Microcomputer implementations of image processing algorithms.

Mahbub Iftekhar Ahmed

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
https://scholar.uwindsor.ca/etd/2724

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000 ext. 3208.
NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS RÉCU
MICROCOMPUTER IMPLEMENTATIONS OF IMAGE PROCESSING ALGORITHMS

by

Mahbub Iftekhar Ahmed

A thesis presented to the University of Windsor in partial fulfillment of the requirements for the degree of M.A.Sc in Department of Electrical Engineering

Windsor, Ontario, 1984

(c) Mahbub Iftekhar Ahmed, 1984
ABSTRACT

An extensive investigation has been made in this work on a class of image processing algorithms for their efficient microcomputer implementations.

Studies have been made on the existing algorithms for their suitability for microcomputer implementations from the viewpoints of memory size requirements and the computational speeds acceptable for different practical applications. Studies also include the choice of the programming language and the image size to reach the acceptable overall efficiencies of the implemented algorithms.

Suitable algorithms have been implemented directly in their existing form. To improve the efficiencies of some of the existing algorithms, modifications and new assumptions have been incorporated in the implementation and their evaluations have been made on the basis of the standards of the original one.

To overcome the limitations of some of the existing algorithms, this work also has developed some new approaches and has discussed in detail how those pointed out limitations have been overcome.

Finally, all the implemented algorithms have been tested on a number of images and their results have been presented.
ACKNOWLEDGEMENT

With all my heart, I remember Abba, my deceased father, who was the primary source of all my encouragement and who thrived on my achievements and successes, and has been finally deprived of his reward. I owe my existence to him.

The author expresses his sincere gratitude towards his supervisors Dr. W.C. Miller and Dr. G.A. Jullien, for all the valuable discussions, suggestions and constructive criticism throughout the study period. Without their active guidance and support, this work could not be accomplished. The valuable advices from Dr. M. Shridhar are also gratefully acknowledged.

Thanks is due to all other faculty members and all the graduate students of this department for their constant assistances.

Special gratitude to Maa and all other family members, for the courage they offered me during my bad times. Special acknowledgement is due for Lucky and Nipu for their sincere moral support.
# CONTENTS

**ABSTRACT** ............................................ ii

**ACKNOWLEDGEMENT** ..................................... iii

## Chapter

<table>
<thead>
<tr>
<th>I.</th>
<th>INTRODUCTION ........................................... 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Digital Image Processing System: ....................... 2</td>
</tr>
<tr>
<td></td>
<td>Microcomputers For Image Processing: ................... 5</td>
</tr>
<tr>
<td></td>
<td>Class Of Image Processing Algorithms Considered: ... 7</td>
</tr>
<tr>
<td></td>
<td>For Implementations: .................................. 7</td>
</tr>
<tr>
<td></td>
<td>Histogram Modification: ................................ 7</td>
</tr>
<tr>
<td></td>
<td>Image Thresholding: .................................... 10</td>
</tr>
<tr>
<td></td>
<td>Border Following: ...................................... 10</td>
</tr>
<tr>
<td></td>
<td>Template Matching: .................................... 11</td>
</tr>
<tr>
<td></td>
<td>Thesis Organisation: ................................... 11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>II.</th>
<th>THEORETICAL DISCUSSIONS OF A CLASS OF IMAGE PROCESSING ALGORITHMS ................................................. 13</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Histogram Modification: ........................................... 13</td>
</tr>
<tr>
<td></td>
<td>Histogram: ................................................................... 14</td>
</tr>
<tr>
<td></td>
<td>Histogram Modification Techniques: ................................ 14</td>
</tr>
<tr>
<td></td>
<td>Histogram Equalisation For Image Enhancement: ................................................. 15</td>
</tr>
<tr>
<td></td>
<td>Histogram Modification By Direct Specification: ................................................. 19</td>
</tr>
<tr>
<td></td>
<td>Image Thresholding: ................................................ 21</td>
</tr>
<tr>
<td></td>
<td>Binary Level Thresholding: ........................................ 22</td>
</tr>
<tr>
<td></td>
<td>Multilevel Thresholding: .......................................... 25</td>
</tr>
<tr>
<td></td>
<td>Selection Of The Threshold Value: ..................................... 26</td>
</tr>
<tr>
<td></td>
<td>Scatter Plotting Technique: ........................................ 27</td>
</tr>
<tr>
<td></td>
<td>Histogram of Points Having Low And High Edge Values: ..................... 28</td>
</tr>
<tr>
<td></td>
<td>Feature Extraction: .................................................. 30</td>
</tr>
<tr>
<td></td>
<td>Boundary Following: ................................................ 31</td>
</tr>
<tr>
<td></td>
<td>Template Matching: .................................................. 32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>III.</th>
<th>IMPROVEMENTS AND MODIFICATIONS OF THE EXISTING ALGORITHMS ................................................................ 38</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Histogram Modification By Direct Specification: ..................... 38</td>
</tr>
<tr>
<td></td>
<td>Automatic Threshold Value Selection: ..................................... 40</td>
</tr>
<tr>
<td></td>
<td>Template Matching: ....................................................... 43</td>
</tr>
</tbody>
</table>
IV. MICROCOMPUTER IMPLEMENTATIONS OF IMAGE PROCESSING ALGORITHMS... 49

Computing Gray Level Density Function: ... 49
Microcomputer Implementation Of Histogram Equalisation: ... 52
Implementation Of The Direct Histogram Modification Technique: ... 58
Implementation Of The Thresholding Technique: ... 67
Microcomputer Implementation Of The Scatter Plot Technique: ... 71
Microcomputer Implementation Of The Border Following Technique: ... 76
Microcomputer Implementation Of Template Matching: ... 81
Direct Detection Of Linear Edge Or Line: ... 83
Implementation Of Spurious Feature Cancellation Algorithm: ... 88

V. SUMMARY OF THE WORK ACCOMPLISHED FOR THIS THESIS ... 91

Summary and conclusions ... 91

Appendix ... page

A. ... 93

REFERENCES ... 96

VITA AUCTORIS ... 98
Chapter I
INTRODUCTION

Information scientists have long recognized that "one picture is worth a thousand words," and they have been continuing their devotion to the extensive research and developments in the area of pattern recognition and analysis by computer. Practical applications of such discipline have already been established in the area of biomedical image processing for diagnosis; recognition of patterns, such as characters, fingerprints and moving objects; remote sensing; industrial inspection for quality control; robotic vision; military intelligence and communication data compression.

In another domain of science, the introduction of microprocessors has caused a tremendous impact upon the field of digital systems design. Microprocessors have opened vast new fields for applications of digital systems where, previously, the introduction of larger systems were not economically feasible.

Development of cost effective intelligent processors is a major concern to research workers, which in turn can promote better man-machine interactions for satisfactory information analysis. The recent developments of microproces-
sors have increased the probability of replacing the minicomputers in various fields of applications by cost effective microcomputers. This also gives an opportunity to investigate the introduction of microcomputer-aided artificial intelligence processors for the applications in the limited fields of automatic information processing.

1.1 DIGITAL IMAGE PROCESSING SYSTEM:

The block diagram of a general purpose image processing system has been shown in figure (1.1.a).

The image, which is to be processed, is digitized by the image digitizer into an integer array. The digitizer may use a TV camera, Flying spot scanner or Microdensitometer to pick up the video signal.

The generated image data are normally transferred into the bulk storage by the computer. While processing, the computer retrieves data from the storage into its main memory and manipulates those according to the requirements.

The images stored in the system, before and after processing, can be displayed in the display media connected to the system. It may be a TV monitor or a hard copy generator.

The whole process is controlled by an operator from the console of the system.
Figure (1.1.a): Block diagram of a general purpose image processing system.
Image processing functions require two items:

1. an online bulk storage system for the image database.

2. fast processing to extract information from the image.

The structure of an image processing system depends upon the priorities of the above requirements. For example, a high resolution image database, in most of the cases, is too large to fit completely in the memory of small computers. In this case, adequate bulk storage capabilities are required to be provided for those systems. One of the popular bulk storage media is high-speed disk.

To meet the required fast processing of the image, in some cases, the existing devices use dedicated hardware to speed up critical functions. The hardware remains connected to the conventional computer which performs the noncritical functions by software.

Depending upon the processing requirements, and cost effectiveness, the processing unit of an image processing system can vary from microcomputer-based systems, in special cases, to large computers capable of handling large variety of functions.
1.2 **MICROCOMPUTERS FOR IMAGE PROCESSING:**

The introduction of microcomputers, in the possible cases, for the purpose of image processing can improve the cost effectiveness of the systems highly. The recent developments of the microcomputers are tremendous and some of the advanced general purpose microprocessors are:

1. Intel 8085
2. Intel 8086
3. Motorola 6800
4. Motorola MC68000
5. Zilog 80
6. Zilog 8000

Earlier, it has been mentioned that two important requirements for an image processing system are large memory size and high computational speed. Table (1.2.a) shows the maximum memory size of the above mentioned microcomputers that can be addressed directly.

The size of the image database depends upon the resolution required by the problem to be solved. The bigger the database, the more the computational time required. Again, the speed of the hardware of the present microcomputers in executing their instruction sets are slower than the larger computers. For this reason, for the microcomputer aided image processing, some sort of optimisation is required between the size of the image database and the processing time, which can be tolerated by the problems to be solved.
<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>MEMORY SIZE IN BYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel 8085</td>
<td>65,536</td>
</tr>
<tr>
<td>Intel 8086</td>
<td>1,048,576</td>
</tr>
<tr>
<td>Motorola 6800</td>
<td>65,536</td>
</tr>
<tr>
<td>Motorola MC 68000</td>
<td>8,388,608</td>
</tr>
<tr>
<td>Zilog Z80</td>
<td>65,536</td>
</tr>
<tr>
<td>Zilog'Z8000</td>
<td>8,388,608</td>
</tr>
</tbody>
</table>

Table (1.2.a): Memory sizes of different microcomputers
An investigation has been carried out using an Intel-8085 based system, and although the system can accommodate a 128 x 128 image database after loading the system routines and user's program, a 64 x 64 image database was finally selected, based upon processing time requirements for various practical applications. It has been found out that the maximum time required for an individual algorithm is about 2 seconds. The 8085 based microcomputer assembler language has been selected for all programming examples.

1.3 **CLASS OF IMAGE PROCESSING ALGORITHMS CONSIDERED FOR IMPLEMENTATIONS:**

The image processing technique, as has been stated earlier, has established its applications in the area of industrial inspections for quality control along with the other areas. In this study, we concentrate on applications of microcomputers for image processing in the area of automatic quality control of manufactured parts. The algorithms, which are frequently used for the above mentioned purpose, have been considered for microcomputer implementations and those have been introduced below.

1.3.1 **Histogram Modification:**

The histogram of an image is the probability density function of grey levels of the pixels. It provides a global description of the appearance of the image. Consider the
histograms shown in the figures (1.3.a) and (1.3.b). The histogram of the first figure represents a dark image, because its grey level density is mainly concentrated in the lower half of the grey level scale. Because of the opposite reason, the second histogram represents an image with a lighter tone.

The characteristics of an image can be enhanced by modifying its histogram. Modifying the grey level densities of an image, so that an approximately uniform histogram results, generates a considerable improvement in the contrast of that image. Again, by modifying the histogram of an image according to direct specifications, information present in certain grey level ranges can be highlighted.
Figure (1.3.a): Sample histogram for a dark image

Figure (1.3.b): Sample histogram for a bright image.
1.3.2 **Image Thresholding:**

Image thresholding is a segmentation technique, which separates different regions of interest present in the image from the rest of its components. It divides the grey level scale into different bands and all the pixels within a particular band are assigned a particular value. In the binary level thresholding technique, the grey level scale is divided into just two bands and for multilevel thresholding technique the number of bands is greater than two.

The choice of the threshold value should be such that a negligible amount of required information is lost. The threshold value should ideally separate the various subpopulations present in the histogram. An automatic thresholding technique is important for implementing a total computer aided image analyser.

1.3.3 **Border Following:**

Although the thresholding technique separates the object from the background, such separated region can not always provide all the information required for total analysis. For example, to calculate the area within the object, the co-ordinates of all the boundary points are required.

The boundary between two different regions can be outlined by various methods. Scanning the image row wise for transitions in grey levels and again scanning the image column wise for the same can locate the border points. Algori
items are also available, which locate the first border point and then trace the entire contour of that region.

1.3.4 Template Matching:

Template matching techniques search for some invariant regional property present in an image. Among the simplest types, the example of templates may be a point detection template, a line detection templates and an edge detection template. Normally, a response is calculated when the template is moved over the image. If this response is less than a specified threshold value, then the region is considered as not being detected and when the response is over the threshold value the region of interest is considered as being detected at the position of the template.

1.4 THESIS ORGANISATION:

All the work for this thesis has been implemented using an Intel-8085 based system. The word 'microcomputer' used throughout in this thesis in relation to the results obtained is implied for the Intel-8085 based system.

Chapter II of this thesis gives the detail background theorems of the class of image processing processing algorithms considered for implementation in the microcomputer. Discussions have been separated in different sections for different algorithms.
Chapter III contains the discussions regarding the improvements, modifications made over the existing algorithms and the new approaches developed. Limitations of some algorithms have been pointed out and their modifications have been suggested. Modifications have also been suggested in the way of application of some existing algorithms. New computational formulae have also been introduced.

Chapter IV deals with the implementations of the algorithms. Efficient ways for the implementations have been followed and they have been pointed out. Some assembler language benefits have been utilised in the implementations. Detailed discussions have been included in this chapter algorithmwise.

Chapter V finally summarises the author's contributions in this thesis and discusses the conclusions derived.
Chapter II

THEORETICAL DISCUSSIONS OF A CLASS OF IMAGE PROCESSING ALGORITHMS

Some image processing algorithms, which have been found suitable for microcomputer implementation, have been discussed in this chapter from the viewpoint of their theoretical background. The discussion is based on their existing form. In some cases some modifications have been suggested, and these are discussed separately in the succeeding chapters.

2.1 HISTOGRAM MODIFICATION:

The histogram of an image is the probability density function of grey levels of pixels. It provides an entire description of the appearance of the image. For example, a histogram which shows the peak value in the range of the lower grey levels, indicates fairly dark characteristics of an image. On the other hand, a peak in the higher grey levels indicates a predominantly light tone. The histogram of an image can provide as important information as the ranges of grey levels occupied by the object in the image and by the background. The characteristics of an image can be enhanced by modifying the histogram of that image. Two methods have been discussed in this chapter to achieve enhancement of an image by modifying its histogram.
2.1.1 **Histogram:**

For computer aided image processing the grey levels assume discrete values and the histogram is a plot of the probability of occurrence of the pixels of different grey levels against the grey level.

Mathematically, probability is defined by

\[ Pr(r_k) = \frac{n_k}{n} \quad 0 \leq r \leq 1 \quad Eq.[2.1.a] \]

\[ k=0,1,2,\ldots\ldots,L-1 \]

Where, \( Pr(r_k) \) is the probability of occurrence in the \( k \)th grey level,

\( n_k \) is the number of occurrences of the \( k \)th grey level

\( n \) is the total number of pixels and

\( L \) is the total number of grey levels.

Although the histogram is defined as a plot of \( Pr(r) \) versus grey level, in this thesis a plot of the number of occurrences of different grey levels against the grey level will also be referred to as a histogram.

2.1.2 **Histogram Modification Techniques:**

Different techniques have already been developed to modify the histogram of an image to improve the image characteristics according to the requirements of the application. It has been found that a considerable improvement in the contrast of the image can be achieved by modifying the grey level densities in the image to result in an approximately
uniform histogram. This modified histogram is referred to as 'equilised'. Useful result can also be obtained by modifying the histogram to a specified shape; for example - to select a threshold value for separating the object of an image from its background. Frequently it has been found very effective to modify the shape of the histogram of the image to the shape of a multimodal Gaussian density function.

Two different histogram modification techniques are described below.

2.1.2.1 Histogram Equalisation For Image Enhancement:

The objective of the technique is, as stated earlier, to transform a given image into one that gives a uniform grey level density function.

Let \( r \) and \( s \) represent the grey level of the pixels in the image to be transformed and the transformed image respectively. For simplicity, in the following discussions, the grey levels \( r \) and \( s \) will be assumed to be normalised, that is

\[
0 \leq r \leq 1 \\
0 \leq s \leq 1
\]

The level 0 represents the black and the level 1 represents the white.

The purpose of the technique is to derive a transformation function

\[
s = T(r). \quad \text{EQ.}[2.1.b]
\]

which fulfills the objective of the technique.
Two imposed constraints for the transformation are

1) $T(r)$ is a single valued and monotonically increasing function in the interval $0 \leq r \leq 1$ to preserve the order from black to white and

2) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$, to guarantee a mapping that is consistent with the allowed range of grey levels.

The inverse transformation from $s$ back to $r$ will be denoted by

$$r = T^{-1}(s) \quad 0 \leq s \leq 1 \quad \text{EQ-[2.1.c]}$$

where, it is assumed that $T^{-1}(s)$ also satisfies conditions (1) and (2) mentioned above with respect to the variable $s$.

The grey levels in an image can be considered to be random variables in the range of $[0,1]$. Assuming them to be continuous variables for the time being, they can be characterised by their probability density functions $Pr(r)$ and $Ps(s)$ respectively.

By probability theory, if $Pr(r)$ and $T(r)$ are known and if $T^{-1}(s)$ satisfies condition (1), then the probability density function of the transformed grey level is given by the relation

$$Ps(s) = \left[ Pr(r) \cdot \frac{dr}{ds} \right]_{r = T^{-1}(s)} \quad \text{EQ-[2.1.d]}$$

Histogram modification techniques modify the probability density function of the grey levels of an image through the transformation function $T(r)$.

For the histogram equalisation technique, the transformation function $T(r)$ has been defined as the cumulative dis-
tribution function of $r$. Mathematically, the definition is as follows:

$$s = T(r) = \int_0^r P_r(w) \, dw \quad 0 \leq r \leq 1$$

Eq. [2.1.e]

where, $w$ is the dummy variable of integration. This transformation has been shown in figure (2.1.a).

Since the cumulative distribution function increases monotonically from 0 to 1 as a function of $r$, it satisfies the two conditions previously imposed on the transformation function.

The probability density function for the transformed image can be calculated by equations (2.1.d) and (2.1.e) as follows:

$$\frac{ds}{dr} = P_r(r)$$

by equation (2.1.e)

Now

$$P_s(s) = \left[ P_r(r) \cdot \left( \frac{dr}{ds} \right) \right]_{r = T^{-1}(s)}$$

$$= \left[ 1 \right]_{r = T^{-1}(s)}$$

$$= 1 \quad 0 \leq s \leq 1$$

Eq. [2.1.f]

which is a uniform density within the interval of definition of the transformed variable $s$. This result is independent of the inverse transformation function and it is important because it is not always possible to calculate $T^{-1}(s)$. 
Figure (2.1.a): Sample transformation function for histogram equalization
So far the variables has been considered as continuous, but for computer implementation the above concepts are required to be formulated in digitized form. In that attempt the probability density has been interpreted as:

\[ \Pr(r_k) = \frac{n_k}{n} \quad 0 \leq r \leq 1 \quad \text{EQ. [2.1.g]} \]

\[ k=0,1,2,\ldots,L-1 \]

The definition of each of the variables is the same as given in section (2.1.1).

The discrete form of the cumulative distribution function can be assumed to be represented by the relation

\[ s = T(r_k) = \frac{1}{n} \sum_{j=0}^{k} n_j = \sum_{j=0}^{k} \Pr(r_j) \quad 0 \leq r \leq 1 \quad \text{EQ. [2.1.h]} \]

\[ k=0,1,2,\ldots,L-1 \]

2.1.2.2 Histogram Modification By Direct Specification:

Although the histogram equalization technique produces useful results for image enhancement, it is not always suitable for interactive image processing applications because of its nature of generating only one result - a uniform histogram approximation.

The above mentioned result improves the contrast in an image, but this may not be the only desired objective in an image processing application using histogram modification. In some important application, it may be desirable to highlight certain grey level ranges. One efficient way to satisfy such a requirement is to directly specify the desired
histogram and then to modify the pixels in the original image according to the requirements of the specified image.

