1982

Image thresholding and feature extraction techniques.

Mohammed Arif Janjua
University of Windsor

0-315-09340-4

National Library of Canada | Bibliothèque nationale du Canada | CANADIAN THESES ON MICROFICHE | *THÈSES CANADIENNES SUR MICROFICHE*

5

57257

NAME OF AUTHOR/*NOM DE L'AUTEUR* Mohammed Arif Janjua

TITLE OF THESIS/*TITRE DE LA THÈSE* Image thresholding and feature extraction techniques.

UNIVERSITY/*UNIVERSITÉ* University of Windsor, Windsor Ont.

DEGREE FOR WHICH THESIS WAS PRESENTED/ *GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE* M.A.Sc.

YEAR THIS DEGREE CONFERRED/*ANNÉE D'OBTENTION DE CE GRADE* May 1982

NAME OF SUPERVISOR/*NOM DU DIRECTEUR DE THÈSE* Dr. G. A. Jullien

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

DATED/*DATÉ* Feb. 25, 1982 SIGNED/*SIGNÉ* Mohammed Arif Janjua

PERMANENT ADDRESS/*RÉSIDENCE FIXE* 437 FULLERTON DR.
SAN JOSE, CA 95111
U.S.A.

NL-91 (3-74)

IMAGE THRESHOLDING
AND FEATURE EXTRACTION TECHNIQUES

by

© Mohammed Arif Janjua

A Thesis

Submitted to the Faculty of Graduate Studies through
the Department of Electrical Engineering in Partial
Fulfillment of the Requirement for the Degree of
Master of Applied Science
at the
University of Windsor

Windsor, Ontario
Canada

1981

770394

APPROVED BY

Dr. G. A. Jullien

Dr. M. Shridhar

Dr. M.A. Sid-Ahmed

Dr. H. E. Toews

ABSTRACT

An approach to transforming an image in order to facilitate threshold selection is presented. A threshold can be selected automatically by examining the histogram of the gray-levels in an image. If the histogram is bimodal, it would be simple to threshold the image. However, the problem becomes a little more complex when the histogram is not clearly bimodal. It has been proposed previously that one could specify a histogram and map it onto an image in order to yield a transformed image whose histogram has the specified distribution.

This thesis presents an approach whereby a bimodal histogram consisting of two Gaussian distributions of different means, but same standard deviations is specified. This histogram is then used to transform the original image into one that has the specified distribution. This transformed image could then easily be thresholded at a single level. The approach of direct histogram specification has also been employed for purposes of multi-level thresholding, and the results achieved have been very encouraging.

This thesis also addresses a technique for feature extraction using template matching. The application of this approach to fault detection of manufactured parts is illustrated. The development of a system is thus proposed (particularly for applications to quality control), which would efficiently threshold an image and extract features from it.

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to his supervisor, Dr. G.A. Jullien, for many valuable discussions and constructive criticism throughout the study period. In addition, sincere thanks go out to my co-supervisor, Dr. Sid-Ahmed for his invaluable suggestions towards carrying out this research. Without his help and constant encouragement, this project would not have gotten off the ground. Thanks are also due to Dr. M. Shridhar and other members of the department who helped the author in different ways.

A special word of gratitude also goes out to Mr. Yousef Safar for utilizing his expertise in photography to help illustrate this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1. An Image Model

The term "image" is defined in Webster's Dictionary as *"a representation, likeness, or imitation of an object or thing, a vivid or graphic description, something introduced to represent something else."* A picture is defined as *"a representation made by a painting, drawing or photography. A vivid, graphic, accurate description of an object or thing so as to suggest a mental image or give an accurate idea of the thing itself."* However, in the content of image processing, the word 'picture' is used synonymous with the word 'image'.

An image is considered as a flat object whose appearance varies from point to point. This variation can be described by a single parameter in the case of 'black and white' pictures, corresponding to the intensity of light emanated at each point. An image is thus a two-dimensional light intensity function $f(x,y)$, where x and y denote spatial coordinates and the value of the picture function f at any point (x,y) is proportional to the total amount of light reaching the observer from the given point. This is also referred to as the gray level of the picture at that point [1,2,3,4].

While processing images on a digital computer, one usually wishes to regard them as discrete arrays of numbers, i.e., as matrices, rather than as functions. The row and column indices of the matrix identify a point in the image, and the corresponding value of the matrix element identifies the gray level of the image at that point. Each element of the matrix is commonly referred to as a pixel.

1.2. Image Thresholding

In digital picture processing, it is often desirable to assume that a picture function can take on only a finite set of values, or in other words, its gray levels are quantized. Thus the gray levels in a picture are spread out into a finite number of discrete values, depending upon the resolution desired and the constraints of computer memory.

A binary-valued picture function which can take on only two values (black and white - with no intermediate gray levels) is usually of particular importance in feature extraction. Thus, a picture is appropriately transformed into two distinct levels by selecting an appropriate 'threshold level'. This process of thresholding an image is performed to separate the objects in an image from its 'background', thereby achieving the desirable information content of an image.

The techniques of thresholding an image can be classified with three broad categories as follows:

1.2.1. Fixed thresholding

In this approach, one threshold level is selected for the entire image. The process of selecting an optimum threshold, so as not to lose any desirable information is by no means an easy one. In past years, several techniques for optimal threshold selection have evolved. It has been shown in various papers that instead of choosing a crude threshold for a given picture, some sort of pre-processing should be performed on it, in order to facilitate the selection of a threshold [1,3,5,9,10,11].

One widely accepted method of threshold selection is where a threshold is selected automatically by examining the probability distribution of the gray levels in an image [1,2,3,6,8], which is also referred to as a histogram of its gray levels. If the histogram of an image suggests that it is feasible to segment the gray level population of an image into two distinct sub-populations, then a threshold is said to exist. Consider the histogram of Figure 1.1. It is quite evident that if we choose a threshold level at the bottom of the valley, separating the two parts, (say, T), we would then be segmenting the gray levels into two sections, one representing the 'objects' and the other, the 'background'. Thus, if the histogram of an image is bimodal, it would be quite straightforward to select a suitable threshold for that image. However, in most situations, this is not the case, and threshold selection could pose a problem.

1.2.2. Variable thresholding

As discussed in the previous sections, more often than not we arrive at a situation where the histogram of an image is not clearly bimodal. This suggests that it is not possible to appropriately threshold the entire image at one fixed level. Chow and Kaneko [14] suggested to segment an image into smaller sections, and to find a threshold for each small segment of the image. In their paper, they suggested to segment an image into 7 x 7 smaller regions, each subsequent region overlapping 50% of the previous one. Histograms were then computed for each region and thresholds were assigned to regions having bimodal histograms. For regions where thresholds could not be assigned, they were interpolated based upon the average value of the gray levels of that region, and upon the thresholds assigned to their neighbouring regions. The entire image was thus thresholded at various levels. Chow and Kaneko [14] displayed the effectiveness of this approach to

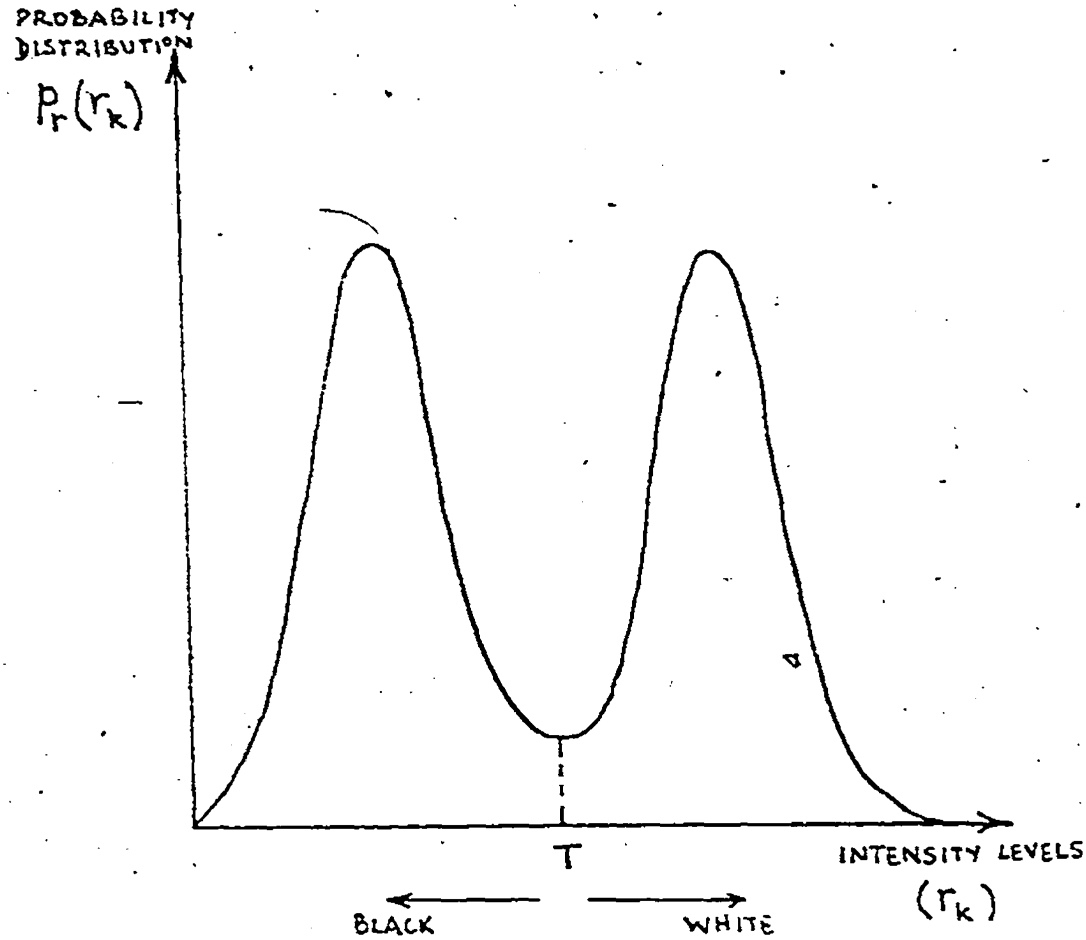


Figure 1.1 A bimodal histogram

medical applications, and Nakagama and Rosenfeld [13] showed that this method was very useful in other applications as well.

1.2.3. Multi-level thresholding

While dealing with the selection of a threshold, one might encounter a situation where it would be difficult, if not totally impossible, to select one good threshold level for the entire image. For example, if the histogram of an image consists of three or four peaks, it would be difficult to choose one level that would separate the image into two classes. If some sort of a crude threshold level was selected, it would result in a considerable loss of information.

In order to cope with situations of this sort, the idea of multi-level thresholding was introduced [24]. As the name suggests, this technique deals with selecting more than one threshold level and thus segmenting the image into more than just two classes. The thresholds can be selected at the intervening valley bottoms between the peaks of the histogram. The various sub-populations in the image can be appropriately segmented by this scheme of thresholding.

As in the case of binary-level thresholding, the main problem in thresholding the image at more than one level arises in situations where the peaks of the histogram are not very well defined, and the detection of a valley between the peaks becomes impossible. This problem could be overcome if one could transform the histogram of an image in such a way that it becomes easier to detect the valley between the peaks.

1.3. Histogram Modification

The idea of modifying a histogram so that the valley between the peaks of a histogram is easier to detect, has been of wide use in image thresholding. Weszka and Rosenfeld [6] have suggested an approach where the 'edge value' of the pixels is used to modify the histogram. The edge value refers to a transition in gray level between adjacent pixels. If the rate of change is high, it indicates the presence of an 'edge' between two different sub-populations in a histogram. Weszka and Rosenfeld have shown the use of scatter plots, a plot between gray level and edge value, to modifying a histogram so that the valley is deeper, and thus easier to detect. The use of edge points for segmentation purposes has been shown to yield satisfactory results [7,17] in most cases.

Histogram modification has also been shown to be of great use in the enhancement of images. Histogram equalization is one of the tools used for this purpose.

1.3.1. Histogram equalization

The process of obtaining a uniform histogram for an image is known as histogram equalization. It has been shown [1,5,16] that by equalizing the histogram, one could enhance the contrast of an image. Hall [16] has suggested to transform a histogram using a transformation function equal to the cumulative distribution function of the pixels. This would yield an almost uniform distribution, thereby increasing the dynamic range of the pixels. This would in turn contribute to increasing the contrast of the image, consequently improving its visual quality.

However, an equalized histogram does not aid in selecting a threshold for the image. Gonzalez and Fittes [15] have extended this approach, and have suggested that it is possible to directly specify a histogram and then transform the image into one whose histogram has that desirable distribution. This thesis presents the application of this concept to threshold selection.

1.3.2. Direct histogram specification

The concept of direct histogram specification as explained in [1], [15] and [23] suggests that we can directly specify a histogram,and map it onto an image, to yield an image whose histogram closely resembles the desired shape. This suggests that if an image does not have a clearly bimodal histogram and selecting a threshold presents a problem, we could directly specify a bimodal histogram and after appropriately transforming the original image, obtain an image whose histogram is bimodal. Since a histogram is the probability distribution of the gray levels, we could specify a histogram consisting of two uni-modal distributions, sufficiently far apart on the intensity axis. Each distribution would then correspond to the object and background populations respectively.

This technique can also be extended for purposes of multi-level thresholding. In situations where the peaks and valleys in the original histogram are not clearly defined, a transformation can be applied on the histogram to appropriately segment it into distinctive sub-populations. In this case too, the histogram specified can be assumed to comprise of three or four uni-modal distributions, depending upon the number of thresholds desired. These distributions should be such that the valley between any two peaks is clearly defined.

It follows from the above discussion that the task of selecting a good threshold in an image can be greatly facilitated by transforming a histogram, using the direct histogram specification technique.

Thresholding is one form of extracting features from an image by segmenting it into two classes, depending upon the gray level of each pixel. However, there are situations where the exact shape and size of the objects is of primary concern. For such situations, there are other techniques of feature extraction, as described in the following section.

1.4. Feature Extraction

It has been mentioned earlier that thresholding the image amounts to extracting features from it, based on the properties of each pixel. Depending upon the intensity level of the pixel, it is classified as an object point or a background point. An alternative to this technique could be the extraction of features based upon local regional properties. In this way, instead of classifying each point to a region, we classify local regional properties. This increases the dimension of the feature vector used for segmenting an image in terms of its objects and backgrounds. One widely accepted approach of image segmentation based on classification of regional properties is template matching.

1.4.1. Template matching

The approach of template matching for purposes of image segmentation has been adopted widely due to its simplicity [1]. This method is based on detecting transitions in gray levels between regions. The various regions are usually characterized by the differences in their gray level content. However, there are other features, namely texture difference [17] and colour difference, which can be used to classify regions.

Template matching is used to identify certain characteristics in an image. A template can be defined, in context to digital image processing applications, as an array designed to detect some invariant regional property [1]. If we have some a priori information about the shape and size of the feature being looked for, a template could be designed to detect that feature. Otherwise, templates designed for various features of interest could be used to extract them.

Template matching can also be considered as correlating the template pattern with the image. At all points where the pattern matches closely to the image, the response of the template will be maximum,thereby indicating a high correlation between the image characteristics and the template pattern at that point.

There are other applications of image processing where it is required to find the boundary between the objects and the background. This information could be very useful in determining the size of the objects. Since a thresholded image comprises of just two levels, following the borders of the objects would yield the desired boundary.

1.4.2. Boundary following

One application of image processing is in the quality control of manufactured parts [19]. Here, it is required to locate the defects in a particular component, and to determine whether the defect is tolerable or not. This is done by taking the image of the part and thresholding it. If the threshold scheme adopted is an efficient one, the defects on the object, which could stem from a dent or a scratch on the surface, would appear as a dark portion on the white object. This is because the light reflected off a dent or a scratch while taking the image would be less as compared to that reflected off the surface of the object. These dark portions can be detected by using a boundary following algorithm, which would follow the edge between the object and the defect. In this way, we could also obtain the size of the defect. Boundary following would also aid in detecting a fault on the edge of an object. If there is a chipped edge on the object, it could be detected while following the borders [19].

Another way of locating faults could be to remove the inherent features in an image and whatever remains would be the fault. Obviously, this would require a previous knowledge of what the image looks like. Moreover, one would need to know the exact size and shape of the inherent features. If this information was available, one could use template matching to remove the features. A template could be designed according to the exact shape and size of the feature, which would detect and remove it.

There are some shortcomings, however, in using template matching for fault detection. Firstly, the size of the inherent feature in the image may make it impractical to implement a template of that size. Secondly, the shape and size of the feature may vary from part to part. Even if it varies slightly, the response of the template could change and thus cause problems. However, template matching could be used to identify a certain known portion of the feature, and then one could follow the borders of that feature to identify it completely. The feature thus detected would be inherent in the image of every part, and after removing it from the image, whatever remains as black would be the defects. In order to identify the defects, one would simply need to count the black pixels inside the object.

1.5. Problem Statement

The problem of threshold selection in cases where the histogram of an image is not bimodal is considered in this thesis. Given the desirable distribution of a histogram for purposes of binary level thresholding, it is required to use a gray level transformation function which maps the gray levels of an image into itself, such that the output image obtained by applying this transformation function to each pixel in the original image would yield the specified distribution. This approach is also extended for purposes of multi-level thresholding.

This thesis also considers the problem of fault detection in manufactured parts. It is required to investigate the possible use of template matching for fault detection, considering its simplicity of operation when compared to standard boundary.following algorithms.

1.6. Thesis Organization

In Chapter II, the theory beyond histogram modification is presented. The techniques of histogram equalization and directly specifying a histogram for an image have been discussed.

Chapter III relates to an explanation of image thresholding. Both single-level and multi-level thresholding techniques are discussed. The application of the direct histogram specification technique for purposes of single and multi-level thresholding has been described.

In Chapter IV, the use of template matching for feature extraction has been discussed. In addition, an algorithm for boundary following has been explained. Finally, a problem oriented example has been considered where the techniques of template matching and boundary following have been used for purposes of fault detection in manufactured parts.

To conclude, Chapter V presents a summary of the research conducted for this thesis, and the conclusions obtained from it.

CHAPTER II

HISTOGRAM MODIFICATION

A histogram of gray levels provides an entire description about the appearance of an image. Since a histogram is the probability distribution of the gray levels, it can provide useful information about the gray level ranges occupied by the "objects" or the background in an image. This information is often of particular importance in image enhancement.

2.1. Construction Of A Histogram

As stated earlier, a histogram gives us the probability distribution of the gray levels in an image. If we plot the number of occurrences versus the gray levels in an image, the resulting graph is called a histogram of gray level content. Mathematically, this can be stated as

$$p_r(r_k) = \frac{n_k}{n} \qquad (2.1\text{-}1)$$

$0 \le r_k \le L-1$

$k = 0, 1, \ldots, L-1$

where

r_k - the k-th intensity level

$p_r(r_k)$ - probability of occurrence of the k-th gray level

n_k - number of times the k-th gray level appears in the image

n - total number of pixels in the image

L - total number of levels in the image

A plot of $p_r(r_k)$ vs r_k is then referred to as a histogram of gray levels in an image. A listing of the computer program for plotting a histogram is provided in Appendix A.

It has been shown in various papers that useful enhancement results can be achieved by modifying the histogram of gray levels in an image. A considerable improvement in the contrast of an image can be achieved by transforming a histogram into one with a more or less uniform distribution. This process of "equalizing" the histogram is explained in the following article.

2.2 Gray Level Equalization [1], [15], [16]:

It should be established at the onset that the concepts being introduced in this article and the next are formulated in the continuous domain. However, since we need the discrete version of these concepts to process images on a digital computer, their equivalence is ascertained in the next chapter.

The object is to transform the original image into one whose histogram has an almost uniform distribution. Let 'r' and 's' be the normalized gray levels in the original and equalized image respectively. The levels have been normalized for simplicity.

$$0 \leq r \leq 1 \quad ; \quad 0 \leq s \leq 1 \qquad (2.2\text{-}1)$$

The level 0 represents black and the level 1 represents white in the gray scale. The 's' levels of the equalized image are obtained after some transformation of the form

$$s = T(r) \qquad (2.2\text{-}2)$$

This would produce a level $s(r)$ for every pixel value in the original image. There are some constraints on the transformation function and it is assumed to satisfy the following conditions:

(i) In order to preserve the order from black to white in the gray scale, the transformation function $T(r)$ should be single-valued and strictly monotonic in the interval $0 \leq r \leq 1$.

