1984

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 519-253-3000ext. 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

—

THÈSES CANADIENNES

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.
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 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                                                                                                     page

I. INTRODUCTION ........................................... 1
    Digital Image Processing System: ......................... 2
    Microcomputers For Image Processing: .................... 5
    Class Of Image Processing Algorithms Considered.
        For Implementations: ................................ 7
        Histogram Modification: ............................. 7
        Image Thresholding: ................................ 10
        Border Following: .................................. 10
        Template Matching: ................................ 11
        Thesis Organisation: ................................ 11

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

III. IMPROVEMENTS AND MODIFICATIONS OF THE EXISTING
    ALGORITHMS ........................................... 38
        Histogram Modification By Direct Specification:    38
        Automatic Threshold Value Selection: ................ 40
        Template Matching: ................................ 43

- iv -
IV. MICROCOMPUTER IMPLEMENTATIONS OF IMAGE PROCESSING ALGORITHMS

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

Summary and conclusions

Appendix

A. REFERENCES VITA AUCTORIS

page

93 96 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 requires two items:

1. an on line 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 cases, is too large to fit completely 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 uses 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 system, 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. Algor-
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

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

$$k=0, 1, 2, \ldots, L-1$$

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

$$s = T(r).$$

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:

\[
s = T(r) = \int_0^r P(r) \, dw \quad 0 \leq r \leq 1 \quad \text{Eq. (2.1.e)}
\]

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:

\[
ds/dr = P(r) \quad \text{by equation (2.1.e)}
\]

Now

\[
P_s(s) = \left[ P(r) \int_C \frac{dr}{ds} \right] \bigg|_{r = T^{-1}(s)}
\]

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

\[
= 1 \quad 0 \leq s \leq 1 \quad \text{Eq. (2.1.f)}
\]

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 have 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

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

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

2.1.2.2 Histogram Modification By Direct Specification:

Although the histogram equalisation 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

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

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

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

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

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

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

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

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

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

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.
Figure (2.3.a - 2.3.e): Sample templates
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 $s = T(r) = v = G(z)$. 

\[ z \]
Since we are dealing with discrete values, a perfect match between s and v cannot 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:

\[ ROB = \frac{1}{4} \max( |E-C|, |B-F| ) \]

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:

\[ \text{OPRTR1} = \frac{1}{16} \left( |E-F| + |E-I| \right) \]

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:

\[ X_{i+1} - X_{i-1} > 0, \text{ scan from right to left}; \]
\[ X_{i+1} - X_{i-1} < 0, \text{ scan from left to right}; \]
\[ X_{i+1} - X_{i-1} = 0, \text{ follow the following conditions:} \]

(i) \[ X_i - X_{i-1} = 0, \text{ skip scanning}; \]
(ii) \[ X_i - X_{i-1} \text{ and } Y_i - Y_{i-1} \text{ are of same sign, scan from left to right}; \]
(iii) \[ X_i - X_{i-1} \text{ and } Y_i - Y_{i-1} \text{ are of opposite sign, skip scanning}. \]

(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.

- 49 -
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 has 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 \( P_k = \sum_{n=0}^{k} N_n \). 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 normalization 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 \( S_k = T(r_k) = \sum_{j=0}^{R-1} P_r(r_j) \).

2. Specifying the desired histogram and equalising the specified histogram by the use of the transfer function \( V_k = G(z_k) = \sum_{j=0}^{\infty} P_z(z_j) \).

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 \( z_k \) 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). Refering 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 (2.2c), 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

$$1/4 \text{ Max}[|E-C|,|E-F|]$$

EQ-[4.5-a]

The above operator refers to figure (2.2c).

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

$$\text{OP\&TR1} = 1/16 \left[ |E-F| + |E-I| \right]$$

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>ROB</th>
<th>OPRTR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK1</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>GAUGE</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>GEAR</td>
<td>112</td>
<td>104</td>
</tr>
<tr>
<td>BLOCK2</td>
<td>40</td>
<td>33</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 step wise 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 \(2, 3, \ldots, 8\). 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 step 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.
### Coordinates of Eight Neighbours

<table>
<thead>
<tr>
<th>Neighbour Number = J</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)

\[
K1 = JD - J1 \quad K2 = ID - I1
\]

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

<table>
<thead>
<tr>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>.2</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

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 has 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 the 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 \( \text{TOPREP} = X(I,J) \). 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 \( X(I,J) = \text{TOPREP} \), 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) $X(I,J)$ 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 $(X_i, Y_i)$ 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 algorithm 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 function</td>
<td>0.48</td>
<td>Piston Head</td>
</tr>
<tr>
<td>Histogram Equalisation</td>
<td>0.71</td>
<td>Dollar</td>
</tr>
<tr>
<td>Direct Histogram Modification Technique</td>
<td>0.95</td>
<td>Piston Head</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 level) Piston Head</td>
</tr>
<tr>
<td>Border Following</td>
<td>1.75</td>
<td>Line in 63rd row having length of 5 pixels Same as above</td>
</tr>
<tr>
<td>Template Matching</td>
<td>12.0</td>
<td></td>
</tr>
<tr>
<td>Direct Line Detection</td>
<td>0.20</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

MICROCOMPUTER PROGRAM LISTINGS.
*FILE NAME: HSGMN*

THIS PROGRAM CALCULATES THE GREY LEVEL DENSITY OF PIXELS OF AN IMAGE.

START: LXI SR 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 LEVELS, THE MAXIMUM VALUE OBTAINED BY THE COUNTER WILL BE 1200H.

; INITIALIZING THE COUNTER LOCATIONS FOR DIFFERENT GREY LEVELS TO ZERO BEFORE STARTING COUNTING.

LXI H 8400H ; STARTING LOCATION FOR GREY LEVEL COUNTERS.

INTLZN; MOV A, 00H
INX H
INX B
MOV A, B
CPI 02H
JNZ INTLZN

LXI D 8400H ; 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 ; SINCE TWO LOCATIONS ARE ALLOTTED FOR EACH GREY LEVEL, IT IS BEING DOUBLED
RAL
PUSH D
LXI D, TEMP
STAX D
MVI A, 00H
RAL
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE</th>
<th>STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>4028</td>
<td>13</td>
<td>54</td>
<td>INX D</td>
<td><strong>; GETTING THE REFERENCE ADDRESS.</strong></td>
</tr>
<tr>
<td>4029</td>
<td>12</td>
<td>55</td>
<td>STAX D</td>
<td><strong>; SAVING THE CURRENT PIXEL ADDRESS.</strong></td>
</tr>
<tr>
<td>402A</td>
<td>D1</td>
<td>56</td>
<td>POP D</td>
<td><strong>; ADDING WITH REFERENCE.</strong></td>
</tr>
<tr>
<td>402B</td>
<td>E5</td>
<td>58</td>
<td>PUSH H</td>
<td><strong>; CORRECT ADDRESS IN &lt;DE&gt;RE</strong></td>
</tr>
<tr>
<td>402C</td>
<td>21FFB0</td>
<td>60</td>
<td>LXI H, TEMP</td>
<td><strong>; INCREMENTING THE COUNTER VALUE.</strong></td>
</tr>
<tr>
<td>402E</td>
<td>AF</td>
<td>61</td>
<td>XRA A</td>
<td><strong>; SAVE COUNTER.</strong></td>
</tr>
<tr>
<td>4030</td>
<td>7E</td>
<td>62</td>
<td>MOV A, M</td>
<td><strong>; ADDRESS OF LSB IN &lt;HL&gt;REG</strong></td>
</tr>
<tr>
<td>4031</td>
<td>03</td>
<td>63</td>
<td>ADD E</td>
<td><strong>; COUNTER INCREMENTED.</strong></td>
</tr>
<tr>
<td>4032</td>
<td>5F</td>
<td>64</td>
<td>MOV E, A</td>
<td></td>
</tr>
<tr>
<td>4033</td>
<td>23</td>
<td>65</td>
<td>INX H</td>
<td></td>
</tr>
<tr>
<td>4034</td>
<td>7E</td>
<td>66</td>
<td>MOV A, M</td>
<td></td>
</tr>
<tr>
<td>4035</td>
<td>8A</td>
<td>67</td>
<td>ADC D</td>
<td></td>
</tr>
<tr>
<td>4036</td>
<td>57</td>
<td>68</td>
<td>MOV D, A</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4037</td>
<td>C5</td>
<td>70</td>
<td>PUSH B</td>
<td></td>
</tr>
<tr>
<td>4038</td>
<td>EB</td>
<td>71</td>
<td>XCHG</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td></td>
<td></td>
<td><strong>; ADDRESS OF LSB IN &lt;HL&gt;REG</strong></td>
<td></td>
</tr>
<tr>
<td>4039</td>
<td>4E</td>
<td>73</td>
<td>MOV C, M</td>
<td></td>
</tr>
<tr>
<td>403A</td>
<td>23</td>
<td>74</td>
<td>INX H</td>
<td></td>
</tr>
<tr>
<td>403B</td>
<td>46</td>
<td>75</td>
<td>MOV B, M</td>
<td></td>
</tr>
<tr>
<td>403C</td>
<td>03</td>
<td>76</td>
<td>INX B</td>
<td></td>
</tr>
<tr>
<td>403D</td>
<td>70</td>
<td>77</td>
<td>MOV M, B</td>
<td></td>
</tr>
<tr>
<td>403E</td>
<td>2B</td>
<td>78</td>
<td>DCH K</td>
<td></td>
</tr>
<tr>
<td>403F</td>
<td>71</td>
<td>79</td>
<td>MOV M, C</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4040</td>
<td>C1</td>
<td>81</td>
<td>POP B</td>
<td><strong>; GETTING BACK THE PIXEL COUNTER.</strong></td>
</tr>
<tr>
<td>4041</td>
<td>E1</td>
<td>82</td>
<td>POP H</td>
<td><strong>; GETTING BACK THE CURRENT PIXEL LOCATION.</strong></td>
</tr>
<tr>
<td>83</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4042</td>
<td>03</td>
<td>85</td>
<td>INX B</td>
<td><strong>; CHECKING WHETHER ALL THE PIXELS HAVE BEEN COVERED.</strong></td>
</tr>
<tr>
<td>4043</td>
<td>78</td>
<td>86</td>
<td>MOV A, B</td>
<td><strong>; IF SO, END OF EXECUTION.</strong></td>
</tr>
<tr>
<td>4044</td>
<td>FE10</td>
<td>87</td>
<td>CPI 10H</td>
<td><strong>; IF NOT, REPEAT THE WHOLE PROCESS FOR NEXT PIXEL.</strong></td>
</tr>
<tr>
<td>88</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4046</td>
<td>E950A0</td>
<td>90</td>
<td>JZ EXT</td>
<td></td>
</tr>
<tr>
<td>4049</td>
<td>23</td>
<td>91</td>
<td>INX H</td>
<td></td>
</tr>
<tr>
<td>404A</td>
<td>110084</td>
<td>92</td>
<td>-LXI D, 8400H</td>
<td></td>
</tr>
<tr>
<td>404D</td>
<td>C1DA0</td>
<td>93</td>
<td>JMP LOOP</td>
<td></td>
</tr>
<tr>
<td>4050</td>
<td>C7</td>
<td>94</td>
<td>EXT: KST 0</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td></td>
<td></td>
<td><strong>END</strong>: START A &amp;ENDA</td>
<td></td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

EXT A 4950 INTLZNA 4909 LOOP A 49ID START A 4790
TEMP A 495F
SOURCE STATEMENT

FILE NAME: HSTEU.M

# 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 LEVELS. 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.

INTLZN: MOV A, 00H
INX H
INX B
MOV A, B
CPI 02H
JNZ INTLZN

LXI D, 7000H ; SETTING COUNTER TO
LXI B, 00H

D

# BY THIS OUTER WILL BE
# 1000H.
LXI H, 7400H ; STARTING ADDRESS OF
# )PIXELS.

# GENERATING CORRECT ADDRESS FOR THE
# COUNTER LOCATIONS.
LOOP: XRA A

MOV A, M
RAL
RAL
PUSH D
# LEVEL, IT IS BEING DOUBLED
LXI D, TEMP
# BEFORE ADDING WITH THE
STAX D
# REFERENCE ADDRESS.
MVX A, 00H
RAL
LOC OBJ  LINE  SOURCE STATEMENT

4023 13  54  INX D
4022 12  55  STAX D
402A D1  56  POP"D
402B E5  58  PUSH H
402C 2'FFB0  60  LXI 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,W
4035 8A  67  ADI D
4036 57  68  MOV D,A