Let $Pr(r)$ and $Pz(z)$ be the probability density of the original image and the desired image respectively. If the original image is histogram equalised, then the new pixels $s'$ in the transformed image can be represented as

$$ s = T(r) = \int_0^r Pr(w) \, dw \quad \text{EQ.} \quad [\text{2.1.i}] $$

If the desired image were available and it were also histogram equalised, the transformed pixel $v'$ of that transformed image would be

$$ v = G(z) = \int_0^z Pz(w) \, dw \quad \text{EQ.} \quad [\text{2.1.j}] $$

The inverse process $z = G^{-1}(v)$ would give back the pixels of the desired image.

Since the histogram equalisation technique leads to a single result, irrespective of the probability density function inside the integration, the result $s'$ of equation (2.1.i) and the result $v'$ of equation (2.1.j) are expected to be the same. Now in the inverse process of getting back $z'$, instead of using $v'$, $s$ can be used (that is $G^{-1}(s)$ is expected to give back the value $z'$ for the desired image).

The above mentioned principle is the backbone of the histogram modification technique by direct specification. The formulation is of course very much hypothetical in the sense that the values $z'$ are precisely the desired values which we are trying to obtain.
2.2 **IMAGE THRESHOLDING:**

Thresholding is a technique of image segmentation to separate different regions of interest in an image from the rest of its components. "Region of interest" is a function of the problem being considered. In this approach the grey scale is divided into bands and the pixels within a particular band are assigned to a particular grey level.

Thresholding is very frequently applied for the purpose of separating an object from its background. If the grey level distribution of the pixels of the object and the background are concentrated into separable bands in the grey scale, then it is possible to choose an appropriate threshold value; pixels having grey levels above the threshold value are assigned with one value and the pixels with grey levels below the threshold value are assigned a different value, thus separating the two bands of grey levels into separate regions in the image. In many applications, more than one threshold value is used where it is necessary to separate more than two bands of grey levels. The choice of the threshold value should be such that a negligible amount of required information is lost.

The selection of an appropriate threshold value is not necessarily straightforward for every image processing application and accurate threshold selection is of paramount importance in many image processing problems. In the following sections the technique of thresholding an image at a single level or at multilevels is discussed in detail.
2.2.1 **Binary Level Thresholding:**

Binary level thresholding transforms an image into one with two grey levels - black and white. In this technique, one threshold value is chosen and the pixels having grey levels higher than that value are set to the white level, otherwise they are set to the black level.

If the average distribution of the pixels of an object is significantly different from that of the background, then this technique can be applied very straightforwardly to separate the object from its background. A threshold value can be chosen by examining the histogram of the image. If two peaks are found in the histogram, it is relatively easy to select the appropriate threshold value which separates the peaks.

The histogram of figure (2.2.a) can be taken as an example. The histogram is bimodal and if the image is of a light object in a dark background, then the first subpopulation of the histogram corresponds to the background of the image and the second subpopulation corresponds to the object. If a threshold value is chosen at the valley of two modes, which has been shown as 'T' in the figure, the background will be transformed into a black region in the processed image and the object will be transformed into a white region.
Figure (2.2.a): Bimodal histogram of an image

Figure (2.2.b): Multimodal histogram of an image
It will be noted always that the threshold value will be different for different images and choosing the threshold value may not be always that straightforward. For example, if the object population and the background population are not comparable in size, that is one is much greater than the other or if their average values are not significantly different, then it is very difficult to set a threshold value that will successfully separate the two populations.

Therefore, for binary level thresholding, the selection of the threshold value will be easiest when the histogram of the image is clearly bimodal with comparable peaks and distinctly separate average values.

The histogram of an image will be bimodal if

1. The image under consideration consists of objects on a background where the probability density function of the grey levels for any small region of the picture consisting only of the object is unimodal.

2. The grey levels of adjacent points interior to the object or to the background are highly correlated, but at the boundary between the object and the background the adjacent points are highly uncorrelated; and

3. The average grey levels of the object and the background are significantly different from one another, these populations are comparable in size and the object population and the boundary population take place on a small range of intensity values.
2.2.2 Multi-level Thresholding:

There may arise some cases where it is extremely difficult to extract required information from an image by selecting only one threshold value. For example, an image whose histogram has more than two modes as shown in figure (2.2.b) can be considered.

For such an image it is required to select more than one threshold value; in the case of figure (2.2.b) three levels $T_1, T_2, T_3$ are required to extract objects from the background. This technique is known as Multi-level Thresholding and the processed image will be represented by more than just two levels. All levels lying between 0 and $T_1$ in the original image would have a certain value, levels between $T_1$ and $T_2$ would have another value and so on. If $N$ is the number of selected threshold values, then the total number of levels in the transformed image will be $N+1$.

As in the case of binary level thresholding, in most of the cases it is not very easy to separate various subpopulations of the histogram of the original image. This makes it difficult to select the threshold values. One of the most commonly encountered problems is that some ranges of grey levels occur a significantly greater number of times than the other ranges. When such a histogram is plotted, those ranges of grey levels play a dominant role in the plot since the entire histogram is plotted on the same scale and observing the grey levels with low probability becomes difficult.
One of the most efficient ways to handle such a difficulty is to modify the histogram of the original image by the method of Histogram Modification by Direct Specification.

2.2.3 Selection Of The Threshold Value:

It has been stated earlier that the selection of a threshold value to separate an object from its background is not always straightforward. If the pixels are distributed almost uniformly over the entire range of the grey level scale, or if the histogram has peaks either very close together or very unequal in size, then it is difficult to detect a valley between the object and the background populations.

In order to overcome this problem, one can go for transforming the original image into one which has distinctly separable subpopulations in the histogram. Previous works[10] of this department suggest the technique of image transformation by direct histogram specification in this connection.

Specifying the desired histogram correctly, so that it enhances the required information in the image, is very much important in this method.

Gaussian density functions have been applied in previous work[11] for specifying the desired histograms, but the means and standard deviations were chosen on a purely trial
and error basis. Further work has been done[11] in the same
department in the area of automatic threshold selection, using Gaussian Fitting Techniques, Iterative Threshold Selection Methods and Scatter Plotting Techniques. Because of the types and sizes of the computations involved in the first two methods, they were not considered suitable for implementing in a microcomputer and only the Scatter Plotting technique has been considered.

2.2.3.1 Scatter Plotting Technique:

The rate of change of grey level at a point is often termed as the edge-value of that point. A plot of the edge value against the grey level is often termed as a Scatter Plot.

A B C
D E F G
H I J K
L M

Figure (2.2.c): Sample image for explaining the edge value operator.

To calculate the edge value of the point E in the sample image shown in figure (2.2.c), different types of edge operators have been described in the literatures. Some of them are:
(a) LAP, the laplacian operator, defined as

\[ |E-\frac{(A+B+C+D+F+H+I+J)}{8}| \quad \text{EQ.-[ 2.2.a ]} \]

(b) ROB, the Roberts Cross, defined as

\[ \max [ |E-C|, |B-F| ] \quad \text{EQ.-[ 2.2.b ]} \]

(c) DIF, the maximum of differences of average grey level in a pair of horizontal and vertically adjacent 2 x 2 neighbourhoods

\[ \frac{1}{4}[ \max [ |B+C+E+F-I-J-L-M|, |D+E+H+I-F-G-J-K| ] ] \quad \text{EQ.-[ 2.2.c ]} \]

2.2.3.2 Histogram of Points Having Low And High Edge Values:

Let us assume that a given image consists of objects on a background, where the grey levels of adjacent points interior to the object or to the background are highly correlated, while across the edges at which the object and the background meet, the adjacent points are significantly uncorrelated.

In the images, which satisfy the above mentioned assumptions, the points interior to the object and the background should generally have low edge values, since they are highly correlated. On the other hand, the points on the boundary of the object and the background should have high edge values.

Two types of modified histograms can be thought of for plotting by the above mentioned edge values:
(1) A weighted histogram, in which the points having low edge values are counted heavily. For example, if $|\Delta|$ is the edge value at a given point then a weight of $1/(1+|\Delta|^2)$ can be used. This gives a maximum weight to points having zero edge value and a negligible weight to the high edge value points.

(2) A weighted histogram, in which points having high edge values are heavily counted. For example, if $|\Delta|$ was the edge value of a point then a weight of $|\Delta|$ should eliminate points with zero edge values.

Let us consider a Scatter plot which gives more weights in counting the points having lower edge values, which are interior to the object or background. Let us also assume that the plot gives minimum weight in counting the points at the boundary of the object and the background, which have the highest edge values. Since a peak in a histogram corresponds to the interior of the object or the background, and a valley corresponds to the points at the border, in the above mentioned Scatter plot, a peak remains essentially the same, but a valley becomes deeper.

In the Scatter plot, which counts heavily the points having high edge values, there will be a peak corresponding to the points near the border between the object and the background, since they contribute mostly to the edge value. So a peak in a Scatter plot which gives more weights to the points having higher edge values and a valley in the Scatter
plot which gives more weights to the points having lower edge values should correspond to the threshold value for the image.

2.3 FEATURE EXTRACTION:

The ultimate objective of image processing is to either improve the information content in an image or to extract information from the image. The information content of the image can be enhanced by, for example, modifying the brightness or contrast. For various image processing applications, extracting information from an image is the ultimate goal. Thresholding an image can separate the object from the background, provided a proper thresholding value has been chosen. From this point of view, thresholding can also be classified as a process of information extraction from an image.

An alternate requirement for feature extraction may be to search for specific type of information within the object. This type of information may be the shape and size of different regions in the object. Two of the popular feature extraction techniques are Boundary Following and Template Matching. The Boundary Following technique detects the border points of a chosen region of the object and the Template Matching technique searches for a specific type of region in the image.
The feature extraction techniques, based on regional properties have a wide range of applications due to their inherent simplicity. In the following sections, some efficient ways of border following and template matching are discussed.

2.3.1 Boundary Following:

One of the very frequent needs of various image processing applications is to detect boundary points between an object and its background. Such information could be of great use for applications such as quality control of manufactured parts where the interest may be to determine the shape and size of the object to see whether it is faulty or not. This also has applications in determining the shape and size of organs in human bodies for biomedical image processing.

The border lines may be outlined using different algorithms. One approach may be to initially threshold the image properly to separate the object and the background. The next step is to scan the image; if a change in grey level occurs between two adjacent pixels, it denotes the presence of a border point. In order to detect boundary points in both horizontal and vertical directions, each pixel is required to be compared with two adjacent pixels—one of which is horizontally adjacent to it and the other is vertically adjacent. This method will determine all boundaries present
in the image. If the border line of only one region is required, then this method will not work. Another method which has proved to generate useful results is discussed in section (4.6).

2.3.2 Template Matching:

An array designed to detect some invariant regional property of an image is termed a Template. Because of the simplicity of using such a template, the technique has been widely accepted in segmentation applications.

Let \( w_1, w_2, \ldots, w_9 \) represents the coefficients in a 3x3 template, and let \( x_1, x_2, \ldots, x_9 \) be the grey levels of the image pixels covered by the template. If we define the vectors \( w \) and \( x \) as:

\[
w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_9 \end{bmatrix}
\]

and

\[
x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_9 \end{bmatrix}
\]
then, the inner product of the vectors $\mathbf{w}$ and $\mathbf{x}$ can be written as

$$\mathbf{w}^T \mathbf{x} = w_1 x_1 + w_2 x_2 + \ldots + w_3 x_3 \quad \text{EQ. [2.3.a]}$$

The region of interest in the image can be detected by setting the relationship

$$\mathbf{w}^T \mathbf{x} > T \quad \text{EQ. [2.3.b]}$$

where, $T$ is the specified threshold value. This procedure can be easily generalised for any arbitrary window size of $n \times n$.

Figure (2.3.a) through figure (2.3.e) show some sample templates for point and line detection.
Point detection a is fairly straightforward procedure. For example, consider the template of figure (2.3.a). If this template is moved around an image, it will give a zero response over a constant background, but when centered over an isolated point on a constant background, it will give the maximum response. The next level of complexity involves the detection of straight lines in an image. Figure (2.3.b) through figure (2.3.e) show examples of line detection templates for lines with different orientation. Line templates may also be used to detect edges by specifying the proper threshold value.

The theoretical discussions of the algorithms end at this point. Figure (2.3.f) through figure (2.3.m) show the images which have been processed by different algorithms for examples.
Figure (2.3.f): Piston head  
Figure (2.3.g): Blocks  
Figure (2.3.h): Strain gauge  
Figure (2.3.i): Dollar
Figure (2.3.j): Skull

Figure (2.3.k): Second strain gauge

Figure (2.3.l): Wooden blocks

Figure (2.3.m): Transmission gear
Chapter III

IMPROVEMENTS AND MODIFICATIONS OF THE EXISTING ALGORITHMS

An extensive study has been carried out on existing algorithms with respect to microcomputer implementation. Some algorithms have been found to be directly suitable for microcomputer implementation, but in other cases either the algorithms have been found not suitable to implement, or modifications in the algorithms were required to make them more efficient. Accordingly, in some cases totally new approaches have been made to solve the problem concerned and in other cases the problems have been tackled by modifying existing algorithms. This chapter gives a detailed insight into the modifications and improvements of the existing algorithms and the new approaches made for solving some image processing problems. Separate discussions for different algorithms are given in the following sections.

3.1 HISTOGRAM MODIFICATION BY DIRECT SPECIFICATION:

Refering to the discussion of section (2.1.2) the grey level z in the desired image by the technique of direct histogram modification is given by \( z = G^{-1}(s) \). Under such conditions \( s = T(r) = v = G(z) \).
Since we are dealing with discrete values, a perfect match between $s$ and $v$ can not always be ensured; therefore, some assumptions are required to be made for the digital computer implementation of the algorithm. Existing algorithm assumes the closest match between $s$ and $v$.

After a thorough investigation, it has been found that if the assumption is slightly modified, it greatly improves the speed of the algorithm. According to the modified assumption, the closest match between $s$ and $v$ has been used in the range of $v$ equal to or greater than $s$; that is, a closer match, if any, for which $v$ is less than $s$, will be ignored.

Taking the modified assumption into account, an algorithm has been developed (discussed further in section (4.3)), to implement the inverse transformation function $G^{-1}(s)$. The software for generating this inverse transformation is totally new and the important property of the algorithm is that it ensures an equal number of computations for each pixel, irrespective of its grey level. For a 64x64 image, the microcomputer takes about one second, in total, to produce the output. This is definitely the introduction of a fast solution to the problem concerned.

The designed algorithm creates a result chart of the inverse transformation $G^{-1}(s)$ for all the levels before the pixels in the original image are remapped. These results are stored in the memory locations whose logical addresses are equal to the grey level values.
3.2 AUTOMATIC THRESHOLD VALUE SELECTION:

The automatic image thresholding technique has been implemented through the Scatter plotting technique. This method is abundantly discussed in the literature, but to ensure fast computation, it has been implemented with slight modifications to the existing formulae.

The Robert's cross operator has been chosen, because of its inherent simplicity, to calculate the edge value at a point. The method has been found more efficient than others discussed in the literature [11]. To limit all the computations within dynamic range of two bytes, a scaling factor has been introduced. The operator has been used, with reference to the figure (2.2.c), in the form:

$$\delta = \frac{1}{4}\text{Max}[|E-C|, |B-F|]$$

As has been suggested by Watanebe [12], a weight equivalent to summing the edge values for each grey level has been used in the implementation. This approach has a disadvantage in that if the areas of the object and background are large, the sum of a large number of low edge values in the interior may be larger than or equal to the sum of the smaller number of high edge values at the border. Since the scaling factor has been implemented by rotating the accumulator, individual edge values which are less than four will not come into consideration, and this can handle such situations on a limited basis.
Figure (3.2.a): Improperly thresholded image of the piston head.

Figure (3.2.b): Improperly thresholded image of the strain gauge.
The edge value is a measure of the rate of change of grey levels at a point. Since the change in grey level can occur either in the horizontal direction or vertical direction, a new formula, which involves fewer computations, has been tested to calculate the edge value. Refering to figure (2.2.c), this formula to calculate the edge value of the point E is:

\[ \text{OPRTE} = \frac{1}{16} \left( |E-F| + |E-I| \right) \]

which takes into consideration one horizontally adjacent pixel and one vertically adjacent pixel only. This formula has been found to give very close results to that of the ROB operator in most cases; in many cases, it has been found to give a closer result to the actual value than the implemented form of the ROB operator.

It has been found that the Scatter plot may give the peak value, in some cases, in the grey level range very near to zero. This is because of the fact that, for the group of images studied, either the photograph was taken against a black background, as was the case of the piston head, or a large amount of distortion was present in the background of the image, as was in the case of the image of the strain gauge. This peak value in the low range of grey level may lead to a wrong threshold value - as has been shown in the cases of the piston head or strain gauge in figure (3.2.a) and figure (3.2.b). After further study, a condition has been imposed in the implemented technique so that no peak in
the range of grey level 0-23 is considered for threshold value.

3.3 **TEMPLATE MATCHING:**

A horizontal line detection template, as suggested in the reference [1], has been found very slow in work. To detect a line at the 63rd row of an image, it takes about 10 seconds by the 8085 microcomputer, even after avoiding the time consuming multiplication operations, as has been described in section (2.3).

In an attempt to improve the efficiency, a totally new algorithm for horizontal line detection has been developed; the algorithm is discussed in full detail in section (4.8).

The newly developed algorithm generates the addresses of the pixels only in the places where there is a possibility of the presence of the feature of interest and searches for the possible match. It totally avoids the places where there is no possibility of the presence of the feature.

The algorithm has been tested to detect the same line in the 63rd row of the image as discussed previously; the time taken was less than a second.
3.4 *Spurious Feature Cancellation Algorithm*

Spurious features are generated due to strong and localized reflections from the surface of highly reflective objects, when the photographs are taken. As has been suggested in reference [9], a simple way to eliminate the spurious features, is to set all the values of the pixels within the confined region of interest to zero, once the border around that region is drawn.

The suggested algorithm starts scanning the interior of the border followed region, from one point of the border until it reaches the boundary at the other end and sets all the elements to the desired value. When all the points on the border are covered this way, the desired result is produced.

Refering to the figure (3.4.a), the suggested algorithm can be applied to the shape shown as follows:

Let the co-ordinates of the different points of the referred shape be

\[ (X_i, Y_i), i=1,2,3,\ldots,27 \]

With \[ X_1 = X_{27} \]

and \[ Y_1 = Y_{27} \]

To choose the direction of scan at the ith border point, the algorithm suggests the following criterions, assuming that the region is traversed clockwise:

If \[ X_{i+1} - X_{i-1} = 0 \], skip scanning;

\[ X_{i+1} - X_{i-1} > 0 \], scan right to left;

\[ X_{i+1} - X_{i-1} < 0 \], scan left to right.
Now, let us consider the figure (3.4.b). If the previously mentioned algorithm is followed for this case, it can be seen that the 'X' marked elements will never be scanned and, as a result, if any spurious feature due to reflection is present in these pixels, they will not be eliminated. For this reason, some modifications are required in this algorithm.
Figure (3.4.a): Scanning scheme for spurious feature cancellation algorithm

Figure (3.4.b): Sample image where the existing algorithm for spurious feature cancellation fails.
With the objectives in mind that the algorithm ensures all the pixels are scanned, the conditions for selecting the direction of scanning have been redefined and the new algorithm stands as:

(1) Outline the border points of the region of interest from which the spurious features are desired to be eliminated. Label these border points with a different number than the background or object.

(2) Start scanning from any ith point on the border. Select the direction for scanning from the following conditions:

- If \( x_{i+1} - x_{i-1} > 0 \), scan from right to left;
- \( x_{i+1} - x_{i-1} < 0 \), scan from left to right;
- for \( x_{i+1} - x_{i-1} = 0 \), follow the following conditions:
  - if (i) \( x_i - x_{i-1} = 0 \), skip scanning;
  - (ii) \( x_i - x_{i-1} \) and \( y_i - y_{i-1} \) are of same sign, scan from left to right;
  - (iii) \( x_i - x_{i-1} \) and \( y_i - y_{i-1} \) are of opposite sign, skip scanning.

(3) Including the border point from which the scanning starts, as the scanning goes forward, substitute a different number for the pixel values.

(4) Stop scanning for a line as soon as a border point on the other end or a pixel with the newly substituted value is reached.
(5) Proceed in a clockwise direction and cover all the points on the border to initiate the scanning process.
Chapter IV

MICROCOMPUTER IMPLEMENTATIONS OF IMAGE PROCESSING ALGORITHMS

Different image processing algorithms have been investigated extensively. Some algorithms, which have been found suitable from the viewpoint of memory size requirements and execution speed, have been implemented in the microcomputer directly in their existing forms. In some cases, some modifications or assumptions have been made or complete new approaches have been developed; they have been discussed in previous chapters. This chapter discusses their implementation techniques in the microcomputer.

