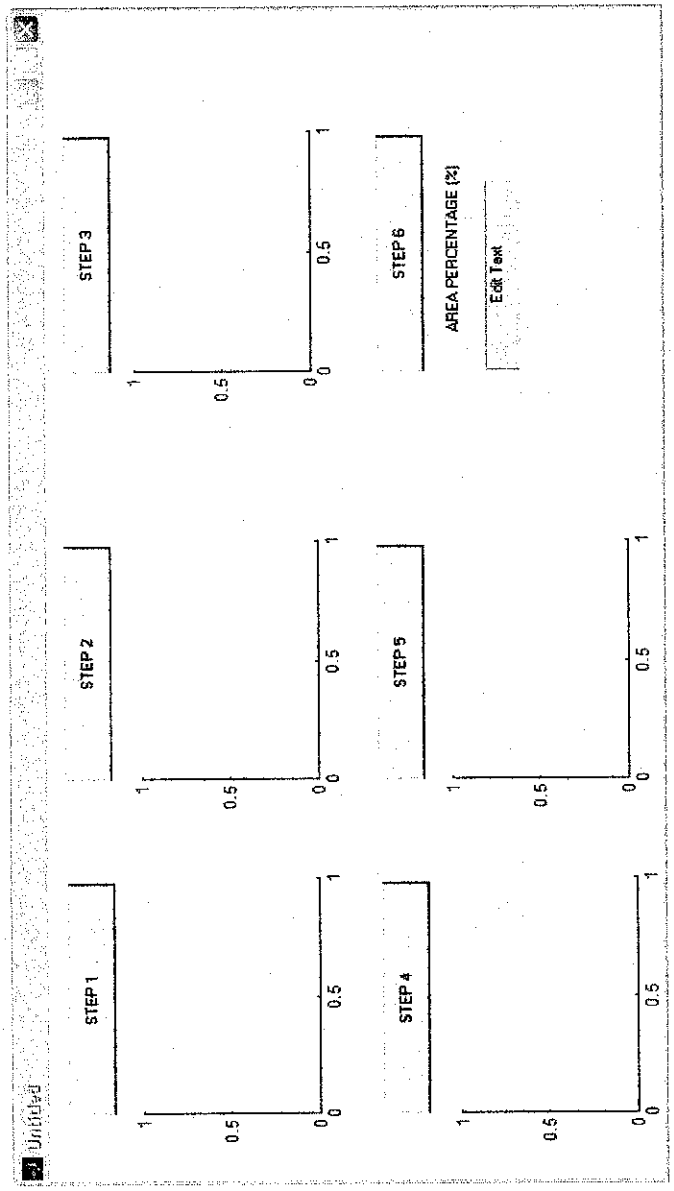


Figure 20 : Graphical User Interface

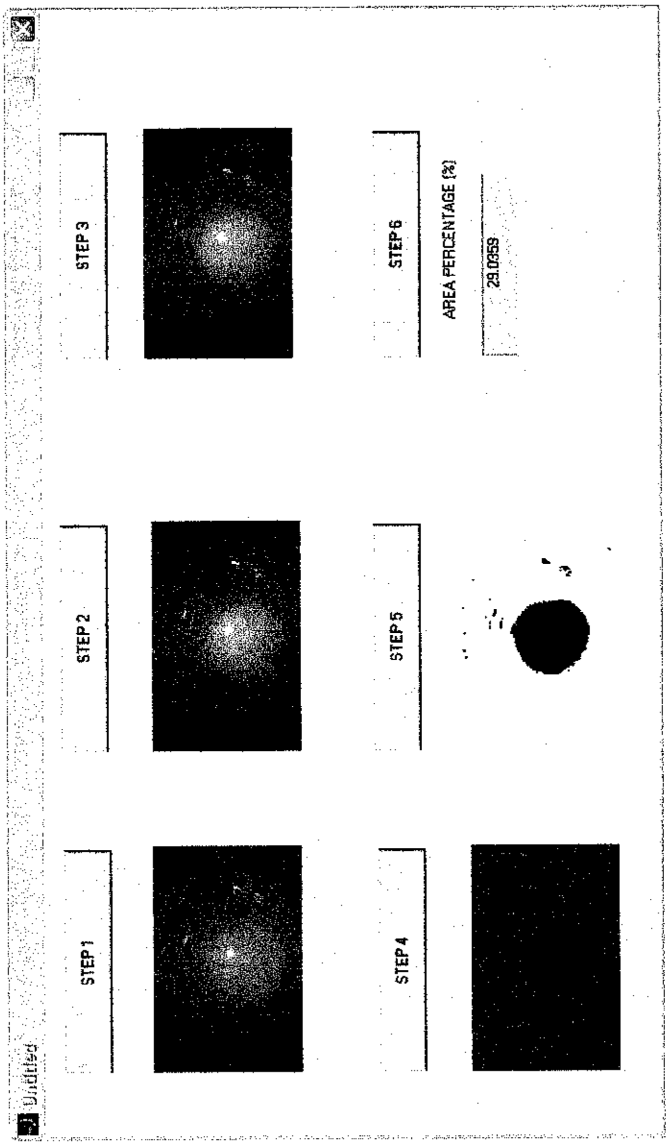


Figure 21 : Graphical User Interface (Patient 1, Image 1)