(ii) In order to ensure that during the transformation, we do not exceed the allowed range of pixel values we need

$$0 \leq T(r) \leq 1 \quad \text{for} \quad 0 \leq r \leq 1$$

Consider the transformation function of the form shown in Fig.2.1. A transformation of this sort satisfies the two conditions imposed on it.

The inverse transformation from 's' back to 'r' can be represented as

$$r = T^{-1}(s) \quad 0 \leq s \leq 1 \tag{2.2-3}$$

where it is assumed that $T^{-1}(s)$ also satisfies the conditions (i) and (ii) previously imposed, with respect to the variable s.

Let $p_r(r)$ and $p_s(s)$ be the probability density functions of the original and transformed image respectively. It has been emphasized that these density functions could relay a great deal of information about the general characteristics of an image. It has been shown from probability theory that if $p_r(r)$ and $T(r)$ are known, and $T^{-1}(s)$ satisfies condition (i), then the probability density function of the transformed gray levels is given by the relation

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r = T^{-1}(s)} \tag{2.2-4}$$

It has been shown in [1], [5], [15], [16] that it is possible to transform the histogram of an image to one having a uniform probability distribution using the following transformation function:

$$s = T(r) = \int_0^r p_r(\omega)\, d\omega \qquad 0 \leq r \leq 1 \tag{2.2-5}$$

This transformation function, which is equivalent to the cumulative distribution function of r, produces an image where $p_s(s)$ is a uniform density in the interval $0 \leq s \leq 1$. Before proving this, one must

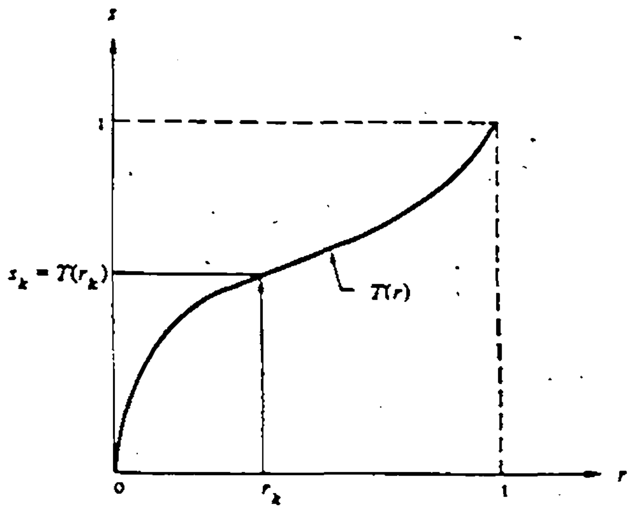


Figure 2.1   A gray level transformation function

appreciate the fact that this transformation function satisfies the conditions imposed on it, since the cumulative distribution function increases montonically as a function of 'r' within the range of values the 'r' levels could take. In order to prove that this transformation indeed does produce a uniform distribution, let us focus our attention on Eq.(2.2-5). The variable 'ω' is a dummy variable of integration, and by taking the derivative of 's' with respect to 'r', we could have from Eq.(2.2-5)

$$\frac{ds}{dr} = p_r(r) \tag{2.2-6}$$

Substituting this value in Eq.(2.2-4), yields

$$\begin{aligned} p_s(s) &= \left[p_r(r) \cdot \frac{1}{p_r(r)} \right]_{r = T^{-1}(s)} \\ &= [\ 1\ ]_{r = T^{-1}(s)} \\ &= 1 \quad 0 \le s \le 1 \end{aligned} \tag{2.2-7}$$

This is a uniform distribution within the interval of definition of the transformed variable S. It should be noted at this stage that this result does not depend upon the density inside the integral of Eq.(2.2-5). This implies that if the transformation function described in Eq.(2.2-5) is performed on any distribution — ideally speaking, the result obtained would be the same, i.e., a uniform density within the given interval. This observation is brought to use in transforming an image such that its histogram acquires a desirable shape.

2.3. Direct Histogram Specification [1], [15]:

The technique of gray level equalization explained in the previous section is somewhat limited in its application because it only generates a histogram with a uniform distribution, thereby improving the contrast of the image. In many cases, this may not be the sole objective. However, this concept could be utilized for purposes of interactive image enhancement, as shown in [1], [15].

It is desirable in certain cases that some ranges of gray levels in an image be highlighted. In order to meet this end, it is possible to directly specify a histogram desired, and transform the original image into one, whose histogram has the desirable shape. The shape of the desired histogram could vary, depending upon the required application.

The procedure of directly specifying a histogram and mapping it onto an image to obtain a transformed image having a histogram of the specified distribution, can be divided into three steps. The first step is to equalize the levels of the original image using the transformation function shown in Eq.(2.2-5). The next step would be to equalize the levels of the specified probability distribution. Let $p_z(z)$ be the probability density function of the desired image, in other words, the specified histogram is a plot between $p_z(z)$ and 'z' levels. The transformation function in this case is the same as the one used for the original image.

$$v = G(z) = \int_o^z p_z(\omega)\, d\omega \qquad (2.3\text{-}1)$$

$$0 \leq z \leq 1$$

'v' denotes the equalized levels of the specified histogram. As proven in Eq.(2.2-7), $p_v(v)$ — the probability distribution of the equalized (specified) histogram $\bar{v}$ would be a uniform density in the interval $0 \leq v \leq 1$.

From Eq.(2.3-1), the inverse process, $z = G^{-1}(v)$ should yield back the desired levels, but this is purely hypothetical since the 'z' levels are exactly what we are trying to obtain. In this case, we are just dealing with the desirable shape of the histogram, and the image does not exist. Therefore, since we do not have the 'z' levels, the inverse process $z = G^{-1}(v)$ would not really carry a meaning. We could obtain a transformed image having the desirable histogram, if we could obtain the 'z' levels.

The preceding argument necessitates the next step in our line of action. We know from Eq.(2.2-7) that the probability distributions $p_v(v)$ and $p_s(s)$ are identical, since they are both uniform densities independent of the density inside the integral in either case. This means that instead of using the 'v' levels obtained from Eq.(2.3-1) in the inverse process, if we use the 's' levels obtained from Eq.(2.2-5), the result would remain the same. In other words, for Eq.(2.3-1), the inverse process $z = G^{-1}(s)$ should yield the same result as using $z = G^{-1}(v)$. The 'z' levels obtained in this manner would be the levels of the desired image — the image whose histogram has the desired distribution.

It follows from the above discussion, that it is possible to specify the desired distribution in a histogram, and transform the original image into one whose histogram closely resembles the specified shape. Thus, whatever our purpose is of transforming the histogram of an image into a desirable shape, it can be achieved by the direct histogram specification technique described in this chapter. The application of this concept on different images is explained in the next chapter.

CHAPTER III

IMAGE THRESHOLDING

Image thresholding is a special case of pattern classification in which a one-dimensional feature space is used, the feature being the gray level of the pixel. If an image consists of dark objects on a light background, or vice versa, the objects can be separated from the background by thresholding the image. By choosing an appropriate threshold level, we could assign each pixel in the image into one of the two classes, light or dark, depending upon whether the gray level of the pixel is below or above the specified threshold level. Often, the choice of selecting an 'appropriate' threshold level poses a problem. The choice has to be such that a negligible amount of information is lost. It might be possible in some cases, that segmenting the image into just two classes, could result in a considerable loss of information. In order to cope with such situations, it may be necessary to choose more than one threshold, and segment the image into more than two classes to adequately describe the picture.

It follows therefore, that selecting a good threshold is of paramount importance in most image processing applications. The techniques of thresholding an image at a single level or at 'multi-levels' are discussed in the following sections.

3.1. Binary Level Thresholding

The process of transforming an image into two levels, black and white, by using an appropriate threshold level, is referred to as binary level thresholding. If the average gray level of the objects is significantly different from that of the background, threshold selection would be fairly straightforward. However, in most situations this is not the case. It is, therefore, not possible to arbitrarily choose a threshold, since the correct threshold may not be the same for all pictures. It is sometimes possible, however, to choose a good threshold for each image, automatically, by examining the histogram of its gray levels. If we find two peaks on the histogram of an image, it would be reasonable to choose a threshold that separates these peaks. Consider for example, a histogram of the shape shown in Fig.3.1. In this case, we would choose a threshold level 'T', at the bottom of the valley between the two peaks, since this threshold appears to separate the gray level population into two distinctive sub-populations, corresponding to the object and background populations. All pixels in the image below the selected threshold level would be rendered black, while those above that level would become white — thus reducing the image to a binary one.

An image would have a bimodal histogram if we make certain assumptions about it, such as:

(a) The given image consists of objects on a background, where the probability distribution of gray levels for any small region of the picture consisting solely of the object on the background is uni-modal.

(b) The gray levels of adjacent points interior to the object, or to the background, are highly correlated, while at the boundary between them, adjacent points differ significantly in gray level.

Under these conditions, the gray level histogram of an image would primarily consist of two uni-modal histograms corresponding to the object and background populations, respectively. If the average gray level of these populations is significantly different from one another, we would have the two distributions spread fairly far apart on the

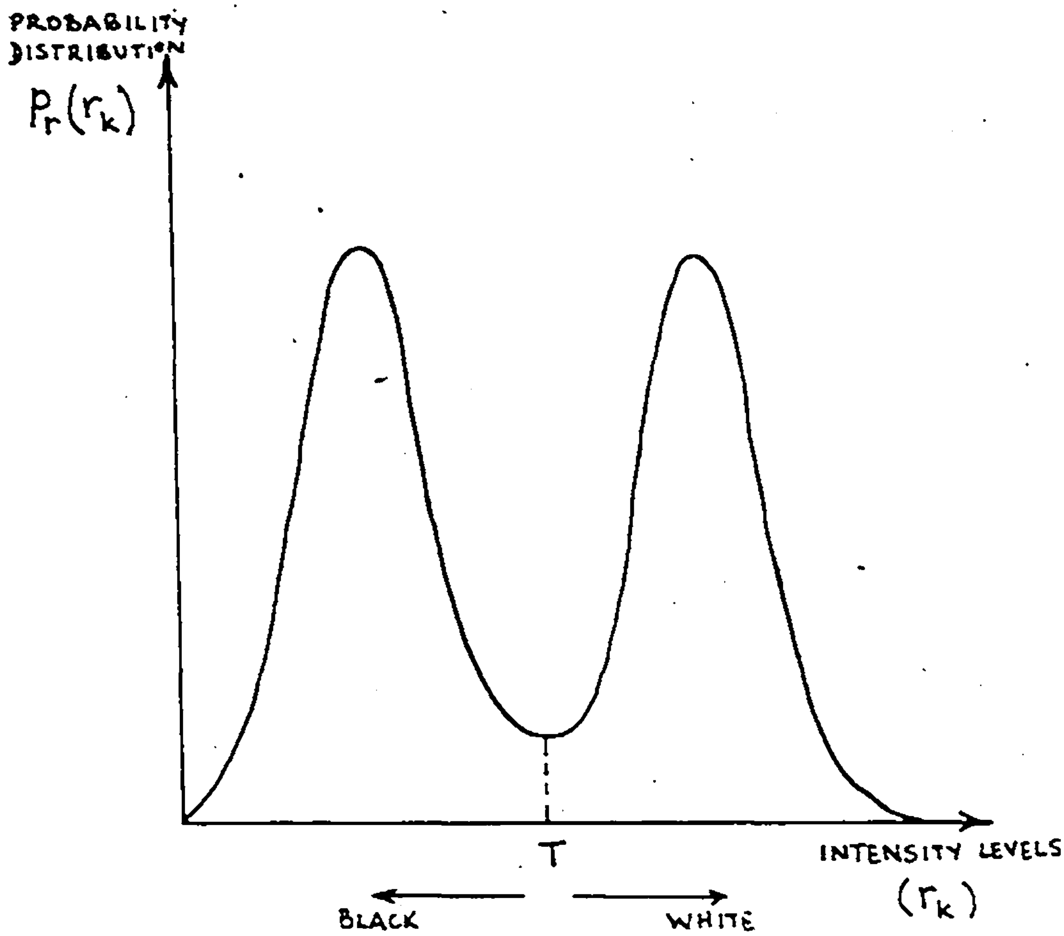


Figure 3.1 A bimodal histogram

intensity axis of the histogram. Moreover, if the object population and the boundary population take on a small range of intensity values, and the two populations are comparable in size, the histogram of the image will be bimodal, like the one shown in Fig.3.1. Otherwise, the resulting histogram may be a mixture of two uni-modal distributions, but it may be difficult to separate the two. For example, if the object population is much greater than the background, or if they lie very close to each other, it would be very difficult to select one level that would separate the two populations into different classes. Therefore, if the histogram of the image is clearly bimodal, the selection of one threshold level for classifying the image in terms of its objects and background would not pose a problem.

(i) Smoothing plus thresholding:

This method suggests the use of an operation that would 'smooth' the image by removing noise, etc. from it [9]. This 'smoothing' operation would then aid in selecting a good threshold for the image.

(ii) Sharpening plus thresholding:

In this technique, one could use an operation that yields high values at all edges in the image and low elsewhere. This method could greatly aid in separating the object from the background, thus facilitating the selection of a good threshold [6], [20], [21], [22].

(iii) Matched filtering plus thresholding:

In this technique, one needs to have some information about the shape and size of the objects in an image. This technique suggests the use of a filtering operation on an image with a certain template pattern. This template acquires the shape of the characteristic being looked for in an image. Thresholding the image would single out points where the image matches the pattern closely.

One widely accepted method deployed for purposes of threshold selection, when the histogram of the image is not clearly bimodal, is the variable thresholding scheme. This technique was developed by Chow and Kaneko [14], for use in the detection of the heart region on chest X-rays. This method was applied to some general images and it proved to give useful results.

3.1.1. Variable thresholding

This technique suggests that instead of using one threshold level for the entire image, we could have different thresholds for different sections of the image. This means that instead of selecting a threshold level for the image based on its global characteristics, we could set different thresholds according to the local characteristics in the image. Based on this idea, Chow and Kaneko [14] have suggested to segment the image into smaller regions and select an appropriate threshold for each region. The segmentation scheme is illustrated in Fig.3.2. The entire image is divided into 7 x 7 regions, with each subsequent region including 50% overlap with the previous one. A histogram is then computed for each region. For regions having bimodal histograms, thresholds can be set automatically at the bottom of the valley between the two peaks. However, for regions which do not have bimodal histograms, thresholds are computed as a weighted average of the neighbouring thresholds. Thus, the entire image is thresholded at different levels, thereby yielding a binary level image.

An example of the successful results achieved by using a variable thresholding scheme is illustrated in Fig.3.3. Fig.3.3(a) shows the original image of a lady. This is a 128 x 128 image having 256 gray levels. Fig.3.3(b) shows the thresholded image obtained after variable thresholding. This image has just two levels, black and white, and we can see that there is a negligible loss of information.

Although the method of variable thresholding seems to give useful results, it does not lend itself to applications where speed of operation is of the essence. This gives rise to the need for investigating simpler and faster techniques of threshold selection.

3.1.2. Image transformation

As mentioned earlier, if the histogram of an image has two peaks close together, or very unequal in size, it would be difficult to detect the valley between them. Furthermore, if the histogram has more than two peaks, it would be impossible to detect a valley, as there will be more than one valley. Therefore in these cases, the selection of a good threshold level for purposes of binary level thresholding would be very difficult, if not impossible.

In order to overcome this problem, one needs to transform the image to one with a histogram that is clearly bimodal. In this connection, an attempt has been made to utilize the concept of direct histogram specification towards its use in the problem of selecting an optimal threshold for the image. This section explains the working of this technique, using different types of images as examples.

Our objective therefore, is to transform an image whose histogram is not suitable for threshold selection into another image whose histogram is clearly bimodal, so that the selection of a threshold does not present a problem. Consider the image shown in Fig.3.4. Fig.3.4(a) is the image of a strain gauge. The 'object' part of this image is the conductor which has a slightly higher intensity level than its insulator

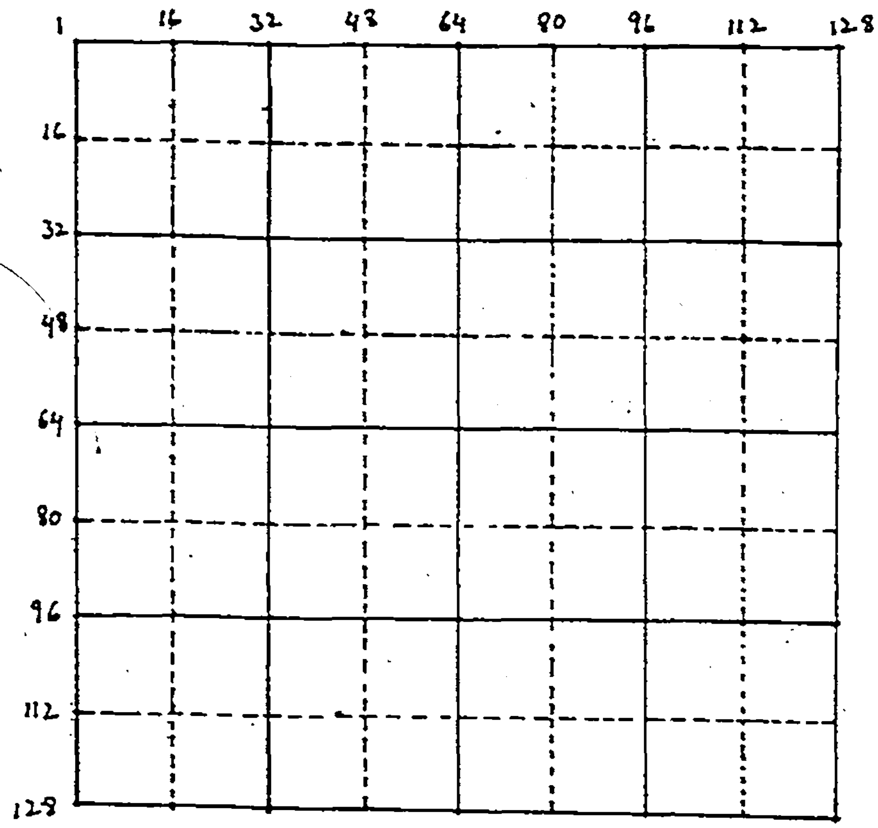


Figure 3.2 Picture segmentation scheme for variable thresholding

(a)

(b)

Figure 3.3 (a) Original image (b) Image after variable thresholding background. It is required to locate the defects in this particular component, and therefore it is imperative that we first obtain a good thresholded image. Fig.3.4(b) shows the histogram of the image. It is clear from the shape of this histogram that it is not possible to select a good threshold level for this image. Therefore, we would need to transform this image for this purpose. The first step towards achieving this objective would be to equalize the levels of the original image.