;CORRECT ADDRESS IN <DE>REG

4037 C5  71  ;INCREMENTING THE COUNTER VALUE.
4038 EB  72  ;PUSH B
4039 4E  73  ;SAVE COUNTER.
403A 23  74  ;ADDRESS OF LSB IN <HL>REG
403B 46  75  ;COUNTER INCREMENTED.
403C 03  76  ;PIXEL LOCATION.
403D 70  77  ;GETTING BACK THE
403E 2B  78  ;PIXEL COUNTER.
403F 71  79  ;GETTING BACK THE CURRENT

4040 C1  82  ;PIXEL LOCATION.
4041 E1  83  ;THE PIXELS HAVE BEEN
4042 03  84  ;COVERED.
4043 78  85  ;CHECKING WHETHER ALL
4044 FE10  86  ;THE PIXELS HAVE BEEN
4046 CA50A0  87  ;INCLUDED.
4049 23  88  ;GENERATION OF CUMULATIVE
404A 110070  89  ;DISTRIBUTION FUNCTION.
404D C31D40  90  ;FIRST LOCATION OF
4050 110070  91  ;DIFFERENT GREY LEVEL COUN
4053 210070  92  ;TEKS.
4056 23  93  ;POINTING TOWARDS THE NEXT
4057 23  94  ;GREY LEVEL.
4058 010000  95  ;INITIALIZING THE CONTVERS.
LOC OBJ LINE SOURCE STATEMENT

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

108 : CONT: XRA A
109 : LDAX D
110 : ADD M
111 : MOV M A
112 : INX D
113 : INX H
114 : LDAX D
115 : ADC M
116 : MOV M A
117 : INX B
118 : CHECKING WHETHER ALL THE
119 : ITEMS HAVE BEEN COVERED.
120 : MOV A;B
121 : CPI $01H'
122 : JZ ROTO'
123 : INX H
124 : INX D
125 : JMP CONT
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 : ROTO: LXI H, $7000H
131 : LXI D, $00H
132 : INITIALIZING THE COUNTER
133 : TO CHECK WHETHER ALL THE
134 : ITEMS HAVE BEEN COVERED.
135 : INITIALIZING THE COUNTER
136 : COUNT THE ROTATION.
137 : PUSH H
138 : INX H
139 : CONT2: PUSH H
140 : INX H
141 : XRA A
142 : MOV A, M
143 : MOV A, M
144 : MOV M A
145 : MOV M A
146 : MOV A, B
147 : CPI $04H
148 : BEEN
149 : JZ CHKEND
150 : LE THE
LOC OBJ LINE SOURCE STATEMENT

4088 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 60 156 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

162 ; SINCEx AFTER ROTATION, THE HIGHEST
163 GREY LEVEL IS 256, EACH GREY LEVEL
164 WILL BE REDUCED BY ONE AND THE
165 TRANSFORMED GREY LEVELS WILL BE STORED
IN
166 A NEW LOCATION.
167 ; A NEW STORAGE LOCATION FOR
168 TRNSFN: LXI, B, 90000H
169 NEW STORAGE LOCATION FOR
170 THE TRANSFORMED GREY LE
171 VELS.
409A 010000 171 LXI D, 00H
409B D5 172 PUSH D
409C 210070 173 LXI H, 7000H
409D 5E 174 CONT4: MOV E, M
409E 23 175 INX H
409F 56 176 MOV-D, M
40A0 1B 177 DEX D
178 ; DECREMENTING THE GREY L
179 EVEL
40A1 7A 179 MOV A, D
40A2 FEFF 180 CPI 0FFH
40A3 C28140 181 JNZ TRNSFR
HANGED.
40A4 110000 182 LXI D, 00H
40A5 78 183 TRNSFR: MOV A, E
40A6 02 184 STAX B
40A7 D1 185 POP D
40A8 13 186 INX D
40A9 7A 187 MOV A, D
40A0 FE01 188 CPI 01H
40AB C4C140 189 JZ LSTSTRP E
190 ; CHECKING WHETHER ALL TH
191 ITEMS HAVE BEEN COVERED
40BB 23 191 INX H
40BC 03 192 INX B
40BD D5 193 "PUSH D
40BE C3A440 194 JMP CONT4
<table>
<thead>
<tr>
<th>LOC OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>195</td>
<td></td>
<td>; SUBSTITUTE BACK THE PIXEL VALUES.</td>
</tr>
<tr>
<td>196</td>
<td></td>
<td>; INITIALIZE COUNTER.</td>
</tr>
<tr>
<td>197</td>
<td></td>
<td>; STARTING LOCATION OF PIXELS.</td>
</tr>
<tr>
<td>198</td>
<td></td>
<td>LSTSTP; LXI B,01H</td>
</tr>
<tr>
<td>199</td>
<td></td>
<td>LXI H, 7400H</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td>LP; LXI D, 9000H</td>
</tr>
<tr>
<td>201</td>
<td></td>
<td>; NEW LOCATION FOR TRANSFORMED GREY LEVELS.</td>
</tr>
<tr>
<td>202</td>
<td></td>
<td>XRA A</td>
</tr>
<tr>
<td>203</td>
<td></td>
<td>MOV A, M</td>
</tr>
<tr>
<td>204</td>
<td></td>
<td>; ADDING THE PIXEL VALUE</td>
</tr>
<tr>
<td>205</td>
<td></td>
<td>RRECT ADDRESS.</td>
</tr>
<tr>
<td>206</td>
<td></td>
<td>ADD E</td>
</tr>
<tr>
<td>207</td>
<td></td>
<td>MOV E, A</td>
</tr>
<tr>
<td>208</td>
<td></td>
<td>MOV A, D</td>
</tr>
<tr>
<td>209</td>
<td></td>
<td>ACI 00H</td>
</tr>
<tr>
<td>210</td>
<td></td>
<td>MOV D, A</td>
</tr>
<tr>
<td>211</td>
<td></td>
<td>LDX D</td>
</tr>
<tr>
<td>212</td>
<td></td>
<td>MOV M, A</td>
</tr>
<tr>
<td>213</td>
<td></td>
<td>MOV A, B</td>
</tr>
<tr>
<td>214</td>
<td></td>
<td>CPI 00H</td>
</tr>
<tr>
<td>215</td>
<td></td>
<td>IXELS</td>
</tr>
<tr>
<td>216</td>
<td></td>
<td>JZ EXT</td>
</tr>
<tr>
<td>217</td>
<td></td>
<td>INX B</td>
</tr>
<tr>
<td>218</td>
<td></td>
<td>INX H</td>
</tr>
<tr>
<td>219</td>
<td></td>
<td>JMP LP</td>
</tr>
<tr>
<td>220</td>
<td></td>
<td>EXT; RST 0</td>
</tr>
<tr>
<td>221</td>
<td></td>
<td>END START</td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

<table>
<thead>
<tr>
<th>CHKEND</th>
<th>A 408D</th>
<th>CONT</th>
<th>A 405B</th>
<th>CONT2</th>
<th>A 4078</th>
<th>CONT3</th>
<th>A 4076</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT4</td>
<td>A 40A4</td>
<td>EXT</td>
<td>A 40DF</td>
<td>INTLZN</td>
<td>A 4009</td>
<td>LOOP</td>
<td>A 4010</td>
</tr>
<tr>
<td>LP</td>
<td>A 40C7</td>
<td>LSTSTP</td>
<td>A 40C1</td>
<td>ROTN</td>
<td>4070</td>
<td>SCNSIP</td>
<td>A 4050</td>
</tr>
<tr>
<td>START</td>
<td>A 4000</td>
<td>TEMP</td>
<td>A B0FF</td>
<td>TRNSFN</td>
<td>A 409A</td>
<td>TRNSFR</td>
<td>A 40B1</td>
</tr>
</tbody>
</table>

ASSEMBLY COMPLETE, NO ERRORS
FILE NAME: HSTMOD

THIS PROGRAM MODIFIES THE HISTOGRAM OF AN IMAGE ACCORDING TO THE GIVEN SPECs.

START: LXI, SP, 0AFFFH

LXI B, 00H

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 259FH.

INITIALIZING THE COUNTER LOCATIONS FOR DIFFERENT GREY LEVELS TO ZERO BEFORE STARTING COUNTING.

LXI H, 7000H

STARTING LOCATION FOR GREY LEVEL COUNTERS.

INTLZN; MVI A, 00H

MOV M, A

INX H

INX B

MOV A, B

CHECK WHETHER ALL THE LOCATIONS HAVE BEEN COVERED.

CPI 02H

JNZ INTLZN

LXI D, 7000H

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

SINCE TWO LOCATIONS ARE ALLOTTED FOR EACH GREY LEVEL, IT IS BEING DOUBLed

MOV A, M

RAL

PUSH D

LXI D, TEMP

BEFORE ADDING WITH THE REFERENCE ADDRESS.

STAX D

MVI A, 00H

RAL

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  
402C 21FF60   58    ADDRESS.
402D AF       59    PUSH H
4030 7E       60    ; SAVING THE CURRENT PIXEL  
4031 63       61    ADDRESS.
4032 5F       62    LXI H, TEMP  
4033 23       63    ; ADDING WITH REFERENCE;
4034 7E       64    XRA A
4035 8A       65    MOV A, M
4036 57       66    ADD E
4037 C5       67    MOV E, A
4038 EB       68    INX H
4039 4E       69    MOV A, M
403A 23       70    XINC B
403B 46       71    ; ADDING ADDRESS OF LSB IN <HL>REG  
403C 03       72    MOV B, M
403D 70       73    ; COUNTER INCEREMENTED.
403E 2B       74    INX B
403F 71       75    MOV M, B
4040 C1       76    DEX H
4041 E1       77    MOV M, C
4042 03.      78    ; PIXEL LOCATION.  
4043 7B       79    POP B
4044 FE10     80    ; GETTING BACK THE  
4046 CA5040   81    PIXEL COUNTER.
4049 23       82    POP H
404A 110070    83    ; GETTING BACK THE CURRENT  
404D C31D40   84    PIXEL LOCATION.  
404F CB04     85    INX B
4050 110070    86    MOV A, B
4051 7B       87    CPI 10H
4052 0000     88    ; CHECKING WHETHER ALL  
4053 120000    89    THE PIXELS HAVE BEEN  
4056 CD548     90    COVERED.  
4059 210000    91    JZ SCNSTP  
405C CD1741   92    INX H
405E CB04     93    ; IF SO, END OF EXECUTION.  
4060 C31D40   94    ; IF NOT, REPEAT THE WHOLE  
4062 010000    95    PROCESS FOR NEXT PIXEL.  
4063 70000H   96    ; GENERATION OF CUMULATIVE  
4065 7E000H   97    DISTRIBUTION FUNCTION.  
4067 7E000H   98    SCNSTP: LXI D, 7000H  
4069 7E000H   99    LXI H, 7000H
406A 7E000H   100    ; FIRST LOCATION OF  
406C 7E000H   101    ; DIFFERENT GREY LEVEL COUN-
406E 7E000H   102    TERS.
4070 CDF89     103    CALL CMFUCN
4072 CD1741   104    LXI H, 7000H
4075 010000    105    CALL ROIN
4077 7E000H   106    LXI D, 7000H
4079 7E000H   107    LXI H, 7000H
407A 7E000H   108    LXI 5, 7000H
407C 7E000H   109    LXI 4, 7000H
407E 7E000H   110    LXI 3, 7000H
4080 7E000H   111    LXI 2, 7000H
4082 7E000H   112    LXI 1, 7000H