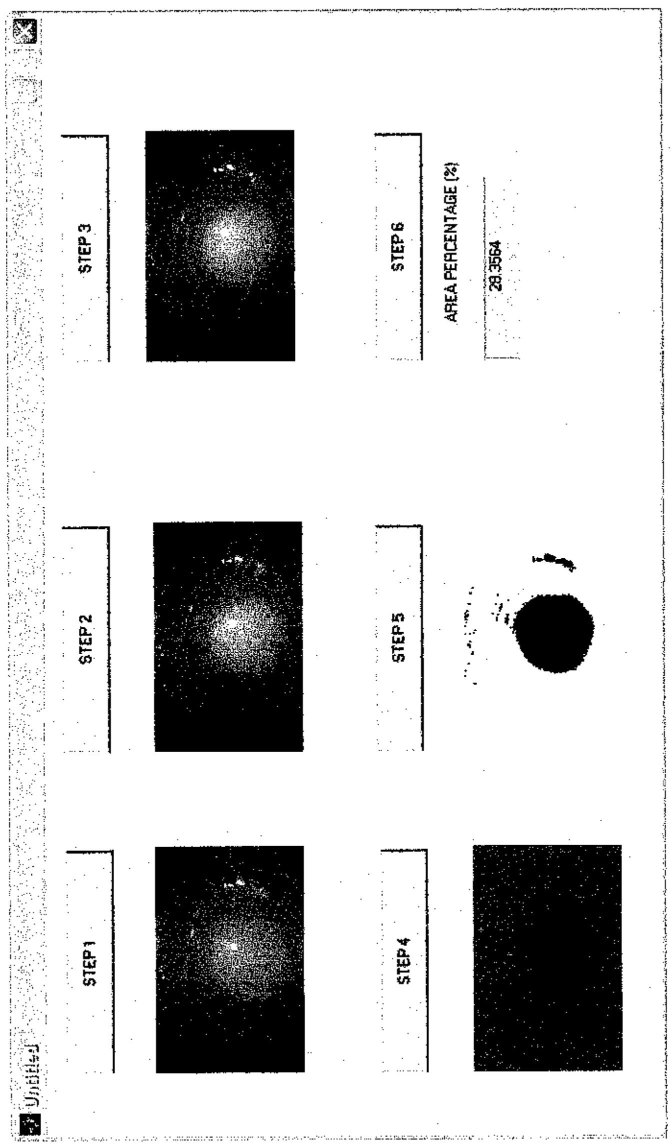


Figure 22 : Graphical User Interface (Patient 1, Image 2)

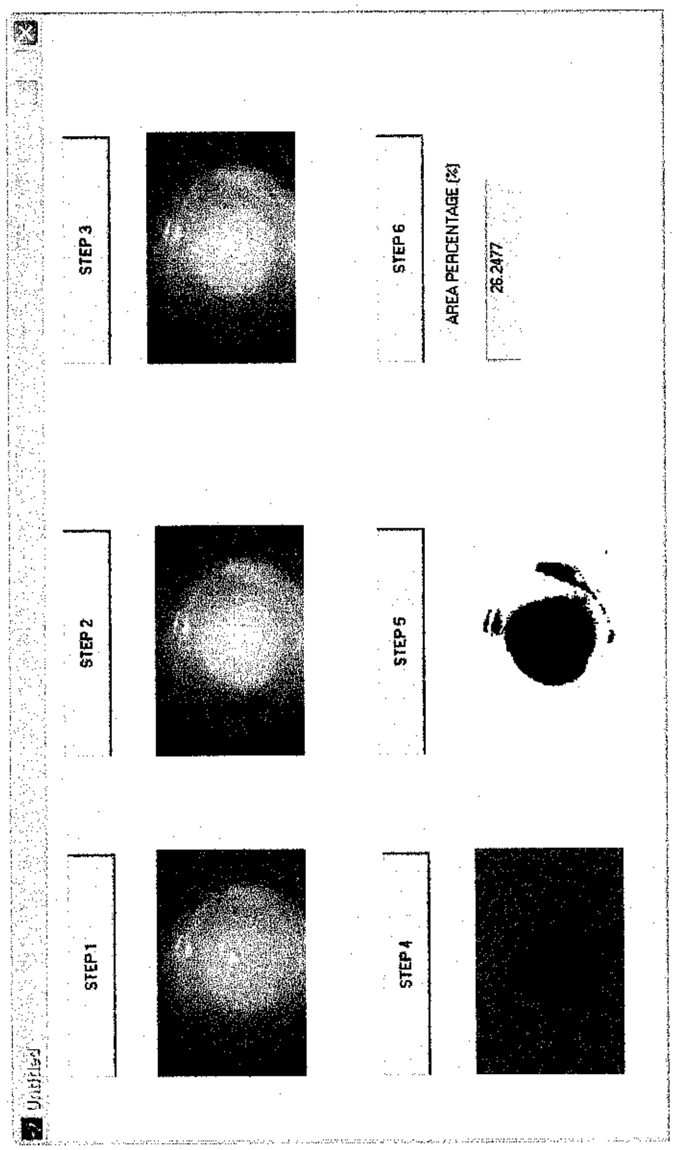


Figure 23 : Graphical User Interface (Patient 1, Image 3)