4.1 COMPUTING GRAY LEVEL DENSITY FUNCTION:

A microcomputer program for computing grey level density of an image has been developed. The algorithm is fairly straightforward:

(1) Assign different counter locations in the memory for different grey levels whose logical addresses are equal to the grey level.

(2) Initialise these locations to zero, and

(3) Scan the whole picture pixel by pixel; as soon as a particular grey level occurs, the corresponding counter location is to be incremented by one.
Figure (4.1.a): Histogram of Piston head in figure (2.3.f)

Figure (4.1.b): Histogram of blocks in figure (2.3.g)
Figure (4.1.c): Histogram of strain gauge in figure (2.3.h)
Some plotted histograms have been shown in figures (4.1.a) through (4.1.c).

4.2 **MICROCOMPUTER IMPLEMENTATION OF HISTOGRAM EQUALISATION:**

The histogram equalization technique requires the following extensions to the original algorithm for computing the grey level density. The extensions consist of (a) generation of a cumulative distribution function, (b) provisions for satisfying the imposed constraints over the transfer function and (c) substitution of the transformed grey levels in the original image.

Two primary requirements for an efficient algorithm are the involvement of a minimum number of memory locations and the minimum number of operation requirements. With this objective in mind, the cumulative distribution function has been generated in the same locations used for computing grey level density. The counter locations for two consecutive grey levels are located in addresses which are logically equal to the grey levels. The steps followed to generate the cumulative distribution function are as follows:

1. Set \( i = 1 \).
2. Add the \( i \)th counter location with the \((i+1)\)th counter location and store the result in the \((i+1)\)th location.
3. Set \( i = i + 1 \). If \( i \) is equal to the total number of counter locations then end the process; otherwise go to step (2).
Each location now contains the value corresponding to the cumulative distribution function according to the definition $P_k = \sum_{i=1}^{k} N_i$. Where $P_k$ is the $k$th discrete value of the cumulative distribution function and $N_n$ is the $n$th discrete value of the density function. A total of 255 additions have been required to generate this function.

After generating the cumulative distribution function, the transformed grey levels are brought within the allowable range. The highest value of the cumulative distribution function is equal to the total number of pixels in the image. Since this value should correspond to the highest allowable grey level, some kind of division operation is necessary to normalize the obtained cumulative distribution function, so that its values correspond to different points in the allowable range of grey level scale. To ensure very fast execution, this normalisation has been implemented by rotating the accumulator right by a specific number of times. Since a 64 x 64 image has been used and the range of grey level is from 0 to 255, the accumulator has been rotated by four times. This leads to a highest grey level of 256. To handle this situation every transformed grey level has been decremented by one, except the zero level. These values are substituted back in the original image.
Figure (4.2.a): Histogram equalized image of dollar in Figure (2.3.i).

Figure (4.2.b): Histogram equalized image of skull in Figure (2.3.j).

Figure (4.2.c): Histogram equalized image of strain gauge in Figure (2.3.k).
Figure (4.2.d): Histogram of the dollar in Figure (2.3.i).

Figure (4.2.e): Equalized histogram of the dollar.
Figure (4.2.f): Histogram of the skull in Figure (2.3.j).

Figure (4.2.g): Equalized histogram of the skull.
Figure (4.2.h): Histogram of the second strain gauge in Figure (2.3.k)

Figure (4.2.i): Equalized histogram of the second strain gauge.
Figures (4.2.a) through (4.2.i) shows some results of applying this algorithm to process some images.

4.3 IMPLEMENTATION OF THE DIRECT HISTOGRAM MODIFICATION TECHNIQUE:

The implementation of the Direct Histogram Modification Technique can be divided into three steps as follows:

1. Equalising the histogram of the original image by the transfer function \( s_k = T(r_k) = \sum_{j} P(r_j) \).

2. Specifying the desired histogram and equalising the specified histogram by the use of the transfer function \( v_k = G(z_k) = \sum_{j} P(z_j) \).

3. Implementing the inverse transformation function \( z = G^{-1}(s) \) to obtain the desired levels for the image pixels. All the symbols have been defined in the section (2.2.2).

Implementation of the first two steps have been discussed in section (4.2). The third transformation \( z = G^{-1}(s) \) requires the most critical implementation in this method from the viewpoint of fast processing.

In order to ensure a very fast evaluation of the inverse transformation function \( G^{-1}(s) \), a method has been developed, which initially creates a look up table for \( z_k = G^{-1}(s_k) \) for the kth grey level in the original image and store in a kth logical address. This method ensures that the same number of executions are required for each pixel in the image irrespective of its grey level.
If the grey levels are continuous, the transformed levels are also continuous. Under such circumstances, the inverse transformation \( G^{-1}(s) \) could be implemented by initially finding out the transformed grey level \( s \), for the \( r \) level in the original image, and matching the value of \( s \) with \( v = G(z) \), i.e. where \( s = v \); and finally by evaluating the inverse function \( G^{-1}(v) \). Graphically, the process can be explained by figure (4.3.1). Referring to this graph, corresponding to the grey level \( r \) in the original image, the transformed grey level \( s \) is to be located by the curve \( s = T(r) \). For the same ordinate value, \( s \), in the curve \( v = G(z) \), the abscissa value, \( z \), is the level in the desired image.

Since the levels dealt with are all discrete, a perfect match of \( v = s \) cannot be always ensured. For this reason, while implementing this method in a digital computer, a closest match between \( v \) and \( s \) is searched for. Since microcomputers are relatively slow machines, in the implemented method, a closest match between \( s \) and \( v \) has been searched for within the allowable range of \( v \) equal to or greater than \( s \). If \( G^{-1}(s) \) reaches its maximum value during the process, then search is to be discontinued and the remaining \( z \) levels are to be substituted by the maximum value of \( G^{-1}(s) \).
Figure (4.3.1): Graphical evaluation of $G^{-1}(S)$
The process of generating the look up table can be outlined by the following points: where a(i) represents initially the array for s=T(r) and b(j) represents the array for v=G(z).

1. Set i=j=0. Go to step(2).

2. Check whether b(j) is equal to or greater than a(i). If true, go to step(3), otherwise go to step(5).

3. Set a(i)=j. If j is less than 255, go to step(4), otherwise to step(6).

4. Set i=i+1. If i=256, then end the process, otherwise go to step(2).

5. Set j=j+1, go to step(2).

6. Substitute a(i+1) through a(255) by j and end the process.

A numerical example of the above technique has been shown in figure (4.3.2), where array sizes have been taken as 7 instead of 256.

Once the look up table is prepared, it is only a matter of scanning the whole image, pixel by pixel, and picking up the corresponding value of z=G^(-1)(s) from the address, which is logically equal to the grey level of the original pixel.

Results of applying this algorithm to different images has been shown in figures (4.3.a) through (4.3.i).
Figure (4.3.2.): Example of the generation technique of the look-up table.
Figure (4.3.a): Transformed image of the piston head in Figure (2.3.f) by direct histogram modification technique.

Figure (4.3.b): Transformed image of the blocks in Figure (2.3.g) by direct histogram modification technique.

Figure (4.3.c): Transformed image of the strain gauge in Figure (2.3.h) by direct histogram modification technique.
Figure (4.3.d): Specified histogram for the piston head in Figure (2.3.f).

Figure (4.3.e): Histogram of the transformed image of piston head in Figure (4.3.a).
Figure (4.3.f): Specified histogram for the blocks in Figure (2.3.g).

Figure (4.3.g): Histogram of the transformed image of the blocks in Figure (4.3.b).
Figure (4.3.h): Specified histogram for the strain gauge in Figure (2.3.h).

Figure (4.3.i): Histogram of the transformed image of the strain gauge in Figure (4.3.c).
4.4 Implementation of the Thresholding Technique:

The implementation of the binary level thresholding technique is straightforward. It requires scanning the image pixel by pixel. The steps followed to implement this technique are given below:

1. Take the first pixel from the memory and compare its grey level with the threshold value.

2. If the pixel value is less than or equal to the threshold value, then substitute the content of the memory location corresponding to the address of that pixel by zero; otherwise substitute by 255.

3. Check whether all the pixels are considered, if true, halt; if not, take the next pixel and go to step (2).

Implementation of the multilevel thresholding technique requires an added level of complexity compared to the binary level technique. In the implemented method, the image is scanned in a similar way to the binary level technique and the steps followed are given below:

1. Take the first pixel of the image and go to step (2).

2. Compare the pixel value with the lowest threshold level. If it is less than or equal to that level, substitute its value by zero and go to step (5). If it is not, go to step (3).

3. Compare the pixel with the next higher threshold level. If it is less than or equal to that level, then sub-
stitute its value by the average of the present and the last threshold level, and go to step (5). If it is not, go to step (4).

(4) If the pixel value is higher than the highest threshold level, substitute its value by 255 and go to step (5). If the present threshold value is not the highest one go to step (3).

(5) Check whether all the pixels are considered, if true, halt; if not, take the next pixel and go to step (2).

Figure (4.4-a) through figure (4.4-f) show examples of some binary and multilevel thresholded images, as performed by the 8085 microcomputer.
Figure (4.4.a): Binary level thresholded image of piston head in Figure (2.3.f)

Figure (4.4.b): Binary level thresholded image of blocks in Figure (2.3.g)

Figure (4.4.c): Binary level thresholded image of the strain gauge in Figure (2.3.h)
Figure (4.4.d): Multilevel thresholded image of transmission gear in Figure (2.3.m).

Figure (4.4.e): Multilevel thresholded image of strain gauge in Figure (2.3.k).

Figure (4.4.f): Multilevel thresholded image of wooden blocks in Figure (2.3.1).
4.5 MICROCOMPUTER IMPLEMENTATION OF THE SCATTER PLOT TECHNIQUE:

Although by using both low and high edge value Scatter Plots, threshold values for an image can be obtained, the implementation of the weight factor for the low edge value Scatter plot involves time consuming computations for the microcomputer. For this reason, an attempt has been made to implement the high edge value Scatter plot only to obtain the required threshold value. Also, it has been shown [11] that the high edge value Scatter plot gives more accurate results than low edge value Scatter plot.

Again, although various operators are available for calculating edge values, consideration was given to define the edge value in such a manner that its computation involves a very small amount of operations. Accordingly, referring to figure (2.2.c), the edge value of the point E was computed by the ROB operator. Again, reference [11] suggests that the ROB operator is the most efficient among the previously mentioned three. To limit all the arithmetic operations within two bytes, a scaling factor $1/4$ was added with the original definition; that is the ROB operator was used in the form

$$1/4 \text{Max}[|E-C|,|E-F|] \quad \text{EQ-[ 4.5.a ]}$$

The above operator refers to figure (2.2.c).

An additional effort was made to calculate the edge values in different efficient ways. Accordingly, another operator was defined as

$$\text{OPATR1} = 1/16 [|E-F|+|E-I|] \quad \text{EQ-[ 4.5.b ]}$$
To give an acceptable result, although the Scatter plot technique requires the assumption that the grey levels of the pixels interior to the object or background should be highly correlated, an attempt has been made to handle the non-uniformity on the background to a limited extent. The study shows that because of the noise in the background, a peak may occur in the Scatter plot in the range of grey level scale near to zero. This peak leads to a wrong threshold value. For the group of the images studied, after further investigation, it was decided to neglect the peak, if any, in the range of 0-23 of grey level scale. The peak occurring in the grey levels greater than 23 was used as the threshold value.

Figure (4.5.a) shows the thresholded image of random sized and shaped blocks. Both the operators, ROB and OPRTR1, gave same result when they were applied to find the thresholded value.

Figure (4.5.b) shows the thresholded image of a strain gauge. Results obtained in this case were also same for both the applied operators. Too much noise in the background has caused the obtained threshold value to be slightly smaller than the actual value. This has caused the presence of some dots in the background.
Figure (4.5.a): Thresholded image of the blocks in Figure (2.3.g) by both the ROB and OPRTRI operators.

Figure (4.5.b): Thresholded image of the strain gauge in Figure (2.3.h) by both the ROB and OPRTRI Operators.
Figure (4.5.c): Thresholded image of the transmission gear in Figure (2.3.m) by ROB operator.

Figure (4.5.d): Thresholded image of the transmission gear in Figure (2.3.m) by OPRTRI operator.

Figure (4.5.e): Thresholded image of the wooden blocks in Figure (2.3.1) by ROB operator.

Figure (4.5.f): Thresholded image of the wooden blocks in Figure (2.3.1) by OPRTRI operator.
Figure (4.5.c) and figure (4.5.d) shows thresholded images of a transmission gear obtained by ROB operator and OPRTR1 respectively. It should be noted here that because of the noise in the object portion, the obtained threshold values have become slightly bigger than the actual value.

Figure (4.5.e) shows the thresholded image of wooden blocks by the ROB operator and figure (4.5.f) shows the thresholded image of the same blocks by OPRTR1.

Table (4.5.a) shows a comparison between the threshold values obtained by the two operators.

<table>
<thead>
<tr>
<th>IMAGE</th>
<th>THRESHOLD VALUE BY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROB</td>
</tr>
<tr>
<td>BLOCK1</td>
<td>32</td>
</tr>
<tr>
<td>GAUGE</td>
<td>25</td>
</tr>
<tr>
<td>GEAR</td>
<td>112</td>
</tr>
<tr>
<td>BLOCK2</td>
<td>40</td>
</tr>
</tbody>
</table>

Table (4.5.a): Threshold values obtained by the operators ROB and OPRTR1 for different images.

It has been found during our study that the Scatter plotting technique works well only if the various subpopulations in the histogram are widely separated.
4.6 MICROCOMPUTER IMPLEMENTATION OF THE BORDER FOLLOWING TECHNIQUE:

The method suggested in the reference [9] has been implemented in the microcomputer because of its established efficiency. Since, at the boundary between the object and the background of a thresholded image, there will be a transition in grey level between two adjacent pixels, the method which is going to be described finds out the first border point by scanning the image for a transition of grey level and then compares all the pixels adjacent to it to locate the next border point. This method is stepwise as follows:

1. To detect the first border element at \((I_1,J_1)\) (a bright pixel) through a row or column scan. The pixel immediately preceding \((I_1,J_1)\) is to be labeled as the first neighbour \((I_0,J_0)\).

2. Starting with \((I_0,J_0)\) to proceeding clock wise, to label the other seven neighbours of \((I_1,J_1)\) as 2, 3, ..., 8. After it, to set \(K=2\).

3. To get the \(K\)th neighbour of \((I_1,J_1)\). If the pixel is a bright one then this pixel is the next border element. Now this element is to be defined as \((I_1,J_1)\) and the preceding element as \((I_0,J_0)\). To go to step 2.

4. If the pixel at the \(K\)th neighbour is a dark one, to set \(K=K+1\) and to go to step 3.

The above mentioned steps are to be followed until the first border element is encountered again. While implementing the above algorithm, provision has been kept to initiate
the process at any transition of grey levels, whether it is from dark to bright or from bright to dark. This provision ensures the improved speed of the algorithm in a general field of application.

The sequence of search for the next border element can be visualised clearly by figure (4.6.1). Say, 'a' is the first border point and 'b' is the previous to the border point. To find the next border point the algorithm will move from 'b' in the clockwise direction to the next neighbour of 'a', which is 'c'. If there is a transition of grey level between 'b' and 'c', 'c' is the next border point. Otherwise the algorithm will continue search for the next border point in the sequence of c-d-e-f-g-h-i. For the quick generation of the addresses of the eight neighbours, the algorithm also suggest a computationally efficient approach. Table (4.6.a) shows how to calculate the co-ordinates of the neighbour points.

The algorithm described above is designed to follow a closed border. It can also detect isolated points in an image. After the border of a closed section has been outlined, the process can either be halted or it can continue scanning for a transition in grey level to locate border points of another closed loop. The described method can be extended to determine the size of a closed boundary by storing the co-ordinates of all the border points.
CO-ORDINATES OF EIGHT NEIGHBOURS

<table>
<thead>
<tr>
<th>NEIGHBOUR NUMBER</th>
<th>X-AXIS=LX(J)</th>
<th>Y-AXIS=LY(J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ID</td>
<td>JD</td>
</tr>
<tr>
<td>2</td>
<td>LX(1) + K1</td>
<td>LY(1) + K2</td>
</tr>
<tr>
<td>3</td>
<td>LX(2) - K2</td>
<td>LY(2) - K1</td>
</tr>
<tr>
<td>4</td>
<td>LX(3) - K2</td>
<td>LY(3) - K1</td>
</tr>
<tr>
<td>5</td>
<td>LX(4) - K1</td>
<td>LY(4) + K2</td>
</tr>
<tr>
<td>6</td>
<td>LX(5) - K1</td>
<td>LY(5) + K1</td>
</tr>
<tr>
<td>7</td>
<td>LX(6) + K2</td>
<td>LY(6) + K1</td>
</tr>
<tr>
<td>8</td>
<td>LX(7) + K2</td>
<td>LY(7) + K1</td>
</tr>
</tbody>
</table>

Co-ordinate of the border element: (I1, J1)
Co-ordinate of the first neighbour (ID, JD)

K1 = JD - J1      K2 = ID - I1

Table (4.6.a): Address generation scheme for the eight neighbour pixels of a border point.

Some examples of this algorithm have been shown in figures (4.6.a) through (4.6.c).
Figure 4.6.1: Sequence of search for the next border point.
Figure (4.6.a): Border followed image of the piston head in Figure (4.4.a).

Figure (4.6.b): Border followed image of the blocks in Figure (4.4.b).

Figure (4.6.c): Border followed image of the strain gauge in Figure (4.4.c).
4.7 MICROCOMPUTER IMPLEMENTATION OF TEMPLATE MATCHING:

Consider the template shown in figure (4.7.a). A line template of this type has been implemented in the microcomputer. This template will give a zero response over a constant background according to equation (2.3.a). Consider the sample image of figure (4.7.b), which represents a binary level image with only a line present on a constant background. When the above mentioned template coincides with the line of this sample image, it will produce the maximum response and this condition can be detected by using the threshold value \( T = 2 \times N \), where \( N \) is the length of the template.
Figure (4.7.a): Line Template

\[
\begin{array}{cccccccc}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2.2 \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
\end{array}
\]

Figure (4.7.b): Binary level image
The template has been implemented in the microcomputer based on addition subtraction operations. The pixel values within the first row and third row of the template have been added together; lets name it as K1. The pixel values within the middle row of the template have been added together and doubled without involving any multiplication routine; lets name it as K2. The value of \((K2-K1)\) gives the response of the template at a point. This response has been compared with the threshold value to arrive at the final decision.

4.8 DIRECT DETECTION OF LINEAR EDGE OR LINE:

Although template matching is very simple to apply, it is very slow when implemented on a microcomputer, in software. The improved algorithm directly generates the addresses of pixels of possible locations, where the line or the linear edge can be present. Since linear edge and line detection are very similar, only edge detection will be discussed in the rest of this section.
Figure (4.8.1): Sample binary level image having linear edge.
To detect a horizontal edge, similar to that shown in the sample binary level image of figure (4.8.1), the sequence of the algorithm can be stated as follows, where \( X(I,J) \) represents the pixel in \( I \)th row and \( J \)th column, \( \text{LENGTH} \) represents the length of the template, \( \text{NR} \) represents the total number of rows and \( \text{NC} \) represents the total number of columns present in the image:

1. Set \( I=0 \). Go to step (2).

2. Set \( I=I+1 \); if \( I<\text{NR} \), then go to step (3), otherwise template is not matched in the image and the process can be terminated.

3. Set \( J=0 \). Go to step (4).

4. Set \( J=J+\text{LENGTH} \). If \( J<\text{NC} \), then go to step (5), otherwise go to step (2).

5. Set \( \text{TOPREF}=X(I,J) \). Check whether \( X(I,J) \neq X(I+1,J) \). If it is, go to step (6); otherwise go to step (4).

6. Set \( \text{INDEX}=\text{LENGTH}-1 \); Check whether \( \text{INDEX}=0 \); if it is, set \( J=J-1 \) and go to step (11); if not, go to step (7).