(a) Equalizing Levels of Original Image

Referring back to Eq.2.2-5, we could use the cumulative distribution function of the intensity levels to transform the original histogram into an almost uniform distribution. However, for the practical implementation of the concepts explained in Chapter II, we would have to approximate them for the discrete domain. Therefore, the results obtained would be an approximate to the ideal case. Eq.2.2-5 can be written in the discrete domain as

$$\begin{aligned} s_k = T(r_k) &= \sum_{j=0}^{k} p_r(r_j) \\ &= \sum_{j=0}^{k} \frac{n_j}{n} \end{aligned} \qquad (3.1\text{-}1)$$

for

$0 \leq r_k \leq 1$

$k = 0, 1, \ldots, L-1$

where

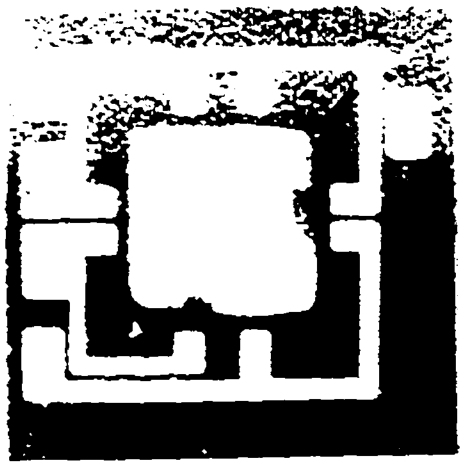
r_k the gray levels are normalized for simplicity,

L is the total number of levels,

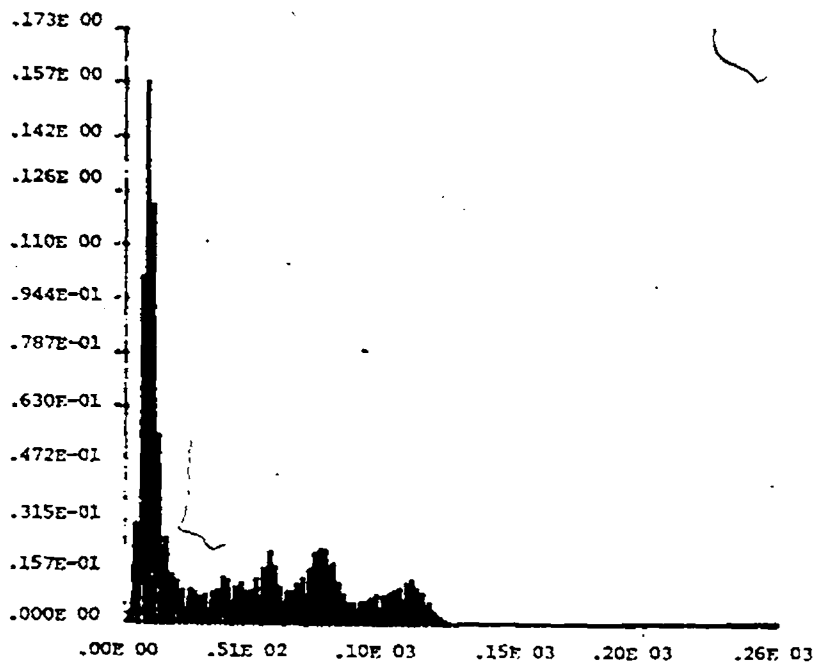
n_j is the number of times the j-th level appears in the image,

n is the total number of pixels in the image.

This produces a transformed level 's' for every level 'r' in the original



(a)



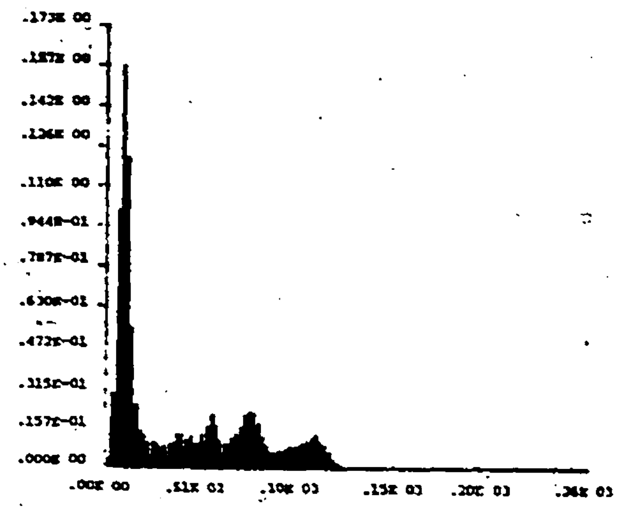
(b)

Figure 3.4 (a) Image of strain gauge (b) Histogram of image

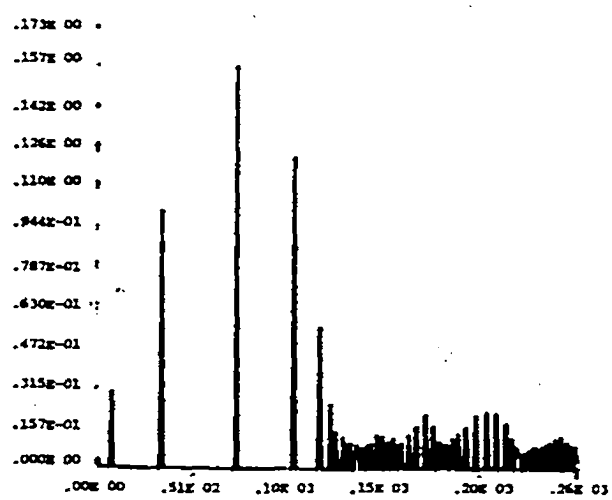
image. These transformed levels are then assigned to their closest valid level thereby yielding a new image (with the original levels transformed), which has an almost uniform distribution. The effect of this transformation can be seen in Fig.3.5.

Fig.3.5(a) shows the original histogram of the image shown in Fig. 3.5(c). After the equalization process, we can see by the histogram of Fig.3.5(b) that we have achieved an almost uniform distribution. The reason this histogram is not exactly uniform is that the image size being used was 128 x 128 pixels. If the image is sampled further into a 256 x 256 size, the result would be a closer approximation to a uniform histogram. The effect of this process on the original image can be seen in Fig.3.5(d). It is noticed that the contrast of the original image has improved, since the dynamic range of its pixels has been increased. A listing of the computer program used to perform histogram equalization is provided in Appendix A.

The process of histogram equalization was carried out on several images. Consider the image of a cable connector in a lab shown in Fig.3.6(c). This is the image of a cable connector and an information plate on a piece of machine — the writing on the plate is not clear because of under sampling. Its histogram is shown in Fig.3.6(a). After the equalization process, we can see in Fig.3.6(b) that the resulting histogram has an almost uniform distribution. In this case also the histogram would have been closely approximating a uniform histogram if a 256 x 256 image were considered. The memory constraint of the computer prevented the implementation of this algorithm on an image of a larger size than 128 x 128. The improvement in the contrast of the original image is also clearly visible in the equalized image shown in Fig.3.6(d).



(a)



(b)

Figure 3.5 (a) Original histogram (b) Equalized histogram

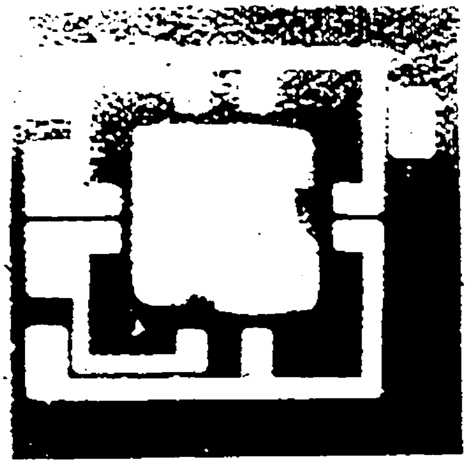
This observation can also be appreciated in the image of a girl shown in Fig.3.7. Fig.3.7(a) is the image and Fig.3.7(b) is obtained after equalizing the histogram of the original image.

It follows from the above examples that equalizing the gray levels in an image would enhance its visual quality, viz-a-viz its contrast. However, improving the contrast of the image is not our ultimate goal, and we therefore have to perform further processing to suit our requirements.

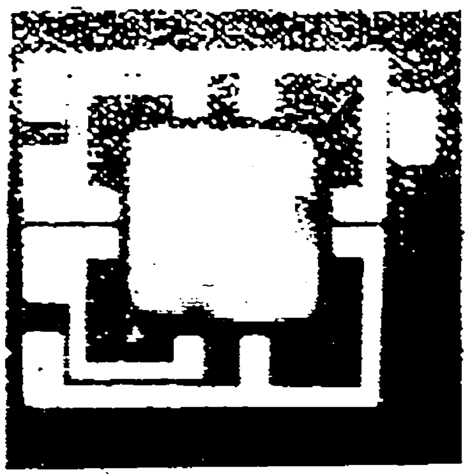
(b) Specifying Desirable Histogram

The next step is to specify the distribution desired in the histogram. Consider the histogram shown in Fig.3.8. This histogram gives the ideal case of the object and background populations being clearly divided into different gray level ranges. If we transform our original image into one whose histogram has this desired distribution, it would be very simple to select a threshold at the bottom of the valley separating the two populations.

The histogram of Fig. 3.8 was generated by combining two Gaussian distributions with sufficiently different means, so that the average gray levels of the populations be significantly far apart on the intensity axis. The standard deviation of both the distributions was taken to be the same, and relatively small so that the two distributions did not overlap. It should be mentioned here that the selection of means and standard deviation of the distribution was carried out purely on a trial and error basis. Further work in the area could be done towards establishing a criterion for automatically selecting the means and standard deviation for a desired histogram of any image.

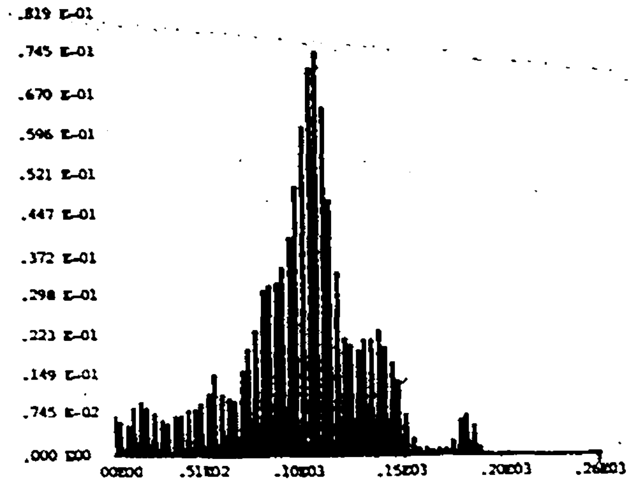


(c)

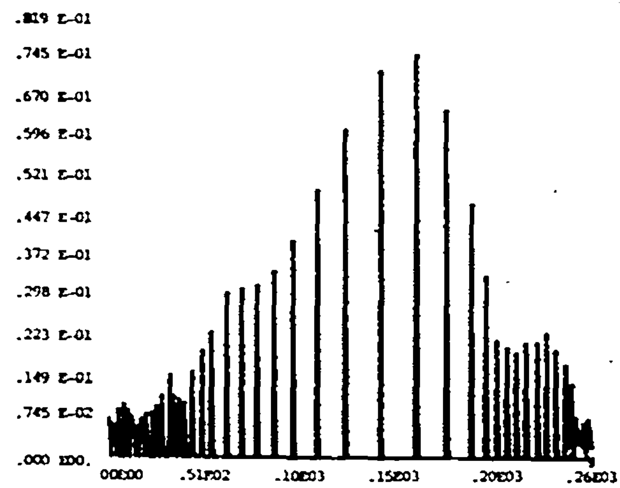


(d)

Figure 3.5 (c) Original image (d) Equalized image

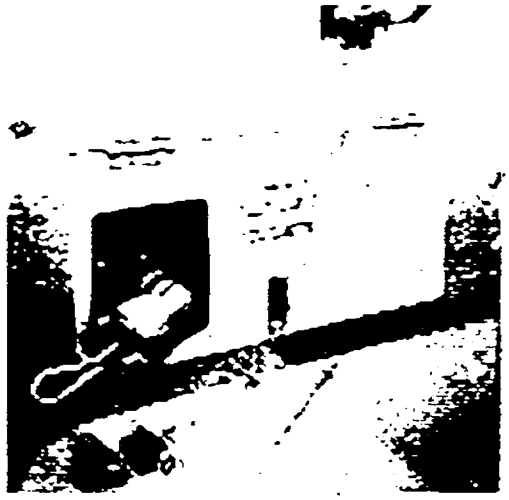


(a)

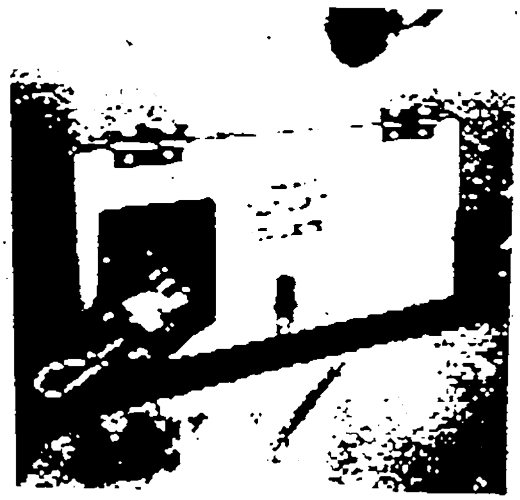


(b)

Figure 3.6 (a) Original histogram (b) Equalized histogram



(c)



(d)

Figure 3.6 (c) Original image (d) Equalized image

(a)

(b)

Figure 3.7 (a) Original image (b) Equalized image

Our objective therefore, is to transform an image into one whose histogram has the distribution shown in Fig.3.8.

(c) Mapping Histogram Onto Image

The next mode of action is to map the desired histogram onto an image. This would require the histogram equalization process to be carried out for the levels of the desired histogram. The transformation function would be the same as in Eq.3.1-1, except that it is now applied on the levels of the desired histogram. Let z_k be the levels of the desired histogram. The transformation would then be

$$v_k = G(z_k) = \sum_{j=0}^{k} p_z(z_j)$$

$$= \sum_{j=0}^{k} \frac{n_j}{n} \qquad (3.1\text{-}2)$$

for

$$0 \le z_k \le 1$$

$$k = 0, 1, \ldots, L-1$$

where all the symbols used have the same connotations as in Eq.3.1-1.

It has been explained in the previous chapter, that in order to obtain an image whose gray levels have the distribution of the desired histogram, we could use the inverse transformation

$$z_k = G^{-1}(s) \qquad (3.1\text{-}3)$$

Eq.3.1-3 suggests that we assign each of the transformed values (v_k) to the 's' level closest to it. (Remember that the 's' levels were obtained after equalizing the levels of the original image). We can therefore obtain the desired levels z_k using the inverse transformation of Eq.3.1-3. After appropriately transforming the levels, we can obtain an image given by the desired levels z_k. This image would then have a histogram whose distribution would closely resemble the specified distribution.

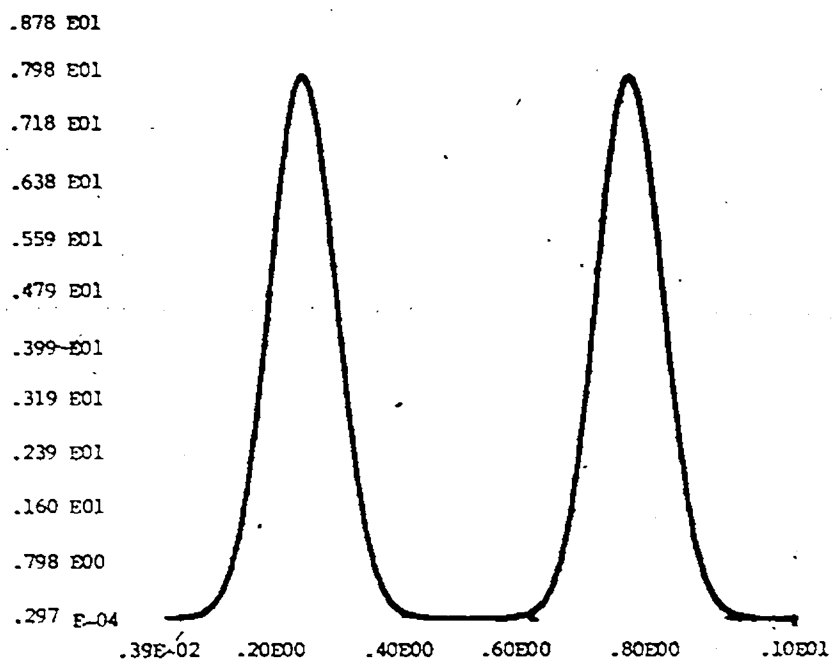


Figure 3.8 Bimodal histogram consisting of two Gaussian distributions

This technique has been applied to several images and the results obtained are illustrated. Fig.3.9 shows the result of this transformation at each step. Fig.3.9(a) is the histogram of the original image which in this case is the strain gauge of Fig.3.4. Fig.3.9(b) is the equalized version of the original histogram. The desired distribution is specified in Fig.3.9(c), and the resultant histogram obtained after the transformation is shown in Fig.3.9(d). It can be seen that although this distribution is not exactly the shape of the specified distribution, yet it has been transformed into two separate classes. The discrepancy between the specified and resultant histogram is due to the fact that this technique is guaranteed to yield exact results only in the continuous case. Here, since we are only approximating the concepts obtained for the continuous domain, the result would also approximate the ideal case. As the number of levels in the image decreases, the error between the specified and resulting histogram would tend to increase. Since the practical limitations of the system have to be borne in mind, one has to contend with the trade-offs required for such situations. In this case, however, the trade-off seems to be a reasonable one as selecting a threshold based on the resulting histogram is no longer a problem, and our objective is reasonably fulfilled.

Fig.3.10 shows the effect of thresholding the transformed image at a single level, thereby reducing the image to a binary one. The threshold level can be easily selected based on the histogram of Fig.3.9(d) it could be any level separating the two populations. Fig.3.10(a) is the original image on which the transformation was carried out, and Fig.3.10(b) is the thresholded image. It can be seen that the original image having 256 gray levels is most adequately described in just two

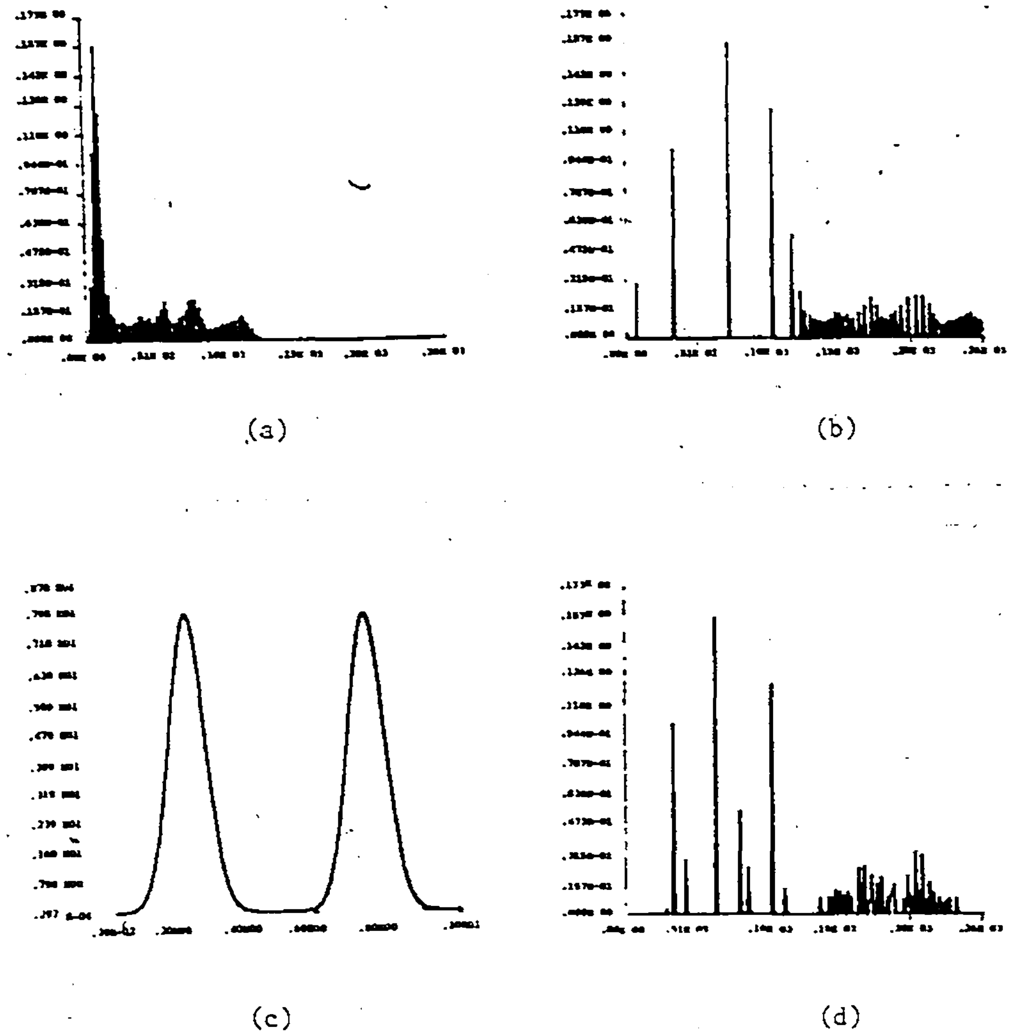


Figure 3.9 (a) Original histogram (b) Equalized histogram
(c) Desired histogram (d) Resulting histogram

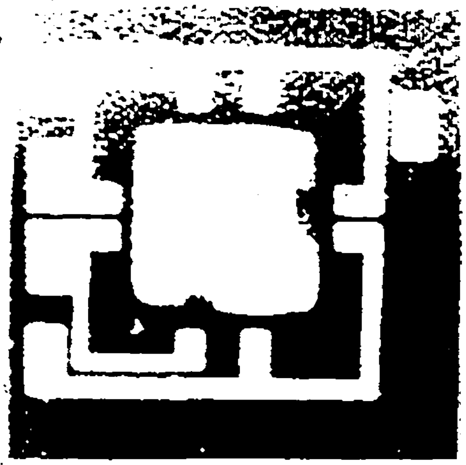
levels, with a negligible loss of information.