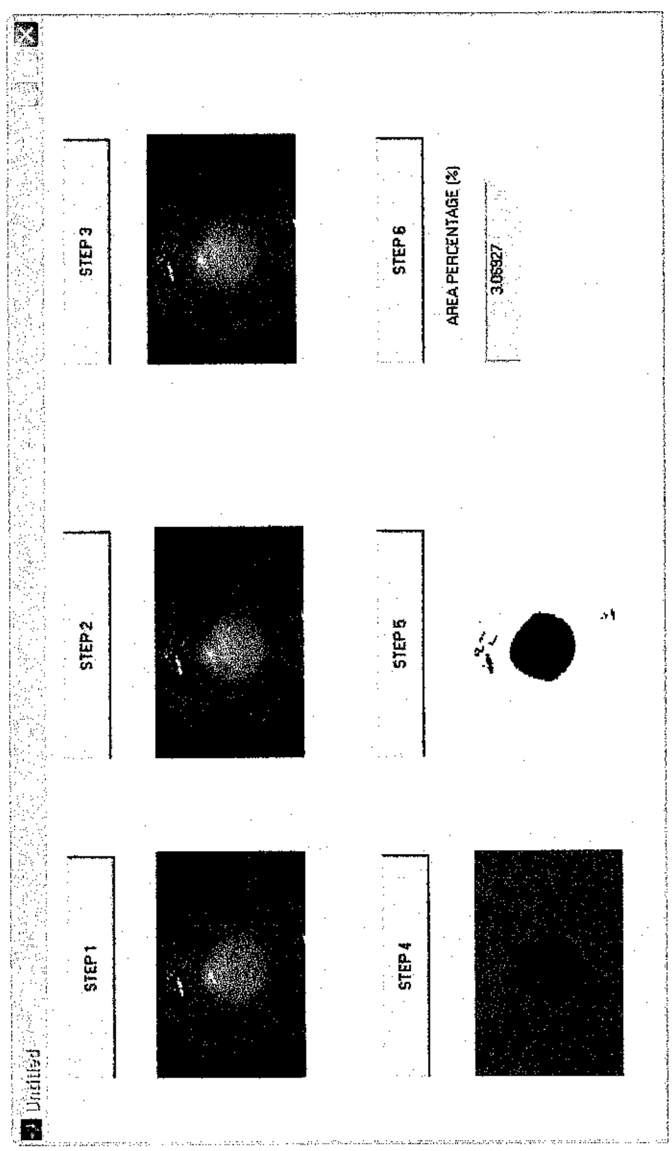


Figure 24 : Graphical User Interface (Patient 2, Image 1)

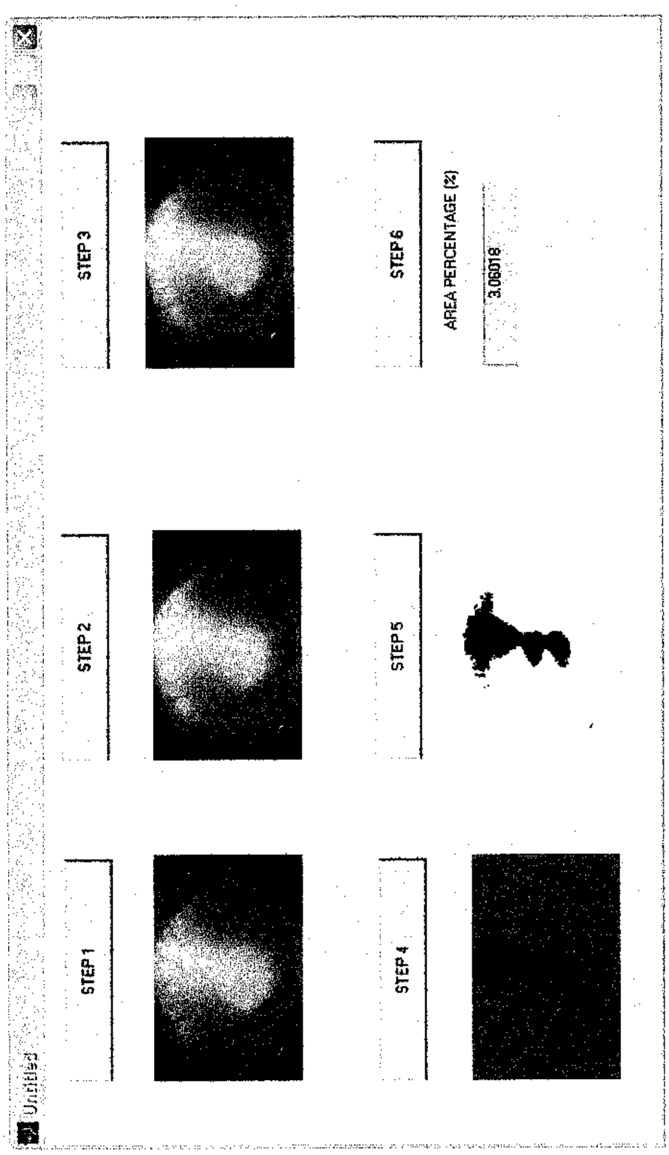


Figure 25 : Graphical User Interface (Patient 2, Image 2)