7. Set \( \text{COUNTER}=\text{INDEX} \); \( J=J-(\text{LENGTH}-1) \); go to step (8).

8. If \( X(I,J)=\text{TOPREF} \), go to step (9); if not, go to step (4).

9. Check whether \( X(I,J) \neq X(I+1,J) \); if it is, go to step (10); if not go to step (4).

10. Set \( \text{COUNTER}=\text{COUNTER}-1 \). If \( \text{COUNTER}=0 \), go to step (11); otherwise, set \( J=J+1 \) and go to step (8).
(11) \( I(I, J) \) now represents the last but one point of the top row of the template. \( J = J + 1 - \text{LENGTH} - 1 \) represents the column where the template starts.

(12) Stop.

The above mentioned technique can be easily used to detect vertical edges by interchanging row and column variables. This can also be extended to detect 45 degree slope lines by simultaneously varying the row and column variables with equal magnitudes and putting simultaneous checks for row and column end.

The above mentioned algorithm has been applied to extract the same features from the image as used in the application of the first type of template.

Some examples of feature extraction by this method are shown in figures (4.8.a) through (4.8.c).
Figure (4.8.a): Feature extracted from the figure (4.4.a) by line detection technique.

Figure (4.8.b): Features extracted from the figure (4.4.a) by line detection technique.

Figure (4.8.c): Feature extracted from the figure (4.4.b) by line detection technique.
4.9 IMPLEMENTATION OF SPURIOUS FEATURE CANCELLATION ALGORITHM:

In implementing the spurious feature cancellation algorithm, the following steps are followed:

(1) The region of interest is border followed first, labelling the border points with a code (32H). The border following algorithm has been discussed in section (4.6). The regions of interest have been detected by the edge detection technique discussed in section (4.8).

(2) A selected point on the border is marked by a different code (33H) to record the starting point of the process. The co-ordinate of this point are saved as \((X_{i-1}, Y_{i-1})\).

(3) Moving in a clockwise direction, the next two points on the border point are found. The co-ordinates of the first point between the latter two are saved as \((X_i, Y_i)\) and the co-ordinates of the last point are stored in the BC register pair in the 8085 microcomputer.

(4) The scanning is initiated from \((X_i, Y_i)\), following the conditions given in section (3.4). The point \((X_i, Y_i)\) is labelled with the code 01H and as the scanning is started, it continues to label the pixels on its way with 01H.

(5) As soon as the border point on the other end or a point labelled with 01H is reached, the scanning stops. This saves computations by preventing the same line from being scanned twice. When the scanning is about to start in a line, which has already been covered from the other end, it
finds the newly assigned value in the very next pixel and the process stops for that line; however, this will not prevent any legitimate scan.

(6) Once a line has been scanned, the algorithm locates the next border point in the clockwise direction, takes its location into BC register pair; before that \((X_{i-1}, Y_{i-1})\) is shifted to \((X_{i-1}, Y_{i-1})\) and the previous content of the register pair BC, to \((X_{i}, Y_{i})\).

The process continues until all the points on the border are covered to initiate a scan. Figure (4.9-a) shows the thresholded image of a piston head with the presence of spurious features in it. Figure (4.9-b) shows the same thresholded image after the removal of the spurious features.
Figure (4.9.a): Image including spurious features due to light reflection.

Figure (4.9.b) Image after the spurious features removed.
Chapter V

SUMMARY AND CONCLUSIONS OF THE WORK ACCOMPLISHED FOR THIS THESIS

5.1 SUMMARY AND CONCLUSIONS:

The purpose of this thesis is to investigate the introduction of microcomputer systems for image processing. We have concentrated our investigation in the area of applications of such systems for the purpose of on line quality control of manufactured parts. The required algorithms have been implemented on an Intel 8085 microcomputer. For 64x64 size images, the execution time required by each of the implemented algorithms has been shown in Appendix-A. The time requirements of individual algorithms, taking the image size also into account, suggest that the microcomputer systems, similar to that which has been used in our study, can be applied for on line quality control applications on a limited basis.

In the cases of Direct histogram modification technique and Spurious feature cancellation algorithm, modifications have been suggested to their existing forms. In case of Direct histogram modification technique, a scheme for generating a look-up table for the desired results has been developed to improve the efficiency of the algorithm. The
modifications suggested to the Spurious feature cancellation algorithm extend it for the application in a general case.

A new approach for detecting the lines in an image has also been developed in this thesis. This approach has been found much more efficient than the conventional window type templates from the viewpoint of the time requirements for detecting the features.
Appendix A

EXECUTION TIME REQUIREMENTS FOR DIFFERENT ALGORITHMS.
**Execution Time Requirements for the Implemented Algorithms.**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Execution Time in Seconds</th>
<th>Example used.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculating grey level density</td>
<td>0.48</td>
<td>Piston Head</td>
</tr>
<tr>
<td>function.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram Equalisation.</td>
<td>0.71</td>
<td>Dollar.</td>
</tr>
<tr>
<td>Direct Histogram Modification</td>
<td>0.95</td>
<td>Piston Head</td>
</tr>
<tr>
<td>Technique.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Level Thresholding.</td>
<td>0.20</td>
<td>Piston Head</td>
</tr>
<tr>
<td>Multilevel Thresholding.</td>
<td>0.35</td>
<td>Transmission Gear (3 levels)</td>
</tr>
<tr>
<td>Border Following.</td>
<td>1.75</td>
<td>Piston Head</td>
</tr>
<tr>
<td>Template Matching.</td>
<td>12.0</td>
<td>Line in 63rd row having length of 5 pixels. Same as above.</td>
</tr>
<tr>
<td>Direct Line Detection.</td>
<td>0.20</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

MICROCOMPUTER PROGRAM LISTINGS.
LOC OBJ LINE SOURCE STATEMENT

1 880FF 1 TEMP EQU. 080FF
2 4000 2 ORG 4000H
3 4000 31FFAF 3 START: LXI SR, 0AFFH ; INITIALIZING
4 4003 010000 4 THE STACK POINTER.
5 9  ; INITIALIZING THE
6 12  ; COUNTER FOR THE
7 14  ; NUMBER OF GREY LEVELS.
8 15  ; FOR 256 GREY LEVELS
9 16  ; AND TWO LOCATIONS FOR
10 17  ; EACH GREY LEVEL. THE
11 18  ; MAXIMUM VALUE OBTAINED
12 19  ; BY THE COUNTER WILL BE
13 20  ; 200H.
14 21  ; INITIALIZING THE COUNTER LOCATIONS
15 22  ; FOR DIFFERENT GREY LEVELS TO ZERO
16 23  ; BEFORE STARTING COUNTING.
17 24  ;
18 4006 210064 25 LXI H, 8400H ; STARTING LOCATION FOR
19 26  ; GREY LEVEL COUNTERS.
20 27 4009 3E00 28 INTLZN; MVI, A, 00H
21 28 400B 77 29 MOV M, A
22 29 400C 23 30 INX H
23 30 400D 03 31 INX B
24 31 400E 78 32 MOV A, B
25 32 400F F92 33 CPI 02H
26 33 4011 C209 40 34 JNZ INTLZN
27 34 4014 110084 35 LXI D, 8400H
28 35 4017 010000 36 LXI B, 00H
29 36 37 37  ; SETTING COUNTER TO
30 37 38 38  ; CHECK WHETHER ALL THE
31 38 39 39  ; LOCATIONS HAVE BEEN
32 39  ; COVERED.
33  ;
34 40 401A 210074 41 LXI H, 7400H ; STARTING ADDRESS OF
35 42  ; THE PIXELS.
36 43 44 44  ; GENERATING CORRECT ADDRESS FOR THE
37 44 45 45  ; COUNTER LOCATIONS.
38 45 46 46  ; LOOP: XRA A
39 46 47 47loop MOV M, A
40 47 48 48 RAL
41 48 49 49 RAL
42 49 50 50 PUS D
43 50 51 51 LDI D, TEMP
44 51 52 52 STA D
45 52 53 53 MVI A, 00H
46 53 54 54 RAL
LOC  OBJ  LINE  SOURCE STATEMENT
4028  13  54  INX D
4029  12  55  STAX D
402A  D1  56  POP D
402B  E5  58  PUSH H
402C \21FFB0  60  LXH H,TEMP
402F  AF  61  XRA A
4030  7E  62  MOV A,M
4031  83  63  ADD E
4032  5F  64  MOV E,A
4033  23  65  INX H
4034  7E  66  MOV A,M
4035  8A  67  ADC D
4036  57  68  MOV D,A
        G
4037  C5  70  PUSH A
4038  EB  72  XCHG
4039  4E  73  MOV C,M
403A  23  74  INX H
403B  46  75  MOV B,M
403C  03  76  INX B
403D  70  77  MOV M,B
403E  2B  78  DCH K
403F  71  79  MOV M,C
4040  C1  81  POP B
4041  E1  82  POP H
4042  03  83  INX B
4043  78  84  MOV A,B
4044  FE10  85  CPI 10H
4046  CA5020  87  JZ EXT
4049  23  88  INX Y
404A  110084  89  -LXI D,8400H
404D  C31DA0  90  JMP LOOP
4050  C7  94  EXT KS T
4000  95  END START

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS
EXT  A  4050  INTLZN A  4009  LOOP  A  4083  START  A  4001
TEMP  A  4084  RAS
SOURCE STATEMENT

FILE NAME: HSTEGU.**
This program equalizes the grey level density function of an image.

** TEMP EQU '0B0FFH
ORG 4000H

START: LXI SP, 0AFFFH ; initializing
; the stack pointer.
LXI B, 00H ; initializing the
; counter for the
; number of grey levels
; for 256 grey levels
; and two locations for
; each grey level. The
; maximum value obtained
; by the counter will be
; 200H.

; initializing the counter locations
; for different grey levels to zero
; before starting counting.
LXI H, 7000H ; starting location for
; grey level counters.

; D
INTLZ M: MVI A, 00H
MOV A, A
INX H
INX B
MOV A, B ; check whether all the
; locations have been
; covered.
JNZ INTLZ
; D
LXI D, 7000H
LXI B, 00H ; setting counter to
; check whether all
; the pixels have been
; covered. Max value obtained
; by this counter will be
; 1000H.

LXI H, 7400H ; starting address of
; pixels.

; generating correct address for the
; counter locations.
LOOP: XRA A
MOV A, M
RAL
PUSH D
; since two locations are
; allotted for each grey
; level, it is being doubled
LXI D, TEMP
; before adding with the
; reference address.
STA D
MVI A, 00H
RAL

LOC OBJ  LINE
0000 31FFAF
4003 010000
4006 210070
4009 3E00
400B 77
400C 23
400D 03
400E 78
400F FE02
4011 C2040
4014 110070
4017 010000
401A 210074
401D AF
401E 7E
401F 17
4020 D5
4021 11FF00
4024 12
4025 3E00
4027 17
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>4028</td>
<td>13</td>
<td>54</td>
<td>INX D</td>
</tr>
<tr>
<td>4029</td>
<td>12</td>
<td>55</td>
<td>STAX D</td>
</tr>
<tr>
<td>402A</td>
<td>D1</td>
<td>56</td>
<td>POP &quot;D&quot;</td>
</tr>
<tr>
<td>402B</td>
<td>E5</td>
<td>57</td>
<td>PUSH H</td>
</tr>
<tr>
<td>402C</td>
<td>21FFB0</td>
<td>58</td>
<td>LXI H, TEMP</td>
</tr>
<tr>
<td>402D</td>
<td>AF</td>
<td>59</td>
<td>XRA A</td>
</tr>
<tr>
<td>4030</td>
<td>7E</td>
<td>60</td>
<td>MOV A, M</td>
</tr>
<tr>
<td>4031</td>
<td>63</td>
<td>61</td>
<td>ADD E</td>
</tr>
<tr>
<td>4032</td>
<td>5F</td>
<td>62</td>
<td>MOV E, A</td>
</tr>
<tr>
<td>4033</td>
<td>37</td>
<td>63</td>
<td>INX H</td>
</tr>
<tr>
<td>4034</td>
<td>7E</td>
<td>64</td>
<td>MOV A, W</td>
</tr>
<tr>
<td>4035</td>
<td>8A</td>
<td>65</td>
<td>ADC D</td>
</tr>
<tr>
<td>4036</td>
<td>57</td>
<td>66</td>
<td>MOV D, A</td>
</tr>
</tbody>
</table>

G:  

69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  

INC: INCREMENTING THE COUNTER VALUE.

PUSH B: SAVE COUNTER.

XCHG ADDRESS OF LSB IN <HL> REG.

MOV C, M COUNTER INCREMENTED.

INX H COUNTER INCREMENTED.

MOV B, M COUNTER INCREMENTED.

JMP LOOP CHECKING WHETHER ALL
THE PIXELS HAVE BEEN
COVERED.

JZ SCNSTP IF SO, END OF EXECUTION.

INX H IF NOT, REPEAT THE WHOLE
PROCESS FOR NEXT PIXEL.

LXI D, 7000H GENERATION OF CUMULATIVE
LXI H, 7000H DISTRIBUTION FUNCTION.

SCNSTP: LXI D, 7000H
LXI H, 7000H FIRST LOCATION OF
DIFFERENT GREY LEVEL COUNT.

INX H POINTING TOWARDS THE NEXT
GREY LEVEL.

LXI B, 00H INITIALIZING THE CONTROLLERS.
LOC OBJ

LINE SOURCE STATEMENT

105 THE TOTAL NUMBER OF GREY
106 LEVELS IS 256, THE FINAL V
107 OF THE COUNTER WILL BE 10

108 OH.

109 CONTI XRA A

110 LDAX D

111 ADD M

112 INX D

113 INX H

114 LDAX D

115 ADX M

116 MOV M A

117 INX B

118 ITEMS CHECKING WHETHER ALL THE

119 MOV A B

120 CPI 01H

121 JZ ROTN

122 INX H

123 INX D

124 JMP CONT

125

126 ROTATE EACH ITEM FOUR TIMES WHICH IS

127 EQUIVALENT TO DIVIDE BY 16, AND THIS

128 MAKES THE HIGHEST TRANSFORMED GREY

129 LEVEL EQUAL TO 256.

130 ROTN LXI H, 7000H

131 LXI D, A0H

132 R

133 ITEMS

134 CONT7 MVI B 01H

135 TO CHECK WHETHER ALL TH

136 CONT2 PUS H

137 COUNT THE ROTATION.

138 INX H

139 MOV A M

140 RAR

141 MOV M A

142 DCX H

143 MOV A M

144 RAR

145 MOV M A

146 MOV A B

147 CPI 04H

148 BEEN

149 JZ CHKEND

150 L THE

151 ITEMS HAVE BEEN COVERED.

152
LOC OBJ LINE SOURCE STATEMENT

408B 04 151 INR 'B
4089 E1 152 POP 'H
408A C37840 153 JMP CONT2 ;IF NO, GO FOR NEXT ROTAT
ION.
408D 13 154 CHKEND: INX 'D
408E 7A 155 MOV 'A,'D
408F FE01 156 CPI 01H
4091 CA9A40 157 JZ TRNSFN
4094 E1 158 POP 'H
4095 23 159 INX 'H
4096 23 160 INX 'H
4097 C37640 161 JMP CONT3

409A 010090 162 ;SINCE AFTER ROTATION, THE HIGHEST
409D 110000 163 GREY LEVEL IS 256, EACH GREY LEVEL
40A0 D5 164 WILL BE REDUCED BY ONE AND THE
40A1 210070 165 TRANSFORMED GREY LEVELS WILL BE STORED
40A4 5E 166 IN A NEW LOCATION.
40A5 23 167 ;A NEW LOCATION.
40A6 56 168 ;
40A7 1B 169 TRNSFN: LXI, B, 9000H ;NEW STORAGE LOCATION FOR

40B0 7A 171 LXI 'D, 00H
40B1 7B 172 PUSH 'D
40B2 02 173 LXI 'H, 7000H ;INITIAL LOCATION.
40B5 7A 174 CONT4: MOV 'E, 'M
40B6 FE01 175 INX 'H
40B7 C2B140 176 MOV 'D, 'M ;DECENDING THE GREY L
40B8 00 177 DEX 'D ;EVEL
40BB 03 178 MXR A, 'D
40BD 05 179 MOV 'A, 'D
40BE C3A440 180 CPI 0FFH ;CHECK WHETHER THE PREVI
40BF C37640 181 JNZ TRNSFR ;US VALUE WAS ZERO, IF IT WA

40C0 7A 182 LXI, D, 00H ;ITS VALUE WILL NOT BE C
40C3 02 183 TRNSFR: MOV 'A,'E
40C5 7A 184 STAX 'B
40C6 13 185 POP 'D
40C7 7A 186 INX 'D
40C8 7A 187 MOV 'A,'D
40C9 FE01 188 CPI 01H ;CHECKING WHETHER ALL TH
40CF C37640 189 JZ LSTSTP ;ITEMS HAVE BEEN COVERED

190
191 INX 'H
192 INX 'B
193 ;PUSH 'D
194 JMP CONT4
LOC OBJ  LINE  SOURCE STATEMENT

195    ; SUBSTITUTE BACK THE PIXEL VALUES.
196    ;
197    ;
40C1 010100 198    LSTSTP: LXI B, 01H ; INITIALIZE COUNTER.
40C4 210074 199    LXI H, 7400H ; STARTING LOCATION OF PIXELS.
40C7 110090 200    LP:  LXI D, 9000H ; NEW LOCATION FOR TRANSFORMED GREY LEVELS.
        201    INNER
        202    - XRA A
        203    MOV A, M ; ADDING THE PIXEL VALUE WITH
        204    RRECT ADDRESS.
        205    ADD E
        206    MOV E, A
        207    MOV A, D
        208    ACI 00H
        209    MOV D, A
        210    LDA X
        211    Y LEVEL
        212    MOV M, A
        213    LUE
        214    CPI 10H ; GET THE TRANSFORMED GREY LEVELS.
        215    JZ EXT
        216    MOV A, E
        217    INX B
        218    INX H
        219    JMP LP
        220    EXT: RST 0

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS
CHKEND A 40BD  CONT A 405B  CONT2 A 407A  CONT3 A 4076
CONT4 A 40A4  EXT A 40DF  INTLZN A 4009  LOOP A 4010
LP A 40C7  LSTSTP A 40C1  ROA A 4070  SCNSIP A 4050
START A 4000  TEMP A B0FF  TRNSFN A 409A  TRNSFR A 40B1

ASSEMBLY COMPLETE, NO ERRORS
LOC  OBJ   LINE  SOURCE STATEMENT

B0FF
4000 31FAF 10  START: LXI, SP; 0AFFFH  \*INITIALIZING
        ORG 4000H.

4003 010000 12  LXI, B, 00H  \*INITIALIZING THE
4006 210070 13  COUNTER FOR THE
4009 3E00 14  NUMBER OF GREY LEVELS.
400B 77 15  FOR 256 GREY LEVELS.
400C 23 16  AND TWO LOCATIONS FOR.
400D 03 17  EACH GREY LEVELS, THE
400E 78 18  MAXIMUM VALUE OBTAINED
400F FE02 19  BY THE COUNTER WILL BE
4011 C20940 20  2FFH.

LXI 'H, 7000H  \*STARTING LOCATION FOR
4014 110070 25  GREY LEVEL COUNTERS.
4017 010000 26  INTLZN; MVI A, 00H
401A 210074 27  MOV M, A
401D AF 28  INX H
401E 7E 29  INX B
401F 17 30  MOV A, B  \*CHECK WHETHER ALL THE
4020 DS 31  CPI 02H  \*LOCATIONS HAVE BEEN
4021 11FFB0 32  JNZ INTLZN  \*COVERED.
4024 12- 33  LXI D, 7000H  \*SETTING COUNTER TO
4025 3E00 34  LXI B, 00H  \*CHECK WHETHER ALL
4027 17 35  THE PIXELS HAVE BEEN
4028 40 36  COVERED, MAX VALUE OBTAINED
402A 41  \*BY THIS COUNTER WILL BE
402C 110070 40  1100H.
402F FE02 41  LXI 'H, 7000H  \*STARTING ADDRESS OF
4031 C20940 42  PIXELS.