LINE  SOURCE STATEMENT

106 CALL TRNSFN
107
108 I SPECIFIED HISTOGRAM IS TO BE
109 EQUALISED NOW.
110
111 LXI D78500H
112 LXI H8500H
113 CALL CHFUCN
114 LXI H8500H
115 CALL ROTN
116 LXI B8500H
117 LXI H8500H
118 CALL TRNSFN
119
120 J INVERSE TRANSFORMATION.
121
122
123 STINV: VVI A00H
124 STA COUNT
125 COUNT CHECKS WHETHER ALL THE LEVELS
126 HAVE BEEN COVERED:
127 LXI B7000H
128 LXI H8500H
129 VVI E00H
130 AGN: LDA X B
131 STA REF
132 LIP: SUB N
133 JZ SUBSTI JIF ZERO OR
134 JC SUBSTI JIF CARRY, THIS IS THE INV. TRA
135 NSFORM.
136 INX H
137 INR E
138 MOV A E
139 CPI 0FFH
140 JZ SUBALL
141 LDA X B
142 JMP LIP
143 SUBSTI: MOV A, E
144 STAX B
145 PUSH H
146 INX B
147 LDA COUNT
148 CPI 0FFH
149 JZ OUTI
150 INR A
151 STA COUNT
152 LDA REF
153 MOV D A
154 LDA X B
155 SUB D
156 JNZ NXTSTP
157 POR H
158 JMP SUBSTI
LOC OBJ | LINE | SOURCE STATEMENT
---|---|---
40C1 E1 | 159 | NEXTSTPT: POP H
40C2 C38D40 | 160 | JMP AGN
40C5 7B | 161 | SUBALL: MOV A. E
40C6 02 | 162 | STAK B
40C7 03 | 163 | INX B
40C8 3ADB40 | 164 | LDA COUNT
40CB FEFF | 165 | CPI 0FFH
40CD CD840 | 166 | JZ EXIT
40D0 3C | 167 | INR A
40D1 32DB40 | 168 | STA COUNT
40D4 C3C540 | 169 | JMP SUBALL
40D7 E1 | 170 | OUTI: POP H
40DB C3DD40 | 171 | EXIT: JMP LSTSTP
40DB | 172 | COUNT: DS I
40E4 | 173 | REF: DS I
40DD 010100 | 174 | ; SUBSTITUTE BACK THE PIXEL VALUES
40E0 210074 | 175 | ; LOCATION FOR TRANSFORMED
40E1 110070 | 176 | ; GRAY LEVELS.
40E6 AF | 177 | XRA A
40E7 7E | 178 | MOV A, M
40E9 83 | 179 | ADD E
40EA 5F | 180 | MOV E, A
40EB 7A | 181 | MOV A, D
40ED 57 | 182 | ADC 00H
40EE 1A | 183 | MOV D, A
40EF 77 | 184 | LDAX D
40F0 78 | 185 | ; GET THE TRANSFORMED GRAY L
40F1 FE10 | 186 | LS
40F3 CAFB40 | 187 | MOV M, A
40F6 03 | 188 | MOV A, B
40F7 23 | 189 | CPI-10H
40FB C7 | 190 | ; CHECK WHETHER ALL THE PIXE.
40FC 23 | 191 | ; HAVE BEEN COVERED.
40FD 23 | 192 | CMSFUCN: INX H
40FE 01960A | 193 | ; POINTING TOWARDS THE NEXT
200 | 194 | ; GREY LEVEL.
201 | 195 | INCE
202 | 196 | INX. H
203 | 197 | LXI B, 00H
204 | 198 | ; INITIALIZING THE CONT.
205 | 199 | S.
206 | 200 | THE TOTAL NUMBER OF GREY
LOC  OBJ  LINE  SOURCE STATEMENT

211  VALUE
212      OF THE COUNTER WILL BE 13

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

224 MOV A, B
225 CPI O1H
226 JZ EXIT
227 INX H
228 INC D
229 JMP CONT

230 EXIT: RET

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

233
234
235
236
237
238

239 ROTN: LXI D, 00H
240 R
241 ; INITIALIZING THE COUNTED ITEMS
242 ; TO CHECK WHETHER ALL THE ITEMS HAVE BEEN COVERED.

243 CONT3: MVI B, 01H
244 R TO
245 CONT2: PUSH H
246 INX H
247 ; MSB WILL BE ROTATED FIRST

248 XRA A
249 MOV A, M
250 DCX H
251 MOV A, M
252 RAR
253 MOV M, A
254 MOV A, B
255 CPI 04H
256 ; CHECK WHETHER DATA HAS BEEN ROTATED FOUR TIMES.

257 JZ CHKEND
258 ; IF YES, CHECK WHETHER ALL
L THE

258 ITEMs HAVE BEEN COVERED.

INR B

259 POP H

260 JMP CONT2 : IF NO, GO FOR NEXT ROTATION.

JMP CONT2

261

262 INX D

263 MOV A, D

264 CPI $01H

265 JZ EXT2

266 POP H

267 INX H

268 INX H

269 JMP CONT3

270 RET

271

272 SUBROUTINE FOR SEQUENTIALLY RESTORING

273 ********************************************

274 SINE AFTER ROTATION, THE HIGHEST

275 GREY LEVEL IS 256, EACH GREY LEVEL

276 WILL BE REDUCED BY ONE AND THE

277 TRANSFORMED GREY LEVELS WILL BE STORED

278 IN A NEW LOCATION.

279

280 TRNSF: LXI D, $00H

281 PUSH D

282

283 CONT4: MOV E, M

284 INX H

285 MOV D, M

286 DCX D

287

288 MOV A, D

289 CPI $0FH

290 JNZ TRNSF

291

292 LXI D, $00H

293 TRNSF: MOV A, E

294 STAX B

295 POP D

296 INX D

297 MOV A, D

298 CPI $01H

299 JZ EXT3

300

301 INX H

302 INX B

303 PUSH D
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>415E</td>
<td>C34441</td>
<td>304</td>
<td>JMP CONT4</td>
</tr>
<tr>
<td>4161</td>
<td>C9</td>
<td>325</td>
<td>RET</td>
</tr>
<tr>
<td>4000</td>
<td></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
- CONTA A 411A
- CONT4 A 4144
- COUNT A 40DE
- EXIT A 40DB
- EXT A 40FB
- EXT1 A 4116
- EXT2 A 413E
- EXT3 A 4161
- INTLN A 4009
- LOOP A 401D
- LIP A 4091
- LP A 40E3
- LSTLP A 40DD
- NXTLP A 40C1
- OUT A 40D7
- REF A 40DC
- RET A 4117
- SCNSTP A 4050
- START A 4000
- STINV A 4080
- SUBALL A 40C5
- SUBJ A 40A4
- TRANSF A 4140
- TRANSFR A 4151

**ASSEMBLY COMPLETE, NO ERRORS**
**FILE NAME: THRSLD.SRC**

**THRESHOLDING**

**MONITOR ROUTINE TO SCAN THE KEYBOARD**

**LAY ON THE CONSOLE.**

**LXI H, MESSAGE**

**ADDRESS TO STORE THE THRESHOLD VALUE.**

**MV B, 00H**

**COUNTER TO COUNT # OF B TIMES IN THRESHOLD VALUE.**

**GET THE MESSAGE FROM THE MEMORY LOCATION.**

**USED FOR COMPARING WITH THE TERMINATING CHARACTER.**

**ASCII CODE FOR THE TERMINATING CHARACTER.**

**FOR NEXT OPERATION.**

**DISPLAY THE MESSAGE ON THE CONSOLE.**

**INX H**

**GO FOR NEXT CHAR IN THE MESSAGE:**

**JMP LP1**

**REPEAT THE PROCESS.**

**CALL CI**

**VALUE FROM THE KEYBOARD.**

**MOV C,A**

**DISPLAY THE THRESHOLD VALUE ON THE CONSOLE.**

**CALL GO**

**GET THE ASCII CODE BACK INTO ACC FOR COMPARISON.**

**CHECK FOR CARRIAGE RETURN.**

**JZ THRESHOLDING**

**IF VALUE ENTERED START THRESHOLDING.**

**STAX D**

**STORE IN THE RESERVE AREA FOR THRESHOLD VALUE.**

**LDAX D**

**SCII CODE BY MASKING**

**ANI 0F0H**

**CPI 40H**

**JZ HEXA**

**ANI 0FH**

**JMP STK**
LOC  OBJ LINE          SOURCE STATEMENT
       4032 1A 36      HEXA: LDA X D
       4033 E60F 37     ANI 0FH
       4035 C609 38     ADI 09H
       4037 12 39     STR: STAX D
       4038 04 40     INR B
       4039 13 41    INX D

       403A C31640 42      JMP NEXT
       403D 0E0A 43     JHC R\H VALUE.
       403F CD00FH 44     THRESH: MVI C,0AH
       4042 78 45     CALL CO\I
       4043 FE01 46     MOV A, B
       4045 CA5540 47     CPI 01H
       4048 217640 48     IF SINGLE DIGIT, GO FOR T
       404B .7E 49     HRESHOLDING.
       404C 07 50     LXI H, STTH\SH
       404D 07 51     TACH THE TWO DIGITS.
       404E 07 52     MOV A,M
       404F 07 53     GET THE FIRST DIGIT.
       4050 23 54     NOW THE 4 LSB'S HAVE BEEN
                     EN TRANSFERED TO THE MSB LOC.
       4051 86 55     INX H
       4055 327640 56     MOVE POINTER TO NEXT DI
                     GIT.
       4055 010074 57     THIS ATTACHES THE SECON
                     D DIGIT TO THE FIRST.
       4055 41 58     STORE THIS THRESHOLD VALUE
                     AT THE SAME LOC.
       4058 217640 59     LUE(HEX) AT THE SAME LOC.
       405B 1640 60     LUE: LXI H, STTH\SH
                     ADDRESS OF STARTING LOC.
       405D 1E40 61     ADDRESS OF THRESHOLD VA
                     EG.
       4060 96 62     MVI D, 64
       4061 DA6940 63     ENTER # OF COLS INTO E
                     REG.
       4064 3EFF 64     LOAD # OF ROWS INTO D REG.
       4066 C36B40 65     GET THE PIXEL OF IMAGE
                     EQU 255.
       4069 3E00 66     REPLAC\E THE PIXEL IN THE
                     SAME LOC.
       406B 02 67     MVI A, 00H
       406C 03 68     REPLAC\E THE PIXEL BACK
                     FOR NEXT PIXEL.
       406D 1D 69     DEC\MEM C\0LMN C\0UNT\R
       406E C25F40 70     DECREMENT C\0LMN C\0UNT\R
       4071 15 71     e PROCESS IN THE SAME ROW.
LOC OBJ  LINE  SOURCE STATEMENT

4072 C25040   72  JNZ LOOP1 ; GO FOR THE NEXT ROW TO
              PROCESS
4075 C7     73  RST 0
4076       74  STIRSH; DS 2
4078 454E5445 75 MESSAG: DB 'ENTER THE THRESHOLD VALUE IN HEXADEC
              IMAL CODE :
407C 52205448
4080 45205448
4084 52455343
4088 4F4C4420
408C 56414C53
4090 4520494E
4094 20484558
4098 41444543
409C 494D414C
40A0 20434F44
40A4 45203A20
40A8 24     76 DB 'S'
40B0       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  MESSAG 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: 5E1D LINE: 1 SOURCE STATEMENT:

1: *************************************************************
2: * FILENAME:  MLTTR. SRC *
3: *************************************************************

4: THIS PROGRAM IS USED TO THRESHOLD A GIVEN IMAGE AT A GIVEN MULTIPLE
5: THRESHOLD VALUES (THRU THE KEY-BOARD.)
6: 
7: 
8: 4000
9: ORG. 4000H
10: CI EQU 0F803H ; MONITOR ROUTINE TO SCAN THE KEY-BOARD.
11: CO EQU 0F809H ; MONITOR ROUTINE TO DISPLAY LAY ON THE CONSOLE.
12: BEGIN: LXI SP, EAFFFH 
13: LXI H, MESAG1 ; TO DISPLAY A MESSAGE ON THE CONSOLE.
14: CALL ACCEPT
15: LXI H, NUM
16: MOV M, A
17: 
18: LXI B, THSLD.
19: LXI H, NUM
20: MOV D, M
21: CONT: PUSH B
22: PUSH D
23: LXI H, MESAGE
24: CALL ACCEPT
25: POP D
26: POP B
27: STAX B
28: DCR D
29: JJZ, PROCES
30: INX B
31: JMP CONT
32: PROCES: LXI B, 7400H
33: MVI D, 64
34: LOOP1: MVI E, 64
35: LOOP2: PUSH D
36: LXI H, NUM
37: MOV D, M
38: LXI H, THSLD.
39: LDAX B
40: SUB M
41: JNC NXT
42: MVI A, 00H
43: JMP STORE
44: NXT: DCR D
45: LXI H, HILVL
46: JZ HILVL
47: INX H
48: LDAX B
49: SUB M
50: JC ASNVAL
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>404A</td>
<td>C34040</td>
<td>51</td>
<td>JMP NXT</td>
</tr>
<tr>
<td>4040</td>
<td>7E</td>
<td>52</td>
<td>ASNV:MOV A,M</td>
</tr>
<tr>
<td>404E</td>
<td>2B</td>
<td>53</td>
<td>DCX H</td>
</tr>
<tr>
<td>404F</td>
<td>B7</td>
<td>54</td>
<td>ORA A</td>
</tr>
<tr>
<td>4050</td>
<td>86</td>
<td>55</td>
<td>ADD M</td>
</tr>
<tr>
<td>4051</td>
<td>1F</td>
<td>56</td>
<td>RAR</td>
</tr>
<tr>
<td>4052</td>
<td>C35740</td>
<td>57</td>
<td>JMP STORE</td>
</tr>
<tr>
<td>4055</td>
<td>3EFF</td>
<td>58</td>
<td>HILV:MYI A,OFFH</td>
</tr>
<tr>
<td>4057</td>
<td>D0</td>
<td>59</td>
<td>STORE:STAX B</td>
</tr>
<tr>
<td>4058</td>
<td>03</td>
<td>60</td>
<td>INX B</td>
</tr>
<tr>
<td>4059</td>
<td>D1</td>
<td>61</td>
<td>POP D</td>
</tr>
<tr>
<td>405A</td>
<td>1D</td>
<td>62</td>
<td>DCR E</td>
</tr>
<tr>
<td>405B</td>
<td>22E49</td>
<td>63</td>
<td>JNZ LOOP2</td>
</tr>
<tr>
<td>405E</td>
<td>15</td>
<td>64</td>
<td>DCR D</td>
</tr>
<tr>
<td>405F</td>
<td>C22C40</td>
<td>65</td>
<td>JNZ LOOP1</td>
</tr>
<tr>
<td>4062</td>
<td>C7</td>
<td>66</td>
<td>RST 0</td>
</tr>
</tbody>
</table>