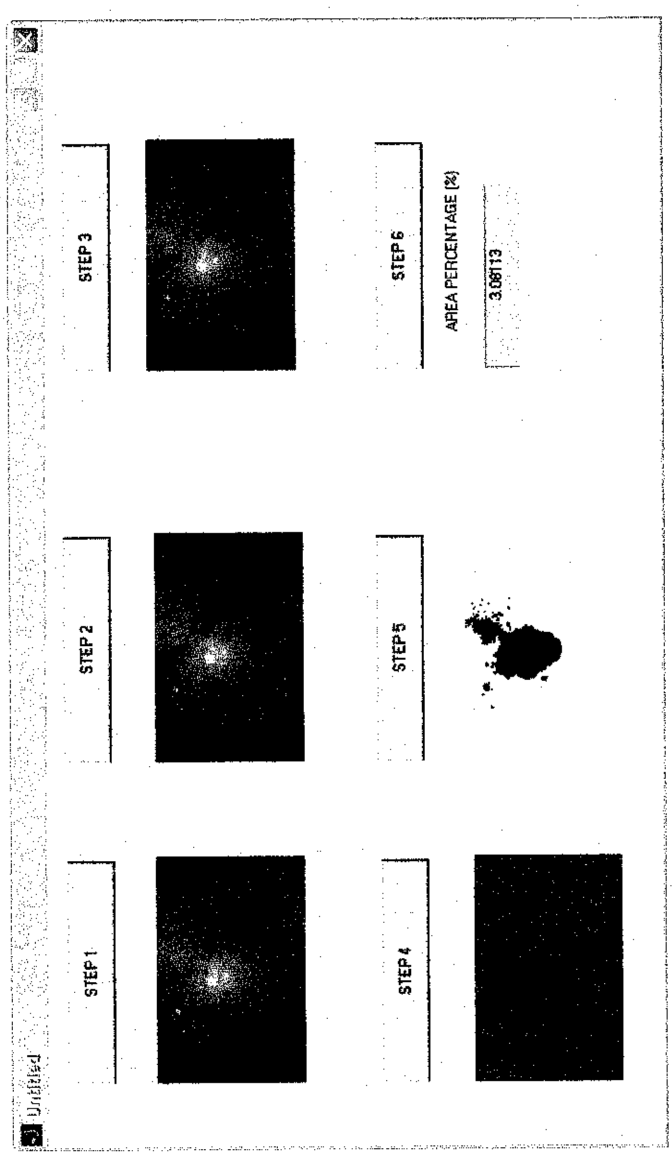


Figure 26 : Graphical User Interface (Patient 2, Image 3)