4034 110074 43  \*GENERATING CORRECT ADDRESS FOR THE
4037 46  LOOP: XRA A  \*SINCE TWO LOCATIONS ARE
403F 7E 44  MOV A, M  \*ALLOTTED FOR EACH GREY
4040 17 45  RAL  \*LEVEL, IT IS BEING DOUBLED
4042 05 46  PUSH D  \*BEFORE ADDING WITH THE
4043 49 47  LXI D, TEMP  \*REFERENCE ADDRESS.
4046 50 48  STAX D
4047 51 49  MVI A, 00H
4049 52 50  RAL
LOC OBJ   LINE    SOURCE STATEMENT

4028 13   54     INX D
4029 12   55     STAX D
402A D1   56     POP D
402B E5   57     ; GETTING THE REFERENCE ADDRESS.
402C 21FF60 58     PUSH H
402D AF   59     ; SAVING THE CURRENT PIXEL ADDRESS.
4030 7E   60     LXI H, TEMP
4031 83   61     XRA A
4032 5F   62     MOV A,M
4033 23   63     ADD E
4034 7E   64     MOV E,A
4035 9A   65     INX H
4036 57   66     MOV A,M
4037 C5   67     ADC D
4038 EB   68     MOV D,A
4039 4E   69     ; CORRECT ADDRESS IN <DE>.
403A 23   70     ; INCREMENTING THE COUNTER VALUE.
403B 46   71     PUSH B
403C 03   72     ; SAVE COUNTER.
403D 70   73     XCHG
403E 2B   74     ; ADDRESS OF LSB IN <HL>.
403F 71   75     ; REG.
4040 03   76     MOV C,M
4041 E1   77     INX H
4042 03   78     MOV B,M
4043 7B   79     INX B
4044 FE10 80     MOV M,B
4045 07   81     ; COUNTER INCREMENTED.
4046 CA5040 82     DCX H
4049 23   83     ; PIXEL LOCATION.
404A 110070 84     POP B
404B 03   85     ; GETTING BACK THE PIXEL COUNTER.
404C 7B   86     POP H
404D FE10 87     ; GETTING BACK THE CURRENT PIXEL LOCATION.
404E 07   88     CPI 10H
404F 07   89     ; CHECKING WHETHER ALL PIXELS HAVE BEEN COVERED.
4050 110070 90     JZ SCNSTP
4053 210070 91     INX H
4056 CDFC40 92     ; IF SO, END OF EXECUTION.
4059 210070 93     IF NOT, REPEAT THE WHOLE PROCESS FOR NEXT PIXEL.
405C CD1741 94     JMP LOOP
405F 010070 95     ; GENERATION OF CUMULATIVE DISTRIBUTION FUNCTION.
4062 213270 96     ; SCNSTP: LXI D, 7000H
4065 110070 97     ; FIRST LOCATION OF DIFFERENT GREY LEVEL COUNTERS.
4068 CDFC40 98     CALL CMFU0
406B 210070 99     LXI H, 7000H
406C CD1741 100     CALL ROTIN
406F 010070 101     LXI 5, 7000H
4072 213270 102     LXI 4, 7000H
I SPECIFIED HISTOGRAM IS TO BE
EQUALISED NOW.

LXI D, 8500H
LXI H, 8500H
CALL CHIFUCN-
LXI H, 8500H
CALL ROTN
LXI B, 8500H
LXI H, 8500H
CALL TRNSFN

J - INVERSE TRANSFORMATION.

STINV;

STA 'COUNT
COUNT CHECKS WHETHER ALL THE LEVELS
HAVE BEEN COVERED.
LXI B, 7000H
LXI H, 8500H
VXI E, 00H

AGN;

LDAX B
STA REF

LIP;

SUB N

JZ SUBSTI JIF ZERO OR
JC SUBSTI JIF CARRY THIS IS THE INV. TRA

INSFORM;

INX H
INR E
MOV A, E
CPI '0FFH'
JZ SUBALL

LDAX 'B
JMP LIP

SUBSTI; MOV A, E
STAX 'B
PUSH H

INX 'B

CHECK WHETHER CONSECUTIVE VALU
ES.

LDA COUNT
CP1 '0FFH'
JZ OUTI

INR 'A

STA COUNT
LDA REF
MOV D 'A
LDAX 'B
SUB D

JNZ NXTSTP
POR 'H

JMP SUBSTI
LOC OBJ   LINE   SOURCE STATEMENT

40C1 E1 159   NISTP:POP H
40C2 C36D40 160   JMP AGN
40C5 7B 161   SUBALL:MOV A,E
40C6 02 162   STAX B
40C7 03 163   INX B
40C8 3ADB40 164   LDA COUNT
40C9 FEFF 165   CPI 0FFH
40CD C0840 166   JZ EXIT
40D0 3C 167   INR A
40D1 32DB40 168   STA COUNT
40D4 C3C540 169   JMP SUBALL
40D7 E1 170   OUT: POP H
40DB C3DD40 171   EXIT: JMP LSTTP
40EE 172   COUNT: DS 1
40EE 173   REF: DS 1

SUBSTITUTE BACK THE PIXEL VALUES

LSTTP:LXI B,01H ;INITIALISING THE COUNTER
LXI H,7400H
LP:LXI D,7000H ;LOCATION FOR TRANSFORMED
GRAY LEVELS.

XRA A
MOV A,M ;ADDITION THE PIXEL VALUE
TO THE REFERENCE ADDRESS
ADD E ;TO GET THE CORRECT LOC.

MOV E,A
MOV A,D
ACI 00H
MOV D,A
LDAX D  ;GET THE TRANSFORMED GRAY L
EVEL.

MOV M,A
MOV A,B
CPI-10H ;CHECK WHETHER ALL THE PIXE.
LS L ;HAVE BEEN COVERED.

JZ EXT,
INX B.
INX-H
JMP LP

EXT: RST 0

SUBROUTINE FOR GENERATING
CUMULATIVE DISTRIBUTION FUNCTION.

CMFUNC:INX H  ;POINTING TOWARDS THE NEXT
INX H  ;GREY LEVEL.
LXI B,00H ;INITIALIZING THE CONTRO.
INCE THE TOTAL NUMBER OF GREY
; LEVELS IS 256, THE FINAL V
; OF THE COUNTER WILL BE 13

211     VALUE
212     0H

213     CONT: XRA A
214     LDX D
215     ADD M
216     MOV M, A
217     INX D
218     INX H
219     LDX D
220     ADC M
221     MOV M, A
222     INX B ; CHECKING WHETHER ALL THE
223     ITEMS HAVE BEEN COVERED.

224     MOV A, B
225     CPI $01H
226     JZ EXT1
227     INX H
228     INX D
229     JMP CONT

230     EXT1: RET

231     ; SUBROUTINE FOR ROTATING THE COUNTED
232     ; GRAY LEVEL'S.
233     ; ROTATE EACH ITEM FOUR TIMES WHICH IS
234     ; EQUIVALENT TO DIVIDE BY 16 AND THIS
235     ; MAKES THE HIGHEST TRANSFORMED GREY
236     ; LEVEL EQUAL TO 256.

237     ROTN: LXI D, 00H ; INITIALIZING THE COUNTER
238     R
239     E ITEMS ; TO CHECK WHETHER ALL TH
240     R TO

241     CONT3: MVI B, 01H ; INITIALIZING THE COUNTER
242     R TO
243     CONT2: PUSH H ; COUNT THE ROTATION.
244     INX H ; MSB WILL BE ROTATED FIRST

245     ST.

246     XRA A
247     MOV A, M
248     RAR
249     MOV M, A
250     DEX H
251     MOV A, M
252     RAR
253     MOV M, A
254     MOV A, B
255     CPI $04H ; CHECK WHETHER DATA HAS
256     BEEN ; ROTATED FOUR TIMES.
257     JZ CHKEND ; IF YES, CHECK WHETHER AL
L THE

ITEMS HAVE BEEN COVERED

INR B

POP H

JMP CONT2

IF NO, GO FOR NEXT ROTATION

INX D

MOV A, D

CPI $01H

JZ EXT2

POP H

INX H

INX H

JMP: CONT13

EXT2: POP H

RET

SUBROUTINE FOR SEQUENTIALLY RESTORING

SINE R AFTER ROTATION, THE HIGHEST

GREY LEVEL IS 256, EACH GREY LEVEL

WILL BE REDUCED BY ONE AND THE

TRANSFORMED GREY LEVELS WILL BE STORED

IN A NEW LOCATION.

TRNSFN: LXI D, $00H

PUSH D

MOV E, M

INX H

MOV D, M

DCX D

DECIMENTING THE GREY LEVEL BY ONE.

CHECK WHETHER THE PREVIOUS VALUE WAS ZERO. IF IT WAS

HANGED, ITS VALUE WILL NOT BE C

CHECKING WHETHER ALL THE ITEMS HAVE BEEN COVERED

INX H

INX B

PUSH D
<table>
<thead>
<tr>
<th>LOC. OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>415E C34441</td>
<td>304</td>
<td>JMP CONT4</td>
</tr>
<tr>
<td>4161 C9</td>
<td>325</td>
<td>RET</td>
</tr>
<tr>
<td>4000</td>
<td>306</td>
<td>END START</td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

AGN A 40BD | CHKEND A 4131 | CMFUCN A 40FC | CONT A 4101
CONT2 A 411C | CONT3 A 411A | CONT4 A 4144 | COUNT A 40DE
EXIT A 40D8 | EXT A 40FB | EXT1 A 4116 | EXT2 A 413E
EXT3 A 4161 | INTLZN A 4009 | L1P A 4091 | LOOP A 40ID
LP A 40E3 | LSTSTP A 40DD | NXTSTP A 40C1 | OUT1 A 40D7
REF A 40DC | ROTN A 4117 | SCNSTP A 4050 | START A 4000
STINV A 4080 | SUBALL A 4065 | SUBST1 A 40A4 | TEMP A 80FF
TRANSF A 4140 | TRANSFR A 4151 |

ASSEMBLY COMPLETE, NO ERRORS
LOC OBJ LINE SOURCE/STATEMENT

1 } ***************
2 * FILENAME: - THRSLD.SRC *
3 ********************************
4 THIS PROGRAM IS USED TO THRESHOLD A GIVEN
5 IMAGE AT A GIVEN
6 THRESHOLD VALUE (THRU THE KEYBOARD.)
7
8 4000
9 ORG 4000H
10 CI EQU 0F83H ; MONITOR ROUTINE TO SCAN
11 THE KEYBOARD.
12 CO EQU 0F809H ; MONITOR ROUTINE TO DISPLAY
13 LAY ON THE CONSOLE.
14 217840 BEGIN: LXI H, MESSAGE ; TO DISPLAY A MESSAGE ON
15 117640 THE CONSOLE.
16 RESHOLD VALUE.
17 0600 MVI B, 00H ; COUNTER TO COUNT # OF B
18 XYES IN THRESHOLD VALUE.
19 4E LP1: MOV A, M ; GET THE MESSAGE FROM THE
20 MEMORY LOCATION.
21 79 MOV A, C ; USED FOR COMPARING WITH
22 THE TERMINATING CHARACTER.
23 AFE24 CPI 24H ASCII CODE FOR THE TERMINATING CHAR "S"
24 CA1640 JZ NEXT ; IF MESSAGE COMPLETE GO
25 FOR NEXT OPERATION.
26 0F CD098 CALL CO ; DISPLAY THE MESSAGE ON
27 123 INX H ; THE CONSOLE.
28 30840 JMP LP1 ; REPEAT THE PROCESS.
29 020 CALL CI ; NOW READ THE THRESHOLD VALUE FROM THE KEYBOARD.
30 4F MOV C, A ; DISPLAY THE THRESHOLD VALUE ON THE CONSOLE.
31 CD09F CALL CO ; GET THE ASCII CODE BACK
32 D79 INTO ACC FOR COMPARISON.
33 FE0D CPI 00H ; CHECK FOR CARRIAGE RETURN.
34 CA3D40 JZ THRESH ; IF VALUE ENTERED START
35 1D THRESHOLDING.
36 12 EA FOR THRESHOLD VALUE.
37 24 LDAX D ; STORE IN THE RESERVE AREA
38 1A ; ASCII CODE BY MASKING
39 E6F ANI 0F0H
40 FE40 CPI 40H
41 CA3240 JZ HEXA
42 1A LDAX D
43 E5F ANI 0FH
44 C3740 JMP STK
LOC OBJ  LINE SOURCE STATEMENT

4032 1A 36 HEXA: LDAX D
4033 E60F 37 ANI 0FH
4035 C609 38 ADI 09H
4037 12 39 STR: STAX D
4038 04 40 INP B
4039 13 41 INX D

403A C31640 42 INCREMENT COUNTER.
403D 0E0A 43 MOVE POINTER TO NEXT LOC.
403F CD09F8 44 JUMP NEXT.
4042 78 45 RECIEVE NEXT DIGIT OF T
4043 FE01 46 THRESHOLD VALUE.
4045 CA5540 47 CHECK VALUE IN COUNTER.
4048 217640 48 CHECK FOR SINGLE DIGIT.
404B 7E 49 IF SINGLE DIGIT, GO FOR T
404C 07 50 HSHOLDING.
404D 07 51 IF DOUBLE DIGIT THEN AT
404E 07 52 TACH THE TWO DIGITS.
404F 07 53 GET THE FIRST DIGIT.

4050 23 54 NOW THE 4 LSB'S HAVE BEEN
4051 86 55 TRANSFERRED TO THE MSB LOC.
4052 327640 56 ADD M DIGIT TO THE FIRST.
4055 010074 E6 57 STORE THIS THRESHOLD VA
4058 217640 58 LUE(HEX) AT THE SAME LOC.
405B 1640 59 ADDRESS OF STARTING LOC.
405D 1E40 60 OF THE IMAGE.
405F 0A 61 ADDRESS OF THRESHOLD VA
4060 96 E6 62 LOOPS: LDAX B
4061 DA6940 63 FOR THRESHOLDING.
4064 3EFF 64 SUB M
4066 C36B40 65 IF LESS THAN THRESHOLD
4069 3E00 66 VALUE THEN SET IT TO ZERO.
406B 02 67 MVI A, 0FFH
406C 03 68 IF ABOVE THRESHOLD THE
406D 1D 69 N SET THE VALUE EQU 255.
406F C25F40 70 JMP STORE
4071 15 71 REPLACE THE PIXEL IN TH
LOC  OBJ  LINE  SOURCE  STATEMENT

4072  C2 5D 04  72  JNZ LOOP1 ; GO FOR THE NEXT ROW TO PROCESS
4075  C7  73  RST 0
4076  26  74  STIRSH: DS 2
4078  45 4E5445  "75 MESSAGE: DB 'ENTER THE THRESHOLD VALUE IN HEX DECIMAL CODE:"
407C  52295448
4080  45205448
4084  52455343
4088  4F4C4420
408C  56414C55
4090  4520494E
4094  20484558
4098  41444543
409C  494D414C
40A0  20434F44
40A4  45233A20
40A8  24  76  DB 'S'
40B0  0  77  END BEGIN

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS
BEGIN  A 4000  CI  A F803  CO  A F809  HEXA  A 4032
LOOP1  A 405D  LOOP2  A 405F  LPI  A 4008  MESSAGE  A 4075
NEXT  A 4016  SINGLE  A 4055  STORE  A 406B  STR  A 4037
STIRSH  A 4076  THRESH  A 403D  ZERO  A 4069

ASSEMBLY COMPLETE, NO ERRORS
LOC: OBJ: 4000   LINE: 1   SOURCE STATEMENT:

1 1  ********************************************
2 1  * FILENAME := MLITR SRC               *
3 1  ********************************************

5 1  THIS PROGRAM IS USED TO THRESHOLD A GIVEN IMAGE AT A GIVEN MULTIPLE
6 1  THRESHOLD VALUES (THRU THE KEY-BOND.)

7 1

8 1

9 1  ORG 4000H

10 1  CI EQU 0F803H; MONITOR ROUTINE TO SCAN THE KEY-BOARD.

11 1  CO EQU 0F809H; MONITOR ROUTINE TO DISPLAY LAY ON THE CONSOLE.

12 1  BEGIN; LXI SP; CAF8H
13 1  LXI H; MESAG1; TO DISPLAY A MESSAGE ON THE CONSOLE.

14 1  CALL ACCEPT

15 1  LXI H; NUM

16 1  MOV M,A;

17 1

18 1  LXI B; THSLD.

19 1  LXI H; NUM

20 1  MOV D,M

21 1  CONT; PUSH B

22 1  PUSH D

23 1  LXI H; MESAGE

24 1  CALL ACCEPT

25 1  POP D

26 1  POP B

27 1  STAX B

28 1  DCR D;

29 1  JZ; PROCES

30 1  INX B

31 1  JMP CONT

32 1

33 PROCES; LXI B; 7400H

34 1  MVI D; 64

35 1  LOOP1; MVI E; 64

36 LOOP2; PUSH D

37 1  LXI H; NUM

38 1  MOV D,M;

39 1  LXI H; THSLD.

40 1  LDAX B

41 1  SUB M

42 1  JNC NXT

43 1  MVI A; 00H

44 1  JMP STORE

45 NXT; DCR D

46 1  JZ; HILVL

47 1  INX H

48 1  LDAX B

49 1  SUB M

50 1  JC ASNV1
LOC    OBJ    LINE    SOURCE STATEMENT
404A C34040    51         JMP NXT
404D 7E        52         ASNVAL:MOV A,M
404E 2B        53         DCX H
404F B7        54         ORA A
4050 56        55         ADD M
4051 1F        56         RAR
4052 C35740    57         JMP STORE
4055 3EFF      58         HILVL:MYI A,OFFH
4057 D2        59         STORE:STAX B
4058 03        60         INX B
4059 D1        61         POP D
405A 1D        62         DCR E
405B C22E49    63         JNZ LOOP2
405E 15        64         DCR D
405F C22C40    65         JNZ LOOP1
4062 C7        66         RST 0
4063 454F45    67         MESSAG:DB 'ENTER NUMBER OF THRESHOLD LEVEL IN H EX CODE: S'
4067 5220E55    68         MESSAG:DB 'ENTER THRESHOLD VALUE (SMALLER ONE FIRST) IN HEX CODE: S'
4068 4D74552    69         SITRSH:DS 2
406F 204F4620    70         NUM: DS 1
4073 54485245    71         THSLD: DS 0AH
4077 53484F4C    72         SUBROUTINE FOR ACCEPTING THRESHOLD VALUE
407B 44204CA5    73         ..................................................:
407F 56454C20    74         .................................:
4083 494E2048    75         ACCEPT:XI D, SITRSH :ADDRESS TO STORE THE THRESHOLD VALUE
4087 45582043    76         MYI B, 00H :COUNTER TO COUNT # OF B
LOC   OBJ   LINE   SOURCE STATEMENT

40DB  4E   77   LPI:   MOV  C,M   ; GET THE MESSAGE FROM TH
40DC  79   78   E MEMORY LOCATION.
40DD  FE24  79   MOV  A,C   ; USED FOR COMPARING WITH
40DF  CAE9 40 79   THE TERMINATING CHARACTER.
40E0  00  80   INATING CHAR "s".
40E0  CD09F6  80   CPI  0AH   ; ASCII CODE FOR THE TERM
40E1  23F6  80   FOR NEXT OPERATION.
40E2  CD09F8  81   JZ  NEXT   ; IF MESSAGE COMPLETE GO
40E3  5E23  81   NEXT:   CALL  CO   ; DISPLAY THE MESSAGE ON
40E4  CD03  81   THE CONSOLE.
40E5  E900  82   INX  H   ; GO FOR NEXT CHAR IN THE
40E6  CD03F8  82   MESSAGE.
40EC  5E4F  83   JMP  LPI   ; REPEAT THE PROCESS.
40ED  CD09F8  84   CALL  CI   ; READ THE THRESHOLD
40F0  79   84   VALUE FROM THE KEYBOARD.
40F0  00F0  85   MOV  C,A   ; DISPLAY THE THRESHOLD V
40FF  F0E0  85   O THE CONSOLE.
40F1  E6F0  86   CALL  CO   ;
40F2  FE40  87   MOV  A,C   ; GET THE ASCII CODE BACK
40FC  CA0541  87   INTO ACC FOR COMPARISON.
40FF  1A   88   CPI  0DH   ; CHECK FOR CARRIAGE RETU
4100  E6F0  88   RN.(CR)   ; PRINTING.
4102  CA0541  89   JZ  THRESH   ; IF VALUE ENTERED START
4104  1A   89   THRESHOLDING.
4105  E6F0  90   STAX  D   ; STORE IN THE RESERVE AR
4106  1A   90   EA FOR THRESHOLD VALUE.
4108  C609  91   LDAX  D   ; GET ACTUAL VALUE FROM A
4109  1A00  91   ASCII CODE BY MASKING
410B  E6F0  92   ANI  0FH   ; INCREMENT COUNTER.
410C  1A00  93   CPI  40H   ; MOVE POINTER TO NEXT LO
410D  C3E940  93   JZ  HEXA   ; RECIEVE NEXT DIGIT OF T
4110  0E9A   94   JZ  HEXA   ; THRESHOLD VALUE.
4112  CD09F8  95   LDAX  D   ; CHECK VALUE IN COUNTER.
4114  78F6  95   ANI  0FH   ; CHECK FOR SINGLE DIGIT.
4116  F0E1  96   JMP  STR   ; IF SINGLE DIGIT GO FOR
4118  C22241  97   STAX  D   ; THRESHOLDING.
411A  21C940  98   INR  B   ; IF DOUBLE DIGIT THEN AT
411C  7E   99   JMP  EXT   ; TACH THE TWO DIGITS.
411F  C32C41  100   MOV  A,M   ;
4122  21C940  100   JMP  EXT   ;
4122  21C940  101   STR:  STAX  D   ;
4124  F0E1  102   INR  B   ;
4126  1A   103   INX  D   ;
4127  C3E940  104   JMP  NEXT   ;
4129  0E9A  105   THRESH: MVI  C,0AH   ;
412B  CD09F8  106   CALL  CO   ;
412D  78F6  107   MOV  A,B   ;
412F  F0E1  108   CPI  01H   ;
4131  C22241  109   JNZ  DOUBLE   ;
4133  21C940  109   THRESHOLDING.
4135  D108  110   LXI  H,STTRSH   ;
4137  7E   110   MOV  A,M   ;
4139  C32C41  111   JMP  EXT   ;
413C  21C940  111   DOUBLE: LXI  H,STTRSH   ;
413F  7E   112   JMP  EXT   ; IF DOUBLE DIGIT THEN AT
4141  C32C41  112   TACH THE TWO DIGITS.
LOC   OBJ   LINE   SOURCE STATEMENT
4125  7E  114  MOV A,M
4126  07  115  RLC
4127  07  116  RLC
4128  07  117  RLC
4129  07  118  RLC
412A  23  119  EN TRANSFERED TO THE MSB LOC.
        120  INX H
        121  GIT
        122  ADD M
412B  86  123  MOVE POINTER TO NEXT DI
412C  C9  124  D DIGIT TO THE FIRST
4000  00  125  THIS ATTACHES THE SECON