The effect of this transformation can also be noticed in the image of the cable connector of Fig.3.6(c). Fig.3.11(a) is the histogram of the original image. The histogram of Fig.3.11(b) is obtained after equalizing the levels of the original image. Fig.3.11(c) is the specified distribution and the resultant histogram is shown in Fig.3.11 (d). It can be appreciated that a threshold can easily be selected for this histogram.

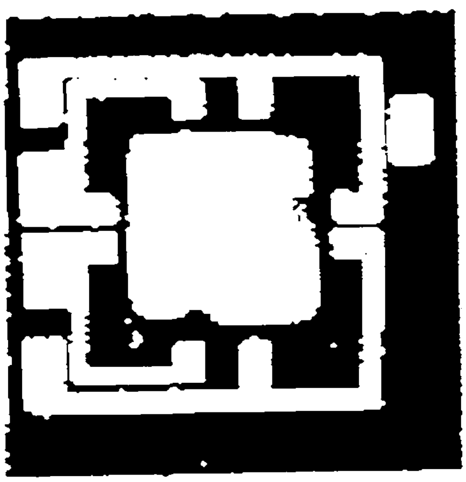
Fig.3.12(b) shows the effect of thresholding the original image at a single level, obtained from the histogram of Fig.3.11(d). The white portion at the top of the image results from the effect of non-uniform illumination on the object while sampling it. This effect could be removed from the image before transforming the histogram, by various means, e.g., homomorphic filtering [19].

Another image on which this transformation technique was applied is shown in Fig.3.13(a). The thresholded image is shown in Fig.3.13(b). Although it is difficult to select one threshold level for this image, a satisfactory result is seen to be achieved from this transformation.

It can be inferred from the previous discussion, that it is possible to transform an image whose histogram is not clearly bimodal, into one whose histogram has a desirable shape which can be specified. This approach was found to yield useful results for purposes of binary level thresholding, however, there could be situations where selecting just a single threshold level would result in a loss of desirable information. For these situations, one needs to investigate the idea of thresholding the image at more than just a single level.



(a)



(b)

Figure 3.10 (a) Original image (b) Image thresholded after bimodal transformation

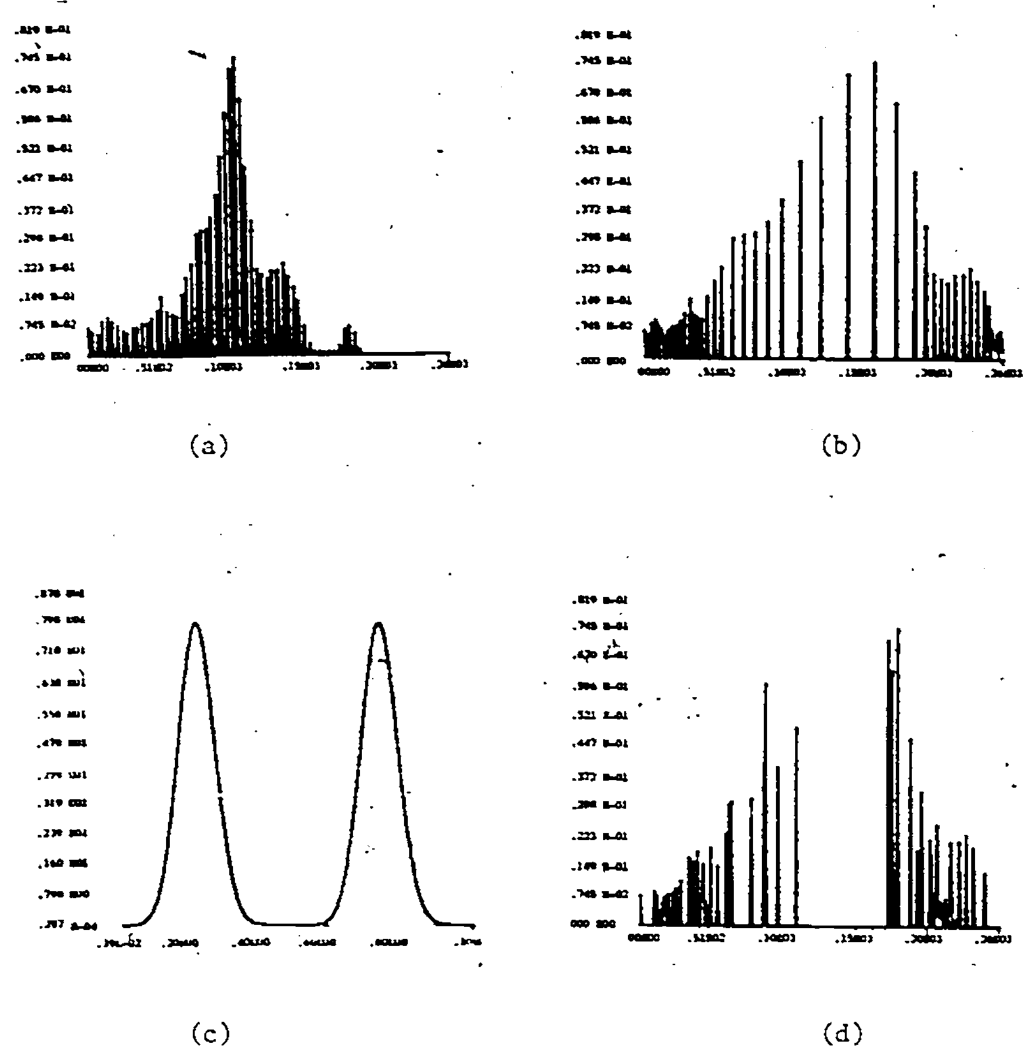
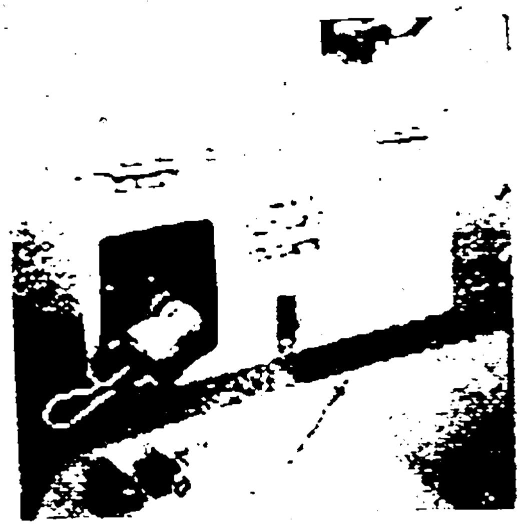


Figure 3.11 (a) Original histogram (b) Equalized histogram
(c) Desired histogram (d) Resulting histogram

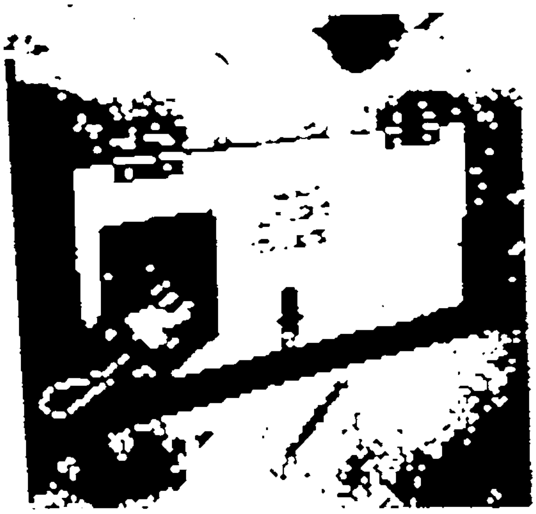
3.2. Multi-Level Thresholding

Consider an image whose histogram has the distribution shown in Fig.3.14. It would be very difficult, if not totally impossible, to select one threshold level for this image, such that a negligible amount of information is lost. It is necessary therefore, that we choose more than one threshold level to extract the objects from the background. For the case of the histogram of Fig.3.14, we would need three threshold levels, T_1, T_2 and T_3, to adequately describe the image. This would mean that the original image will be represented by more than just two levels. All levels lying between 0 and T_1 would have a certain value; levels lying between T_1 and T_2 would have another value, and so on. Thus, for the case where we have three threshold levels, the image after thresholding will contain four levels. This approach of thresholding the image at more than just a single level is called multi-level thresholding.

As in the case of binary level thresholding, the shape of the histogram is mostly not conducive to multi-level thresholding. It is not always easy to separate the various sub-populations. One commonly encountered difficulty is that some ranges of gray levels may occur a significantly greater number of times compared to the rest, so that when plotting the entire histogram, these gray level ranges play a dominant role and since the scale on the axis is uniform, observing the distribution of the lesser occuring gray levels becomes a problem. In order to overcome this, a section of the histogram is considered at a time, and a bimodality is searched for. If a bimodality is found [4], a threshold level is selected appropriately for that section. All pixels having intensity value less than this threshold are assigned one gray level.



(a)



(b)

Figure 3.12 (a) Original image (b) Image thresholded after bimodal transformation

(a)

(b)

Figure 3.13 (a) Original image (b) Image thresholded after bimodal transformation

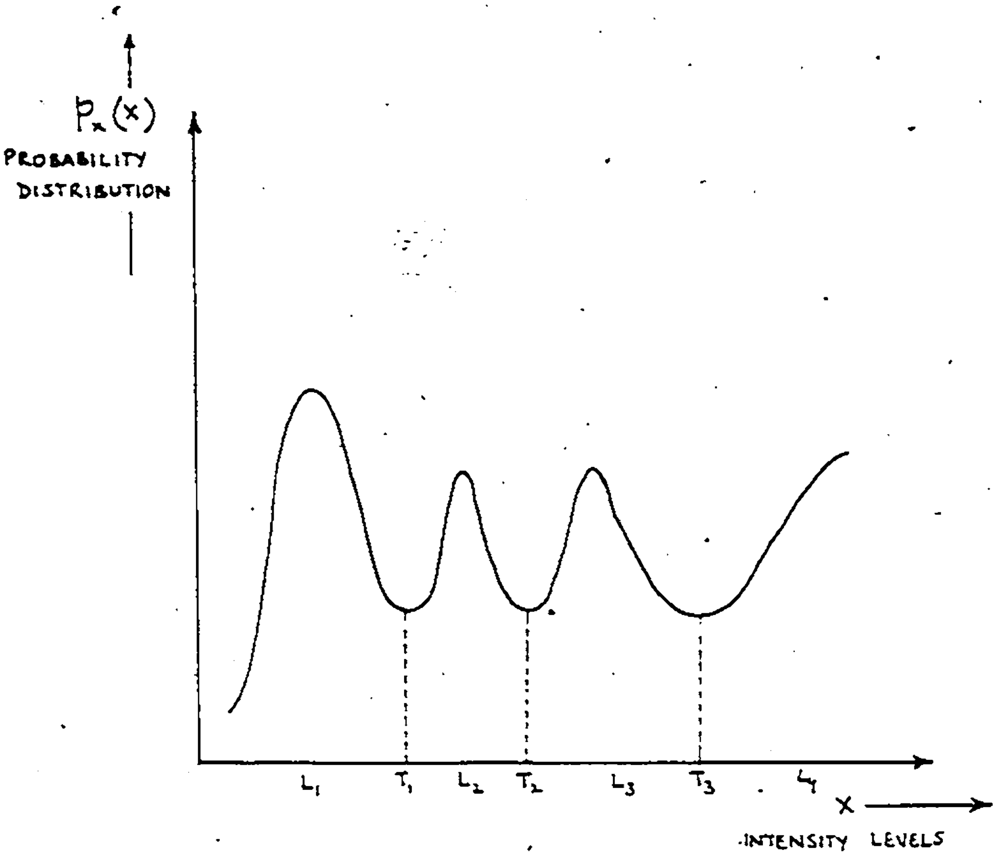


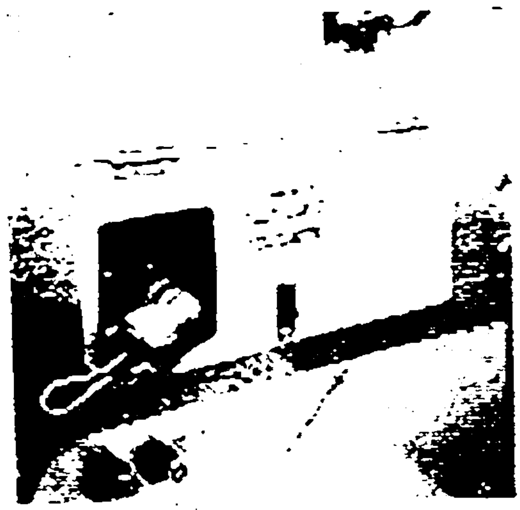
Figure 3.14 A multi-modal histogram

From this threshold, another section of the histogram is subjected to the bimodality test. If a threshold can be selected for this section, all pixels with intensity value lying between the first and second thresholds are assigned another gray level. This process is carried on until the entire histogram has been scanned to check local bimodality. Thus an image having a histogram of the shape of Fig.3.14 can be appropriately represented in four levels L_1, L_2, L_3 and L_4.

The process of selecting thresholds by sectioning the histogram would work if a bimodality is detected in each section being considered. However, in most instances, this is not the case. If a small section of the histogram is being considered, we might not encounter a bimodality. If the size of the section is increased, we might come across a situation where there are more than two peaks, or the valley between the two peaks may not be very distinguishable. In such cases, this thresholding scheme could result in a loss of information. This is illustrated in the thresholded image of the cable connector shown in Fig.3.15. Fig.3.15(b) was obtained after thresholding the original image at three levels, after sectioning the histogram. It can be seen that this image misses out on some detail from the original image.

The limitation of selecting thresholds after sectioning the histogram prompted the need to investigate the idea of direct histogram specification for purposes of multi-level thresholding. This technique was used on various images and the results proved to be quite encouraging.

The only difference in the case of multi-level thresholding with that described in Section 3.1.2, is in the specification of the desirable histogram. Since binary level thresholding desires a bimodal histogram, the combination of two Gaussian distributions was specified. However,



(a)

(b)

Figure 3,15 (a) Original image (b) Multi-level thresholded image

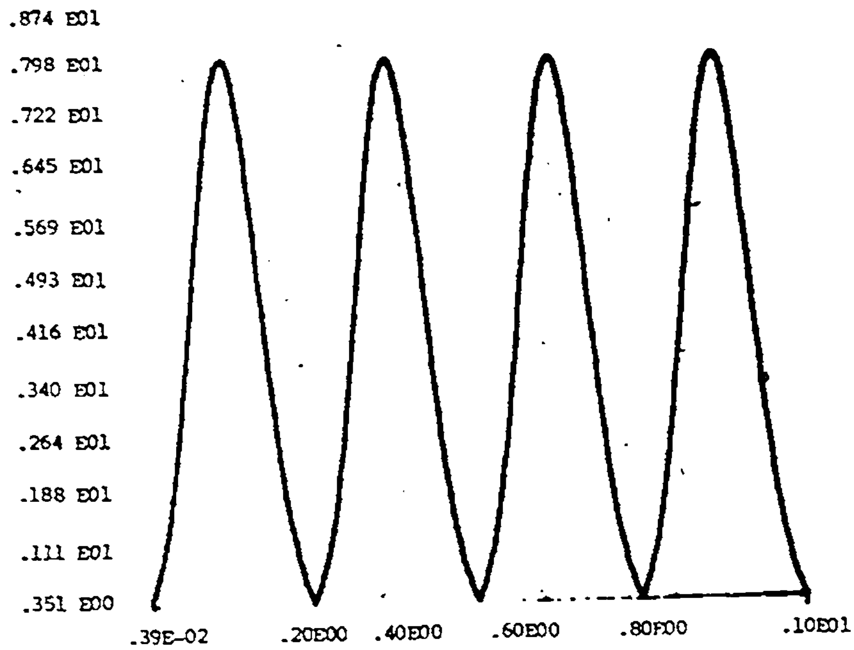
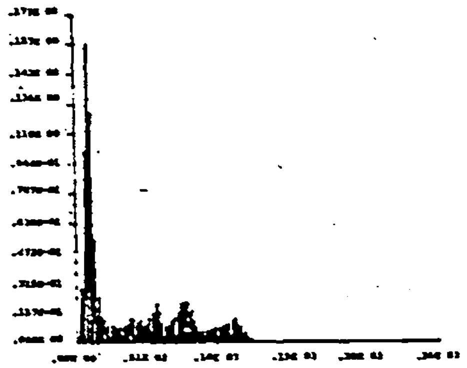


Figure 3.16 A multi-modal histogram consisting of four Gaussian distributions

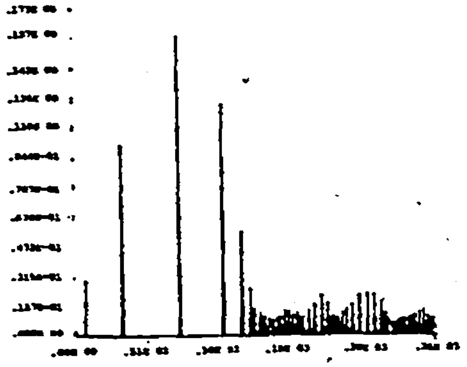
in this case, a multi-modal histogram is desired, with each of the peaks being separated from the other by a sufficient amount. Therefore, a combination of four Gaussian distributions with different means and same standard deviation, of the form shown in Fig.3.16 is specified. As in the case of binary level thresholding, the means and standard deviation of the specified distribution was arbitrarily chosen. The rest of the process is exactly the same as described in Section 3.1.2.

The result of this transformation on the histogram of the image of the strain gauge is illustrated in Fig.3.17. Fig.3.17(a) is the histogram of the original image. After equalization, the histogram has an almost uniform distribution shown in Fig.3.17(b). Fig.3.17(c) is the desired histogram we specified, and the resultant histogram is shown in Fig.3.17(d). It can be seen that the histogram has been transformed into four different sub-populations, and thresholds can be selected appropriately to separate these sub-populations. Fig.3.18(b) shows the image of the strain gauge represented in four levels after being thresholded at three different levels.

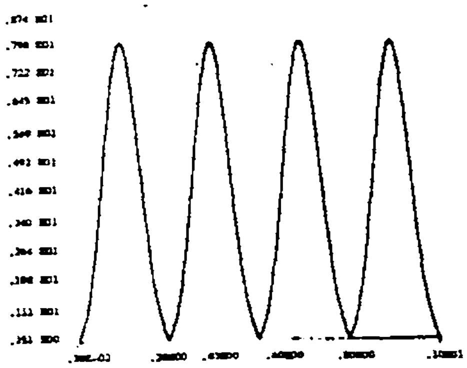
The use of multi-level thresholding cannot be fully appreciated by the image of Fig.3.18(b). However, consider another image of the strain gauge shown in Fig.3.19(a). This image consists of two backgrounds so to say, one being the insulator background having a low average intensity level and the other being the background at the outer borders of the image having a relatively higher average intensity level. Fig.3.19 (b) shows this image thresholded at a single level, and we can see that all the desirable information in the image (the conductor part) is lost. However, thresholding the image at multi-levels would yield the desired information, as shown in Fig.3.19(d).