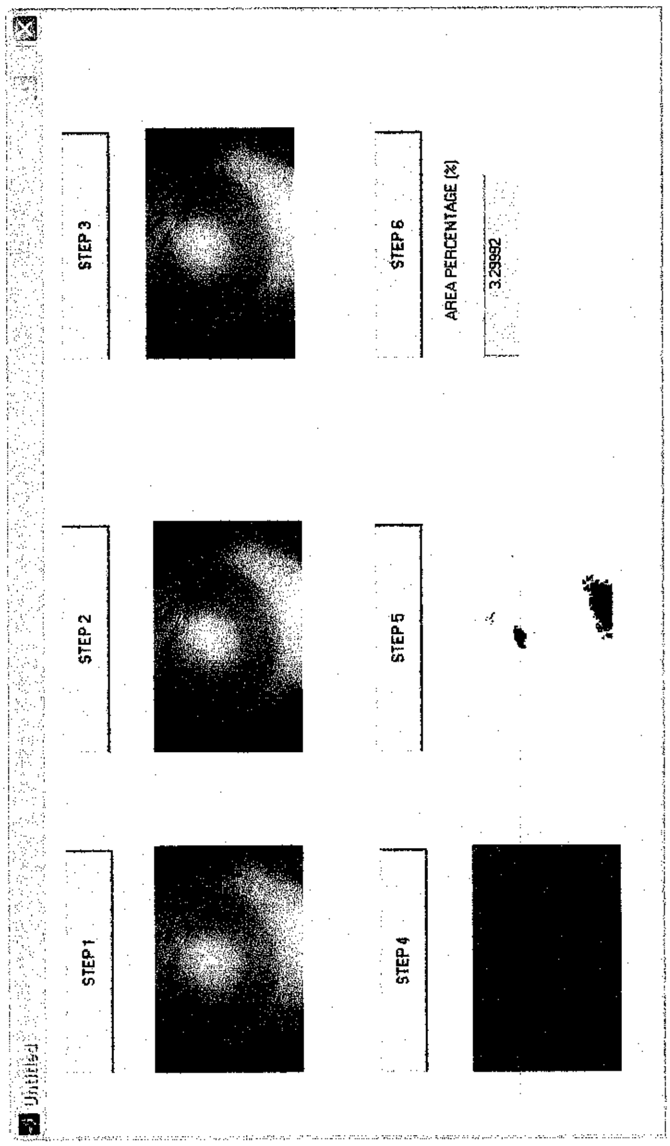


Figure 27 : Graphical User Interface (Patient 2, Image 4)

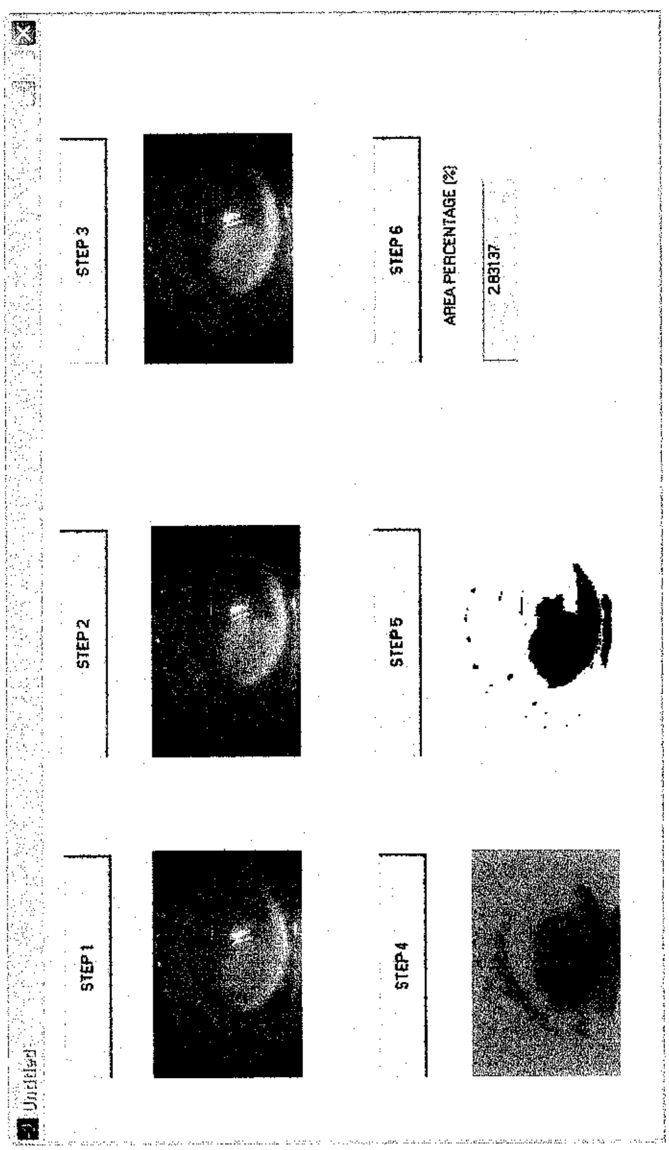


Figure 28 : Graphical User Interface (Patient 3, Image 1)