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS
ACCEPT A 40D6
CO   A F809
HEXA A 4105
LP1  A 40DB
NUM  A 40CB
STR  A 410A
ASNV A 404D
CONT A 4014
HILVL A 4055
MESAG1 A 4063
NXT  A 4040
STTRSH A 40C9
BEGIN A 4000
DOUBLE A 4122
LOOPI A 402C
MESSAGE A 4091
PROCESS A 4027
THRESH A 4110
CI    A F823
EXT   A 412C
LOOP2 A 402E
NEXT  A 40E9
STORE A 4057
THS-LD A 40C0

ASSEMBLY COMPLETE, NO ERRORS
LOC OBJ LINE SOURCE STATEMENT

***************************
* FILENAME: 8FLG.SRC *
***************************

CSEG
PUBLIC PICTRE, BREG, CREG
EXTERN NABOR, ADREG

START: MVI B, 02H ; INITIAL ROW FOR BORDER FOLLOW
LPI: MVI C, 01H ; INITIAL COL OF BORDER FOLLOW
SRCH: MOV A, B ; TO SAVE THE PREVIOUS ROW INFORMATION.
STA BREG ; SAVE THE ROW INFORMATION
NLOC BREG ; TO SAVE COL INFORMATION.
MOV A, C ; SAVE THE COL INFORMATION
STA CREG
NLOC CREG
LHLD PICTRE ; LOAD STARTING ADDRESS OF THE PICTURE IN HL REG.
CALL ADREG ; CALL THE SUBROUTINE TO GENERATE PROPER ADDRESS.

NEXTL: MOV D, M ; GET THE PIXEL VALUE INTO D REG.

JAT LOC CREG ; SAVE THE PREVIOUS COL VALUE
STA CREG
INR C ; GO FOR NEXT PIXEL THAT IS
NEXT COL.

MOVC C
CPI 40H ; CHECK FOR END OF THAT ROW.
JZ NXTROW ; IF PRESENT ROW IS OVER GO

LHLD PICTRE ; AGAIN GET THE ADDRESS INTO HL REGS.
CALL ADREG ; GENERATE THE ADDRESS OF NEXT PIXEL.

MOV A, D ; GET THE FIRST PIXEL INTO A
CMP M ; COMPARE WITH THE NEXT ADJACENT PIXEL.
CENT PIXEL.

JZ NEXT ; IF BOTH PIXELS ARE SAME THEN
EN TRY THE NEXT PAIR.

CHK: MOV A, M.

ANSATION.

CPI 00FH
JNZ SCNCHK
MOV A, D
CPI 00H
JZ STARTB

JMP NEXT

SCNCHK: MOV A, M.
LOC OBJ       LINE   SOURCE STATEMENT
0039 FE00    38    CPI 00H
003B C21200   39    JNZ NEXT
003E 7A       40    MOV A, D
003F FEFF     41    CPI 0FFH
0041 CA5100   42    JZ STRTBG
0044 C31200   43    JMP NEXT
0047 04       44    NXTROW: INR B ; TRY THE NEXT ROW OF PIXELS
0048 78       45    MOV A, B
0049 FE3F     46    CPI 3FH ; CHECK FOR END OF PICTURE.
004B CA5900   47    JZ OVER ; IF END OF PICTURE THEN STOP
004E C30200   48    JMP LP1 ; IF NOT END OF PICTURE THEN REPEAT.
0051 C5       49    STRTBG: PUSH B
0052 C0000    50    CALL NABOR ; CALL THE SUBROUTINE TO DO THE BORDER FOLLOWING.
0054 C1       51    POP B
0056 C50400    52    JMP SRCH
0059 C7       53    DVER: RST 0
005A 54       54    BREG: DS 1
005B 55       55    CREG: DS 1
005C-0074     56    PICTRE: DW 7400H
0077 C000     57    END

PUBLIC SYMBOLS
BREG C 005A  CREG C 005B  PICTRE C 005C

EXTERNAL SYMBOLS
ADREG E 0000  NABOR E 0000

USER SYMBOLS
ADREG E 0000  BREG C 005A  CHK? SC 0029  CREG C 005B
LP1 C 0002  NABOR E 0000  NEXT C 0012  NXTROW C 0047
OVER C 0059 PICTRE C 005C  SONCHK C 003B  SRCH C 0004
START C 0000 STRTBG C 0051

ASSEMBLY COMPLETE, NO ERRORS
SOURCE STATEMENT

1. ***********
2. * FILENAME: NABOR.SRC *
3. ***********
4. CSEG
5. BOUNRY MACRO
6. LHLDPIC:TRE : STARTING ADDRESS OF TH
7. E PICTURE
8. CALL ADREGEN : CALL SUBROUTINE TO GEN
9. ERATE ADDRESS REQUIRED.
10. LDA REFVAL : LOAD THE REFERENCE VAL-
11. UE INTO ACC.
12. CMP M : COMPARE IT WITH THE LA
13. TEST PIXEL OBTAINED. : IF NOT EQUAL, IT INDICA-
14. TES A BOUNDARY POINT. SO
15. ENDM
16. XPLUS MACRO REGSTR
17. MOV A, B :LOAD THE ACC WITH THE
18. CURRENT HOWN INFORMN.
19. STA BREG : ALSO SAVE THIS ROW INF
20. RMN FOR LATER USE.
21. ADD REGSTR : ADD THE CONTENTS OF RE-
22. GISTER SPECIFIED.
23. MOV B, A : REPLACE THE ROW INFRMN
24. BACK INTO B REG.
25. ENDM
26. XMINUS MACRO REGSTR
27. MOV A, B
28. STA BREG
29. SUB REGSTR
30. MOV B, A
31. YPLUS MACRO REGSTR
32. MOV A, C : LOAD ACC. WITH THE CURR
33. ENIT COL. INFRMN.
34. STA CREG : ALSO SAVE THIS VALUE F
35. OR LATER USE.
36. ADD REGSTR : ADD THE CONTENTS OF RE-
37. G SPECIFIED.
38. MOV C, A : REPLACE THE COLMN INFR
39. MN. BACK INTO C REG.
40. ENDM
41. YMINUS MACRO REGSTR
42. MOV A, C
43. STA CREG
44. SUB REGSTR
45. MOV C, A
LOC OBJ  LINE  SOURCE STATEMENT

42    ENDM
43 ;
44    PUBLIC NABOR
45    EXTRN ADRGEN, BREG, CREG, PICTRE
46 ;
47    NABOR:
48    MOV A, D
49    L VALUE FOR FUTURE REFERANCE.
50    STA REFVAL
51    MVI M, 33H
52    ; SET THE FIRST BOUNDARY POINT EQUAL TO "3".
53    ; THIS IS TO GET INFORMATION NEEDED IN; FINDING NEIGHBOURS.
54    LDA BREG
55    SUB B
56    ; THIS GIVES THE VALUE 0
57    F "K2".
58    MOV D, A
59    ; SAVE VALUE OF "K2" IN D REG.
60    LDA CREG
61    SUB C
62    ; THIS IS TO GET VALUE 0
63    F "K1"
64    MOV E, A
65    ; SAVE THIS VALUE OF "K1"
66    "IN E REG.
67    LDA BREG
68    MOV B, A
69    ; LOAD THE ACC WITH THE CURRENT ROW INFORMN.
70    LDA CREG
71    MOV C, A
72    ; ALSO SAVE THIS ROW INF RMN FOR LATER USE.
73    STA BREG
74    ADD E
75    ; ADD THE CONTENTS OF REGISTER SPECIFIED.
76    R SPECIFIED.
77    MOV B, A
78    ; REPLACE THE ROW INF RMN BACK INTO B REG.
79    YMINUS D
80    ; STARTING ADDRESS OF THE
81    MOV A, C
82    ; CALL ADRGEN.
83    STA CREG
84    ; CALL SUBROUTINE TO GENERATE ADDRESS REQUIRED.
85    SUB D
86    LDA REFVAL
87    CMP M
88    ; LOAD THE REFERANCE VALUE.
89    ; COMPARE IT WITH THE LAST
90    TEST PIXEL OBTAINED.
91    JNZ NEXT
92    ; IF NOT EQUAL IT INDICATES A BOUNDARY POINT. SO
93    XMINUS D
94    ; GO FOR THE NEXT STEP.
95    MOV A, B
96    ; STA BREG
LOC  OBJ  LINE  -SOURCE STATEMENT
0036  47  81+  MOV B,A
0037  79  82+  YMINUS E
003B  320000  83+  MOV A,C
003B  93  84+  STA CREG
203C  4F  85+  SUB E
003D  2A0000  86+  MOV C,A
87+  BOUNRY

E PICTURE.

CALL ADREG.
CALL SUBROUTINE TO GENERATE ADDRESS REQUIRED.
LOAD THE REFERENCE VALUE.
UE INTO ACC.
COMPARE IT WITH THE LAST TEST PIXEL OBTAINED.
IF NOT EQUAL, IT INDICATES A BOUNDARY POINT. SO GO FOR THE NEXT STEP.

KMINUS D
MOV A,B
STA BREG
SUB D
MOV B,A
YMINUS E
MOV A,C
STA CREG
SUB E
MOV C,A

E PICTURE.
CALL ADREG.
CALL SUBROUTINE TO GENERATE ADDRESS REQUIRED.
LOAD THE REFERENCE VALUE.
UE INTO ACC.
COMPARE IT WITH THE LAST TEST PIXEL OBTAINED.
IF NOT EQUAL, IT INDICATES A BOUNDARY POINT. SO GO FOR THE NEXT STEP.

XMINUS E
MOV A,B
STA BREG
SUB E
MOV B,A
YPLUS D
MOV A,C
ENT COL. INFRMN.
LOAD ACC WITH THE CURRENT COLUMN INFORMATION.
ALSO SAVE THIS VALUE FOR LATER USE.
ADD D
ADD THE CONTENTS OF REG SPECIFIED.

MOV C,A
MN BACK INTO C REG.
LOC OBJ

0066 2A0000  E 122+ CALL HLD PICTRE ; STARTING ADDRESS OF THE
0067 CD0000  E 123+ CALL ADRGEN ; CALL SUBROUTINE TO GENERATE
0068 3D0000  C 124+ LDA REFL ; LOAD THE REFERENCE VALUE
0069 3D0000  C 125+ CMP M ; COMPARE IT WITH THE LAST VALUE
006A C2A000  C 126+ JNZ NEXT ; IF NOT EQUAL IT INDICATES
006A D70000  E 127+ MOV A, B ; GO FOR THE NEXT STEP
006B 610000  E 128+ MOV C, A ; LOAD ACC WITH THE CURRENT
006C 3C0000  E 129+ STA BREG ; ALSO SAVE THIS VALUE FOR LATER USE
006D 820000  E 130+ ADD D ; ADD THE CONTENTS OF REG SPE
006E 410000  E 131+ MOV C, A ; REPLACE THE COLUMN INFRMN
006F 200000  E 132+ LDA REFL ; STARTING ADDRESS OF THE
0070 030000  E 133+ CMP M ; CALL SUBROUTINE TO GENERATE
0071 070000  E 134+ JNZ NEXT ; LOAD THE REFERENCE VALUE
0072 3A0000  E 135+ MOV A, C ; IF NOT EQUAL IT INDICATES
0073 0C0000  E 136+ STA CREG ; GO FOR THE NEXT STEP
0074 3D0000  C 137+ MOV C, A ; LOAD ACC WITH THE CURRENT
0075 320000  E 138+ STA BREG ; ALSO SAVE THIS ROW INF
0076 820000  E 139+ ADD D ; ADD THE CONTENTS OF REG SPE
0077 410000  E 140+ MOV C, A ; REPLACE THE COLUMN INFRMN
0078 200000  E 141+ LDA REFL ; STARTING ADDRESS OF THE
0079 030000  E 142+ CMP M ; CALL SUBROUTINE TO GENERATE
007A 070000  E 143+ JNZ NEXT ; LOAD THE REFERENCE VALUE
007B 3A0000  E 144+ MOV A, C ; IF NOT EQUAL IT INDICATES
007C 0C0000  E 145+ XPLUS D ; GO FOR THE NEXT STEP
007D 0E0000  E 146+ MOV A, B ; LOAD ACC WITH THE CURRENT
007E 320000  E 147+ STA BREG ; ALSO SAVE THIS ROW INF
007F 820000  E 148+ ADD D ; ADD THE CONTENTS OF REG SPE
0080 410000  E 149+ MOV C, A ; REPLACE THE COLUMN INFRMN
0081 200000  E 150+ LDA REFL ; STARTING ADDRESS OF THE
0082 030000  E 151+ CMP M ; CALL SUBROUTINE TO GENERATE
0083 070000  E 152+ JNZ NEXT ; LOAD THE REFERENCE VALUE
0084 3A0000  E 153+ MOV A, C ; IF NOT EQUAL IT INDICATES
0085 0C0000  E 154+ STA CREG ; GO FOR THE NEXT STEP
0086 410000  E 155+ MOV C, A ; LOAD ACC WITH THE CURRENT
0087 200000  E 156+ LDA REFL ; ALSO SAVE THIS ROW INF
0088 030000  E 157+ CMP M ; ADD THE CONTENTS OF REG SPE
0089 070000  E 158+ JNZ NEXT ; REPLACE THE COLUMN INFRMN
008A 3A0000  E 159+ MOV A, C ; LOAD ACC WITH THE CURRENT
008B 0C0000  E 160+ STA CREG ; ALSO SAVE THIS VALUE FOR LATER USE
008C 410000  E 161+ MOV C, A ; ADD E
008D 200000  E 162+ LDA REFL ; REPLACE THE COLUMN INFRMN
008E 030000  E 163+ CMP M ; ADD THE CONTENTS OF REG SPE
008F 070000  E 164+ JNZ NEXT ; REPLACE THE COLUMN INFRMN
00A1 2A0000 E 155 BOUNRY
     156 LHLH PICTRE
00A4 CD0000 E 157 CALL ADREGEN
     158 ERATE ADDRESS REQUIRED.
00A7 3AD800 C 159 UE INTO ACC.
     160 LDA REFVAL
00AA BE 161 CMP M
     162 TEST PIXEL OBTAINED.
00AB C2CA00 C 163 JNZ NEXT
     164 TES A BOUNDARY POINT SO
00AE 78 165 GO FOR THE NEXT STEP.
00AF 320000 E 166 XPLUS D
     167 MOV A,B
     168 CURRENT ROW INFO.
00B2 82 169 STA BREG
     170 R MN FOR LATER USE.
00B3 47 171 ADD D
     172 R SPECIFIED.
00B4 79 173 MOV B,A
     174 BACK INTO B REG.
00B5 320000 E 175 YPLUS E
     176 MOV A,C
     177 ENT COL INFO.
00B8 83 178 OR LATER USE.
00B9 4F 179 ADD E
     180 CIFIED
00BA 2A0000 E 181 MOV C,A
     182 MN BACK INTO C REG.
00BD CD0000 E 183 LHLH PICTRE
     184 BOUNRY
00C0 3AD800 C 185 CALL ADREGEN
     186 ERATE ADDRESS REQUIRED.
00C3 BE 187 UE INTO ACC.
     188 CMP M
     189 TEST PIXEL OBTAINED.
00C4 C2CA00 C 190 JNZ NEXT
     191 TES A BOUNDARY POINT SO
00C7 C30700 C 192 JMP OVER
     193 BOUNDARY POINT IS PRESENT
     194 THEN IT IS AN ISOLATED
00CA 7E 195 POINT.
00CB FE33 196 MOV A,M
00CD CD5000 C 197 CPI 33H
00DD 3632 198 ST BOUNDARY POINT
00D2 C35600 C 199 JPALM AST
00D0 3032 200 R THE LAST STEP.
00D1 3232 201 NDARY POINT, EQUAL TO 32H.
00D2 C35600 C 202 JMP LP1
LOC OBJ LINE SOURCE STATEMENT

FINDING NEXT BOUNDARY POINT.