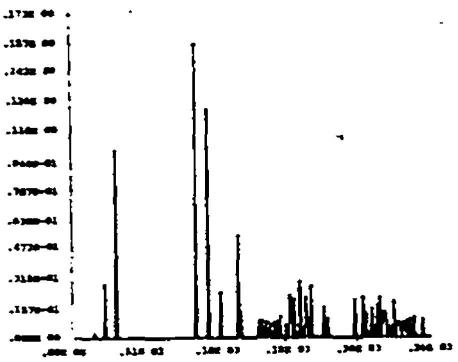
(a)



(b)

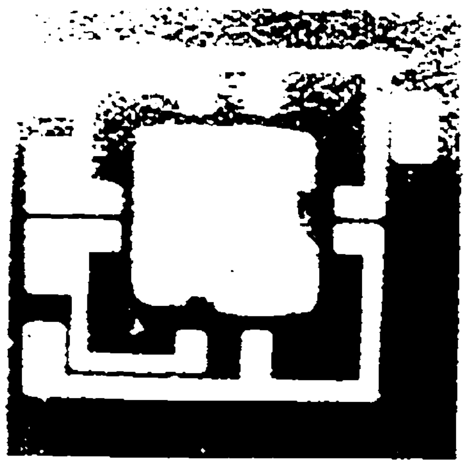


(c)

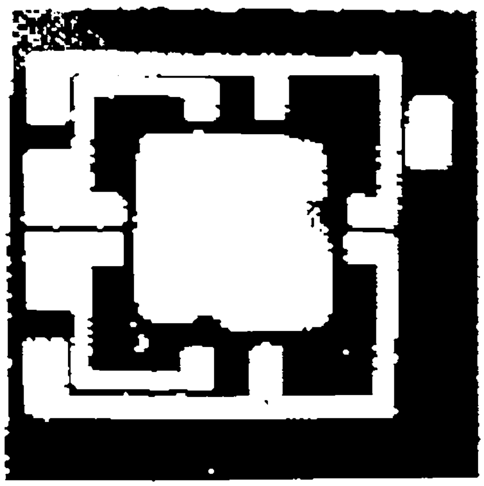


(d)

Figure 3.17 (a) Original histogram (b) Equalized histogram
(c) Desired histogram (d) Resulting histogram

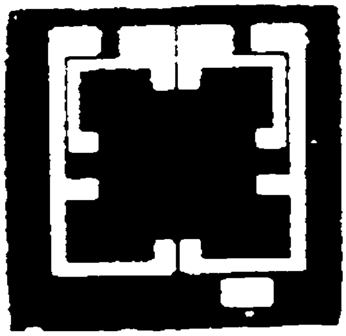


(a)



(b)

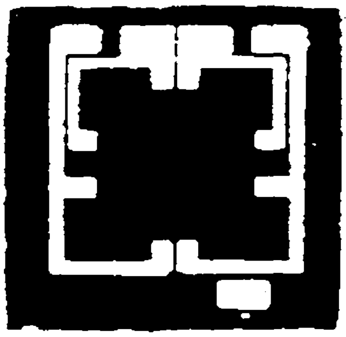
Figure 3.18 (a) Original image (b) Multi-level thresholded image after transformation



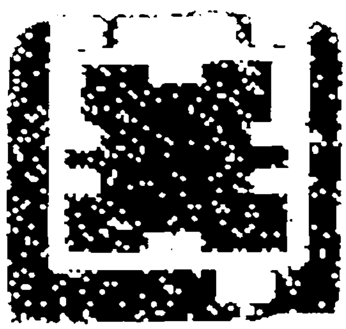
(a)

(b)

Figure 3.19 (a) Original image (b) (mage thresholded at a single level



(c)



(d)

Figure 3.19 (c) Original image (d) Image after multi-level thresholding

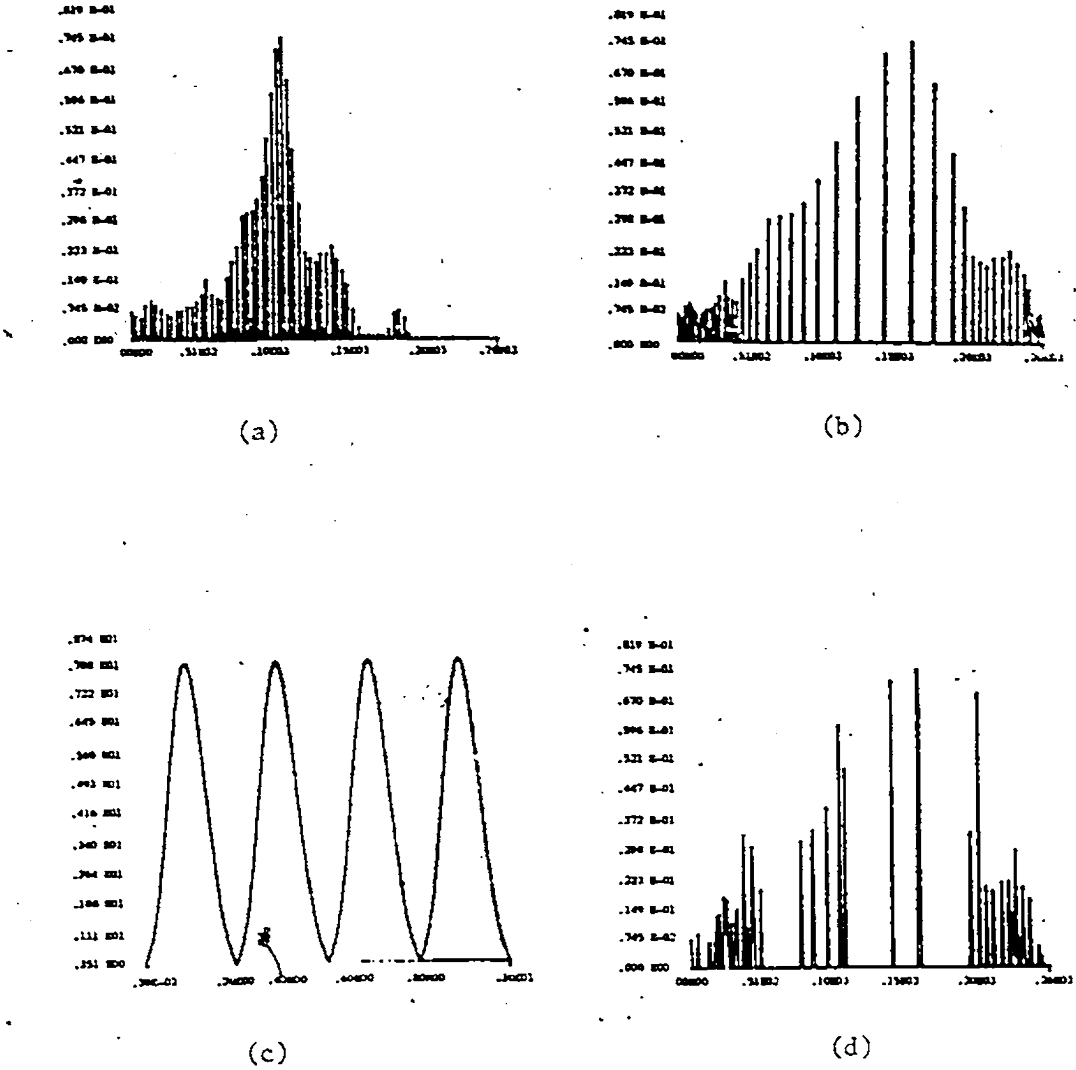
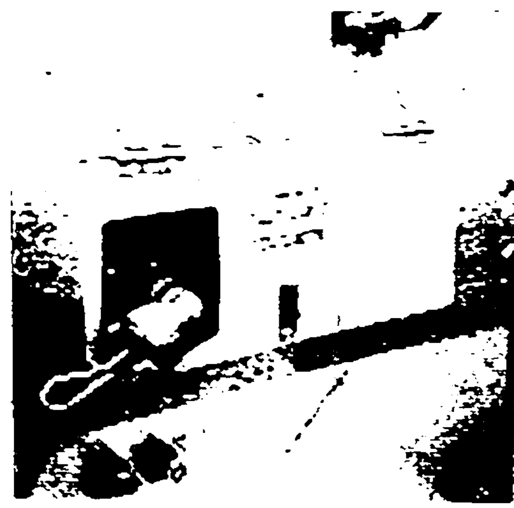
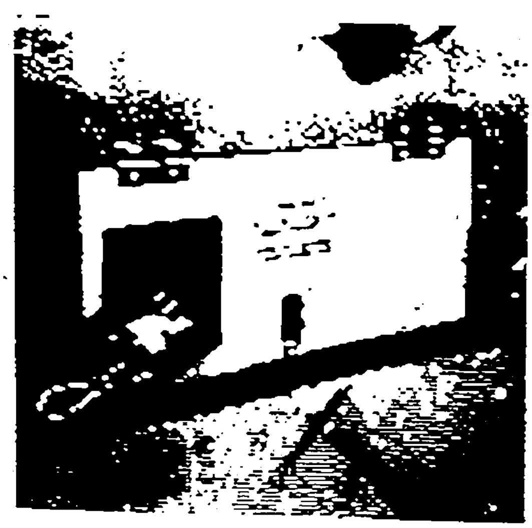


Figure 3.20 (a) Original histogram (b) Equalized histogram
(c) Desired histogram (d) Resulting histogram



(a)



(b)

Figure 3.21 (a) Original image (b) Multi-level thresholded image after transformation

This transformation was also carried out on the image of the cable connector. Fig.3.20(a) shows the histogram of the original image. The resulting histogram after the transformation is shown in Fig.3.20(d). It is evident that the original histogram has been transformed into a shape that resembles the specified shape, thereby greatly aiding the task of selecting appropriate thresholds for the image. Fig.3.21(b) shows the effect of thresholding the original image of Fig.3.21(a) at three different levels. Comparing Fig.3.21(b) with Fig.3.15(b), we can appreciate the greater amount of detail present in the image thresholded using the image transformation technique.

Another example of the use of the direct histogram specification technique to aid multi-level thresholding is illustrated in the image of Fig.3.22(a). The same process of transformation was carried out, and the image was thresholded at three levels to yield the thresholded image of Fig.3.22(b).

These examples suggest that the technique of transforming the original image into one whose histogram has a desirable distribution can be used for purposes of single and multi-level thresholding to yield very satisfactory results. A listing of the computer program used to implement this technique is provided in Appendix A.

(a)

(b)

Figure 3.22 (a) Original image (b) Multi-level thresholded image after transformation

CHAPTER IV

FEATURE EXTRACTION

In most image processing applications, the eventual goal is to extract the desirable information in an image. If the original image is of a poor quality, various processes could be carried out to enhance the visual quality of the image. It is also possible to extract features in an image by various means. Thresholding the image efficiently, amounts to classifying the image into one of two classes, object and background. This is one way of extracting the object from its background, and a need for a good thresholding scheme is emphasized, otherwise some inherent features of the image may be lost.

Thresholding is a form of feature extraction whereby an image is segmented into different classes based on the properties of each pixel. Depending upon the intensity level of the pixel, it is classified into being an object point or a background point. An alternative to this form could be the extraction of features in an image based on local regional properties. Thus, instead of classifying each point to a region, we classify local regional properties. This increases the dimension of the feature vector used for segmenting an image in terms of its object or background. One widely accepted region-dependent approach for segmentation of images is template matching. This concept has wide applications due to its pragmatic feasibility and simplicity.

4.1. Template Matching

This method of feature extraction is based on detecting transitions in gray levels between regions. The various regions are usually characterized by the differences in their gray level content. However, this is not necessarily the only feature that can be used for establishing region characteristics. Texture differences or colour differences (when dealing with coloured images) can also be used to differentiate between regions.

Template matching is used to identify certain characteristics in an image. A template is defined, in context to digital image processing applications, as an array designed to detect some invariant regional property. A template is sometimes also referred to as a mask or a window. In using template matching for feature extraction, it would be advantageous to have some prior information regarding the approximate shape and size of the characteristic being identified. This would help in choosing an appropriate size of the template. Each template would then be designed to detect the property being looked for. The following discussion explains the process of template matching to detect three different characteristics in an image.

4.1.1. Point template

Consider a simple example of detecting isolated points on a constant intensity background. A 3 x 3 template of the form shown in Fig. 4.1 can be used for this purpose. The centre of the template (marked 8) is moved around the image from point to point. At every position, each point in the image lying inside the template area is multiplied by the corresponding element of the template and the results are added. Notice that the sum of all the elements in the template is zero. Therefore, when the template occupies an area of constant background, meaning thereby that the gray levels of points inside the template area are the same, the sum of the operation performed at each position would be zero. However, if the centre of the template lies on a particle point, the sum

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 4.1 Point template

would be different from zero, since the gray level of the point would be different from its surrounding background. The sum would also be different from zero if the particle point lies inside the template area, but does not correspond to the centre of the template. In this case, the magnitude of the response would be weaker than if the point occurred at the centre. In other words, since only the point lying at the centre of the template is multiplied by 8, the sum in this case would be more than if the particle were located at any other point. In order to exactly locate the point, a threshold level would be specified in such a way that the weaker responses can be eliminated. It can be said, therefore, that a particle point is located at the centre of the template when the response of the template at that position exceeds the threshold level. By moving the template throughout the image, we could detect all isolated points in the image. This process can be explained mathematically [1] as follows:

We would represent the template and the image pixels lying inside the template as a vector. Let $z_1, z_2, \ldots, z_q$ be the weights in a 3 x 3 template. A 3 x 3 template size is being chosen for convenience; the procedure could be generalized for an m x n template. Therefore,

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_q \end{bmatrix}, \qquad (4.1\text{-}1)$$

where the first three elements of Z are the elements in the first row of the template, the next three are from the second row, and so on.

Let x_1, x_2, ..., x_q be the gray levels of the pixels inside the template area. It should be emphasized here that the element z_1 of the template should be on x_1, z_2 on x_2, and so on. Thus

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_q \end{bmatrix} \tag{4.1-2}$$

The preceding discussion on template matching suggests that it is basically a cross correlation between the elements of the template and the image points. Therefore, the response of the template at each position would be given by the inner product of the two vectors X and Z, which is defined as

$$Z'X = z_1x_1 + z_2x_2 + \ldots\ldots z_qx_q \tag{4.1-3}$$

It can be seen that Eq.(4.1-3) is the same as taking the sum of products as explained earlier.

If we specify a threshold level T, to eliminate all weak responses, we can detect a particle point at the centre of the template if

$$Z'X > T \tag{4.1-4}$$

This was a simple case involving the detection of a single isolated point against a constant background. This idea could be extended to identify other characteristics as well.

4.1.2. Line template

Consider the four templates shown in Fig.4.2. These templates are designed to detect straight lines, one pixel thick in an image. The lines could be oriented in a vertical or horizontal direction, or slanted

-1	-1	-1
2	2	2
-1	-1	-1

(a)

-1	-1	2
-1	2	-1
2	-1	-1

(b)

-1	2	-1
-1	2	-1
-1	2	-1

(c)

2	-1	-1
-1	2	-1
-1	-1	2

(d)

Figure 4.2 Line templates designed to detect (a) Horizontal line (b) Slanting line (c) Vertical line (d) Slanting line at 45°. For example, consider the template of Fig. 4.2(a). The sum of the elements of this template is zero. Therefore, if this template was to be moved around the image from point to point, its response would be zero for a constant background and maximum for a horizontal line one pixel thick. Using a threshold as in the previous case, the line can be identified to lie in the middle row of the template. Similarly, the template shown in Fig.4.2(b) would detect a line slanted at an angle of 45°. The template of Fig.4.2(c) would detect a vertical line and the template of Fig. 4.2(d) would identify lines slanted at an angle of -45°.

If we are interested in the identification of any one of the four features, we could use the appropriate template. Since in each template the preferred direction is weighted with a larger coefficient, one template could only identify the feature it is designed for. However, if we do not have a priori knowledge of the shape of the feature, we would not know which template to use. Therefore, it may be necessary to use all four templates and determine the closest match to the image section. For example, let Z_1, Z_2, Z_3 and Z_4 be four nine-dimensional vectors of the form of Eq. 4.1-1. These would stem from the elements of each of the four templates shown in Fig. 4.2. As described earlier, the response of each template would be given by

$$Z_i'X \quad \text{for } i = 1, 2, 3, 4 \tag{4.1-5}$$

where X is the vector of the image points lying inside the template area. If we wish to determine the closest match between the image section under review and each of the four templates of Fig. 4.2, we can say that the vector X is closest to the i-th template if the response of this template given by Eq.4.1-5 is the maximum. This can be stated as

$$Z_i'X > Z_j'X \tag{4.1-6}$$

for all values of j, except j = i.

If the condition given by Eq.4.1-6 is satisfied, we can say that the region under question matches closest to the i-th template.

This was a situation where templates were designed to detect straight lines. It should be understood that templates could be designed for any feature, and all the arguments presented above would hold true.

4.1.3 Edge-detection template

The concept of edge-detection is based on detecting transitions in gray level between regions. We would say that an "edge" has been detected if there is a significant change in gray level between adjacent regions. This information would aid in segmenting the image into different classes by delineating the boundaries between them. Using some sort of a two-dimensional differentiation process, we could detect these transitions in gray levels. Implementing a two-dimensional derivative function would yield high values at all edges in the image, and low elsewhere. The edge strengths produced by the differentiation process depend upon the local contrast in the image. Since a thresholded image just contains two levels, corresponding to the objects and the background, detecting the edges in such an image would yield the outline of the objects.

In order to find the gradient at a point, consider a 3 x 3 template shown in Fig. 4.3. This template can be used to find the gradient at the point e. We can define G_x, the gradient in the x-direction to be

$$G_x = (g + 2h + i) - (a + 2b + c) \quad .... \qquad (4.1\text{-}7)$$

We can see by looking at the image region given in Fig.4.3, that G_x is the difference between the first and third rows of this image

a	b	c
d	e	f
g	h	i

Figure 4.3 3 x 3 template used for edge detection

section, where the elements closer to e (b and h) are given twice as much weight as compared to the other elements. This weighting is based more on intuitive grounds than anything else. Therefore, G_x would represent an estimate of the derivative in the x-direction.

This argument could be extended for calculating the derivative in the y-direction as well. G_y, the gradient in the y-direction would be the difference between the first and third columns of the image section of Fig. 4.3.

$$G_y = (c + 2f + i) - (a + 2d + g) \;\ldots. \qquad (4.1\text{-}8)$$

The gradient at the point e would then be defined as

$$G = [G_x{}^2 + G_y{}^2]^{\frac{1}{2}} \qquad (4.1\text{-}9)$$

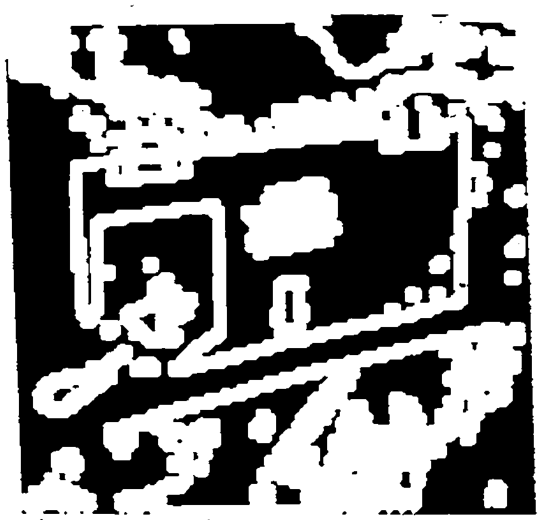
In order to simplify this expression for use on the computer, an alternative definition, using absolute values would be

$$G = |G_x| + |G_y| \qquad (4.1\text{-}10)$$

The absolute sum of the derivatives in the x and y direction would then give us the gradient at the point e. If this technique is applied on a binary level image, we can say that an edge occurs when G is non-zero, otherwise the point e lies interior to the object or the background.

This technique of edge detection can be used on a thresholded image, in applications where it is desired to outline the objects from the background. This process is applied to locate the boundaries between objects and the background on several images and the results can be seen in Fig. 4.4 through Fig. 4.6. The images on the top are the original images and after thresholding them and carrying out the edge detection technique described above, we can obtain the images at the bottom which have the object part of the image outlined from the background.