67 MESSAGI:DB 'ENTER NUMBER OF THRESHOLD LEVEL IN H

EX CODE: $4

68 MESSAG:DB 'ENTER THRESHOLD VALUE (SMALLER ONE FIRST) IN HEX CODE: $5

69 SITRSH:DS 2

70 NUM: DS 1

71 THSLD: DS 0AH

72

73 SUBROUTINE FOR ACCEPTING THRESHOLD VALUE

74

40D8 110940 75 ACCEPT LXI D, SITRSH ;ADDRESS TO STORE THE THRESHOLD VALUE.

40D9 25A9 76 MVI B, 00H ;COUNTER TO COUNT # OF B
LOC OBJ TLINE SOURCE STATEMENT

40DB 4E 77 LP1: MOV C,M ; GET THE MESSAGE FROM TH
40DC 79 E MEMORY LOCATION.
40DD FE24 78 MOV A,C ; USED FOR COMPARING WITH
40DF CAE9 40 THE TERMINATING CHARACTER.
40DE INAT CHAR "S" : ASCII CODE FOR THE TERM
40DF 80 CPI 24H FOR NEXT OPERATION.
40E2 CD09F8 JZ NEXT ; IF MESSAGE COMPLETE GO
40E5 23 FOR NEXT OPERATION.
40E6 CD0BH0 CALL CO ; DISPLAY THE MESSAGE ON
40E9 CD03F8 THE CONSOLE.
40EC 4F INX H ; GO FOR NEXT CHAR IN THE
40ED CD09F8 MESSAGE.
40F0 79 JMP LP1 ; REPEAT THE PROCESS.
40F1 FE0D NEXT: CALL CI ; READ THE THRESHOLD
40F3 CA1041 VALUE FROM THE KEY BOARD. /
40F6 12 MOV C,A ; DISPLAY THE THRESHOLD V
40F7 F60F ALUE ON THE CONSOLE.
40FA FE40 CALL CO 86 GET THE ASCII CODE BACK
40FC CA0541 INTO ACC FOR COMPARISON.
40FF 1A CPI 0DH 87 mn( CR)
4100 E60F JZ THRESH 88 CHECK FOR CARRIAGE RETU
4102 C30A41 RD THRESHOLING;
4105 1A STAX D 89 STORE IN THE RESERVE AR
4106 E60F EA FOR THRESHOLD VALUE.
4109 C609 LDAX D 90 ; GET ACTUAL VALUE FROM A
410A 12 ASCII CODE BY MASKING
410B 04 ANI 0F0H 91 INCR COUNTER.
410C 13 CPI 4BH ; MOVE POINTER TO NEXT LO
410D C3E940 JMP NEXT ; RECEIVE NEXT DIGIT OF T
4110 0E0A THRESHOLING.
4112 CD09F8 THRESH: MVI C100AH
4115 7B ; CHECK VALUE IN COUNTER.
4116 FE01 4 MOV A,B ; CHECK FOR SINGLE DIGIT.
4118 C22241 ; IF SINGLE DIGIT GO FOR
411A 21C940 THRESHOLING.
411C 7E LXI H,STRSH
411E 0F MOV A,M
4120 00 JMP EXT
4122 21C940 DOUBLE: LXI H,STRSH ; IF DOUBLE DIGIT THEN AT
TACH THE TWO DIGITS.
LOC  OBJ  LINE  SOURCE STATEMENT
4125 7E   114  MOV A,M  GET THE FIRST DIGIT.
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.
412B 66   120  INX H
412C 09   121  GIT
4000     122  ADD M

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS
ACCEPT A 40D6  ASNV A 404D  BEGIN A 4000  CI A F823
C0 A F809  CONT A 4014  DOUBLE A 4122  EXT A 412C
HEXA A 4105  HILVL A 4055  LOOP1 A 402C  LOOP2 A 40D2
LP1 A 40DB  MSAG1 A 4063  MESSAGE A 4091  NEXT A 40E9
LM A 40CB  NXT A 4048  PROCES A 4027  STORE A 4057
STR A 410A  STTRSH A 40C9  THRESH A 4110  THSLL A 406C

ASSEMBLY COMPLETE, NO ERRORS
LOC OBJ LINE SOURCE STATEMENT

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

CSEG

PUBLIC PICTRE,BREG,CREG
EXTERN NABOR,ADRGEN

START: MVI B,02H ;INITIAL ROW FOR BORDER FOLLOWING ALGORITHM.
LP1: MVI C,01H ;INITIAL COL OF BORDER FOLLOWING ALGORITHM.
SRCH: MOV A,B ;TO SAVE THE PREVIOUS ROW INFORMATION.
STA BREG ;SAVE THE ROW INFORMATION.
N LOC BREG ;TO SAVE COL INFORMATION.
MOV A,C ;SAVE THE COL INFORMATION.
STA CREG ;
N LOC CREG ;LOAD STARTING ADDRESS OF THE PICTURE IN HL REG.
LHLD PICTRE ;CALL ADRGEN ERA TE PROPER ADDRESS.
NEXT . MOV D,M ;GET THE PIXEL VALUE INTO D REG.
MOV A,C ;SAVE THE PREVIOUS COL VALUE
AT LOC CREG ;
STA CREG ;GO FOR NEXT PIXEL THAT IS
INR C ;CHECK FOR END OF THAT ROW.
NEXT COL.
MOV A,C ;IF PRESENT ROW IS OVER GO
CPI 40H ;TO NEXT ROW.
JZ NXTROW ;AGAIN GET THE ADDRESS INTO HL REGS.
LHLD PICTRE ;CALL ADRGEN XT PIXEL.
NEXT . MOV A,D ;GET THE FIRST PIXEL INTO A
CCUMULATOR.
CMP M ;COMPARE WITH THE NEXT ADJACENT PIXEL.
CENT PIXEL.
JZ NEXT ;IF BOTH PIXELS ARE SAME THEN
ENTRY. THE NEXT PAIR.
CHK ;MOV A,M.
ANITION.
CPI 0FFH ;CHECKING FOR 0-1 OR 1-0 TRANSITION.
JNZ SCNCHK
MOV A,D
CPI 00H
JZ STRTAG
JMP NEXT
SCNCHK: MOV A,M.
LOC OBJ LINE SOURCE STATEMENT

0000 FE00 0039 CPI 00H
000B C2100 003B JNZ NEXT
003E 7A 003F MOV A, D
003F FEFF 0041 CPI 0FFH
0041 CA5100 0044 JZ STRTBG
0044 C31200 0047 JMP NEXT
0047 04 0048 NXTROW: INR B ; TRY THE NEXT ROW OF PIXELS

0048 78 0049 MOV A, B 
0049 FE3F 004A CPI 3FH ; CHECK FOR END OF PICTURE.
004A CA5900 004B JZ OVER ; IF END OF PICTURE THEN STO

004E C30200 0051 JMP LP1 ; IF NOT END OF PICTURE THEN

0051 C5 0052 STRTBG: PUSH B REPEAT.
0052 CD0000 0053 CALL NABOR ; CALL THE SUBROUTINE TO DO

0055 C1 0056 THE BORDER FOLLOWING.
0056 C30400 0057 JMP SRCH

0059 C7 005A DVER: RST 0
005A 005B 005C BREG: DS 1
005B 005D CREG: DS 1
005C 0074 005E PICTRE: DW 7400H

005F 57 END

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

EXTERNAL SYMBOLS
ADRGEN E 0000 NABOR E 0000

USER SYMBOLS
ADRGEN E 0000 BREG C 005A CHK? = C 0029 CREG C 005B
LP1 C 0002 NABOR E 0000 NEXT C 0012 NXTROW C 0047
OVER C 0059 PICTRE C 005C SONCHK C 0038 SRCH C 0004
START C 0000 STRTBG C 0051