00D5 3632 186 ALMOST: MVI M,32H
00D7 C9 187 OVER: RET
00D8 188 REVAL: DS 1
189 END

PUBLIC SYMBOLS
NABOR C 0000

EXTERNAL SYMBOLS
ADREG E 0000 BREG E 0000 CREG E 0000 PICTRE E 0000

USER SYMBOLS
ADREG E 0000 ALMOST C 00D5 BOUNHY + 0000 BREG E 0000
CREG. E 0000 LP1. C 0006 NABOR C 0000 NEXT C 00CA
OVER C 00D7: PICTRE E 0000 REVAL C 00D8 XMINUS + 0005
XPLUS + 0003 YMINUS + 0008 YPLUS + 0006

ASSEMBLY COMPLETE, NO ERRORS
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>D5</td>
<td>9</td>
<td>ADRGEN: PUSH D ; SAVE PIXEL OF D REGISTER.</td>
</tr>
<tr>
<td>0001</td>
<td>C5</td>
<td>10</td>
<td>PUSH B ; SAVE ROW AND COL. INFORMATIONS</td>
</tr>
<tr>
<td>0002</td>
<td>E5</td>
<td>11</td>
<td>PUSH H ; SAVE ADDRESS IN &lt;HL&gt; REG. PAIR</td>
</tr>
<tr>
<td>0003</td>
<td>05</td>
<td>12</td>
<td>DCR B ; GET THE EXACT NUMBER OF ROWS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>TO BE ADVANCED FROM THE REFERE</td>
</tr>
<tr>
<td>0004</td>
<td>48</td>
<td>14</td>
<td>MULTIPLICATION WILL BE DONE NOW. MULTIPLI</td>
</tr>
<tr>
<td>0005</td>
<td>1E</td>
<td>15</td>
<td>ER ; IS THE NUMBER OF ROWS TO BE ADVANCED AN</td>
</tr>
<tr>
<td>0007</td>
<td>06</td>
<td>16</td>
<td>D ; THE MULTIPLICAND IS THE NUMBER OF PIXEL</td>
</tr>
<tr>
<td>0009</td>
<td>16</td>
<td>17</td>
<td>S IN EACH ROW.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18</td>
<td>MOV C, B ; MULTIPLIER IN C REG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19</td>
<td>MVI E, 40H; MULTICANT IN E REG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>MVI B, 08H; BIT COUNTER.</td>
</tr>
<tr>
<td>000D</td>
<td>AF</td>
<td>21</td>
<td>MULT: MVI D, 0H</td>
</tr>
<tr>
<td>000E</td>
<td>79</td>
<td>22</td>
<td>MOV H, D</td>
</tr>
<tr>
<td>000F</td>
<td>B7</td>
<td>23</td>
<td>MOV L, D</td>
</tr>
<tr>
<td>0010</td>
<td>CA</td>
<td>24</td>
<td>MULT: XRA A</td>
</tr>
<tr>
<td>0013</td>
<td>1F</td>
<td>25</td>
<td>MOV A, C</td>
</tr>
<tr>
<td>0014</td>
<td>4F</td>
<td>26</td>
<td>ORA A</td>
</tr>
<tr>
<td>0015</td>
<td>21</td>
<td>27</td>
<td>JZ ADECOL</td>
</tr>
<tr>
<td>0018</td>
<td>19</td>
<td>28</td>
<td>RAR</td>
</tr>
<tr>
<td>0019</td>
<td>EB</td>
<td>29</td>
<td>MOV C, A</td>
</tr>
<tr>
<td>001A</td>
<td>29</td>
<td>30</td>
<td>JNC MULTI</td>
</tr>
<tr>
<td>001B</td>
<td>EB</td>
<td>31</td>
<td>DAD D</td>
</tr>
<tr>
<td>001C</td>
<td>05</td>
<td>32</td>
<td>MULTI: XCHG</td>
</tr>
<tr>
<td>001D</td>
<td>20</td>
<td>33</td>
<td>DAD H</td>
</tr>
<tr>
<td>0020</td>
<td>D1</td>
<td>34</td>
<td>XCHG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35</td>
<td>DCR B</td>
</tr>
<tr>
<td>0021</td>
<td>19</td>
<td>36</td>
<td>JNZ MULTI; ADDRESS OF ROW IS IN &lt;HL&gt;</td>
</tr>
<tr>
<td>0022</td>
<td>C1</td>
<td>37</td>
<td>ADCOL: POP D ; POP THE REFERENCE ADDRESS INTO</td>
</tr>
<tr>
<td>0023</td>
<td>C5</td>
<td>38</td>
<td>&lt;DE&gt;</td>
</tr>
<tr>
<td>0024</td>
<td>59</td>
<td>39</td>
<td>REG. PAIRS WHICH WERE IN &lt;HL&gt; P</td>
</tr>
<tr>
<td>0025</td>
<td>1D</td>
<td>40</td>
<td>AIR</td>
</tr>
<tr>
<td>0026</td>
<td>16</td>
<td>41</td>
<td>DAD D ; STARTING ADDRESS OF A ROW IS IN</td>
</tr>
<tr>
<td></td>
<td>00</td>
<td>42</td>
<td>POP B</td>
</tr>
<tr>
<td>0028</td>
<td>19</td>
<td>43</td>
<td>PUSH B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>44</td>
<td>MOV E, C ; COLUMN INFORMATION IN E REG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>45</td>
<td>DCR E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>46</td>
<td>MVI D, 0H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>47</td>
<td>DAD D ; CORRECT ADDRESS IS IN &lt;HL&gt; REG. F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>48</td>
<td>AIR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>49</td>
<td>POP B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>PUSH B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>51</td>
<td>MOV E, C ; COLUMN INFORMATION IN E REG.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>52</td>
<td>DCR E</td>
</tr>
<tr>
<td></td>
<td></td>
<td>53</td>
<td>MVI D, 0H</td>
</tr>
<tr>
<td></td>
<td></td>
<td>54</td>
<td>DAD D ; CORRECT ADDRESS IS IN &lt;HL&gt; REG. F</td>
</tr>
</tbody>
</table>
LOC  OBJ  LINE  SOURCE STATEMENT

0029  C1  .  47  POP B
002A  D1  .  48  POP D
002B  C9  .  49  RET
       50  END

PUBLIC SYMBOLS
ADRG C 0000

EXTERNAL SYMBOLS

USER SYMBOLS
ADCOL C 0020  ADRG C 0000  MULTI C 0009  MULT C 00AD

ASSEMBLY COMPLETE; NO ERRORS
SOURCE STATEMENT

PROGRAM FOR LINE DETECTION

FILENAME: EDGDET.*

CO EQU 0F009H
LENGTH EQU 04H
PUBLIC ROW, COLMN
EXTRN ADRGN1, BORDR

CSEG

START: LXI SP, 0AFFFH
MVI A, 00H
STA COUNT1
MVI A, 01H
DO1: STA ROW
MVI A, 00H
DO: MVI B, LENGTH
ADD B
STA COLMN
FSTPNT: CALL ADRGN1.

APPROPRIATE PIXEL ADDRESS IS IN 
<HL> REG PAIR.
CALL TRANS
LDA Flag:
CPI 01H ; CHECK WHETHER THERE IS 
ANY TRANSITION.

JZ CHKBAK
ADVANCE: LDA COLMN
MVI B, LENGTH
ADD B ; IF THERE IS NO TRANSITION 
GO TO NEXT SEARCH-POINT.

CPI 64...
JNC NXTROW
STA COLMN
JMP FSTPNT
NXTROW: LDA ROW
INR A
CPI 64
JNZ 001
JMP EXIT

CHKBAK: MVI A, LENGTH ; CHECK WHETHER OTHER 
POINTS ALSO SATISFY 
REQUIRED TRANSITION.

DCR A
CPI 00H
JNZ STCNTR
LDA COLMN
DCR A
STA COLMN
JMP MESSAGE
STCNTR: LDA COUNT
LOC  OBJ  LINESOURCE STATEMENT
005C 32C800  C  55CONT STA COLMN
005F CD0000 E  56"CALL ADHGN1"
0062 3ACD00 C  57LDA TOPREF
0065 96   58SUB N ;CHECK WHETHER THE PRESENT
0066 CA6F00 C  59POINT IS SAME AS THE LAST
0067 60   60POINT OF THE TEMPLATE.
0068 C9F000 C  61JZ TRTRNS ;IF IT IS CHECK WHETHER
0069 61   62THERE IS THE REQUIRED TRANS-
006A ACB00 C  63ITION PRESENT.
006C C32300 C  64LDA COLMN
006F CDF7001 C  65JMP ADVANCE
0072 3ACC00 C  66TRTRNS CALL TRANS
0075 FE01   67LDA FLAG
0077 CA9000 C  68CPI '01H
007A ACB00 C  69JZ NXTPXL
007D C32300 C  70LDA COLMN
0080 3ACD00 C  71JMP ADVANCE
0083 3D   72NXTPXL LDA COUNT
0084 32C000 C  73DCR A
0087 FE001  74STA COUNT
0089 CA9300 C  75CPI '00H
008C ACB00 C  76JZ MESSAEGE
008F 3C   77LDA COLMN
0090 C35C00 C  78JMP CONT
0093 ACB00 C  79MESAGE LDA COLMN
0096 3C   80INR A
0097 9694   81MVI B LENGTH
0099 05   82DCR B
009A 90   83SUB B
009B 32CB00 C  84STA COLMN
009E 4F   85MOV C A
009F 3AC000 C  86LDA ROW
00A2 47   87MOV B A
00A3 CD0000 E  88CALL BORDR
00A6 3ACF00 C  89LDA COUNT1
00A9 3C   90INR A
00AA 32CF00 C  91STA COUNT1
00AD FE04   92CPI '04H
00AF CAC900 C  93JZ EXIT2
00B2 3ACB00 C  94LDA COLMN
00B5 C32300 C  95JMP ADVANCE
00BB 21DE00 C  96EXIT1 LXI H NEW1
00BE 4E   97LOOP MOV C M
00BC 79   98MOV A C
00BD FE24   99CPI 24H
00BE CAC900 C 100JZ EXIT2
00C2 CD09FB   101CALL CO
00C5 23   102INX H
00C6 C3BB00 C 103JMP LOOP
00C9 C7   104EXIT3 RST 0
00CA   105ROW 'DS 1
00CB   106"COLUMN 'DS 1
00CC   107"FLAG 'DS 1
00CD   108TOPREF 'DS 1
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>00CE</td>
<td>-</td>
<td>110</td>
<td>COUNT: DS 1</td>
</tr>
<tr>
<td>00CF</td>
<td>-</td>
<td>111</td>
<td>COUNT: DS 1</td>
</tr>
<tr>
<td>00DF</td>
<td>20205445</td>
<td>112</td>
<td>NEWS1: DB 'TEMPLATE IS NOT MATCHED IN THE IMAGES'</td>
</tr>
<tr>
<td>00D4</td>
<td>4D504C41</td>
<td>113</td>
<td>TRANS: PUSH H</td>
</tr>
<tr>
<td>00D8</td>
<td>54452049</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00DC</td>
<td>53204E4F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00E0</td>
<td>54204D41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00E4</td>
<td>54434845</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00E8</td>
<td>4420494E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00EC</td>
<td>20544845</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00F0</td>
<td>28494D41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00F4</td>
<td>474524</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00F7</td>
<td>55</td>
<td>114</td>
<td>MOV D,H</td>
</tr>
<tr>
<td>00F8</td>
<td>54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00F9</td>
<td>5D</td>
<td>115</td>
<td>MOV E,L</td>
</tr>
<tr>
<td>00FA</td>
<td>AF</td>
<td>116</td>
<td>XRA A</td>
</tr>
<tr>
<td>00FB</td>
<td>1A</td>
<td>117</td>
<td>LDAX D</td>
</tr>
<tr>
<td>00FC</td>
<td>FFF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00FE</td>
<td>C21101</td>
<td>C</td>
<td>119</td>
</tr>
<tr>
<td>0101</td>
<td>32CD00</td>
<td>C</td>
<td>120</td>
</tr>
<tr>
<td>0104</td>
<td>114000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0107</td>
<td>12</td>
<td>121</td>
<td>LXI D,64</td>
</tr>
<tr>
<td>0108</td>
<td>7E</td>
<td>122</td>
<td>DAD D</td>
</tr>
<tr>
<td>0109</td>
<td>EE00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>010B</td>
<td>C21101, C</td>
<td>123</td>
<td>MOV A,M</td>
</tr>
<tr>
<td>010C</td>
<td>C31601, C</td>
<td>124</td>
<td>CPI 00H</td>
</tr>
<tr>
<td>0111</td>
<td>3E00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0113</td>
<td>C31801</td>
<td>C</td>
<td>125</td>
</tr>
<tr>
<td>0116</td>
<td>3E01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0118</td>
<td>32CD00</td>
<td>C</td>
<td>126</td>
</tr>
<tr>
<td>011B</td>
<td>E1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>011C</td>
<td>C9</td>
<td>127</td>
<td>LDFLAG, STFLAG: MVI A,01H</td>
</tr>
<tr>
<td>00C0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PUBLIC SYMBOLS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLMN C 00CB</td>
<td>ROW C 00CA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EXTERNAL SYMBOLS
ADRGN1 E 0000 BORDR E 0000

USER SYMBOLS
ADRGN1 E 0000 ADVANCE C 0023 BORDR E 000A CHKBAK C 0040
CO A F809 COLMN C 00CB CONT C 005C COUNT C 00CE
COUNT1 C 00CF DO C 000F D01 C 000A DNFLAG C 0111
EXIT C 0088 EXIT2 C 0089 FLAG C 00CC FSTPNT C 0015
LDFLAG C 0118 LENGTH A 0004 LJMP C 00BB MESSAGE C 0233
NEWS1 C 00D0 NXTPXLC C 0050 NXTROW C 0034 NOV C 00CA
START C 0000 STRCONTR C 0052 STFLAG C 0116 TOPREF1 C 006D
TRANS C 0000 TRRNS C 006F

ASSEMBLY COMPLETE, NO ERRORS
SOURCE STATEMENT

**FILENAM**E: ADRGN1.SRC

**SEG**

PUBLIC ADRGN1
EXTRN ROW, COLMN

0000 3A0000  E  10 ADRGN1: LDA ROW ; GET ROW INFORMATION.
0003 47     11 MOV B, A
0004 3A0000  E  12 LDA COLMN
0007 4F     13 MOV C, A
0008 C5     14 PUSH B ; SAVE ROW AND COL. INFORMATION.
0009 210074 15 LXI H, 7400H
000C E5     16 PUSH H ; SAVE ADDRESS IN <HL> REG. PAIR
000D 05     17 DCR B ; GET THE EXACT NUMBER OF ROWS.
000E 48     18 ; TO BE ADVANCED FROM THE REFERENC
0011 0608  19 ; MULTIPICATION WILL BE DONE NOW. MULTIPLI
0013 1600  20 ; IS THE NUMBER OF ROWS TO BE ADVANCED AN
0015 6A     21 ; THE MULTIPLICAND IS THE NUMBER OF PIXEL
0016 AF     22 ; IN EACH ROW.
0018 79     23 MOV C, B ; MULTIPLIER IN C REG.
0019 B7     24 MVI E, 40H; MULTIPLICANT IN E REG.
001A CA2A00 25 MVI B, 08H; BIT COUNTER.
001D 1F     26 MULT: MUL D,ON
001E 4F     27 MOV H, D
001F D22300 C 28 MOV L, D
0022 49     29 MULTO: XRA A
0023 EB     30 MOV A, C
0024 29     31 ORA A
0025 EB     32 JZ ADCOL
0026 05     33 RAR
0027 C2,1700 C 34 MOV C, A
0028 D1     35 JNC MULTI
0029 49     36 DAD D
002A EB  37 MULT: XCHG
002B 29  38 DAD H
002C EB  39 XCHG
002D 05  40 DCR B
002E 05  41 JNZ MULTO ; ADDRESS OF ROW IS IN <HL>
002F 04  42 ADCOL: POP D ; POP THE REFERENCE ADDRESS INTO
0030 04  43 <DE>
0031 04  44 ; REG. PAIRS WHICH WAS IN <HL> P-
0032 19  45 AIR.
0033 C1  46 DAD D ; STARTING ADDRESS OF A ROW IS IN
0034 59  47 <HL>.

0035 46  48 POP B
0036 C1  49 MOV E, C ; COLUMN INFORMATION IN E REG.
<table>
<thead>
<tr>
<th>LOC OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>'002E 1D</td>
<td>48</td>
<td>DCR E</td>
</tr>
<tr>
<td>'002F 1600</td>
<td>49</td>
<td>MVI D, #H</td>
</tr>
<tr>
<td>'0031 19</td>
<td>50</td>
<td>DAD D</td>
</tr>
<tr>
<td>'0032 C9</td>
<td>51</td>
<td>AIR</td>
</tr>
<tr>
<td>'0032 C9</td>
<td>52</td>
<td>RET</td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS
ADRN1 C 0000

EXTERNAL SYMBOLS
COLMN E 0000, ROW E 0000

USER SYMBOLS
ADCOL C 002A, ADRGN1 C 0000, COLMN E 0000, MUL C 0013
MULT C 0017, MULTI C 0023, ROW E 0000

ASSEMBLY COMPLETE, NO ERRORS
:PROGRAM FOR HORIZONTAL
:TEMPLATE MATCHING

FILE NAME: IMPL.**

***

EXTERN ADDRG,N,ROWSUM,BORDR.
CO, EQU 0F809H
TRSH, EQU 09F6H
LENGTH, EQU 05H

CSEG.

START: LXI SP, 0BFFFH

MVI B, 03H
LOAD THE NUMBER OF
ROWS OF THE TEMPLATE
INTO THE B-REG.

RSTRT: MVI C, LENGTH
LOAD THE NUMBER OF
COLUMNS OF THE TEMPLATE
INTO THE C-REG.

RSTRT1: PUSH B

CALL ADDRG
GET THE LOCATION FOR
STARTING THE ACTION.

CALL ADDRG
GET THE ADDRESS IN THE
<DE> REG PAIR.

CALL ROWSUM
CALCULATE THE SUM OF
THE ROWS OF THE PIXELS
WITHIN THE Template.
RESULT IS IN <HL> REG PAIR.

IR.

POP B
GET BACK THE ROW-COL IN
FORMATIONS.

PUSH B
SAVE THE PREVIOUS RESULT.

T.

INR B
MOVE NEXT TO NEXT ROW.

INR B
GET THE PROPER ADDRESS.

CALL ADDRG
ADDRESS IN <DE>.

CALL ROWSUM
POP THE 1ST ROW'S SUM INTO
<BC> REG PAIR. SUM OF 3R

D ROW.
IS IN <HL> REG PAIR.
SUM OF 1ST ROW + SUM OF
3RD ROW IS IN <HL> REG

PAIR.

POP B
GET THE ROW-COL. INFORM
FORMATIONS.

PUSH H
SAVE THE PREVIOUS RESULT.
LOC    OBJ    LINE    SOURCE STATEMENT
0050  036780  C  91    JMP RSTRT1
0053  078     C  92    CHKROW:MOV A,B
0054  FE40    C  93    CPI 40H
0056  065100  C  94    JZ NOMTCH
0059  04      C  95    INX B
005A  035000  C  96    JMP RSTRT
005D  218100  C  97    MATCH: LXI H, RMESAG
0060  366000  C  98    JMP SHOW
0063  219500  C  99    NOMTCH: LXI H, NMESAG
0066  04      C  100  SHOW: MOV C, M
0067  79      C  101  MOV A, C
0068  FE24    C  102  CPI 24H
006A  074000  C  103  JZ EXT
006D  C00000  C  104  CALL CO
0070  23      C  105  INX H
0071  366000  C  106  JMP SHOW
0074  01      C  107  EXT: POP B
0075  05      C  108  DCR B
0076  05      C  109  DCR B
0077  79      C  110  MOV A, C
0078  1605    C  111  MVI D, LENGTH
007A  15      C  112  DCR D
007B  92      C  113  SUB D
007C  04      C  114  MOV C, A
007D  C00000  E  115  CALL BORDR
0080  07      E  116  RST 0
0081  54454050 107  RMESAG: DB 'TEMPLATE IS MATCHED'
0085  4C415445 111  RMESAG: DB 'TEMPLATE IS NOT MATCHED'
0089  2E495320
008D  4D415443
0091  4A544424
0095  54454450 118  RMESAG: DB 'TEMPLATE IS NOT MATCHED'
0099  4C415445
009D  2E495320
00A1  4E4F5420
00A5  4D415443
00A9  4A544424
0000  C  119    - END START -

PUBLIC SYMBOLS

EXTERNAL SYMBOLS
ADDRGN E 0000  BORDR E 0000  ROWSUM E 0000

USER SYMBOLS
ADDRGN E 0000  BORDR E 0000  CHKROW C 0A53  CMPLNT C 0034
CO A F699  EXT C 0074  LENGTH A 0005  MATCH C 005D
NMESAG C 0095  NOMTCH C 0063  RMESAG C 0051  ROWSUM E 0000
RSTRT C 0005  RSTRT1 C 0007  SHOW C 0066  START C 0000
TRSH A 09F6  TRSLS D 003B

ASSEMBLY COMPLETE, NO ERRORS.
; PROGRAM FOR CALCULATING THE
; SUM OF THE PIXELS IN A ROW
; WITHIN THE TEMPLATE.
; *******************************************
; FILE NAME: ROWSUM.*
; *******************************************

CSEG
PUBLIC ROWSUM

0000 DS:
0001 0E00
0003 210000
0006 1A
0007 D5
0009 1600
000B 19
000C D1
000E 79
000F FE05
0011 CA1600
0014 13
0015 C30600
0018 D1
0019 C9

0000 DS:
0001 0E00
0003 210000
0006 1A
0007 D5
0009 1600
000B 19
000C D1
000E 79
000F FE05
0011 CA1600
0014 13
0015 C30600
0018 D1
0019 C9

PUBLIC SYMBOLS
ROWSUM C 0000

EXTERNAL SYMBOLS

USER SYMBOLS
LP1 C 0006
LSTISP C 001B
ROWSUM C 0000

ASSEMBLY COMPLETE, NO ERRORS
LOC OBJ LINE SOURCE STATEMENT

*******************************

FILENAME: FILL*
***************

PROGRAM FOR SCANNING THE
INTERIOR OF A CLOSED
REGION AND REMOVING THE,
SPURIOUS FEATURES PRESENT
IN IT.

PUBLIC FILL

EXTERNAL MINUSB, MINUSC, PRSNTB, PRSNTC, PICTRE

ADRGEN

CSEG
MINUSB AND MINUSC ARE THE
X AND Y CO-ORDINATES OF THE
PREVIOUS POINT, PRSNTB AND
PRSNTC ARE THE PRESENT CO-
ORDINATES, AND ADRGEN
GENERATES THE ADDRESS OF THE
PIXELS.