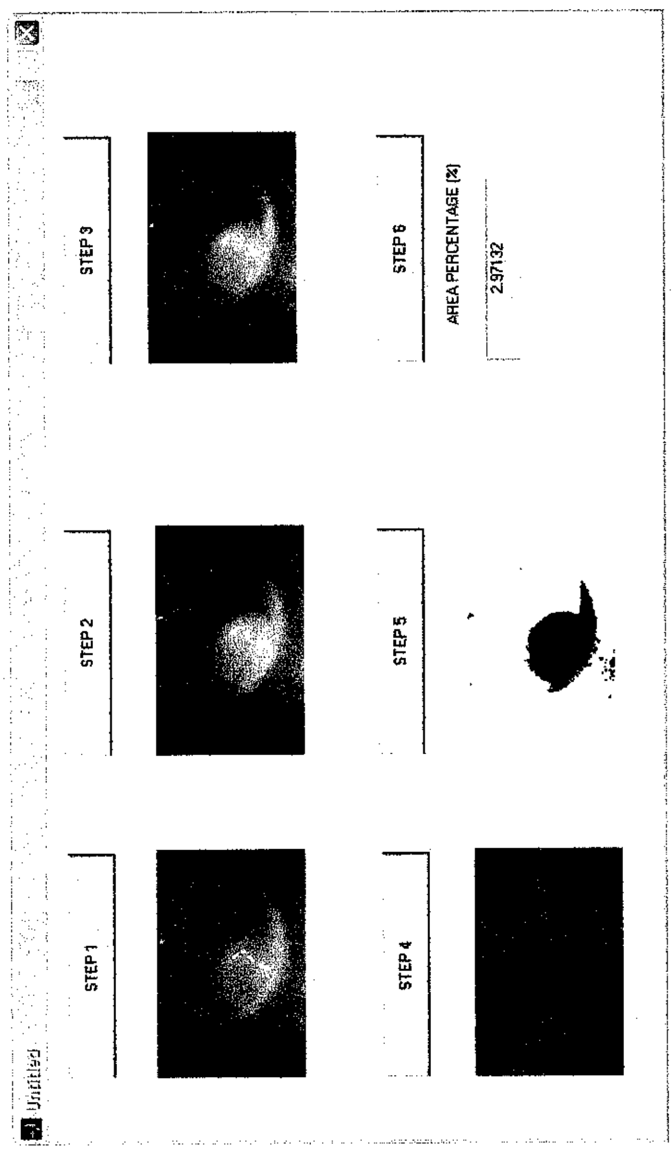


Figure 29 : Graphical User Interface (Patient 3, Image 2)

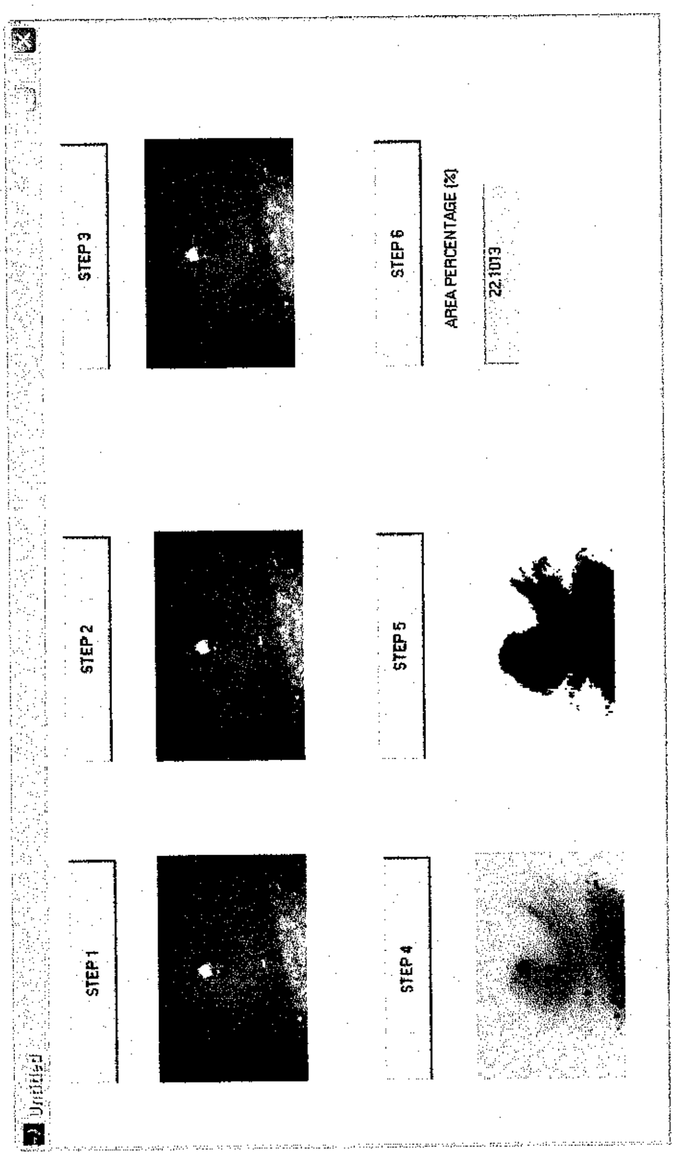


Figure 30 : Graphical User Interface (Patient 3, Image 3)