ASSEMBLY COMPLETEA, NO ERRORS
LOC OBJ LINE SOURCE STATEMENT
1 **********
2 * FILENAME : NABOR.SRC *
3 **********
4 
5 CSEG
6 
7 BOUNRY MACRO
8 LHLD.PICTRE ; STARTING ADDRESS OF TH
9 E PICTURE.
10 CALL ADREGN ; CALL SUBROUTINE TO GEN
11 ERATE ADDRESS REQUIRED.
12 ; LDA REFVAL LOAD THE REFERENCE VAL-
13 UE INTO ACC.
14 CMP M ; COMPARE I T WITH THE LA
15 TEST PIXEL OBTAINED. ; IF NOT EQUAL IT INDICA
16 ; TES A BOUNDARY POINT. SO
17 ; GO FOR THE NEXT STEP.
18 
19 ENDM
20 
21 ; XPLUS MACRO REGSTR
22 MOV A, B ; LOAD THE ACC WITH THE
23 CURRENT HOW INFORMN.
24 ; STA BREG ; ALSO SAVE THIS ROW INF
25 RMN FOR LATER USE.
26 ; ADD REGSTR ; ADD THE CONTENTS OF RE
27 REGISTER SPECIFIED.
28 ; MOV B, A ; REPLACE THE ROW INFRMN
29 30 YPLUS MACRO REGSTK
31 ; MOV A, C ; LOAD ACC. WITH THE CURR
32 ENTI COL INF
33 ; STA CREG ; ALSO SAVE THIS VALUE F
34 OR LATER USE.
35 ; ADD REGSTR ; ADD THE CONTENTS OF RE
36 G SPECIFIED.
37 ; MOV C, A ; REPLACE THE COLMN INF
38 MN. BACK INTO C REG.
39 ENDM
40 
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 NADOR
45 EXTRN ADRGEN, BREG, CREG, PICTRE
47 NADOR.
48 ; MOV A, D
49 ; SAVE THE PREVIOUS PIXEL VALUE FOR FUTURE REFERENCE.
50 ; STA REFVAL
51 ; MVI H, 33H
52 ; SET THE FIRST BOUNDARY
54 POINT EQUAL TO "3".
55 ; LDA BREG
56 ; THIS IS TO GET INFORMATION NEEDED IN FINDING NEIGHBOURS.
57 ; SUB B
58 ; THIS GIVES THE VALUE 0
59 D REG.
60
61 ; MOV D, A
62 ; SAVE VALUE OF "K2" IN D REG.
63 ; LDA CREG
64 ; THIS IS TO GET VALUE 0 IN E REG.
65
66 ; LDA BREG
67 ; ADD E
68 ; ADD THE CONTENTS OF REGISTER SP^EIFIED.
69 ; MOV C, A
70 ; REPLACE THE ROW INFORMATION CURRENT ROW INFORMATION.
71 ; STA BREG
72 ; ALSO SAVE THIS ROW INFORMATION FOR LATER USE.
73 ; XPLUS E
74 ; MOV, A, B
75 ; LOAD THE ACC WITH THE CURRENT ROW INFORMATION.
76 ; YMINUS D
77 ; MOV C, A
78 ; BACK INTO B REG.
79 ; STA, CREG
80 ; BOUNDARY
81 ; SUB D
82 ; LRLD PICTURE.
83 ; STARTING ADDRESS OF THE PICTURE.
84 ; CALL ADRGEN
85 ; CALL SUBROUTINE TO GENERATE ADDRESS REQUIRED.
86 ; LDA REFVAL
87 ; UPGRADE INTO ACC.
88 ; CMP M
89 ; TEST PIXEL OBTAINED.
90 ; JNZ NEXT TEST, A BOUNDARY POINT.
91 ; IF NOT EQUAL, IT INDICATE
92 ; GO FOR THE NEXT STEP.
<table>
<thead>
<tr>
<th>LOC OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0036 A7</td>
<td>81+</td>
<td>MOV B,A</td>
</tr>
<tr>
<td>0037 79</td>
<td>82+</td>
<td>YMINUS E</td>
</tr>
<tr>
<td>0038 320000</td>
<td>83+</td>
<td>MOV A,C</td>
</tr>
<tr>
<td>0039 93</td>
<td>84+</td>
<td>STA CREG</td>
</tr>
<tr>
<td>003E 4F</td>
<td>85+</td>
<td>SUB. E</td>
</tr>
<tr>
<td>003D 2A0000</td>
<td>86+</td>
<td>MOV C,A</td>
</tr>
<tr>
<td>003F 87</td>
<td>87+</td>
<td>BOUNRY</td>
</tr>
<tr>
<td>0045 2A0000</td>
<td>88+</td>
<td>LHLDD PICTRE</td>
</tr>
<tr>
<td>0040 CD0000</td>
<td>89+</td>
<td>E PICTURE</td>
</tr>
<tr>
<td>0043 3AD800</td>
<td>90+</td>
<td>CALL ADGEN</td>
</tr>
<tr>
<td>0046 BE</td>
<td>91+</td>
<td>LOAD THE REFERANCE VAL</td>
</tr>
<tr>
<td>0047 02CA00</td>
<td>92+</td>
<td>CMP M</td>
</tr>
<tr>
<td>0049 79</td>
<td>93+</td>
<td>CMP M</td>
</tr>
<tr>
<td>0050 320000</td>
<td>94+</td>
<td>STA CREG</td>
</tr>
<tr>
<td>0051 93</td>
<td>95+</td>
<td>STA BREG</td>
</tr>
<tr>
<td>0054 79</td>
<td>96+</td>
<td>STA BREG</td>
</tr>
<tr>
<td>0055 4F</td>
<td>97+</td>
<td>SUB E</td>
</tr>
<tr>
<td>0056 2A0000</td>
<td>98+</td>
<td>TEST PIXEL OBTAINED.</td>
</tr>
<tr>
<td>0058 79</td>
<td>99+</td>
<td>YMINUS E</td>
</tr>
<tr>
<td>005B 0000</td>
<td>100+</td>
<td>MOV A,C</td>
</tr>
<tr>
<td>005A 93</td>
<td>101+</td>
<td>STA CREG</td>
</tr>
<tr>
<td>0055 4F</td>
<td>102+</td>
<td>MOV C,A</td>
</tr>
<tr>
<td>0057 2A0000</td>
<td>103+</td>
<td>LHLDD PICTRE</td>
</tr>
<tr>
<td>0059 CD0000</td>
<td>104+</td>
<td>E PICTURE</td>
</tr>
<tr>
<td>005C 3AD800</td>
<td>105+</td>
<td>CALL ADGEN</td>
</tr>
<tr>
<td>005F 87</td>
<td>106+</td>
<td>LOAD THE REFERANCE VAL</td>
</tr>
<tr>
<td>0060 02CA00</td>
<td>107+</td>
<td>CMP M</td>
</tr>
<tr>
<td>0062 79</td>
<td>108+</td>
<td>CMP M</td>
</tr>
<tr>
<td>0064 320000</td>
<td>109+</td>
<td>STA CREG</td>
</tr>
<tr>
<td>0067 93</td>
<td>110+</td>
<td>STA BREG</td>
</tr>
<tr>
<td>0068 47</td>
<td>111+</td>
<td>SUB E</td>
</tr>
<tr>
<td>0069 79</td>
<td>112+</td>
<td>MOV A,C</td>
</tr>
<tr>
<td>006A 320000</td>
<td>113+</td>
<td>MOV A,C</td>
</tr>
<tr>
<td>006D 82</td>
<td>114+</td>
<td>MOV B,A</td>
</tr>
<tr>
<td>006E 4F</td>
<td>115+</td>
<td>MOV B,A</td>
</tr>
<tr>
<td>116+</td>
<td>YPLUS D</td>
<td></td>
</tr>
<tr>
<td>117+</td>
<td>MOV A,C</td>
<td></td>
</tr>
<tr>
<td>118+</td>
<td>MOT C,A</td>
<td></td>
</tr>
<tr>
<td>119+</td>
<td>ADD D</td>
<td></td>
</tr>
<tr>
<td>120+</td>
<td>ADD THE CONTENTS OF REG SPE</td>
<td></td>
</tr>
<tr>
<td>121+</td>
<td>REPLC THE COLUMN INFR</td>
<td></td>
</tr>
</tbody>
</table>
LOC OBJ LINE 001 "SOURCE STATEMENT"

006F 2A0000 E 1221 LHL PICRE: STARTING ADDRESS OF THE "PICRE"
0072 3D0000 E 123+ CALL ADGEN+ CALL SUBROUTINE TO GENERATE ADDRESS REQUIRED.
0075 3AD600 C 124+ LDA REFVAL: LOAD THE REFERENCE VALUE
0078 BE E 125+ CMP M: COMPARE IT WITH THE LA
0079 C2CA00 C 126+ JNZ NEXT: IF NOT EQUAL IT INDICATES
007C 78 E 127+ XMINUS E
007D 320000 E 128+ STA BREG: SAVE THE REG.
0080 93 E 129+ SUB E:
0081 47 E 130+ MOV B'A:
0082 79 E 131+ YPLUS D:
0083 326000 E 132+ STA CREG: ...
0086 62 E 133+ ADD D: ADD THE CONTENTS OF REG SPE.
0087 4F E 134+ MOV C'A:
0088 2A0000 E 135+ LHL PICRE: STARTING ADDRESS OF THE "PICRE"
008B CC0000 E 136+ CALL ADGEN: CALL SUBROUTINE TO GENERATE
008E 3AD500 C 137+ EPC: LOAD THE REFERENCE VALUE REQUIRED.
0091 BE E 138+ CMP M: COMPARE IT WITH THE LA
0092 C2CA00 C 139+ JNZ NEXT: IF NOT EQUAL IT INDICATES
0095 78 E 140+ XMINUS E
0096 320000 E 141+ STA BREG: SAVE THE REG.
0099 82 E 142+ ADD D: ADD THE CONTENTS OF REG SPE.
009A 47 E 143+ MOV B'A:
009B 79 E 144+ YPLUS E:
009C 320000 E 145+ STA CREG: ...
009F 83 E 146+ ADD E: ADD THE CONTENTS OF REG SPE.
00A0 4F E 147+ MOV C'A:

CURRENT ROW INFORMATION: ALSO SAVE THIS ROW INF.

MIN FOR LATER USE: ADD D: ADD THE CONTENTS OF REG SPE.

BACK INTO R REG: REPLACE THE ROW INFRM.

LOAD ACC WITH THE CURRENT ROW INFORMATION:

ENT COL INFRMN: ALSO SAVE THIS VALUE FOR LATER USE.

LOAD ACC WITH THE CURRENT ROW INFORMATION:

MOV A.C: REPLACE THE COLMN INFR.
LOC OBJ-ECHOING SOURCE STATEMENTS

00A1 2A0000  E 155  BOUNRY
00A4 CD0000  E 157  LBLD PICTRE
00A7 3AD800  C 158  CALL ADRGEN
00AA BE      159  ERATE ADDRESS REQUIRED.
00AB C2CA00  C 160  LDA REFVAL
00AE 7E      161  UE INTO ACC.
00AF 320000  E 162  CMP M
00B2 82      163  TEST PIXEL OBTAINED.
00B3 47      164  TES A BOUNDARY POINT SO
00B4 76      165  GO FOR THE NEXT STEP.
00B5 320000  E 166  XPLUS D
00B8 83      167  MOV A,B
00B9 4F      168  CURRENT ROW INFORMTN.
00BA 2A0000  E 169  STA BREG
00BD CD0000  E 170  RRM FOR LATER USE.
00C0 3AD800  C 171  ADD D
00C3 BE      172  R SPECIFIED.
00C4 C2CA00  C 173  MOB A,B
00C7 C3D700  C 174  BACK INTO B REG.
00CA 7E      175  BACK INTO D REG.

STARTING ADDRESS OF TH-
CALL ADRGEN
ERATE ADDRESS REQUIRED.
LOAD THE REFERENCE VAL
COMPARE IT WITH THE LA
IF NOT EQUAL IT INDICA
GO FOR THE NEXT STEP.
LOAD THE ACC WITH THE
ALSO SAVE THIS Row INF
ADD THE CONTENTS OF REGISTE
REPLACE THE ROW INFRMN
LOAD ACC WITH THE CURR
ALSO SAVE THIS VALUE F
ADD THE CONTENTS OF REG SPE
REPLACE THE COLMN INFR
REPLACE THE COLUMN
STARTING ADDRESS OF TH-
CALL ADRGEN
ERATE ADDRESS REQUIRED.
LOAD THE REFERENCE VAL
COMPARE IT WITH THE LA
IF NOT EQUAL IT INDICA
GO FOR THE NEXT STEP.
IF NO OTHER NEIGHBOURING
THEN IT IS AN ISOLATED
POINT.
NEXT: MOB A,M
BRING THE PIXEL INTO A
ST BOUNDARY POINT
JZ ALMOST
THE LAST STEP.
BOUNDARY POINT IS PRESENT.
CHECK IF IT IS THE FIR
IF IT IS SO THEN GO Fo.
OTHERWISE MAKE THE BOU
REPEAT THE PROCESS FOR.
LOC  OBJ  LINE  SOURCE STATEMENT