0000 0C5
0001 AF
0002 3A0000 E 23
0005 90
0006 C5A4 00 C 25
0009 220000 C 26
000C 3A0000 E 27
000F 47
0010 3A0000 E 28
0013 4F
0014 2A0000 E 29
0017 CD0000 E 30
001A 3601
001C 2B
001D 1E FE33
0020 C6500 C 31
0023 FE01
0025 C6500 C 32
0028 FE32
002A C6500 C 33
002D C31000 C 34
0030 3A0000 E 35
0033 47
0034 3A0000 E 36
0037 4F
003B 2A0000 E 37
003F CD0000 E 38
0043 3601
0048 29
0041 7E
0042 FE33
0044 C6500 C 39

FILL: PUSH B
XRA A
LDA MINUSB
SUB B
JZ SCNCHK
JNC LIK

RIOL: LDA PRSNTB
MOV B A
LDA PRSNTC
MOV C A
LHLD PICTRE
CALL ADRGEN

LOOP: MVI M 01H
DCX H
MOV A M
CPI 33H
JZ ENDI
CPI 01H
JZ ENDI
CPI 32H
JZ ENDI
JMP LOOP

LTOR: LDA PRSNTB
MOV B A
LDA PRSNTC
MOV C A
LHLD PICTRE
CALL ADRGEN

LOOP: MVI M 01H
INX H
MOV A M
CPI 33H
JZ ENDI
```
LOC   OBJ    LINE    SOURCE STATEMENT

0047 F0E1   54    CPI 01H
0049 CA8500 55    JZ END1
004C F032   56    CPI 03H
004E CA6500 57    JZ END1
0051 C33E00 58    JMP LOOP1
0054 3A0000 59    SCNCHK LDA PRSNTB
0057 47     60    MOV B A
0058 3A0000 61    LDA PRSNTC
005B 4F     62    MOV C A
005C 2A0000 63    LHL DL PICHE
005F CD0000 64    CALL ADRGEN
0062 3601   65    MVI $A 01H
0064 AF     66    XRA A
0065 3A0000 67    LDA MINUSB
0068 90     68    SUB B
0069 CA8500 69    JZ END1
006C DATA00 70    JC CHK2
006F AF     71    XRA A
0070 3A0000 72    LDA MINUSC
0073 91     73    SUB C
0074 D2300 74    JNC LTOR
0077 C3500 75    JMP ENDi
007A AF     76    CHK2 XRA A
007B 3A0000 77    LDA MINUSC
007E 91     78    SUB C
007F DA0000 79    JC LTOR
0082 C3500 80    JMP ENDi
0085 C1     81    END1 POP B
0086 C9     82    RET
0087 C9     83    END

PUBLIC SYMBOLS
FILL C 0000

EXTERNAL SYMBOLS
ADRGEN E 0000 MINUSB E 0000 MINUSC E 0000 PICTRE E 0000
PRSTB E 0000 PRSNTC E 0000

USER SYMBOLS
ADRGEN E 0000 CHK2 C 007A END1 C 0085 FILL C 0000
LOOP C 001A LOOP1 C 003E LTOR C 0030 MINUSB E 0000
MINUSC E 0000 PICTRE E 0000 PRSNTB E 0000 PRSNTC E 0000

ASSEMBLY COMPLETE: NO ERRORS
```
SOURCE STATEMENT

************

1 PROGRAM FOR TRANSFERRING DATA FROM
2 BLUE BOX TO OTHER COMPUTER.
3 THE NUMBER OF DATA HAS BEEN
4 ASSUMED TO BE 4096.
5 TO SEND DIFFERENT NUMBER OF DATA
6 THE CHECK FOR THE MAXIMUM NUMBER
7 IN THE CONTAINER IS TO BE CHANGED.
8 FIRST THE LEAST SIGNIFICANT
9 4 BITS ARE SENT BY MASKING
10 AND THEN THE MOST SIGNIFICANT
11 4 BITS ARE SENT ALSO BY MASKING.
0000 D
0000 A
F809
0003
0040
00CE
0004
00FA
00F7
00F6
00F7
00F6
0004
0020
0020
0002
0070
4000
4000 31FF0A
4003 3E40
4005 D3F7
4007 3ECE
4009 D3F7

CR EQU 0DH
LF EQU 0AH
CO EQU 0F809H
EOF EQU 03H; END OF FILE
RESET EQU 0100000B; RESET CHIP
MODEL EQU 1101110B; MODE INSTRUCTION
RXE EQU 0000100B; RECEIVE ENABLE
RTCT EQU 250; READER TIME OUT CONSTANT
TTYC EQU 0F7H; OUTPUT CONTROL PORT
TTVI EQU 0F6H; INPUT DATA PORT
TTVS EQU 0F7H; INPUT STATUS PORT
TTYO EQU 0F6H; OUTPUT DATA PORT
TXBE EQU 0000100B; TRANSMIT BUFFER EMPT
CTSI EQU 0010000B; SIMULATE CLEAR TO SEND
RTS EQU 0100000B; SET REQ TO SEND OUT
RRDY EQU 0000100B; RECEIVE BUFFER EMPT
ONEMS EQU 112; 1 MS TIME OUT CONSTANT
ORG 4000H

START: LXI SP, 0AFFH; INITIALIZING STACK POINTER

***************

INITIALIZATION OF 8251 CHIP.

***************

INITIALIZATION OF 8253 CHIP.
MVI A,$076H; LOAD 3253 COUNTER 2 FOR MODE AND LSB FOLLOWED BY MSB.
        OUT $0F3H
LXI H,$008H; BAUDE CODE FOR 9600 RATE.
MOV A,L
        OUT $0F1H; LOAD LSR.
MOV A,H; MSB OF BAUDE RATE.
        OUT $0F1H

MAIN PROGRAM

MVI A,$025H
OUT $014H; ENABLE TRANSMIT, RECEIVE ANDREQ TO SEND.
LXI H,$7400H; DATA LOADING STARTS FROM HE RE.

LXI B,$00H; INITIALIZE THE COUNTER.
IN TTYI; CLEAR BY INITIAL READING.

DDTAMOVI A,$W
        CPI $03H
        JNZ LOOP
MVI A,$04H.

LOOP: CALL WRITE
MOVI A,$B.
        CPI $10H
        JNZ RDTA
MVI A,$03H.
        CALL WRITE
        RST 0
WRITE; PUSH B.
PUSH H.
PUSH PSW
SEERDY; IN TTYS.
        ANI RRDY
        JZ SEERDY.
BEGIN; IN TTYI.
        CPI $05H; SEARCH FOR ENQ.
        JNZ SEERDY
AGAIN; IN TTYS; READ STATUS.
        ANI TXE; CHECK FOR EMPTY RECEIVE BUFFER.
        JZ AGAIN
MVI A,$06H; SEND ACKNOWLEDGEMENT.
        OUT TTYO
GOOPL; IN TTYS
        ANI TXE
        JZ GOOPL
        ANI TXE
        JZ GOOPL

400B 3E76
400C 3DF3
400E 210800
4012 7D
4013 D3F1
4015 7C
4016 D3F1
4018 3E25
401A D3F7
401C 210074
401F 010000
4022 DBFE
4024 7E
4025 FE03
4027 C22C40
402A 3E94
402C CD3B40
402F 78
4030 FE10
4032 C22440
4035 3E03
4037 CD3B40
403A 07
403B 05
403C 05
403D 05
403E DBF7
4040 E602
4042 CA3E40
4045 DBF6
4047 FE05
4049 C23E40
404C DBF7
404E E604
4050 CA4C40
4053 3E96
4055 D3F6
4057 DBF7
4059 E604
405B CA5740
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A056</td>
<td>F1</td>
<td>97</td>
<td>POP PSW</td>
</tr>
<tr>
<td>A05F</td>
<td>F5</td>
<td>98</td>
<td>CONT: PUSH PSW</td>
</tr>
<tr>
<td>A060</td>
<td>E00F</td>
<td>99</td>
<td>ANI 0FH LEAST SIGNIFICANT 4 BITS WILL BE SENT FIRST</td>
</tr>
<tr>
<td>A062</td>
<td>F630</td>
<td>100</td>
<td>ORI 30H</td>
</tr>
<tr>
<td>A064</td>
<td>D3F6</td>
<td>101</td>
<td>OUT TTYO</td>
</tr>
<tr>
<td>A066</td>
<td>CDBA40</td>
<td>102</td>
<td>CALL CHECK CHECK FOR EMPTY TRANSMIT BUFFER</td>
</tr>
<tr>
<td>A069</td>
<td>4F</td>
<td>103</td>
<td>MOV C,A</td>
</tr>
<tr>
<td>A06A</td>
<td>CD09F8</td>
<td>104</td>
<td>CALL CO</td>
</tr>
<tr>
<td>A06D</td>
<td>F1</td>
<td>105</td>
<td>POP PSW MOST SIGNIFICANT 4 BITS WILL BE SENT FIRST</td>
</tr>
<tr>
<td>A06E</td>
<td>E6F0</td>
<td>106</td>
<td>ANI 0FH</td>
</tr>
<tr>
<td>A070</td>
<td>B7</td>
<td>107</td>
<td>ORA A</td>
</tr>
<tr>
<td>A071</td>
<td>F9</td>
<td>108</td>
<td>RAR</td>
</tr>
<tr>
<td>A072</td>
<td>1F</td>
<td>109</td>
<td>RAR</td>
</tr>
<tr>
<td>A073</td>
<td>1F</td>
<td>110</td>
<td>RAR</td>
</tr>
<tr>
<td>A074</td>
<td>1F</td>
<td>111</td>
<td>RAR</td>
</tr>
<tr>
<td>A075</td>
<td>F630</td>
<td>112</td>
<td>OHI 30H</td>
</tr>
<tr>
<td>A077</td>
<td>D3F6</td>
<td>113</td>
<td>OUT TTYO</td>
</tr>
<tr>
<td>A079</td>
<td>CDBA40</td>
<td>114</td>
<td>CALL CHECK</td>
</tr>
<tr>
<td>A07C</td>
<td>4F</td>
<td>115</td>
<td>MOV C,A</td>
</tr>
<tr>
<td>A07D</td>
<td>CD09F8</td>
<td>116</td>
<td>CALL CO</td>
</tr>
<tr>
<td>A080</td>
<td>02E0</td>
<td>117</td>
<td>MJVI C 20H</td>
</tr>
<tr>
<td>A082</td>
<td>CD09F8</td>
<td>118</td>
<td>CALL CO</td>
</tr>
<tr>
<td>A085</td>
<td>E1</td>
<td>119</td>
<td>POP H</td>
</tr>
<tr>
<td>A086</td>
<td>23</td>
<td>120</td>
<td>INX H GET THE NEXT ADDRESS</td>
</tr>
<tr>
<td>A087</td>
<td>C1</td>
<td>121</td>
<td>POP B</td>
</tr>
<tr>
<td>A088</td>
<td>03</td>
<td>122</td>
<td>INX B INCREMENT THE COUNTER TO CHECK FOR THE END OF PROCESS</td>
</tr>
<tr>
<td>A089</td>
<td>C9</td>
<td>123</td>
<td>RET</td>
</tr>
<tr>
<td>A08A</td>
<td>F5</td>
<td>124</td>
<td>CHECK = PUSH PSW</td>
</tr>
<tr>
<td>A08B</td>
<td>DBFT</td>
<td>125</td>
<td>CHECK = IN TTY</td>
</tr>
<tr>
<td>A08D</td>
<td>E604</td>
<td>126</td>
<td>CALL TKBET</td>
</tr>
<tr>
<td>A08F</td>
<td>C8840</td>
<td>127</td>
<td>JZ CHECK</td>
</tr>
<tr>
<td>A092</td>
<td>F1</td>
<td>128</td>
<td>POP PSW</td>
</tr>
<tr>
<td>A093</td>
<td>C9</td>
<td>129</td>
<td>RET</td>
</tr>
<tr>
<td>4000</td>
<td></td>
<td>130</td>
<td>END START</td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

LOC  OBJ  LINE  SOURCE STATEMENT
000D  000A  1  PROGRAM FOR TRANSFERRING DATA FROM OTHER COMPUTER TO BLUE BOX.
000B  000A  2  CR EQU 0DH.
0003  000A  3  LF EQU 0AH.
000C  000A  4  CO EQU 0F809H.
000D  000A  5  EOF EQU 03H ; END OF FILE.
000C  000A  6  RESET EQU 01000000B ; RESET CHIP.
000F  000A  7  MODE1 EQU 11001110B ; MODE INSTRUCTION.
0010  000A  8  RXE EQU 00000100B ; RECEIVE, ENABLE.
0013  000A  9  T1OCT EQU 250 ; READER TIME OUT CONST.
0015  000A  10  TTYC EQU 0F7H ; OUTPUT CONTROL PORT.
0017  000A  11  TTXE EQU 0F6H ; INPUT DATA PORT.
0019  000A  12  TTYS EQU 0F7H ; INPUT STATUS PORT.
001B  000A  13  TTYO EQU 0F6H ; OUTPUT DATA PORT.
001D  000A  14  TXBE EQU 00000100B ; TRANSMIT BUFFER EMPTY.
0020  000A  15  CTS EQU 00100000B ; SIMULATE CLEAR TO SEND.
0023  000A  16  RTS EQU 00100000B ; SET REG TO SEND OUT.
0026  000A  17  RRDY EQU 00000100B ; RECEIVE BUFFER EMPTY.
0029  000A  18  ONEMS EQU 112 ; 1 MS TIME OUT CONST.
002C  000A  19  ORG 0400H.
002D  000A  20  START: LXI SP,0AFFH ; INITIALIZING STACK POINTER.
0030  000A  21  ; INITIALIZATION OF 8251 CHIP.
0033  000A  22  ; INITIALIZATION OF 8253 CHIP.
0036  000A  23  ; OUT TTYC ; OUTPUT THE RESET CODE.
0039  000A  24  ; MVI A,RESET ; MODE INSTRUCTION FOR 8251 CHIP.
003C  000A  25  ; OUT TTYC.
003F  000A  26  ; INITIALIZE 8253 CHIP.
0042  000A  27  ; MVI A,076H; LOAD 8253 COUNTER 2 FOR MODE.
0045  000A  28  ; AND LSB FOLLOWED BY MSB.
0048  000A  29  ; OUT 0F3H.
004A  000A  30  LXI H,008H; BAUDE CODE FOR 9600 RATE.
004C  000A  31  ; MOV A,L.
004E  000A  32  ; OUT 0F1H; LOAD LSB.
4015 7C 45 MOV A, H ; MSB OF BAUDE RATE.
4016 D3F1 46 OUT 0F1H

48

49 ; MAIN PROGRAM

50

51

52

53 MVI A, 02H
54 OUT TTYC ; ENABLE TRANSMIT, RECEIVE
55 ; AND REG TO SEND.
56 LXI H, 7400H ; DATA LOADING STARTS FROM HE
57 RE.
58

59 SEERDY: IN TTYS
60 ANI RRDY,
61 JZ SEERDY.
62 BEGN: IN TTYI
63 CPI 05H ; SEARCH FOR ENGO
64 JNZ SEERDY.
65 AGAIN: IN TTYS ; READ STATUS.
66 ANI TXBE ; CHECK FOR EMPTY
67 ; RECEIVE BUFFER.
68 JZ AGAIN.
69 OUT TTYO
70 GOLOOP: IN TTYS
71 ANI TXBE
72 JZ GOLOOP
73 LOOP: CALL READ.
74 MOV M, A
75 CPI EOF
76 JZ EXIT.
77 INX H
78 JMP FURO.
79 EXIT: RST 0

80

81 ; SUBROUTINE

82

83 READ: PUSH H
84 PUSH B
85 IN TTYI ; CLEAR RECEIVE BUFFER
86 ; BY INITIAL READING.
87 PROMPT: MVI A, 36H ; SEND PROMPT CHARACTER.
88 OUT TTYO.
89 RI: IN TTYS ; READ STATUS
90 ANI TXBE ; CHECK FOR EMPTY BUFFER.
91 JZ RI;
92 TMOUT: MVI B, RTOC ; LOAD TIME OUT CONCT.
93 CHKRDY: IN TTYS ; READ STATUS
94 ANI RRDY ; CHECK FOR READY RECEIVE
95 ; BUFFER.
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>4064</td>
<td>C271</td>
<td>4C</td>
<td>JNZ RI3</td>
</tr>
<tr>
<td>4067</td>
<td>CD8A</td>
<td>40</td>
<td>CALL DELAY</td>
</tr>
<tr>
<td>4068</td>
<td>05</td>
<td>98</td>
<td>DCR B</td>
</tr>
<tr>
<td>406B</td>
<td>C260</td>
<td>40</td>
<td>JNZ CHKRDY</td>
</tr>
<tr>
<td>406E</td>
<td>C35E</td>
<td>40</td>
<td>JMP TMOUT</td>
</tr>
<tr>
<td>4071</td>
<td>DBF6</td>
<td>6</td>
<td>RI3 FN TTYI ; READ DATA</td>
</tr>
<tr>
<td>4073</td>
<td>F5</td>
<td>2</td>
<td>PUSH PSW</td>
</tr>
<tr>
<td>4074</td>
<td>E60F</td>
<td>3</td>
<td>ANI 0FH</td>
</tr>
<tr>
<td>4076</td>
<td>F630</td>
<td>4</td>
<td>ORI 30H</td>
</tr>
<tr>
<td>4078</td>
<td>4F</td>
<td>5</td>
<td>MOV C,A</td>
</tr>
<tr>
<td>4079</td>
<td>CD09</td>
<td>F8</td>
<td>CALL CO</td>
</tr>
<tr>
<td>407C</td>
<td>B00D</td>
<td>6</td>
<td>MVI C,CR</td>
</tr>
<tr>
<td>407E</td>
<td>CD09</td>
<td>F8</td>
<td>CALL CO</td>
</tr>
<tr>
<td>4081</td>
<td>0E0A</td>
<td>8</td>
<td>MVI C,LF</td>
</tr>
<tr>
<td>4083</td>
<td>CD09</td>
<td>F8</td>
<td>CALL CO</td>
</tr>
<tr>
<td>4086</td>
<td>F1</td>
<td>101</td>
<td>POP PSW</td>
</tr>
<tr>
<td>4087</td>
<td>C1</td>
<td>102</td>
<td>RI3B1 : POP B</td>
</tr>
<tr>
<td>4088</td>
<td>E1</td>
<td>103</td>
<td>POP H</td>
</tr>
<tr>
<td>4089</td>
<td>C9</td>
<td>104</td>
<td>RET</td>
</tr>
<tr>
<td>408A</td>
<td>0E70</td>
<td>105</td>
<td>DELAY : MVI C, ONEMS</td>
</tr>
<tr>
<td>408C</td>
<td>0D</td>
<td>106</td>
<td>DLY1 : DCR C</td>
</tr>
<tr>
<td>409D</td>
<td>C28C</td>
<td>40</td>
<td>JNZ DLY1</td>
</tr>
<tr>
<td>4099</td>
<td>C9</td>
<td>108</td>
<td>RET</td>
</tr>
<tr>
<td>4099</td>
<td>C9</td>
<td>109</td>
<td>END START</td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

AGAIN A 402F BEGN A 4028 CHKRDY A 4060 CO A FR09
CR A 000D CTS A 0020 DELAY A 408A DLYI A 408C
EDF A 0003 EXIT A 404E GOLoop A 403A LF A 000A
LOOP A 4041 MODE1 A 00CE ONEMS A 0070 PRMPT A 4053
READ A 404F RESET A 0040 RI0 A 4057 RI3 A 4071
RI3B A 4087 RTOCT A 0002 RTS A 0020
RXE A 0004 SEERDY A 4021 START A 4000 TMOUT A 405E
TTYC A 00F7 TTYL A 00F6 TTYO A 00F6 TTYS A 00F7
TXXE A 0004

ASSEMBLY COMPLETE, NO ERRORS
REFERENCES


VITA AUCTORIS

Mahbub Iftekhar Ahmed

1958 Born on June 13th in Dacca, Bangladesh.

1974 Completed Secondary School Education at West End High School, Dacca, Bangladesh.

1976 Completed Higher Secondary Education at Dacca College, Dacca, Bangladesh.

1981 Graduated from the University of Engineering and Technology, Dacca, Bangladesh.

1984 Candidate for the degree of Master of Applied Science in Electrical Engineering at the University of Windsor, Windsor, Ontario, Canada.