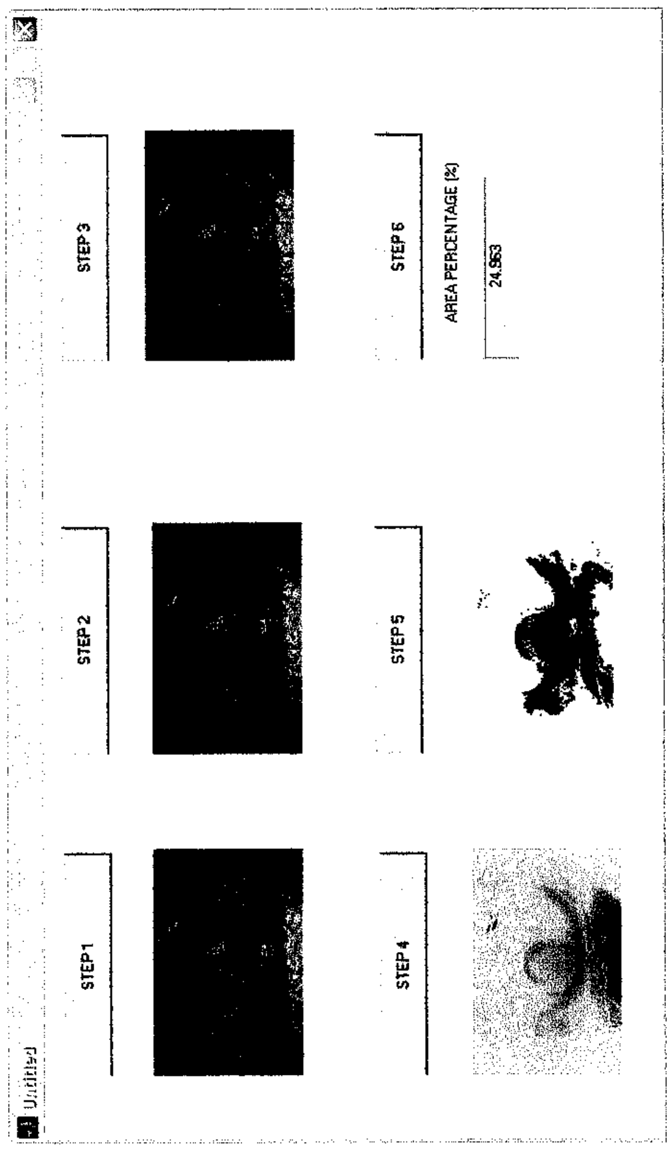


Figure 31 : Graphical User Interface (Patient 3, Image 4)

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In conclusion, the project successfully introduces digital imaging method as a method of monitoring corneal ulcer progression. Software used is MATLAB software version 6.1, SCION software and other photo editor software. Images had been enhanced properly for a better quality images. Parameters of area, vertical and horizontal had been obtained for each image. Since MATLAB's results produces large errors, back up calculations had been done using SCION software. Next, some analysis had been done to determine the corneal ulcer progression. Finally, graphical user interface had been built to help ophthalmologist monitoring the progression.

5.2 Recommendations

It is suggested a few recommendations for future enhancement in this project:

1) Develop C or C++ programs for the project

Currently the project only develops MATLAB programs in monitoring the corneal ulcer progression. Thus, it is expected for the future to develop other language programs, such as C or C++ for a depth programming of digital image processing application.

2) Comparisons of existed monitoring corneal ulcer progression and digital image processing method.

Currently the project only provides data obtained from digital image processing method. Thus, it is expected for the future to compare data obtained from digital image processing method with existed methods, clinical assessment and bacteriological identification for validity.

3) Perform full analysis on the corneal ulcer disease

Currently the project does not perform full analysis on the corneal ulcer disease. Thus, it is expected for the future to perform full analysis with strength literature studies on corneal ulcer disease.

4) Perform image processing in RGB

Currently, the project performs image processing by first converting to the grayscale. It affects results of the project because of existing unwanted noise. Thus, it is expected for the future to perform image processing in RGB to reduce the effect from unwanted noise.

REFERENCES

[1] http://www.tedmontgomery.com/the_eye

[2] http://seniorhealth.about.com/library/vision/blcornea.htm

[3] http://www.mathworks.com

[4] Gonzalez, Woods, *Digital Image Processing Second Edition*, Prentice Hall, 2002.

[5] *Image Processing Toolbox User's Guide*, Math Works Inc, 1997

[6] Kenneth R. Castleman, *Digital Image Processing*, Prentice Hall, 1996

[7] Randy Crane, *Simplified Approach to Image Processing*, Prentice Hall, 1997

[8] Dr Ezanee Mokhtar, 2003, *A comparative study on the Agreement of using digital imaging with analysis software for assessment of corneal ulcer size progression*. Thesis Master Proposal. Universiti Sains Malaysia.

[9] Khairul Nisak Md Hasan, June 2004, *Detection of Microcalcifilication using Mammograms*, Universiti Teknologi PETRONAS.

[10] Farah Aini Binti Hj. Amin Nordin, June 2004, *Medical Imaging of Skin Disease*, Universiti Teknologi PETRONAS

APPENDICES

APPENDIX A

FIRST SEMESTER GANTT CHART

GANTT CHART FYP1

Detail/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Selection of Project Topic																
Topic assigned To students																
Preliminary Researh Work																
Introduction & Objective																
List of reference/literature																
Project planning																
Submission of Preliminary Report				•												
Project Work																
Reference/literature																
Obtaining data (images)																
Obtaining info on image processing tools																
Laboratory work (MATLAB)																
Submission of Progress Report								•								
Project Work Continue																
Processing images																
Research on glaucoma																
Laboratory work (MATLAB)																
Submission of Interim Report Final Draft												•				
Submission of Interim Report														•		
Oral Presentation															•	

Legend •
Planning Work

APPENDIX B

SECOND SEMESTER GANTT CHART

GANTT CHART FYP2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Development of Program									•							
a) Image Processing					•											
b) Monitoring Image Progression								•								
c) Graphical User Interface									•							
Evaluation of result											•					
a) Analysis										•						
b) Analysis with Ophthalmologist											•					
Submission of Progress Report 1				•												
Submission of Progress Report 2								•								
a) Assessment of Progress Report 2								•								
Pre-EDX												•				
Submission of Final Report															•	
a) Assessment of Final Report														•		
Oral Presentation																
a) Presentation material preparation																
b) Oral Presentation														•		