3632  186  ALMOST:  MVI  M, 32
C9    187  OVER:  RET
00D8  188  REFVAL:  DS 1

PUBLIC SYMBOLS
NABOR  C  0000

EXTERNAL SYMBOLS
ADRG E  0000  BREG E  0000  CREG E  0000  PICTRE E  0000

USER SYMBOLS
ADRG E  0000  ALMOST C  00D5  BOUNKY + 0000  BREG E  0000
CREG E  0000  LP1  C  0006  NABOR C  0000  NEXT C  00CA
OVER  C  00D7  PICTRE E  0000  REFVAL C  00D8  XMINUS + 0005
XPLUS + 0003  YMINUS + 0003  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 D5</td>
<td></td>
<td>9</td>
<td>ADREGEN: PUSH D ; SAVE PIXEL OF D REGISTER.</td>
</tr>
<tr>
<td>0001 C5</td>
<td></td>
<td>10</td>
<td>PUSH B ; SAVE ROW AND COL. INFORMATIONS</td>
</tr>
<tr>
<td>0002 E5</td>
<td></td>
<td>11</td>
<td>PUSH H ; SAVE ADDRESS IN &lt;HL&gt; REG. PAIR</td>
</tr>
<tr>
<td>0003 05</td>
<td></td>
<td>12</td>
<td>DCR B ; GET THE EXACT NUMBER OF ROWS</td>
</tr>
<tr>
<td>0004 48</td>
<td></td>
<td>13</td>
<td>TO BE ADVANCED FROM THE REFERE</td>
</tr>
<tr>
<td>0005 1E40</td>
<td></td>
<td>14</td>
<td>NCE. MULTIPLICATION WILL BE DONE NOW. MULTIPLI</td>
</tr>
<tr>
<td>0007 0608</td>
<td></td>
<td>15</td>
<td>ER. IS THE NUMBER OF ROWS TO BE ADVANCED AN</td>
</tr>
<tr>
<td>0009 1600</td>
<td></td>
<td>16</td>
<td>D. THE MULTIPLICAND IS THE NUMBER OF PIXEL</td>
</tr>
<tr>
<td>000B 62</td>
<td></td>
<td>17</td>
<td>IN EACH ROW.</td>
</tr>
<tr>
<td>000C 6A</td>
<td></td>
<td>18</td>
<td>MOV C, B ; MULTIPLIER IN C REG.</td>
</tr>
<tr>
<td>000E 79</td>
<td></td>
<td>19</td>
<td>MOV E, 40H; MULTIPLICAND IN E REG.</td>
</tr>
<tr>
<td>000F B7</td>
<td></td>
<td>20</td>
<td>MVI B, 0B8H; BIT COUNTER.</td>
</tr>
<tr>
<td>0010 CA2000</td>
<td></td>
<td>21</td>
<td>MULT: MVI D, 0H</td>
</tr>
<tr>
<td>0013 1F</td>
<td></td>
<td>22</td>
<td>MOV H, D</td>
</tr>
<tr>
<td>0014 4F</td>
<td></td>
<td>23</td>
<td>MOV L, D</td>
</tr>
<tr>
<td>0015 D21900</td>
<td></td>
<td>24</td>
<td>MULT2: XRA A</td>
</tr>
<tr>
<td>0018 19</td>
<td></td>
<td>25</td>
<td>MOV A, C</td>
</tr>
<tr>
<td>0019 EB</td>
<td></td>
<td>26</td>
<td>ORA A</td>
</tr>
<tr>
<td>001A 29</td>
<td></td>
<td>27</td>
<td>JZ ADDCOL</td>
</tr>
<tr>
<td>001B EB</td>
<td></td>
<td>28</td>
<td>RAR</td>
</tr>
<tr>
<td>001C 05</td>
<td></td>
<td>29</td>
<td>MOV C, A</td>
</tr>
<tr>
<td>001D C20D00</td>
<td></td>
<td>30</td>
<td>JNC MULTI</td>
</tr>
<tr>
<td>0020 D1</td>
<td></td>
<td>31</td>
<td>DAD D</td>
</tr>
<tr>
<td>0021 19</td>
<td></td>
<td>32</td>
<td>MULT2: XCHG</td>
</tr>
<tr>
<td>0022 C1</td>
<td></td>
<td>33</td>
<td>DAD H</td>
</tr>
<tr>
<td>0023 C5</td>
<td></td>
<td>34</td>
<td>XCHG</td>
</tr>
<tr>
<td>0024 59</td>
<td></td>
<td>35</td>
<td>DCR B</td>
</tr>
<tr>
<td>0025 1D</td>
<td></td>
<td>36</td>
<td>JNZ MULT2; ADDRESS OF ROW IS IN &lt;HL&gt;</td>
</tr>
<tr>
<td>0026 1600</td>
<td></td>
<td>37</td>
<td>ADCOL: POP D; POP THE REFERENCE ADDRESS INTO</td>
</tr>
<tr>
<td>0027 19</td>
<td></td>
<td>38</td>
<td>&lt;DE&gt;</td>
</tr>
<tr>
<td>0022 C1</td>
<td></td>
<td>39</td>
<td>REG. PAIRS WHICH WAS IN &lt;HL&gt; P</td>
</tr>
<tr>
<td>0023 C5</td>
<td></td>
<td>40</td>
<td>AIR. DAD D; STARTING ADDRESS OF A ROW IS IN</td>
</tr>
<tr>
<td>0024 59</td>
<td></td>
<td>41</td>
<td>&lt;HL&gt;</td>
</tr>
<tr>
<td>0025 1D</td>
<td></td>
<td>42</td>
<td>POP B</td>
</tr>
<tr>
<td>0026 1600</td>
<td></td>
<td>43</td>
<td>PUSH B</td>
</tr>
<tr>
<td>0027 19</td>
<td></td>
<td>44</td>
<td>MOV E, C; COLUMN INFORMATION IN E REG.</td>
</tr>
<tr>
<td>0028 19</td>
<td></td>
<td>45</td>
<td>DCR E</td>
</tr>
<tr>
<td>0029 D0H</td>
<td></td>
<td>46</td>
<td>MVI D, 0H</td>
</tr>
<tr>
<td>002A D0H</td>
<td></td>
<td>47</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
002C  50  END

PUBLIC SYMBOLS
ADRG C 0000

EXTERNAL SYMBOLS

USER SYMBOLS
ADCOL C 0020  ADRG C 0000  MULT C 0009  MULT0 C 000D
MULTI C 0019

ASSEMBLY COMPLETE: NO ERRORS
FILE: EDGEDET.*

PROGRAM FOR LINE DETECTION

PUBLIC ROW, COLMN
EXTERNAL ADRGN1, BORDR

SEG

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.
DCA A
CPI 0H
JNZ STCNTK
LDA COLMN
DCA A
STA COLMN
JMP MESSAGE
STCNTK: LDA COUNT
MVI B, LENGTH
DCR B
SUB B
MEMO
SOURCE STATEMENT

005C 32CB00 C 55  CONT STA COLMN
005F CD0000 E 56  CALL ADJHGN1
0062 3ACD00 C 57  LDA TOPREF
0065 96 C 58  SUB 'N
0066 96 C 59  CHECK WHETHER THE PRESENT POINT IS SAME AS THE LAST
0067 60  POINT OF THE TEMPLATE.
0068 60 JZ TRNRS  IF IT IS, CHECK WHETHER THERE IS THE REQUIRED TRANS
0069 61  SITION PRESENT.
0069 62 3ACB00 C 64  LDA COLMN
006C C32300 C 65  JMP ADVNCE
0066 CDF7001 C 66  TRNRS: CALL TRANS
0072 3ACC00 C 67  LDA FLAG
0075 FE01 C 68  CPI '01H
0077 CA3000 C 69  JZ NXTPKL
007A 3ACB00 C 70  LDA COLMN
007D C32300 C 71  JMP ADVNCE
0080 3ACE00 C 72  NXTPKL: LDA COUNT
0083 3D 73  DCR A
0084 32CE00 C 74  STA COUNT
0087 FE00 C 75  CPI '0H
0089 CA3000 C 76  JZ MESHAGE
008C 3ACB00 C 77  LDA COLMN
008F 3C 78  INR A
0090 C35C00 C 79  JMP CONT
0093 3ACB00 C 80  MESHAGE: LDA COLMN
0096 3C 81  INR A
0097 964 E 82  MVI B.LNCH
0099 05 E 83  DCR B
009A 90 E 84  SUB B
009B 32CB00 C 85  STA COLMN
009E 4F 86  MOV C,A
009F 3ACA00 C 87  LDA ROW
00A2 47 88  MOV B,A
00A3 CD0000 E 89  CALL BORDR
00A6 3ACF00 C 90  LDA COUNT!
00A9 3C 91  INR A
00AA 32CF00 C 92  STA COUNT!
00AD FE04 C 93  CPI '04H
00AF CAC900 C 94  JZ EXIT2
00B2 3ACB00 C 95  LDA COLMN
00B5 C32300 C 96  JMP ADVNCE
00BB 21D000 C 97  EXIT: LXI H,NEWS1
00BB 4E 98  LOOP: MOV C,M
00BC 79 99  MOV A,C
00BD FE24 100  CPI '24H
00BE CAC900 C 101  JZ EXIT2
00C2 CD09F8 C 102  CALL CO
00C5 23 103  INX: H
00C6 C3BB00 C 104  JMP: LOOP
00C9 C7 105  EXIT: RST 0
00CA 106  .ROW: DS 1
00CB 107  .COLMN: DS 1
00CC 108  .FLAG: DS 1
00CD 109  TOPREF: DS 1
LOC OBJ  LINE  SOURCE STATEMENT

00CE  110  COUNT: DS 1
00CF  111  COUNT: DS 1
00D0  20205445  112  NEWS1: DB 'TEMPLATE IS NOT MATCHED IN THE I
0    MAGES'
03D4  4D504C41
00DB  54452049
00DC  53204E4F
00E0  54204D41
00E4  54434845
00EB  4420494E
00EC  20544845
00E0  28494D41
00FA  474524
00F7  55  113  TRANS: PUSH H
00FB  54  114  MOV D, H
00F9  5D  115  MOV E, L
00FA  AF  116  XOR A
00FB  1A  117  LDAX D
00FC  FEFF  118  CPI 0FFH
00FE  C21101  C  119  JNZ DWNFLG
0101  32C000  C  120  STA TOPREF
0104  114000  C  121  LAXI D, 64
0107  12  122  DAD D
0108  7E  123  MOV A, M
0109  EE00  124  CPI 00H
010B  C21101  C  125  JNZ DWNFLG
010E  C31501  C  126  JMP STFLAG
0111  3E00  127  DWNFLG: MVI A, 00H
0113  C31801  C  128  JMP LDFLAG
0116  3E01  129  STFLAG: MVI A, 01H
0118  32C000  C  130  LDFLAG: STA FLAG
011B  E1  131  PCH H
011C  C9  132  RET
0090  C  133  END START

PUBLIC SYMBOLS
COLMN C 000B RCN C 00CA

EXTERNAL SYMBOLS
ADRGNI E 0000 BORDR E 0000

USER SYMBOLS
ADRGNI E 0000 ADVNCE C 0023 BORDIR E 000A CHKBAK C 0040
CO A FB09 COLMN C 00C0 CONT C 002C COUNT C 00CE
COUNT1 C 00CF DO C 000F DO1 C 00A0 DWNFLG C 0111
EXIT C 00AB EXIT2 C 00C9 FLAG C 00CC FSTPTN C 0015
LDFLAG C 0118 LENGTH A 0004 LJOJP C 00BB MESSAGE C 0273
NEWS1 C 00D0 NXTPLX C 0030 NXTROW C 0034 ROW C 00CA
START C 0000 STCTR C 0052 STFLAG C 0116 TOPREF C 00CD
TRANS C 0000 TRTRNS C 006F

ASSEMBLY COMPLETE, NO ERRORS
FILE_NAME: ADRGN1.SRC

SOURCE STATEMENT

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

CSEG

PUBLIC ADRGN1
EXTRN ROW, COLUMN

0000 3A0000 E  10 ADRGN1:  LDA ROW ; GET ROW INFORMATION.
0003 47      11 MOV B, A
0004 3A0000 E  12 LDA COLUMN
0007 4F      13 MOV C, A
0008 C5      14 PUSH B ; SAVE ROW AND COLUMN INFORMATIONS
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 TO BE ADVANCED FROM THE REFERENCE.
18 ; MULTIPLICATION WILL BE DONE NOW. MULTIPLIER IS THE NUMBER OF ROWS TO BE ADVANCED AND THE MULTIPLICAND IS THE NUMBER OF PIXELS IN EACH ROW.
19
20
21

000E 48      22 MOV C, B ; MULTIPLIER IN C REG.
000F 1E40      23 MVI B@40H; MULTIPLICANT IN E REG.
0011 0608      24 MVI B@O8H; BIT COUNTER.
0013 1600      25 MULTI: MUL D, DH
0015 62      26 MOV H, D
0016 6A      27 MOV L, D
0017 AF      28 MULTI0: XRA A
0018 79      29 MOV A, C
0019 B7      30 ORA A
001A CA2A00 C  31 JZ ADCCOL
001D 1F      32 RAR C
001E 4F      33 MOV C, A
001F DD2300 C  34 JNC MULTI
0022 09      35 DAD D
0023 EB      36 MULTI: XCHG
0024 29      37 DAD H
0025 EB      38 XCHG
0026 05      39 DCR B
0027 C21700 C  40 JNZ MULTI0; ADDRESS OF ROW IS IN <HL>
002A D1      41 ADCOL: POP D ; POP THE REFERENCE ADDRESS INTO <DE>
42          43  
44          45 REG. PAIRS WHICH WAS IN <HL> P.
45          46 STAIRING ADDRESS OF A ROW IS IN <HL>.
46          47 POP B ; COLUMN INFORMATION IN E REG.
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>'002E</td>
<td>1D</td>
<td>48</td>
<td>DCR E</td>
</tr>
<tr>
<td>'002F</td>
<td>1600</td>
<td>49</td>
<td>MVI D, 2H</td>
</tr>
<tr>
<td>'0031</td>
<td>19</td>
<td>50</td>
<td>DAD D</td>
</tr>
<tr>
<td>'0032</td>
<td>C9</td>
<td>51</td>
<td>CORRECT ADDRESS IS IN &lt;HL&gt; REG P</td>
</tr>
<tr>
<td>'0032</td>
<td>C9</td>
<td>52</td>
<td>RET</td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS
ADRGN1 C 0000

EXTERNAL SYMBOLS
COLMN E 0000  ROW E 0000

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

ASSEMBLY COMPLETE, NO ERRORS
; PROGRAM FOR HORIZONTAL TEMPLATE MATCHING
; FILE NAME: IMPLT
; EXTRN ADDRGN, ROWSUM, BORDR
; CO EQU 0F009H
; TRSH EQU 09F6H
; LENGTH EQU 05H