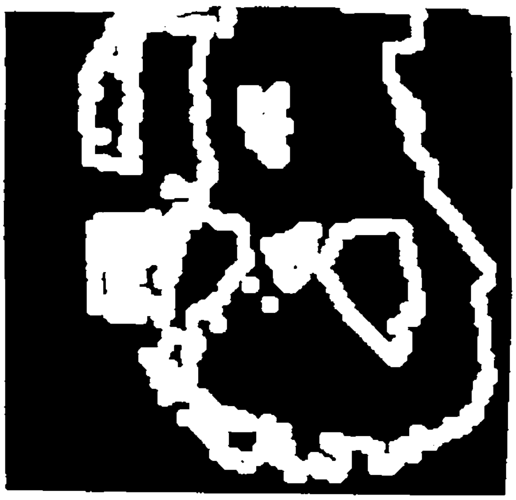
(a)



(b)

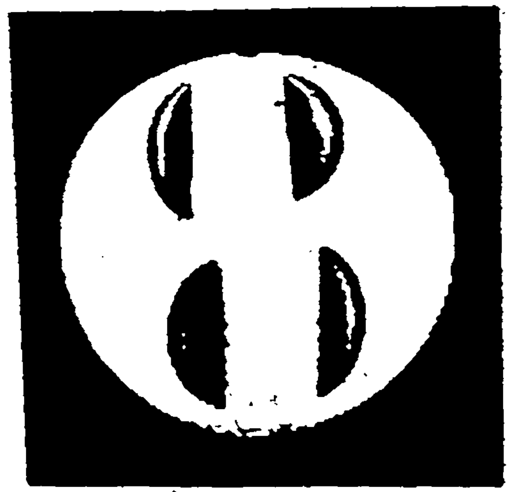
Figure 4.4 (a) Original image (b) Boundaries of objects in image

(a)

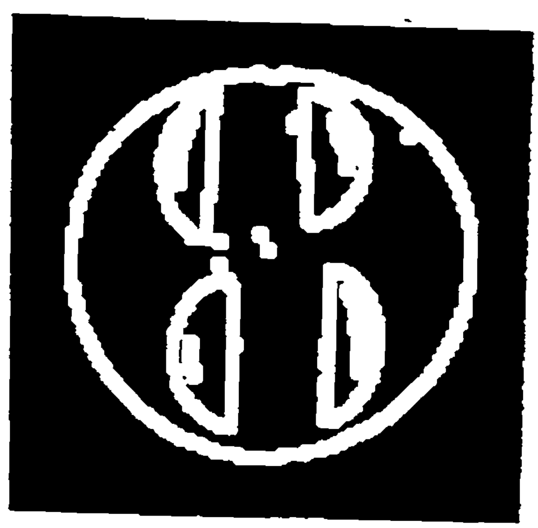


(b)

Figure 4.5 (a) Original image (b) Boundaries of objects in image



(a)



(b)

Figure 4.6 (a) Original image (b) Boundaries of objects in image

This was just one technique of boundary delineation in images where the edge detection template was brought to use. However, there are several other techniques available for this purpose, some of which will be discussed in the next section.

4.2. Boundary Following

As mentioned earlier, some image processing applications require the detection of the boundary between the objects and the background. One way of achieving this objective is to determine the edges in a thresholded image as shown earlier. Basically, a thresholded image is one where the objects have been separated from the background, and edge detection on such an image would just outline the objects. This information could be of great use in applications like quality control of manufactured parts. In such applications, we are usually interested in the shape and size of the object in order to determine whether it is faulty or not. This necessitates the process of outlining the boundaries of the object, to locate defects, if any, in it.

It has been described earlier that boundary following is based on locating two regions of higher and lower intensities, and in case of a thresholded image this would amount to locating the transition from black to white. There are several boundary following algorithms which can be used for the detection of these transitions. The working of one algorithm which has proved to give useful results [4], is explained below.

Since our objective is to determine whether or not there is a transition in gray-level between adjacent pixels, at every point we will have to consider its gray level and compare it to the gray level of all pixels surrounding it. The first step would be to obtain the first border point by scanning the thresholded image and looking for a transition in gray level. The first transition reached would be the starting point for the process. At the first border point, we compare its gray level to the gray level of the eight pixels surrounding it, as shown in Fig.4.7. Say, point a is the first border point and point b is the previous point which was not a border point. Now, we move clockwise from b and check the next pixel, which in this case would be c. If there is a transition between b and c, c would be the next border point, and b would still be the previous point. If however, c is not a border point, we proceed further in the same direction and search until a transition is reached. Whenever a transition is known to occur, we will have a border point and a previous point. At the new found border point we will continue the search process again starting from the previous point. It is important though that the search process be conducted sequentially in one direction only.

This method of searching around an already found border point would outline the boundary of a closed section. After the borders of a closed section have been obtained, we start the scanning procedure again of searching for a transition in gray level to give us a border point of another closed section. The search procedure, as explained above, is then used to outline the borders of this closed section. The whole image is thus scanned and the objects can be outlined from their background.

The algorithm described above is designed to follow closed borders. This means that starting from the first point, we follow the border points by the search process, and this process ends when we arrive at the point we originally started from. If the border is disconnected

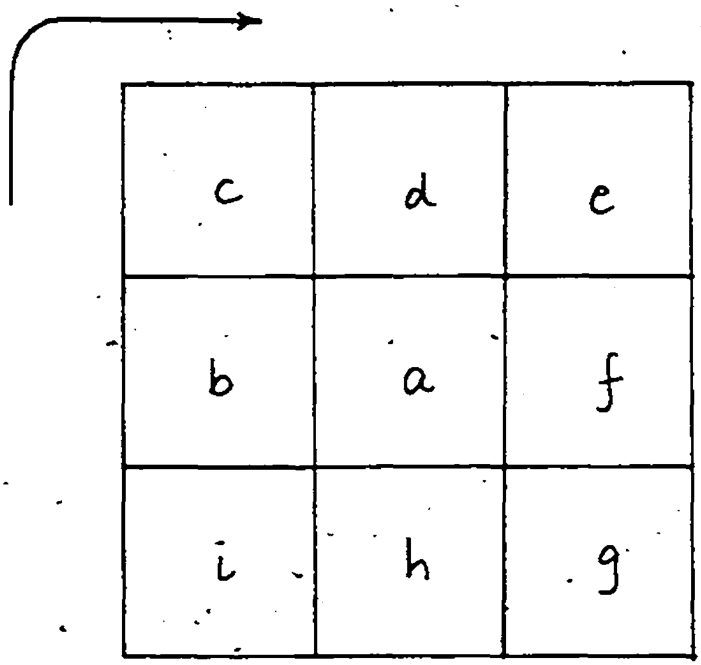


Figure 4.7 Sequence of search to follow borders

however, this method would outline a closed section of the border, and in order to outline the other disconnected section, we would have to consider it as another closed border. This process would also detect all isolated points in the image.

This method of border following is quite versatile in nature. Apart from just outlining the boundaries, we can also calculate the area occupied by the objects in the image. This is done by storing the co-ordinates of all border points obtained in the search process, and from these co-ordinates the perimeter of each closed border can be determined. This information is very useful in applications where besides the shape, we also need to determine the size of the objects. One application where this information is of immense importance is the detection of faults in manufactured parts.

The application of image processing to quality control is finding wide acceptance in industry today. The next section considers a problem oriented example and explains the use of template matching and boundary following to extract the desirable, information from the image.

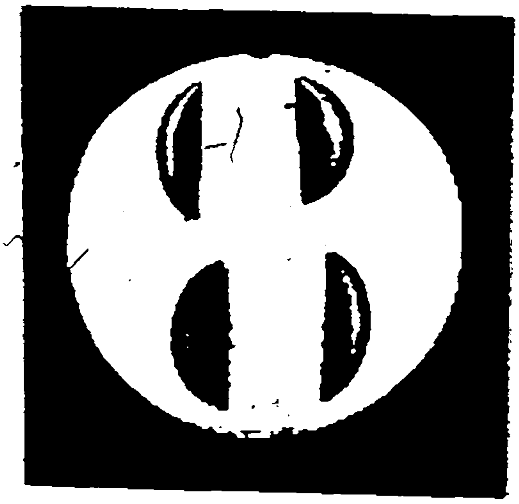
4.3. A Problem Oriented Example

In this section we will consider the application of image processing to an important aspect of industry, namely quality control. The concepts of template matching and boundary following are shown to be of great use in the detection of faults in manufactured parts. The image of each component coming down a manufacturing line is studied, and depending upon whether the particular component is faulty or not, a decision is made based on that component fulfilling the required specifications or not.

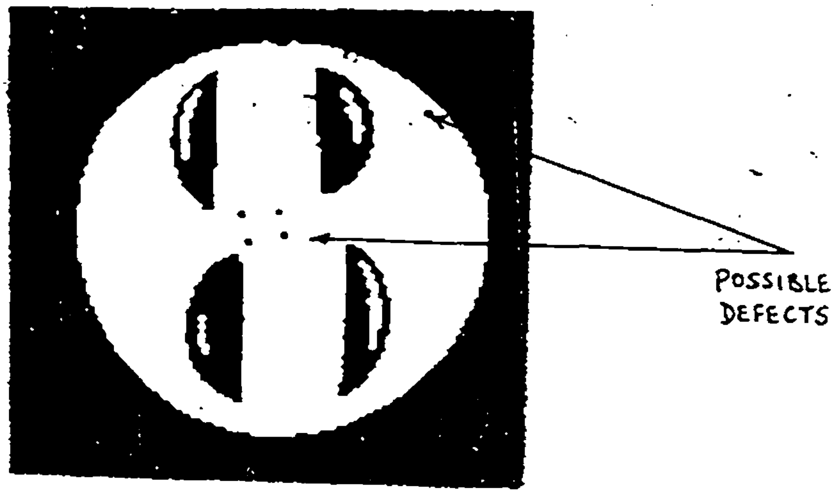
Consider the image of a piston head shown in Fig.4.8(a) used in the automobile industry. It is required to detect the faults if any, on the surface of the piston head. This fault may stem from a dent or a scratch or even a chipped edge. These piston heads are manufactured by the thousands every day, and it is necessary to have an automated system of fault detection. In some large plants a 2 second detection time is required. This imposes a speed factor in the algorithms used in the fault detection process.

It can be appreciated that while obtaining the image of the piston head, all irregularities on the surface in the form of a dent or a scratch would appear to be darker than the rest of the surface. This is obvious since a lesser amount of light would be reflected from a depression in the surface. These irregularities can be detected by an efficient thresholding scheme, as illustrated by the thresholded image shown in Fig.4.8(b). Notice that the dark spots shown are the possible defects on the surface. The "eye-brows" (getting the name from their shape) however, are inherent features of the piston head. The importance of obtaining a good threshold level is clearly evident in this example, because if the threshold level selected was not a good one, it was possible that the dark spots could have been merged with the rest of the surface, thus giving a wrong picture of what the image is actually like.

After obtaining a thresholded image and detecting the faults, our next step is to determine whether the particular part is acceptable or not. The faults in the part may be of a minor nature and fall within its tolerance limit. Secondly, the fault may occur at the edge of the part, in which case we would not be able to distinguish it from the



(a)



(b)

Figure 4.8 (a) Image of piston head (b) Thresholded image with possible defects indicated background. Therefore, it is necessary to follow the boundaries of the piston head and compare it to the boundaries obtained for a standard part.

Referring back to the image of Fig.4.8(b), it seems logical that if the eye-brows on the surface (which as mentioned before are inherent features of the piston head) be removed, whatever black portion would remain inside the piston head would be the faults. Therefore, a simple count of all the black pixels inside the piston head would give us the number of faults. This deduction seems quite simple, but the process of removing the inherent features in the image could pose to be a problem.

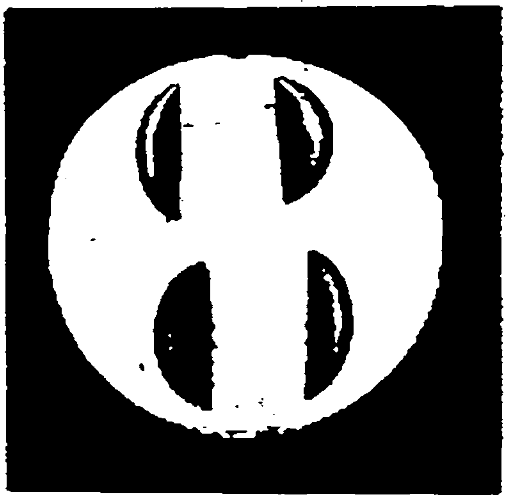
In applications of the form described above, we have the image of a standard part to go by and thus have prior knowledge about the inherent features of the image. The shape and size of these features could vary from part to part, but not to a great degree. We therefore have an idea about the approximate shape and size of these features. Going back to our discussion on extracting features through template matching, we know that we could design a template for whatever feature being looked for, if we had prior information about the shape and size of the feature. It seems reasonable therefore, that if we designed a template giving appropriate weight to suit the shape and size of the eye-brows, we could extract them from the image, and only the defects would remain. There are two problems however, in this approach. Firstly, since the shape and size of the eye-brows vary from part to part, it would be difficult to design one template that would give the desired result in every image. Secondly, the area occupied by the eye-brows in the image is quite large and designing a template of those dimensions would be very impractical. If we came across another situation where the feature under question was relatively small in size, and we knew its exact shape, we could extract the feature using template matching.

In this example, we notice from Fig. 4.8(b) that the eye-brows consists of one straight vertical line and a curved portion. Now in each part the eye-brows may be oriented slightly different, or their size may vary, but essentially all of them would have a straight vertical line and a curved portion alongside it. Therefore, if we can locate the straight line of the eye-brow, we could obtain a point on the border of the eye-brow and the rest of the surface. This can be achieved by using a template designed to locate vertical lines. Consider the template shown in Fig. 4.9. This template pattern when correlated with the image of Fig. 4.8(b), would detect all vertical lines in the image, ten pixels long. The length of ten pixels was arbitrarily chosen, so that the template does not locate small scratches on the surface, and misinterpret them as a border.

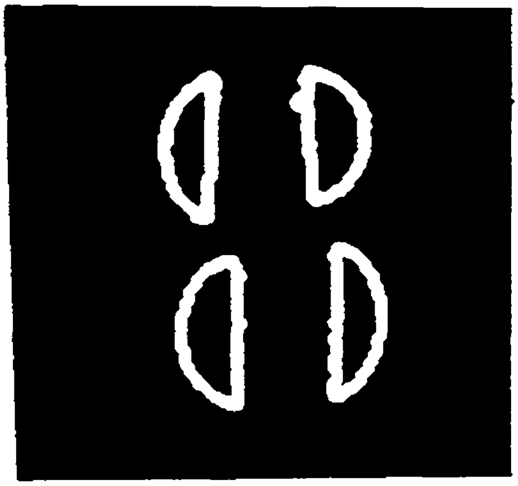
Once all the four vertical lines of the eye-brows have been detected, we can apply the boundary following algorithm explained in the previous section to remove the eye-brows from the image. Since the vertical lines detected by the templates are known to lie on the border, the utilization of that technique would not be a problem. This approach was carried out on the image of Fig. 4.8(b), and the eye-brows were extracted from the image. Fig. 4.10 shows just the eye-brows of the image extracted by the use of template matching and then boundary following. Removing the eye-brows from the image reduced the image of the piston head to the shape of Fig. 4.11. It can easily be noticed that the earlier defects are still present, only the inherent features of the image have been removed. The number of faults

-1	2	-1
-1	2	-1
-1	2	-1
-1	2	-1
-1	2	-1
-1	2	-1
-1	2	-1
-1	2	-1
-1	2	-1
-1	2	-1

Figure 4.9 Template designed to detect vertical line ten pixels long

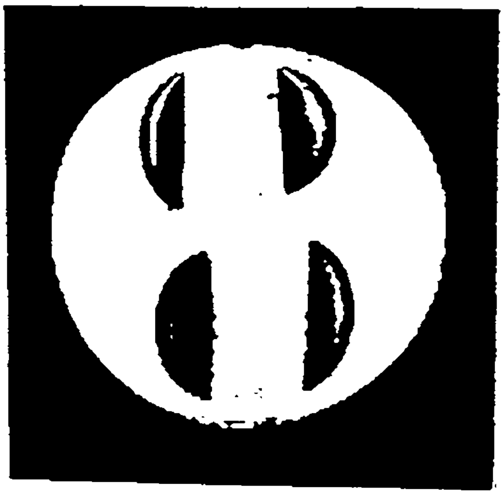


(a)

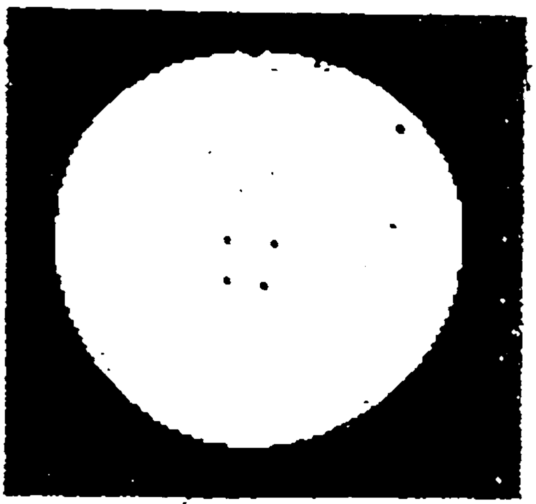


(b)

Figure 4.10 (a) Original image (b) Features extracted.



(a)



(b)

Figure 4.11 (a) Original image (b) Image with inherent features removed can now be determined by a simple count of all the pixels having zero intensity level (black) inside the piston.

In the example discussed above, it is clear that if a boundary following algorithm was to be employed, the process of template matching seems redundant. This is because the area of the defects in this case are much smaller than the inherent feature of the image, and the eye-brows can easily be detected even without template matching. However, it is emphasized that the example discussed above was just to illustrate the possible use of template matching in fault detection. Consider for example, an image whose features are small in comparison to the defects, and its exact shape and size were known. In this case, the inherent features in the image could be removed by template matching and a boundary following procedure need not be utilized.

It can therefore be concluded from this example that it is possible to use template matching for purposes of fault detection in manufactured parts, if the feature being considered is of a relatively small size and we have a prior knowledge of its exact shape. If that is not the case, a point on the border of the feature can be located through template matching, and then a boundary following algorithm can be utilized to extract the feature from the image. Removing the inherent features from an image would reduce the task of fault detection to a simple count of the black pixels in the image.

CHAPTER V

SUMMARY AND CONCLUSIONS

5.1. Summary

Image thresholding has been known to be a useful tool for extracting features from an image. The problem of selecting a good threshold level therefore, has often been the focal point of research for many image processing applications. Several techniques have evolved in past years for threshold selection. One widely accepted technique is where a threshold is selected automatically by examining the histogram of the gray levels in an image. If the histogram is bimodal, a threshold can be selected to lie at the bottom of the valley between the two peaks. However, the problem becomes a little more complex when the histogram is not bimodal.

The research conducted for this thesis was based upon the extraction of features in an image through thresholding and boundary following. For situations where the histogram of the image was not clearly bimodal, the approach of variable thresholding was verified. In this approach, thresholds are selected region-wise and the entire image is thresholded at various levels depending upon the local properties of the regions. These regions are obtained after segmenting the original image into smaller sections.

It has been shown earlier that it is possible to transform an image into one whose histogram has a specified distribution. This thesis discusses the use of the approach of direct histogram specification for purposes of binary level thresholding. A bimodal histogram is specified and the original image is transformed into one whose histogram has the desirable shape, thus facilitating the task of threshold selection. This approach was also extended for purposes of multilevel thresholding. This is used in cases where it is not possible to select one threshold level for the entire image, and it is necessary to select more than one threshold for the image. In this case, the histogram specified was a multi-modal histogram, and transforming the image appropriately yielded an image having a histogram of approximately the same distribution as the one specified.

Research was also carried out for specific applications to quality control. The concept of template matching was utilized to obtain boundaries of a thresholded image. It was also shown that template matching could be a useful tool for the extraction of features.

Finally, a problem oriented example in quality control was considered where it was desired to locate the defects in a manufactured part. A boundary following algorithm was utilized after template matching to extract the inherent features in the image. A template was designed for the particular feature being looked for, and since a large sized template would have been impractical, the concept of boundary following was also employed to detect the faults. The inherent features in the image were removed, leaving behind the possible defects.

5.2. Conclusions

In cases where an image does not have a bimodal histogram and threshold selection, poses to be a problem. The approach of variable thresholding was verified, and the results obtained were very satisfactory. However, this approach required extensive computation and was not suitable for purposes where efficiency in terms of time was a great concern.