Legend •
Planning Work

APPENDIX C

MATLAB PROGRAM

```
-----------------------------------------------patient 1---------------------------------------
---------------------------------------------- image 1-----------------------------------------
I = imread ( 'btul1.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)

---------------------------------------------- image 2-----------------------------------------
I = imread ( 'btul2.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **image 3** --

```
I = imread ( 'btul3.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **patient 2** --

-- **image 1** --

```
I = imread ( 'SSI1.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **image 2**---

```
I = imread ( 'SSI2.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **image 3**---

```
I = imread ( 'SSI3.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **image 4**--

```
I = imread ( 'SS14.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

---**patient 3**---

--- **image 1**---

```
I = imread ( 'MNY1.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **image 2**--

```
I = imread ( 'MNY2.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **image 3**--

```
I = imread ( 'MNY3.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

-- **image 4**--

```
I = imread ( 'MNY4.tif' )
n = rgb2gray (I)
j = imadjust ( n, stretchlim (n), [ ] )
figure, imshow (j)
imhist (j)
gI = imadjust (im2double (J), [ ], [0 1] )
marker1 = imsubtract (incomplement (g1) ,0.8 )
recon = imreconstruct (marker, incomplement (g1) )
figure,  imshow (recon)
impixel (recon)
BW = im2bw ( recon, 0.2196 )
figure, imshow (BW)
bwarea (BW)
```

APPENDIX D

GUI PROGRAM

```
function varargout = mny1(varargin)
if nargin == 0  % LAUNCH GUI
        fig = openfig(mfilename,'reuse');
        handles = guihandles(fig);
        guidata(fig, handles);
        if nargout > 0
                varargout{1} = fig;
        end
elseif ischar(varargin{1})
        try
                if (nargout)
                        [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
                else
                        feval(varargin{:}); % FEVAL switchyard
                end
        catch
                disp(lasterr);
        end

end


% --------------------------------------------------------------------
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
axes(handles.axes1)
I = imread ( 'MNY1.tif' );
imshow(I);


% --------------------------------------------------------------------
function varargout = pushbutton2_Callback(h, eventdata, handles, varargin)
axes(handles.axes2)
I = imread ( 'MNY1.tif' );
n = rgb2gray (I);
imshow(n);


% --------------------------------------------------------------------
function varargout = pushbutton3_Callback(h, eventdata, handles, varargin)
axes(handles.axes3)
I = imread ( 'MNY1.tif' );
n = rgb2gray (I);
j = imadjust ( n, stretchlim (n), [ ] );
imshow(n);
```

```
% -----------------------------------------------------------------
function varargout = pushbutton4_Callback(h, eventdata, handles, varargin)
axes(handles.axes4);
I = imread ( 'MNY1.tif' );
n = rgb2gray (I);
j = imadjust ( n, stretchlim (n), [ ] );
gI = imadjust (im2double (j), [ ], [0 1] );
marker1 = imsubtract (imcomplement (gI) ,0.65 );
recon = imreconstruct (marker1, imcomplement (gI) );
imshow(recon);


% -----------------------------------------------------------------
function varargout = pushbutton5_Callback(h, eventdata, handles, varargin)
axes(handles.axes5)
I = imread ( 'MNY1.tif' );
n = rgb2gray (I);
j = imadjust ( n, stretchlim (n), [ ] );
gI = imadjust (im2double (j), [ ], [0 1] );
marker1 = imsubtract (imcomplement (gI) ,0.65 );
recon = imreconstruct (marker1, imcomplement (gI) );
BW = im2bw ( recon, 0.1922);
imshow(BW);

% -----------------------------------------------------------------
function varargout = pushbutton6_Callback(h, eventdata, handles, varargin)
I = imread ( 'MNY1.tif' );
n = rgb2gray (I);
j = imadjust ( n, stretchlim (n), [ ] );
gI = imadjust (im2double (j), [ ], [0 1] );
marker1 = imsubtract (imcomplement (gI) ,0.65 );
recon = imreconstruct (marker1, imcomplement (gI) );
BW = im2bw ( recon, 0.1922 );
lesion = bwarea (BW);
image = 1312*2000*3;
percentage= (lesion/image)*100;
set(handles.edit1,'string',percentage);


% -----------------------------------------------------------------
function varargout = edit1_Callback(h, eventdata, handles, varargin)
set(handles.edit1,'string',percentage);
I = imread ( 'MNY1.tif' );
n = rgb2gray (I);
j = imadjust ( n, stretchlim (n), [ ] );
gI = imadjust (im2double (j), [ ], [0 1] );
marker1 = imsubtract (imcomplement (gI) ,0.65 );
recon = imreconstruct (marker1, imcomplement (gI) );
BW = im2bw ( recon, 0.1922 );
lesion = bwarea (BW);
image = 1312*2000*3;
percentage= (lesion/image)*100;
```