CSEG

START: LXI SP, 0BFFEH ; INITIALIZING THE STACK POINTER.
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.
RSTRT ; PUSH B
RSTRT2 ; PUSH B
CALL ADDRGN ; GET THE LOCATION FOR STARTING THE ACTION.
XCHG ; TAKE THE ADDRESS IN THE <DE> REG PAIR.
CALL ROWSUM ; CALCULATE THE SUM OF THE ROW OF THE PIXELS WITHIN THE TEMPLATE.
; RESULT IS IN <HL> REG PAIR.
POP B ; GET BACK THE ROW-COL IN
FORMATIONS.
PUSH B ; SAVE THE PREVIOUS RESULT.
IN
; MOVE NEXT TO NEXT ROW.
IN
; GET THE PROPER ADDRESS.
XCHG ; ADDRESS IN <DE>
CALL ROWSUM ; POP THE 1ST ROW'S SUM INTO <BC> REG PAIR.
; SUM OF 3RD ROW IS IN <HL> REG PAIR.
PUSH B ; GET THE ROW-COL-INFO.
T ; SAVE THE PREVIOUS RESULT.
LOC: OBJ    LINE    SOURCE STATEMENT

0020 04    47    INR B
0021 CD0000 48    CALL ADDHGN ;GET ADDRESS OF 2ND ROW.
0022 EB             49    XCHG
0023 CD0000 50    CALL ROWSUM ;GET THE SUM OF 2ND ROW.
0024 29             51    DAD H ;SUM IS DOUBLED BECAUSE
0025                  52    THE
0026                  53    ;MID-ROW ELEMENTS OF THE
0027                  54    ;TEMPLATE ARE ALL 2.
0028                  55    ;SUBTRACTION OF THE RESULT OF SECOND ROW
0029                  56    ;FROM
002A                  57    ;THE SUM OF THE RESULTS OF 1ST AND 3RD ROW
002B                  58    ;AND THE
002C                  59    ;ONE DATA IS IN <HL> REGS
002D                  60    ;OTHER DATA IS IN <BE> REGS
002E                  61    XRA A
002F                  62    MOV A,C
0030 47             63    SUB L
0031 D23500 64    MOV C,A
0032 E9             65    MOV A,B
0033 47             66    SBB H
0034 0B             67    MOV B,A
0035 79             68    ;FINAL RESULT IS IN <BC> REGS
0036 2F             69    JNC TRSLD ;IF THE RESULT IS NEGATIVE, GET THE COMPLEMENT.
0037 4F             70    CMPLNT: DGO X B
0038 78             71    MOV A,C
0039 2F             72    CMA
003A 47             73    MOV C,A
003B 21F609          74    MOV A,B
003C AF             75    CMA
003D 79             76    MOV B,A
003E 95             77    SBB H
003F 47             78    MOV A,B
0040 78             79    MOV A,B
0041 9C             80    MOV B,A
0042 CA5000 81    SBB H
0043 38             82    MOV B,A
0044 0C             83    JNC MATCH
0045 D25D00 84    TRSLD: LXI H, TRSH ;THRESHOLD VALUE IS IN <HL>
0046                  85    JNC MATCH
0047                  86    IF THE TEMPLATE DOES NOT MATCH, TRY AGAIN
0048 C1             87    BY SHIFTING IT.
0049                88    POP B
004A FE40             89    MOV A,C
004B CA5000 90    CPI 40H
004C                91    Z CHKROW ;IF ALL THE COLUMNS HAVE
004D                92    BEEN
004E                93    INR C ;COVERED, TRY NEXT ROW.
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>SOURCE STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0050</td>
<td>C30780</td>
<td>C</td>
<td>91</td>
</tr>
<tr>
<td>0053</td>
<td>78</td>
<td>C</td>
<td>92</td>
</tr>
<tr>
<td>0054</td>
<td>FE40</td>
<td>C</td>
<td>93</td>
</tr>
<tr>
<td>0056</td>
<td>C6300</td>
<td>C</td>
<td>94</td>
</tr>
<tr>
<td>0059</td>
<td>04</td>
<td>C</td>
<td>95</td>
</tr>
<tr>
<td>005A</td>
<td>C30500</td>
<td>C</td>
<td>96</td>
</tr>
<tr>
<td>005D</td>
<td>218100</td>
<td>C</td>
<td>97</td>
</tr>
<tr>
<td>0060</td>
<td>C36600</td>
<td>C</td>
<td>98</td>
</tr>
<tr>
<td>0063</td>
<td>219500</td>
<td>C</td>
<td>99</td>
</tr>
<tr>
<td>0066</td>
<td>4E</td>
<td>C</td>
<td>100</td>
</tr>
<tr>
<td>0067</td>
<td>79</td>
<td>C</td>
<td>101</td>
</tr>
<tr>
<td>0068</td>
<td>FE24</td>
<td>C</td>
<td>102</td>
</tr>
<tr>
<td>006A</td>
<td>C47400</td>
<td>C</td>
<td>103</td>
</tr>
<tr>
<td>006D</td>
<td>CD09FB</td>
<td>C</td>
<td>104</td>
</tr>
<tr>
<td>0070</td>
<td>23</td>
<td>C</td>
<td>105</td>
</tr>
<tr>
<td>0071</td>
<td>C36600</td>
<td>C</td>
<td>106</td>
</tr>
<tr>
<td>0074</td>
<td>C1</td>
<td>C</td>
<td>107</td>
</tr>
<tr>
<td>0075</td>
<td>05</td>
<td>C</td>
<td>108</td>
</tr>
<tr>
<td>0076</td>
<td>05</td>
<td>C</td>
<td>109</td>
</tr>
<tr>
<td>0077</td>
<td>79</td>
<td>C</td>
<td>110</td>
</tr>
<tr>
<td>0078</td>
<td>1605</td>
<td>C</td>
<td>111</td>
</tr>
<tr>
<td>007A</td>
<td>15</td>
<td>C</td>
<td>112</td>
</tr>
<tr>
<td>007B</td>
<td>92</td>
<td>C</td>
<td>113</td>
</tr>
<tr>
<td>007C</td>
<td>4F</td>
<td>C</td>
<td>114</td>
</tr>
<tr>
<td>007D</td>
<td>CD0000</td>
<td>E</td>
<td>115</td>
</tr>
<tr>
<td>0080</td>
<td>C7</td>
<td>C</td>
<td>116</td>
</tr>
<tr>
<td>0081</td>
<td>54454050</td>
<td>E</td>
<td>117</td>
</tr>
<tr>
<td>0085</td>
<td>4C415445</td>
<td>E</td>
<td>118</td>
</tr>
<tr>
<td>0089</td>
<td>2D4F3220</td>
<td>E</td>
<td>119</td>
</tr>
<tr>
<td>008D</td>
<td>40415443</td>
<td>E</td>
<td>120</td>
</tr>
<tr>
<td>0091</td>
<td>48454424</td>
<td>E</td>
<td>121</td>
</tr>
<tr>
<td>0095</td>
<td>54454050</td>
<td>E</td>
<td>122</td>
</tr>
<tr>
<td>0099</td>
<td>4C415445</td>
<td>E</td>
<td>123</td>
</tr>
<tr>
<td>009D</td>
<td>2D4F3220</td>
<td>E</td>
<td>124</td>
</tr>
<tr>
<td>00A1</td>
<td>4E4F3220</td>
<td>E</td>
<td>125</td>
</tr>
<tr>
<td>00A5</td>
<td>40415443</td>
<td>E</td>
<td>126</td>
</tr>
<tr>
<td>00A9</td>
<td>48454424</td>
<td>E</td>
<td>127</td>
</tr>
</tbody>
</table>

**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 F809 | EXT C 0074 | LENGTH A 0005 | MATCH C 005D |
| NOMTCH C 0063 | RMESAG C 0081 | ROWSUM E 0000 |
| RSTRT C 0005 | RSTRT1 C 0007 | SHOW C 0066 | START C 0000 |
| TRSH A 09F6 | TRSLD C 003B | |

**ASSEMBLY COMPLETE, NO ERRORS.**
LOC OBJ ) LINE SOURCE STATEMENT

1  ; PROGRAM FOR CALCULATING THE
2  ; SUM OF THE PIXELS IN A ROW
3  ; WITHIN THE TEMPLATE...
4  ;******************************************************************************
5  ; FILE NAME: ROWSUM.*
6  ;******************************************************************************
7  ;
8  CSEG
9  PUBLIC ROWSUM
10  ;
11  ROWSUM: PUSH D ; SAVE THE INITIAL ADDRESS.
12  MVI C, 00H ; INITIALIZATION OF COUNTER
13  LXI H, 00H
14  LP1: LDAX D ;
15  HUSH D
16  MOV E, A
17  MVI D, 00H
18  DAD D ; SUM IS STORED IN <HL> REGS.
19  POP D
20  INR C
21  MOV A, C
22  CFI 05H ; LENGTH OF TEMPLATE IS FIVE PIXEL LONG.
23  JZ LSTSTP
24  INX D
25  JMP LP1
26  LSTSTP: POP D.
27  RET
28  END

PUBLIC SYMBOLS
ROWSUM C 0000

EXTERNAL SYMBOLS

USER SYMBOLS
LP1 C 0006 LSTSTP C 0018 ROWSUM C 0000

ASSEMBLY COMPLETE: NO ERRORS
LOC OBJ LINE SOURCE STATEMENT

1 **************************
2 FILENAME: FILL.*
3 **************************
4 PROGRAM FOR SCANNING THE
5 INTERIOR OF A CLOSED
6 REGION AND REMOVING THE.
7 SPURIOUS FEATURES PRESENT
8 IN IT.
9 PUBLIC FILL
10 EXTRN MINUSB, MINUSC, PRSNTB, PRSNTC, PICTRE, ADRGEN
11 CSEG
12 MINUSB AND MINUSC ARE THE
13 X AND Y CO-ORDINATES OF THE
14 PREVIOUS POINT, PRSNTB AND
15 PRSNTC ARE THE PRESENT CO-
16 ORDINATES, AND ADRGEN
17 GENERATES THE ADDRESS OF THE
18 PIXELS.
19
20 0000 C5 21 FILL: PUSH B
21 0001 AF 22 XRA A
22 0002 3A0000 23 LDA MINUSB
23 0005 90 24 SUB B
24 0006 CA4000 25 JZ SCNCHK
25 0009 D2000 26 JNC LTK
26 000C 3A0000 27 RIOL: LDA PRSNTB
27 000F 47 28 MOV B, A
28 0010 3A0000 29 LDA PRSNTC
29 0013 4F 30 MOV C, A
30 0014 CA0000 31 LHLB PICTRE
31 0017 CD0000 32 CALL ADRGEN
32 001A 3601 33 LOOP: MVI M, 01H
33 001C 2B 34 DCH H
34 001D 7E 35 MOV A, M
35 001E FE33 36 CPI 33H
36 0020 CA8500 37 JZ ENDI
37 0023 FE01 38 CPI 01H
38 0025 CA8500 39 JZ ENDI
39 0028 FE32 40 CPI 32H
40 002A CA8500 41 JZ ENDI
41 002D C31000 42 JMP LOOP
42 0030 3A0000 43 LTOR: LDA PRSNTB
43 0033 47 44 MOV B, A
44 0034 3A0000 45 LDA PRSNTC
45 0037 4F 46 MOV C, A
46 0038 2A0000 47 LHLB PICTRE
47 003B CD0000 48 CALL ADRGEN
48 003E 3601 49 LOOP: MVI M, 01H
49 0040 23 50 INX H
50 0041 7E 51 MOV A, M
51 0042 FE33 52 CPI 33H
52 0044 CA8500 53 JZ ENDI
LOC OBJ   LINE  SOURCE STATEMENT
0047 FE01  54  CPI 01H
0049 CA6500 55  JZ ENDI
004C FE32  56  CPI 32H
004E CA6500 57  JZ ENDI
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  LHLD PICTRE
005F CD0000 64  CALL ADRGEN
0062 3601  65  MVI M,01H
0064 AF  66  XRA A
0065 3A0000 67  LDA MINUSB
0068 90  68  SUB B
0069 CA5000 69  JZ ENDI
006C DATA00 70  JC CHK2
006F AF  71  XRA A
0070 3A0000 72  LDA MINUSC
0073 91  73  SUB C
0074 D23000 74  JNC LTOR
0077 C35000 75  JMP ENDI
007A AF  76  CHK2 XRA A
007B 3A0000 77  LDA MINUSC
007E 91  78  SUB C
007F DA3000 79  JC LTOR
0082 C35000 80  JMP ENDI
0085 CF  81  END1 POP B
0086 C9  82  RET
83  END