The concept of direct histogram specification was used for purposes of binary level thresholding. It is known that an image can be transformed into one whose histogram has a specified shape. This concept was applied for binary level thresholding by specifying a bimodal histogram consisting of different means and same standard deviations. The selection of means and standard deviations was totally arbitrary and no criteria has been established for their selection. It was shown that an image whose histogram is not clearly bimodal can be transformed into one whose histogram closely resembles the specified bimodal distribution. This greatly aids the selection of a good threshold level.

This approach of image transformation was also extended for purposes of multi-level thresholding, and it was shown that for cases where the selection of one threshold level for the entire image results in a loss of information, more than one threshold can be selected, thereby segmenting the image into more than two levels.

The concept of template matching was used for purposes of fault detection of manufactured parts. It was seen that this approach could only be used where the size of the feature being detected was small and its exact shape and size were known. Otherwise, a portion of the feature could be detected through template matching, and a boundary following scheme utilized to extract the feature from the image. It was also noticed that by removing the inherent features in an image, fault detection would be greatly simplified. These features can be removed simply by template matching, or combining it with boundary following, depending upon the size of the feature in question.

In conclusion, we can envisage the development of a system which would efficiently threshold an image using the direct histogram specification technique and then extract features from it using template matching, and if necessary, boundary following.

APPENDIX A

```
C         ************************
C             FILENAME PLOTDEN
C         ************************

C         THIS PROGRAM PLOTS THE HISTOGRAM OF GRAY LEVELS IN AN IMAGE
C
          DIMENSION NAME(5),IP(128,128),PR(256),X(256),IFILE(5)
          DATA IBLNK/'  '/
1000      ACCEPT '# OF PIXELS PER LINE=',NSIZE
          ACCEPT '# OF LINES           =',NLINE
          ACCEPT '# OF LEVELS          =',NP
          NT=NLINE*NSIZE
          DO 1 I=1,5
1         NAME(I)=IBLNK
          WRITE(10,2)
2         FORMAT(' ',1X,'INPUT FILENAME:-',Z)
          READ(11,3)(NAME(I),I=1,5)
3         FORMAT(5A2)
          OPEN 2,NAME,LEN=2*NSIZE,REC=NLINE
          DO 4 I=1,NLINE
4         READ(2)(IP(I,J),J=1,NSIZE)
          CLOSE 2
C
C         CALCULATING THE PROBABILITY DENSITY FUNCTION
C
          DO 7 K=1,NP
7         PR(K)=0.0
          DO 8 I=1,NLINE
          DO 8 J=1,NSIZE
          K=IP(I,J)+1
8         PR(K)=PR(K)+1.0
          DO 20 K=1,NP
20        PR(K)=PR(K)/NT
C
C         PLOTTING THE PROBABILITY DISTRIBUTION
C
          DO 30 I=1,NP
          X(I)=FLOAT(I-1)
30        CONTINUE
          OPEN 0,"$TTO1"
          CALL PLTEKS(X,PR,NP)
          CLOSE 0
          ACCEPT 'WISH TO CONTINUE(1)---OR NOT(0)---->',INT
          IF(INT.EQ.1)GO TO 1000
          STOP
          END
```

```
C         ***************************
C              FILENAME PROBDEN
C         ***************************
C         THIS PROGRAM IS USED TO EQUALIZE THE HISTOGRAM OF AN IMAGE
C
          DIMENSION IP(128,128),PR(256),S(256),X(256),NAME(5),IFILE(5)
          DATA IBLNK/'  '/
1000      ACCEPT '# OF PIXELS PER LINE=',NSIZE
          ACCEPT '# OF LINES          =',NLINE
          ACCEPT '# OF LEVELS         =',NP
          NT=NLINE*NSIZE
          DO 1 I=1,5
1         NAME(I)=IBLNK
          WRITE(10,2)
2         FORMAT('   ',1X,'INPUT FILENAME:-',Z)
          READ(11,3)(NAME(I),I=1,5)
3         FORMAT(5A2)
          OPEN 1,NAME,LEN=2*NSIZE,REC=NLINE
          DO 4 I=1,NLINE
4         READ(1)(IP(I,J),J=1,NSIZE)
          CLOSE 1
C
C         QUANTIZING LEVELS TO LIE BETWEEN '0' & '255'
C         '0' REPRESENTS BLACK AND '255' REPRESENTS WHITE
C
          IBIG=IP(1,1)
          ISMALL=IP(1,1)
          DO 5 I=1,NLINE
          DO 5 J=1,NSIZE
          IF(IP(I,J).GT.IBIG)IBIG=IP(I,J)
          IF(IP(I,J).LT.ISMALL)ISMALL=IP(I,J)
5         CONTINUE
          IDIFF=IBIG-ISMALL
          DO  6 I=1,NLINE
          DO 6 J=1,NSIZE
          RIP=((255.0*FLOAT(IP(I,J)-ISMALL))/IDIFF)+0.5
6         IP(I,J)=RIP
C
C         CALCULATING THE PROBABILITY DENSITY FUNCTION
C
          DO 7 K=1,NP
          PR(K)=0.0
7         CONTINUE
          DO 8 I=1,NLINE
          DO 8 J=1,NSIZE
          K=IP(I,J)+1
8         PR(K)=PR(K)+1.0
          DO 20 K=1,NP
20        PR(K)=PR(K)/NT
```

```
C
C       COMPUTATION OF TRANSFORMATION FOR UNIFORM PROBABILITY DENSITY
C
        S(1)=PR(1)
        BIG=S(1)
        SMALL=S(1)
        DO 9 J=2,NP
        S(J)=S(J-1)+PR(J)
        IF(S(J).GT.BIG)BIG=S(J)
        IF(S(J).LT.SMALL)SMALL=S(J)
9       CONTINUE
C
C       ASSIGNING TRANSFORMED LEVELS TO THEIR CLOSEST
C       VALID LEVEL
C
        DIFF=BIG-SMALL
        DO 10 I=1,NP
10      S(I)=((S(I)-SMALL)/DIFF)*(NP-1)
C
C       SUBSTITUTING NEW LEVELS INTO PICTURE
C
        DO 12 I=1,NLINE
        DO 12 J=1,NSIZE
        IP(I,J)=S(IP(I,J)+1)+0.5
12      CONTINUE
        DO 16 I=1,5
16      IFILE(I)=IBLNK
        WRITE(10,17)
17      FORMAT(' ',1X,'TRANSFORMED IMAGE FILENAME:-',Z)
        READ(11,18)(IFILE(I),I=1,5)
18      FORMAT(5A2)
        OPEN 2,IFILE,LEN=2*NSIZE,REC=NLINE
        DO 19 I=1,NLINE
19      WRITE(2)(IP(I,J),J=1,NSIZE)
        CLOSE 2
        TYPE"
        ACCEPT 'PROGRAM :-  STOP(0)--------CONTINUE(NE.0)---->',IWT
        IF(IWT.NE.0)GO TO 1000
        STOP
        END
```

```
C         **************************
C             FILENAME MODHIST
C         **************************
C
C         THIS PROGRAM TRANSFORMS THE ORIGINAL IMAGE INTO ONE
C         WHOSE HISTOGRAM  RESEMBLES A SPECIFIED DISTRIBUTION
C
          DIMENSION IP(128,128),PR(256),S(256),NAME(5),IFILE(5)
          DIMENSION X(256),PZ(256),V(256),IV(256)
          DATA IBLNK/'  '/
1000      ACCEPT '# OF PIXELS PER LINE=',NSIZE
          ACCEPT '#OF LINES            =',NLINE
          NT=NLINE*NSIZE
          DO 1 I=1,5
1         NAME(I)=IBLNK
          WRITE(10,2)
2         FORMAT(' ',1X,'INPUT FILENAME=',Z)
          READ(11,3)(NAME(I),I=1,5)
3         FORMAT(5A2)
          OPEN 1,NAME,LEN=2*NSIZE,REC=NLINE
          DO 4 I=1,NLINE
4         READ(1)(IP(I,J),J=1,NSIZE)
          CLOSE 1
C
C         QUANTIZING GRAY LEVEL  INTENSITIES TO LIE BETWEEN '0' & '255'
C         '0' REPRESENTS BLACK AND '255' REPRESENTS WHITE
C
          IBIG=IP(1,1)
          ISMALL=IP(1,1)
          DO 5 I=1,NLINE
          DO 5 J=1,NSIZE
          IF(IP(I,J).GT.IBIG)IBIG=IP(I,J)
          IF(IP(I,J).LT.ISMALL)ISMALL=IP(I,J)
5         CONTINUE
          IDIFF=IBIG-ISMALL
          DO 6 I=1,NLINE
          DO 6 J=1,NSIZE
          RIP=((255.0*FLOAT(IP(I,J)-ISMALL))/IDIFF)+0.5
6         IP(I,J)=RIP
C
C         CALCULATING THE PROBABILITY DENSITY FUNCTION
C
          DO 7 K=1,256
7         PR(K)=0.0
          DO 8 I=1,NLINE
          DO 8 J=1,NSIZE
          K=IP(I,J)+1
8         PR(K)=PR(K)+1.0
          DO 20 K=1,256
20        PR(K)=PR(K)/NT
```

```
C
C       COMPUTATION OF TRANSFORMATION FOR UNIFORM PROBABILITY DENSITY
C
        S(1)=PR(1)
        BIG=S(1)
        SMALL=S(1)
        DO 9 J=2,256
        S(J)=S(J-1)+PR(J)
        IF(S(J).GT.BIG)BIG=S(J)
        IF(S(J).LT.SMALL)SMALL=S(J)
9       CONTINUE
C
C       ASSIGNING  TRANSFORMED VALUES TO THEIR CLOSEST VALID LEVEL
C
        DIFF=BIG-SMALL
        DO 10 I=1,256
10      S(I)=((S(I)-SMALL)/DIFF)*255
C
C       GENERATING THE DESIRED PROB.  DENSITY FUNCTION
C
        ACCEPT 'DESIRED DIST. :- BIMODAL(1)--MULTIMODAL(2)--->',IFT
        IF(IFT.EQ.1)GO TO 201
        IF(IFT.EQ.2)GO TO 202
201     CALL GAUSS(PZ,X)
        GO TO 203
202     CALL TEST(PZ,X)
C
C       COMPUTING TRANSFORMATION FOR DESIRED PROB.  DENSITY FUNCTION
C
203     V(1)=PZ(1)
        BIG=V(1)
        SMALL=V(1)
        DO 11 I=2,256
        V(I)=V(I-1)+PZ(I)
        IF(V(I).GT.BIG)BIG=V(I)
        IF(V(I).LT.SMALL)SMALL=V(I)
11      CONTINUE
        DIFF=BIG-SMALL
        DO 12 I=1,256
        V(I)=255*((V(I)-SMALL)/DIFF)
12      IV(I)=V(I)+0.5
C
C       TRANSFORMING PICTURE TO ONE HAVING THE DESIRED
C       PROBABILITY DENSITY FUNCTION
C
        DO 15 I=1,NLINE
        DO 15 J=1,NSIZE
        IP(I,J)=S(IP(I,J)+1)+0.5
        DO 13 L=1,256
        IF(IP(I,J).EQ.IV(L))GO TO 14
        GO TO 13
14      IP(I,J)=L-1
        GO TO 15
13      CONTINUE
15      CONTINUE
```

```
      DO 16 I=1,5
16    IFILE(I)=IBLNK
      WRITE(10,17)
17    FORMAT('  ',1X,'TRANSFORMED IMAGE FILENAME:-',Z)
      READ(11,18)(IFILE(I),I=1,5)
18    FORMAT(5A2)
      OPEN 2,IFILE,LEN=2*NSIZE,REC=NLINE
      DO 19 I=1,NLINE
19    WRITE(2)(IP(I,J),J=1,NSIZE)
      CLOSE 2
      TYPE'     '
      ACCEPT 'PROGRAM :-  STOP(0) ----  CONTINUE(NE.0)----->',IWT
      IF(IWT.NE.0)GO TO 1000
      STOP
      END
```

```
      SUBROUTINE GAUSS(F,X)
C     ************************
C         FILENAME GAUSS
C     ************************
C
C     THIS SUBROUTINE PLOTS A BIMODAL  DISTRIBUTION
C     CONSISTING OF TWO GAUSSIAN DISTRIBUTIONS
C
      DIMENSION F(256),X(256)
      ACCEPT 'FIRST MEAN VALUE=',ETA1
      ACCEPT 'SECOND MEAN VALUE=',ETA2
      ACCEPT 'FIRST STANDARD DEVIATION=',SIGMA1
      ACCEPT 'SECOND STANDARD DEVIATION=',SIGMA2
      PI=3.141593
      Y=2*3.141593
      DO 1 N=1,128
      X(N)=FLOAT(N)/FLOAT(256)
      F(N)=(1/(SIGMA1*SQRT(Y)))*EXP(-((X(N)-ETA1)**2)/(2*(SIGMA1**2)))
1     CONTINUE
      DO 2 N=129,256
      X(N)=FLOAT(N)/FLOAT(256)
      F(N)=(1/(SIGMA2*SQRT(Y)))*EXP(-((X(N)-ETA2)**2)/(2*(SIGMA2**2)))
2     CONTINUE
      OPEN 0, "$TTO1"
      CALL PLTEKS(X,F,256)
      CLOSE 0
      RETURN
      END
```

```
      SUBROUTINE TEST(F,X)
C     ************************
C         FILENAME TEST
C     ************************
C     THIS SUBROUTINE PLOTS A MULTIMODAL DISTRIBUTION
C     CONSISTING OF FOUR GAUSSIAN DISTRIBUTIONS
C
      DIMENSION F(256),X(256)
      ACCEPT 'FIRST MEAN VALUE=',ETA1
      ACCEPT 'SECOND MEAN VALUE=',ETA2
      ACCEPT 'THIRD MEAN VALUE =',ETA3
      ACCEPT 'FOURTH MEAN VALUE=',ETA4
      ACCEPT 'FIRST STANDARD DEVIATION=',SIGMA1
      ACCEPT 'SECOND STANDARD DEVIATION=',SIGMA2
      ACCEPT 'THIRD STANDARD DEVIATION =',SIGMA3
      ACCEPT 'FOURTH STANDARD DEVIATION=',SIGMA4
      PI=3.141593
      Y=2*3.141593
      DO 1 N=1,64
      X(N)=FLOAT(N)/FLOAT(256)
      F(N)=(1/(SIGMA1*SQRT(Y)))*EXP(-((X(N)-ETA1)**2)/(2*(SIGMA1**2)))
1     CONTINUE
      DO 2 N=65,128
      X(N)=FLOAT(N)/FLOAT(256)
      F(N)=(1/(SIGMA2*SQRT(Y)))*EXP(-((X(N)-ETA2)**2)/(2*(SIGMA2**2)))
2     CONTINUE
      DO 3 N=129,192
      X(N)=FLOAT(N)/FLOAT(256)
      F(N)=(1/(SIGMA3*SQRT(Y)))*EXP(-((X(N)-ETA3)**2)/(2*(SIGMA3**2)))
3     CONTINUE
      DO 4 N=193,256
      X(N)=FLOAT(N)/FLOAT(256)
      F(N)=(1/(SIGMA4*SQRT(Y)))*EXP(-((X(N)-ETA4)**2)/(2*(SIGMA4**2)))
4     CONTINUE
      OPEN 0, "$TTO1"
      CALL PLTEKS(X,F,256)
      CLOSE 0
      RETURN
      END
```

```
C          *************************
C              FILENAME THTEST
C          *************************

C          THIS PROGRAM IS USED  FOR THRESHOLDING THE IMAGE AT LEVELS
C          OBTAINED FROM THE TRANSFORMED HISTOGRAMS
C
           DIMENSION NAME(5),IFILE(5),IP(128,128),IFTH(10)
           DATA IBLNK/'  '/
1000       ACCEPT '# OF PIXELS PER LINE=',NLINE
           ACCEPT '# OF LINES           =',NSIZE
           ACCEPT '# OF LEVELS          =',NP
           DO 1 I=1,5
1          NAME(I)=IBLNK
           WRITE(10,2)
2          FORMAT('  ',1X,'INPUT FILENAME :-',Z)
           READ(11,3)(NAME(I),I=1,5)
3          FORMAT(5A2)
           OPEN 1,NAME,LEN=2*NLINE,REC=NSIZE
           DO 4 I=1,NLINE
4          READ(1)(IP(I,J),J=1,NSIZE)
           CLOSE 1
           ACCEPT 'TOTAL NO. OF THRESHOLDS=',MAN
           DO 5 I=1,MAN
           ACCEPT 'THRESHOLD LEVEL=',ITH
           IFTH(I)=ITH
5          CONTINUE
           M=1
           DO 6 I=1,128
           DO 6 J=1,128
           IF(IP(I,J).LE.IFTH(M))IP(I,J)=0
6          CONTINUE
10         M=M+1
           IF(M.GT.MAN)GO TO 9
           DO 7 I=1,NLINE
           DO 7 J=1,NSIZE
           IF(IP(I,J).GT.IFTH(M-1).AND.IP(I,J).LE.IFTH(M))GO TO 8
           GO TO 7
8          IP(I,J)=(IFTH(M-1)+IFTH(M))/2
7          CONTINUE
           GO TO 10
9          DO 11 I=1,NLINE
           DO 11 J=1,NSIZE
           IF(IP(I,J).GT.IFTH(MAN))IP(I,J)=NP-1
11         CONTINUE
```

```
      DO 12 I=1,5
12    IFILE(I)=IBLNK
      WRITE(10,13)
13    FORMAT(' ',1X,'THRESHOLDED IMAGE FILENAME:-',Z)
      READ(11,14)(IFILE(I),I=1,5)
14    FORMAT(5A2)
      OPEN 2,IFILE,LEN=2*NLINE,REC=NSIZE
      DO 15 I=1,NLINE
15    WRITE(2)(IP(I,J),J=1,NSIZE)
      CLOSE 2
      TYPE' '
      ACCEPT 'WISH TO CONTINUE(1)----OR NOT(0)---->',IWT
      IF(IWT.EQ.1)GO TO 1000
      STOP
      END
```

```
C       ****************************
C           FILENAME MULTHRESH
C       ****************************
C       THIS PROGRAM IS USSED FOR MULTI-LEVEL THRESHOLDING
C       CONSIDERING SECTIONS OF THE HISTOGRAM
C
        DIMENSION NAME(5),IP(128,128),PR(256),X(256),IFILE(5)
        DIMENSION X1(256),PR1(256),IFTH(10)
        DATA IBLNK/'  '/
1000    ACCEPT '# OF PIXELS PER LINE=',NSIZE
        ACCEPT '# OF LINES          =',NLINE
        ACCEPT '# OF LEVELS         =',NP
        DO 1 I=1,5
1       NAME(I)=IBLNK
        WRITE(10,2)
2       FORMAT('   ',1X,'INPUT FILENAME :-',Z)
        READ(11,3)(NAME(I),I=1,5)
3       FORMAT(5A2)
        OPEN 1,NAME,LEN=2*NSIZE,REC=NLINE
        DO 4 I=1,NLINE
4       READ(1)(IP(I,J),J=1,NSIZE)
        CLOSE 1
        NT=NLINE*NSIZE
C
C       CALCULATING THE PROBABILITY DENSITY FUNCTION
C
        DO 7 K=1,NP
7       PR(K)=0.0
        DO 8 I=1,NLINE
        DO 8 J=1,NSIZE
        K=IP(I,J)+1
8       PR(K)=PR(K)+1.0
        DO 20 K=1,NP
        PR(K)=PR(K)/NT
        X(K)=FLOAT(K-1)
20      CONTINUE
C
C       PLOTTING SECTIONS OF HISTOGRAM
C
        M=0
        ITH=0
110     IT=ITH+1
        ACCEPT 'SIZE OF SECTION OF HIST.=',NUM
        L3=ITH+NUM
        DO 15 I=1,NUM
        I1=I+ITH
        X1(I)=FLOAT(X(I1))
        PR1(I)=PR(I1)
15      CONTINUE
```

```
      OPEN 0,"$TTO1"
      CALL PLTEKS(X1,PR1,NUM)
      CLOSE 0
      ACCEPT 'HISTOGRAM BIMODAL(1)---OR NOT(0)---->',IBM
      IF(IBM.EQ.1)GO TO 22
      IF(IBM.EQ.0)GO TO 55
C
C     SELECTING A THRESHOLD
C
22    ACCEPT 'LEVEL TO SEPARATE PEAKS=',L1
C
C     FINDING MAX. BETWEEN 0&L1.
C
      NO1=IT
      PEAK1=PR(NO1)
      NOM=NO1+1
      DO 100 I=NOM,L1
      IF(PR(I).GT.PEAK1)GO TO 50
      GO TO 100
50    PEAK1=PR(I)
      NO1=I
100   CONTINUE
      WRITE(10,18)NO1
18    FORMAT('    ',1X,'FIRST PEAK OCCURS AT :-    ',I3)
C
C     FINDING MAX. BETWEEN L1 & END OF SECTION
C
      L2=L1+2
      J=L1+1
      PEAK2=PR(J)
      NO2=J
      DO 200 I=L2,L3
      IF(PR(I).GT.PEAK2)GO TO 60
      GO TO 200
60    PEAK2=PR(I)
      NO2=I
200   CONTINUE
      WRITE(10,19)NO2
19    FORMAT('  ',1X,'SECOND PEAK OCCURS AT :-    ',I3)
C
C     FINDING VALLEY BETWEEN PEAKS
C
      JO=NO1
      MO=NO1+1
      VALLI=PR(JO)
      DO 300 I=MO,NO2
      IF(PR(I).EQ.0)GO TO 300
      IF(PR(I).LT.VALLI)GO TO 70
      GO TO 300
70    VALLI=PR(I)
      NTH=I-1
300   CONTINUE
```

```
C
C         TESTING FOR VALIDITY OF THRESHOLD
C
          ACCEPT 'ACCEPTED RATIO OF PEAK TO VALLEY=',TRATIO
          ACCEPT 'SMALL NO.  TO PREVENT DIV.  BY 0  =',EPS
          IF(PEAK1-PEAK2)16,16,21
16        RATIO=PEAK1/(VALLI+EPS)
21        RATIO=PEAK2/(VALLI+EPS)
          IF(RATIO.GT.TRATIO)GO TO 80
          GO TO 90
80        WRITE(10,17)NTH
17        FORMAT('   ',1X,'THRESHOLD LEVEL=',I5)
          M=M+1
          IFTH(M)=NTH
          ITH=NTH
          GO TO 55
90        TYPE 'NO THRESHOLD AVAILABLE'
          GO TO 55
55        ACCEPT 'WISH TO GIVE ARBITRARY THRESH(1)---OR NOT(0)--->',IDM
          IF(IDM.EQ.1)GO TO 500
          IF(IDM.EQ.0)GO TO 501
500       ACCEPT 'ARBITRARY THRESHOLD=',ITH
          GO TO 110
501       IF(L3.EQ.NP)GO TO 600
          GO TO 110
C
C         ASSIGNING LEVELS TO SEGMENTED SECTIONS
C

600       MAN=M
          KO=1
          DO 601 I=1,NLINE
          DO 601 J=1,NSIZE
          IF(IP(I,J).LE.IFTH(KO))IP(I,J)=0
601       CONTINUE
400       KO=KO+1
          IF(KO.GT.MAN)GO TO 605
          DO 602 I=1,NLINE
          DO 602 J=1,NSIZE
          IF(IP(I,J).GT.IFTH(KO-1).AND.IP(I,J).LE.IFTH(KO))GO TO 650
          GO TO 602
650       IP(I,J)=(IFTH(KO-1)+IFTH(KO))/2
602       CONTINUE
          GO TO 400
605       DO 700 I=1,NLINE
          DO 700 J=1,NSIZE
          IF(IP(I,J).GT.IFTH(MAN))IP(I,J)=NP-1
700       CONTINUE
```

```
13      DO 9 I=1,5
9       IFILE(I)=IBLNK
        WRITE(10,10)
10      FORMAT('   ',1X,'THRESHOLDED IMAGE FILENAME :-    ',Z)
        READ(11,11)(IFILE(I),I=1,5)
11      FORMAT(5A2)
        OPEN 2,IFILE,LEN=2*NSIZE,REC=NLINE
        DO 12 I=1,NLINE
12      WRITE(2)(IP(I,J),J=1,NSIZE)
        CLOSE 2
        TYPE '  '
        ACCEPT 'WISH TO CONTINUE(1)----OR STOP(0)----->',IWT
        IF(IWT.EQ.1)GO TO 1000
        STOP
        END
```

```
C         ***********************
C             FILENAME TEMP
C         ***********************

C         THIS PROGRAM IS USED TO DETECT EDGES USING TEMPLATE MATCHING
C
          DIMENSION NAME(5),IFILE(5),IP(128,128),IT1(128),IT2(128)
          INTEGER A,B,C,D,E,F,G,H,GX,GY,GRAD
          DATA IBLNK/'  '/
1000      ACCEPT ' # OF PIXELS PER LINE =',NLINE
          ACCEPT ' # OF LINES           =',NSIZE
          ACCEPT ' # OF LEVELS          =',NP
          DO 1 I=1,5
1         NAME(I)=IBLNK
          WRITE(10,2)
2         FORMAT('   ',1X,'INPUT FILENAME :- ',Z)
          READ(11,3)(NAME(I),I=1,5)
3         FORMAT(5A2)
          OPEN 1,NAME,LEN=2*NSIZE,REC=NLINE
          DO 4 I=1,NLINE
4         READ(1)(IP(I,J),J=1,NSIZE)
          CLOSE 1
C
C         CALCULATING THE GRADIENT
C
          DO 5 I=2,127
          DO 6 J=2,127
          A=IP(I-1,J-1)
          B=IP(I-1,J)
          C=IP(I-1,J+1)
          D=IP(I,J-1)
          E=IP(I,J+1)
          F=IP(I+1,J-1)
          G=IP(I+1,J)
          H=IP(I+1,J+1)
          GX=(F+2*G+H)-(A+2*B+C)
          GY=(C+2*E+H)-(A+2*D+F)
          GRAD=ABS(GX)+ABS(GY)
C
C         TRANSFORMING PIXEL VALUES DEPENDING UPON WHETHER
C         GRADIENT IS NON-ZERO OR NOT
C
          M=((I+1)/2)-(I/2)
          IF(M.EQ.0)GO TO 7
          GO TO 8
7         IF(GRAD.NE.0)GO TO 9
          GO TO 10
9         IT1(J)=NP-1
          GO TO 6
10        IT1(J)=0
          GO TO 6
8         IF(GRAD.NE.0)GO TO 11
```

```
      GO TO 12
11    IT2(J)=NP-1
      GO TO 6
12    IT2(J)=0
6     CONTINUE
      K=I-3
      IF(K.GT.0)GO TO 13
      GO TO 5
13    IF(M.EQ.0)GO TO 14
      GO TO 15
14    DO 16 J=2,127
      IP(I-2,J)=IT1(J)
16    CONTINUE
      GO TO 5
15    DO 17 J=2,127
      IP(I-2,J)=IT2(J)
17    CONTINUE
5     CONTINUE
      DO 22 J=2,127
      I=126
      IP(I,J)=IT1(J)
22    CONTINUE
      DO 23 J=2,127
      I=127
      IP(I,J)=IT2(J)
23    CONTINUE
      DO 18 I=1,5
18    IFILE(I)=IBLNK
      WRITE(10,19)
19    FORMAT('   ',1X,'OUTPUT FILENAME :- ',Z)
      READ(11,20)(IFILE(I),I=1,5)
20    FORMAT(5A2)
      OPEN 2,IFILE,LEN=2*NSIZE,REC=NLINE
      DO 21 I=1,NLINE
21    WRITE(2)(IP(I,J),J=1,NSIZE)
      CLOSE 2
      TYPE '   '
      ACCEPT 'WISH TO CONTINUE(1)---OR STOP(0)---->',IWT
      IF(IWT.EQ.1)GO TO 1000
      STOP
      END
```

```
C          ******************************
C              FILENAME REMOVBROW
C          ******************************

C          THIS PROGRAM IS USED TO DETECT THE STRAIGHT LINE  OF
C          THE 'EYE-BROWS' IN THE PISTON HEAD BY TEMPLATE MATCHING.
C          THE BOUNDARY OF THE EYE-BROW IS THEN FOLLOWED AND THIS
C          FEATURE IS REMOVED
C
           DIMENSION NAME(5),IFILE(5),IP(128,128),IR(64),IC(64)
           INTEGER XCORD,YCORD,LXT(8),LYT(8)
           DIMENSION KA(10),KB(10),KC(10)
           COMMON/B1/LXT,LYT
           DATA IBLNK/'  '/
1000       ACCEPT ' # OF PIXELS PER LINE =',NLINE
           ACCEPT ' # OF LINES           =',NSIZE
           DO 1 I=1,5
1          NAME(I)=IBLNK
           WRITE(10,2)
2          FORMAT('  ',1X,'INPUT FILENAME :- ',Z)
           READ(11,3)(NAME(I),I=1,5)
3          FORMAT(5A2)
           OPEN 1,NAME,LEN=2*NSIZE,REC=NLINE
           DO 4 I=1,NLINE
4          READ(1)(IP(I,J),J=1,NSIZE)
           CLOSE 1
C
C          'ITH' GIVES THE THRESHOLD LEVEL FOR TEMPLATE RESPONSE
C
           ACCEPT 'THRESHOLD LEVEL = ',ITH
C
C          SIZE OF TEMPLATE AND STARTING POSITION IS SPECIFIED
C
5000       ACCEPT 'BEG ROW OF TEMP=',INITR
           ACCEPT 'END ROW OF TEMP=',IFINR
           ACCEPT 'BEG COL OF TEMP=',INITC
           ACCEPT 'END COL OF TEMP=',IFINC
C
C          IMAGE DIVIDED INTO FOUR QUADRANTS.EACH QUADRANT
C          CONSIDERED SEPARATELY
C
           ACCEPT 'BEG ROW OF QUAD=',IRBQ
           ACCEPT 'END ROW OF QUAD=',IREQ
           ACCEPT 'BEG COL OF QUAD=',ICBQ
           ACCEPT 'END COL OF QUAD=',ICEQ
```

```
C
C       DETECTING A STRAIGHT LINE USING A LINE  TEMPLATE
C
        IF(ICBQ.EQ.1)GO TO 600
        GO TO 601
600     JUM=1
        GO TO 602
601     JUM=-1
602     N1=0
        DO 10 I=INITR,IFINR
        N1=N1+1
        IR(N1)=I
        N2=0
        DO 15 J=INITC,IFINC
        N2=N2+1
        IC(N2)=J
        ISUMA=0
        DO 100 IA=1,10
        IMA=IA+I
        KA(IA)=(-1)*IP(IMA,J-1)
        ISUMA=ISUMA+KA(IA)
100     CONTINUE
        ISUMB=0
        DO 200 IB=1,10
        IMB=IB+I
        KB(IB)=2*IP(IMB,J)
        ISUMB=ISUMB+KB(IB)
200     CONTINUE
        ISUMC=0
        DO 300 NC=1,10
        IMC=I+NC
        KC(NC)=(-1)*IP(IMC,J+1)
        ISUMC=ISUMC+KC(NC)
300     CONTINUE
        IRESP=ISUMA+ISUMB+ISUMC
        IF(IRESP.GT.ITH)GO TO 20
15      CONTINUE
10      CONTINUE
C
C       BOUNDARY FOLLOWING OF  EYE-BROW
C
C       I1,J1 ARE CO-ORDS OF FIRST BORDER POINT
C
20      I1=IR(N1)+5
        J1=IC(N2)
C
C       STORE STARTING POINT
C
        XCORD=I1
        YCORD=J1
```

```
C
C       ID, JD ARE CO-ORDS OF PREVIOUS POINT
C
        ID=I1
        JD=J1+JUM
        IP(I1,J1)=400
202     CALL NABOR(I1,J1,ID,JD,LXT,LYT)
        DO 101 K=2,8
        L=K
        IF(IP(LXT(L),LYT(L)).EQ.0.OR.IP(LXT(L),LYT(L)).NE.255)GO TO 102
101     CONTINUE
102     IX1=LXT(L)
        IY1=LYT(L)
        JON=J1+1
        IF(IY1.EQ.JON.AND.IX1.EQ.I1)GO TO 400
        GO TO 401
400     I1=IX1
        J1=IY1
        IP(I1,J1)=300
        IF(I1.EQ.XCORD.AND.J1.EQ.YCORD)GO TO 500
        ID=LXT(L-1)
        JD=LYT(L-1)
        GO TO 202
401     I1=IX1
        J1=IY1
        IP(I1,J1)=400
        IF(I1.EQ.XCORD.AND.J1.EQ.YCORD)GO TO 500
        ID=LXT(L-1)
        JD=LYT(L-1)
        GO TO 202
C
C       REMOVING INHERENT FEATURE (EYE-BROWS) FROM IMAGE
C
500     DO 501 I=IRBQ,IREQ
        J=ICBQ
503     IF(IP(I,J).EQ.400)GO TO 502
        J=J+1
        IF(J.GT.ICEQ)GO TO 501
        GO TO 503
502     IP(I,J)=255
        J=J+1
        IF(J.GT.ICEQ)GO TO 501
        IF(IP(I,J).EQ.400.OR.IP(I,J).EQ.300)GO TO 502
        IF(IP(I,J).EQ.255)GO TO 505
        GO TO 504
505     JM=J+1
507     IF(IP(I,JM).EQ.400.OR.IP(I,JM).EQ.300)GO TO 509
        GO TO 510
509     J=JM
        GO TO 502
510     JM=JM+1
        IF(JM.GT.15)GO TO 501
        GO TO 507
```

```
504     IP(I,J)=255
        J=J+1
        IF(J.GT.ICEQ)GO TO 501
        IF(IP(I,J).EQ.400.OR.IP(I,J).EQ.300)GO TO 506
        GO TO 504
506     IP(I,J)=255
        J=J+1
        IF(J.GT.ICEQ)GO TO 501
        IF(IP(I,J).NE.255)GO TO 506
        KMC=J+3
        IF(IP(I,KMC).NE.255)GO TO 504
501     CONTINUE
        ACCEPT 'WISH TO CHANGE CO-ORDS(1)--OR NOT(0)--->',IST
        IF(IST.EQ.1)GO TO 5000
        DO 11 I=1,5
11      IFILE(I)=IBLNK
        WRITE(10,12)
12      FORMAT('  ',1X,'OUTPUT FILENAME :- ',Z)
        READ(11,13)(IFILE(I),I=1,5)
13      FORMAT(5A2)
        OPEN 2,IFILE,LEN=2*NSIZE,REC=NLINE
        DO 14 I=1,NLINE
14      WRITE(2)(IP(I,J),J=1,NSIZE)
        CLOSE 2
        ACCEPT 'WISH TO CONTINUE(1)---OR STOP(0)---->',IWT
        IF(IWT.EQ.1)GO TO 1000
        STOP
        END
```

```
      SUBROUTINE NABOR(I1, J1, ID, JD, LXT, LYT)
C     --------------------------------
C     FILE NAME :- NABOR
C     --------------------------------
C
C     THIS  SUBROUTINE COMPUTES THE COORDINATES OF EIGHT
C     NEIGHBOURING PIXELS AROUND A BORDER POINT, STARTING
C     SEQUENTIALLY FROM A 'PREVIOUS' POINT
C
      DIMENSION LXT(8),LYT(8)
      K1=JD-J1
      K2=ID-I1
      LXT(1)=ID
      LXT(2)=LXT(1)+K1
      LXT(3)=LXT(2)-K2
      LXT(4)=LXT(3)-K2
      LXT(5)=LXT(4)-K1
      LXT(6)=LXT(5)-K1
      LXT(7)=LXT(6)+K2
      LXT(8)=LXT(7)+K2
      LYT(1)=JD
      LYT(2)=LYT(1)-K2
      LYT(3)=LYT(2)-K1
      LYT(4)=LYT(3)-K1
      LYT(5)=LYT(4)+K2
      LYT(6)=LYT(5)+K2
      LYT(7)=LYT(6)+K1
      LYT(8)=LYT(7)+K1
      RETURN
      END
```