PUBLIC SYMBOLS
FILL C 0000

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

USER SYMBOLS
ADRGEN E 0000  CHK2 C 007A  END1 C 0085  FILL C 0000
LOOP C 001A  LOOP1 C 005E  LTOR C 0030  MINUSB E 0000
MINUSC E 0000  PICTRE E 0000  PRSNB E 0000  PRSNTC E 0000
KTOL C 000C  SCNCHK C 0054

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 CONTER 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£Q 0DH
0001 LF.EQU 0AH
0002 CO.EQU 0F809H
0003 EOF EQU 03H ; END OF FILE
0004 RESET. EQU 01000000B ; RESET CHIP
0005 MODEI. EQU 11001110B ; MODE INSTRUCTION
0006 RXE. EQU 0000100B ; RECEIVE ENABLE
0007 RTOCT. EQU 250 ; READER TIME OUT CONST.
0008 TTYC EQU 0F7H ; OUTPUT CONTROL PORT
0009 TTYV. EQU 0F6H ; INPUT DATA PORT
000A TTYS. EQU 0F7H ; INPUT STATUS PORT
000B TXBE. EQU 0000100B ; TRANSMIT BUFFER EMPT
000C CTS. EQU 0000000B ; SIMULATE CLEAR TO S
000D END ; BY REQ. TO SEND
000E RTS. EQU 01000000B ; SET REQ. TO SEND OUT
000F RRDI. EQU 0000010B ; RECEIVE BUFFER EMPT
0010 OEMS. EQU 112 ; 11 MS TIME OUT CONST.
0011 ORG 4000H
0012 START: LXI SP,0AFFH ;INITIALIZING STACK POINTER

INITIALIZATION OF 8251 CHIP.

INITIALIZATION OF 8253 CHIP.
<table>
<thead>
<tr>
<th>LOC</th>
<th>OBJ</th>
<th>LINE</th>
<th>STATEMENT</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>405E.F1</td>
<td>97</td>
<td>POP PSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>405F.F5</td>
<td>98</td>
<td>CONT: PUSHP SW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4060.E0F</td>
<td>99</td>
<td>ANI 0FH ; LEAST SIGNIFICANT 4 BITS WILL BE SENT FIRST.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4062.F630</td>
<td>100</td>
<td>ORI 30H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4064.D3F6</td>
<td>101</td>
<td>OUT TTYO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4066.CDA4A0</td>
<td>102</td>
<td>CALL CHECK ; CHECK FOR EMPTY TRANSMIT BUFFER.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4069.4F</td>
<td>103</td>
<td>MOV C,A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>406A.CD89F8</td>
<td>104</td>
<td>CALL CO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>406D.F1</td>
<td>105</td>
<td>POP PSW ; MOST SIGNIFICANT 4 BITS WILL BE SENT FIRST.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>406E.E6F0</td>
<td>106</td>
<td>ANI 0FH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4070.F7</td>
<td>107</td>
<td>ORA A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4071.F5</td>
<td>108</td>
<td>BAR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 4072.F7  | 109  | RAR /
| 4073.F5  | 110  | RAR |
| 4074.F7  | 111  | RAR |
| 4075.F630 | 112  | OHI 30H |
| 4077.D3F6 | 113  | OUT TTYO |
| 4079.CDA4A0 | 114  | CALL CHECK |
| 407C.4F | 115  | MOV C,A |
| 407D.CD89F8 | 116  | CALL CO |
| 4080.0E20 | 117  | JVI C 20H |
| 4082.CD89F8 | 118  | CALL CO |
| 4085.E1  | 119  | POP H |
| 4086.23  | 120  | INX H ; GET THE NEXT ADDRESS. |
| 4087.C1  | 121  | POP B |
| 4088.03  | 122  | INX B ; INCREMENT THE COUNTER TO CHECK FOR THE END OF PROCESS. |
| 4089.C9  | 123  | RET |
| 408A.F5  | 124  | CHECK: PUSH PSW |
| 408B.DBF7 | 125  | CHECK1: IN TTYS |
| 408D.E60A | 126  | ANI TXBE |
| 408F.CA8B40 | 127  | JZ CHECK1 |
| 4092.F1  | 128  | POP PSW |
| 4093.C9  | 129  | RET |
| 4090  | 130  | END START |

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

AGAIN : A 404C  BEGN : A 4045  CHECK: A 408A  CHECK1: A 408B
C0 : A F809  CONT: A 405F  CR : A 000D  CTS : A 4020
DF : A 0003  GOALOP : A 4057  LF : A 000A  LOOP : A 402C
MODE: A 00CE  ONEN : A 007B  XXDA : A 4224  RESET : A 944D
NDY : A 0002  RIO:CT : A 00FA  RTS : A 0020  RXE : A 0004
SEEK : A 403E  START : A 4000  TT:YC : A 0047  TTYI : A 00F6
TTY0 : A 00F6  TTYS : A 00F7  TXRE : A 0004  WHITE : A 403B
**SOURCE STATEMENT**

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

*PROGRAM FOR TRANSFERRING DATA FROM
*OTHER COMPUTER TO BLUE BOX.

OCR EQU 0DH.
LF EQU 0AH.

CO EQU 0F809H.
EOF EQU 03H ; END OF FILE.

RESET EQU 01000000B ; RESET CHIP.

MODEL EQU 11101110B ; MODE INSTRUCTION.
RXE EQU 00000100B ; RECEIVE, ENABLE.
MTGCT EQU 250 ; READ TIME OUT CONST.
TTYC EQU 0F7H ; OUTPUT CONTROL PORT.
TTYI EQU 0F6H ; INPUT DATA PORT.
TTYS EQU 0F7H ; INPUT STATUS PORT.
TTYO EQU 0F6H ; OUTPUT DATA PORT.
TXBE EQU 00000100B ; TRANSMIT BUFFER EMPTY.

CIS EQU 00100000B ; SIMULATE CLEAR TO S
END

BY REG. TO SEND.

RTS EQU 00100000B ; SET REG TO SEND OUT

RRDY EQU 00000010B ; RECEIVE BUFFER EMPTY.

ONEMS EQU 1121 ; 1 M.S. TIME OUT CONST.
ORG 4000H

LXI SP,0AFFH ; INITIALIZING STACK POINTER

; INITIALIZATION OF 8251 CHIP.

;...

MVI A,RESET

OUT TTYC ; OUTPUT THE RESET CODE.

MVI A,MODEL ; MODE INSTRUCTION FOR 8251.

;...

OUT TTYC

; INITIALIZATION OF 8253 CHIP.

;...

MVI A,076H ; LOAD 8253 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 LSB.
LOOPE: 45  MOV A, 4  ; MSB OF BAUD RATE.
        46  OUT 0F1H
        47  
        48  
        49  ; MAIN PROGRAM
        50  
        51  
        52  
        53  MVI A, 02H
        54  OUT ITYI, ; ENABLE TRANSMIT, RECEIVE
        55  AND REG. TO SEND.
        56  LXI H, 7400H ; DATA LOADING STARTS FROM HE-
        57  RE.
        58  IN ITYI ; CLEAR BY INITIAL READING.
        59  SEERDY: IN ITYS
        60  ANI RRDY
        61  JZ SEERDY
        62  BEG: IN ITYI
        63  CPI $05H ; SEARCH FOR ENO.
        64  JNZ SEERDY
        65  AGAIN: IN ITYS ; READ STATUS.
        66  ANI TXBE ; CHECK FOR EMPTY
        67  RECEIVE BUFFER.
        68  JZ AGAIN
        69  MVI A, 06H ; SEND ACKNOWLEDGEMENT.
        70  OUT ITYO
        71  GOLOOP: IN ITYS
        72  ANI TXBE
        73  JZ GOLOOP
        74  LOOP: CALL READ
        75  MOV M, A
        76  CPI $0F
        77  INK H
        78  J4P LOOP
        79  EXIT: RST 0
        80  
        81  ; SUBROUTINE
        82  
        83  READ: PUSH H
        84  PUSH B
        85  IN ITYI ; CLEAR RECEIVE BUFFER.
        86  ; BY INITIAL READING.
        87  PRMPT: MVI A, 36H ; SEND PROMPT CHARACTER.
        88  OUT ITYO
        89  RIG: IN ITYS ; READ STATUS
        90  ANI TXBE ; CHECK FOR EMPTY BUFFER.
        91  JZ RIG
        92  TMOUT: MVI B, @TOCT ; LOAD TIME OUT CONST.
        93  CHKRDY: IN ITYS ; 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>.96</td>
<td>JNZ RI3</td>
</tr>
<tr>
<td>4067</td>
<td>CDBA</td>
<td>.97</td>
<td>CALL DELAY</td>
</tr>
<tr>
<td>406A</td>
<td>05</td>
<td>.98</td>
<td>DCR B</td>
</tr>
<tr>
<td>406B</td>
<td>C260</td>
<td>.99</td>
<td>JNZ CHKRDY</td>
</tr>
<tr>
<td>406E</td>
<td>C35E</td>
<td>100</td>
<td>JMP TMOUT</td>
</tr>
<tr>
<td>4071</td>
<td>DBF6</td>
<td>101</td>
<td>RI3 IN TTYI ; READ DATA</td>
</tr>
<tr>
<td>4073</td>
<td>F5</td>
<td>102</td>
<td>PUSH PSW</td>
</tr>
<tr>
<td>4074</td>
<td>E60F</td>
<td>103</td>
<td>ANI 0FH</td>
</tr>
<tr>
<td>4076</td>
<td>F630</td>
<td>104</td>
<td>ORI 30H</td>
</tr>
<tr>
<td>4078</td>
<td>4F</td>
<td>105</td>
<td>MOV C,A</td>
</tr>
<tr>
<td>4079</td>
<td>C09F</td>
<td>106</td>
<td>CALL CO</td>
</tr>
<tr>
<td>407C</td>
<td>0E0D</td>
<td>107</td>
<td>MVI C,CR</td>
</tr>
<tr>
<td>407E</td>
<td>C09F</td>
<td>108</td>
<td>CALL CO</td>
</tr>
<tr>
<td>4081</td>
<td>0E0A</td>
<td>109</td>
<td>MVI C,LF</td>
</tr>
<tr>
<td>4083</td>
<td>C09F</td>
<td>110</td>
<td>CALL CO</td>
</tr>
<tr>
<td>4086</td>
<td>F1</td>
<td>111</td>
<td>POP PSW</td>
</tr>
<tr>
<td>4087</td>
<td>C1</td>
<td>112</td>
<td>RI3B1 POP B</td>
</tr>
<tr>
<td>4088</td>
<td>E1</td>
<td>113</td>
<td>POP H</td>
</tr>
<tr>
<td>4089</td>
<td>C9</td>
<td>114</td>
<td>RET</td>
</tr>
<tr>
<td>408A</td>
<td>0E70</td>
<td>115</td>
<td>DELAY MVI C,ONEMS</td>
</tr>
<tr>
<td>408D</td>
<td>0D</td>
<td>116</td>
<td>DLY1 : DCR C</td>
</tr>
<tr>
<td>4093</td>
<td>C28C</td>
<td>117</td>
<td>JNZ DLY1</td>
</tr>
<tr>
<td>4099</td>
<td>C9</td>
<td>118</td>
<td>RET</td>
</tr>
<tr>
<td></td>
<td></td>
<td>119</td>
<td>END START</td>
</tr>
</tbody>
</table>

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

AGAIN  A 402F  BEGN  A 402B  CHKRDY  A 4060  CO  A 8089
CR     A 000D  CTS   A 0020  DELAY   A 408A  DLYI   A 408C
EDF    A 0003  EXIT  A 404E  GOLoop  A 403A  LF     A 808A
LOOP   A 4041  MODE1 A 00CE  ONEMS  A 0070  PRMPT  A 4053
READ   A 404F  RESET  A 0040  R10    A 4057  RI3    A 4071
RI3B   A 4087  SEERDY A 4002  RTOCT   A 80FA  RTS    A 0020
RXE    A 0004  SEERDY A 402I  START  A 4000  TMOUT  A 405E
TTYC   A 00F7  TTYI   A 00F6  TTYO    A 00F6  TTYS    A 00F7
TXBE   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.