REFERENCES

[1] Gonzalez, R.C. and Wintz, P., "Digital Image Processing", Addison-Wesley Publishing Company, Inc., 1977.

[2] Castleman, Kenneth R., "Digital Image Processing", Prentice-Hall Publishing Company, Inc., 1979.

[3] Rosenfeld, A., "Digital Picture Processing" , Academic Press, New York, 1976.

[4] Rosenfeld, A., "Picture Processing by Computer", Academic Press, New York, 1969.

[5] Aggarawal, J.C., et al, "Computer Methods in Image Analysis", IEEE Press, 1977.

[6] Weszka, Joan S. and Rosenfeld, A., "Histogram Modification for Threshold Selector", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-9, No.1, January 1979.

[7] Perkins, W.A., "Area Segmentation of Images Using Edge Points", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, No.1, January 1980.

[8] Otsu, Nobuynki, "A Threshold Selection Method from Gray-level Histograms", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-9, No.1, January 1979.

[9] Davis, Larry S., et al, "Region Extraction by Averaging and Thresholding", IEEE Trans. on Systems, Man and Cybernetics, May 1975.

[10] Ridler, T.W. and Calvard, S., "Picture Thresholding Using An Iterative Selection Method", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-8, No.8, August 1978.

[11] Ullman, Julian R., "Bimarization Using Associative Addressing". Pattern Recognition, Vol.6, pp.127-135, 1974.

[12] Weszka, Joan S., and Rosenfeld, A., "Threshold evaluation techniques", IEEE Trans. on Systems, Man and Cybertetics, Vol. SMC-8, No.8, August 1978.

[13] Nakagawa, Yasmo and Rosenfeld, A., "Some Experiments on Variable Thresholding", Pattern Recognition, Vol.11, pp.191-204, 1979.

[14] Chow, C.K. and Kaneko, T., "Automatic Boundary Detection of the Left Ventricle From Cincangiograms", Comput. and Biomed. Res., Vol.5, pp.388-410.

[15] Gonzalez, R.C. and Fittes, B.A., "Gray Level Transformation for Interactive Image Enhancement", Proc. Second Conf. on Remotely Manned Systems, pp.17-19.

[16] Hall, E.L., "Almost Uniform Distributions for Computer Image Enhancement", IEEE Trans. Computers, Vol.C-23, No.2, pp.207-208.

[17] Sklansky, J., "Image Segmentation and Feature Extraction", IEEE Trans. on Systems, Man and Cybernetics, Vol.SMC-8, No.4, April 1978.

[18] Prager, John M., "Extracting and Labelling Boundary Segments in Natural Scenes", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.PAMI-2, No.1, January 1980.

[19] Chottera, A., et al, "Image Processing for Quality Control of Manufactured Parts", IEE Conference on Digital Processing of Signals in Communications, 1981, Loughborough, U.K.

[20] Weszka, Joan S. and Rosenfeld, A., "Threshold Selection 4", University of Maryland Computer Science Center TR-336, Oct.1974.

[21] Weszka, Joan S. and Rosenfeld, A., "Threshold Selection 5", Univ. of Maryland Computer Science Center, Tech.Rep.-349, Feb.1975.

[22] Weszka, Joan S., et al, "A Technique for Facilitating Threshold Selection for Object Extraction From Digital Pictures", Univ. of Maryland Computer Science Center, TR-241, May 1973.

[23] Humsnel, R.A., "Histogram Modification Techniques", Univ. of Maryland Computer Science Center, TR-329, Sept. 1974.

[24] Prewitt, J.M.S. and Mendelsohn, M.L., "The Analysis of Cell Images", Annals N.Y. Acad.Sci. 128, pp.1031-1053, 1966.

[25] Rosenfeld, A. and Kak, A.C., "Digital Picture Processing", New York: Academic, 1975.

VITA AUCTORIS

Mohammed Arif JANJUA

1956 Born on January 19th, in London, England.

1967 Completed Elementary School education at Karachi Grammar School, Karachi, Pakistan.

1973 Completed High School at Cadet College Hasan Abdal, Hasan Abdal, Pakistan.

1979 Graduated from the University of Engineering and Technology, Lahore, Pakistan, with the Degree of Bachelor of Science (Engineering) in Electronics and Communications.

1981 Candidate for the Degree of Master of Applied Science in Electrical Engineering at the University of Windsor, Windsor, Ontario, Canada.