A STUDY OF HISTOGRAM SEGMENTATION TECHNIQUES

by

TROY K. MOORE

B.S., Kansas State University, 1979

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1981

Approved by:

Nasir Ahmad

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

During the summer of 1980, the Department of Electrical Engineering at Kansas State University purchased and received a Grinnell GMR 270 Digital Image Processor and Display System. The Grinnell System is interfaced with an existing Data General Nova 4/X Processor and various peripheral devices. The Nova 4/X provides system control as well as numerical computational facilities for image evaluation and modification. Mass program and image file storage is supplied via 5 megabyte hard disk pack.

In order to become familiar with system operation and provide useful algorithms for future users, the author has pursued the development of simple histogram generation and image segmentation techniques. The concept of image histograms is discussed first. The second topic is the creation of a segmented image. The derivation of the ideal threshold concept is considered in the third section. Experimental results illustrating histogram and segmented image algorithms are presented in the fourth section. The fifth section is devoted to conclusions related to the research. Appendix A, consisting of FORTRAN V computer programs developed during the course of this research, is also included. Observations resulting from generating image data using a television camera are presented in Appendix B. Appendix C contains the guidelines for photographing images on the Grinnell display.

II. IMAGE HISTOGRAM

In any given digital image, each pixel (picture element) takes a range of intensity values. These intensity values are commonly called gray levels. Due to the binary representation of an image, the gray levels for an application are usually an integer multiple of 2, i.e.,

$$\text{number of gray levels} = 2^n$$

where $n = 1, 2, 3 \ldots$

This produces the actual dynamic range of intensities as

$$0 \leq P \leq 2^n - 1$$

where P = pixel intensity.

Each individual pixel in an image can represent any intensity value within this range. An image histogram is defined as the number of pixels in an image that have a specified gray level. All gray levels within the dynamic range will be considered. The results are plotted against the pixel intensity to illustrate the distribution of the gray levels in an image.

Three FORTRAN V programs have been developed to calculate the histogram of a specified part of an image. Listings of these programs are given in Appendix A, and are called LHP.FR, THP.FR, and SHP.FR.

Although each of the above programs calculate different histograms, all of them must manipulate the stored digital image data into a useable form. In the image digitization process, adjacent pixels on the same line are assigned a gray level and then "packed" together to form a data word. For example, consider an image that is represented by 256 gray levels. Each pixel will require 8 bits (1 byte) to represent each possible intensity level. When two pixels are paired together, a 16-bit image data word is created. This requires each program to "unpack" the 16 bit word into 2 bytes representing the two individual pixels. This process is accomplished by a logic shift right and a logical AND operation. An array whose size is dimensioned by the number of gray levels in the image is used to calculate the histogram. For each pixel in an image, the correct array element is indexed by the gray level of a pixel and incremented.

III. SEGMENTATION [1], [2]

Before pattern recognition or scene analysis techniques can be applied to a given image, the given image must be broken up into meaningful subsections, which consist of an object and the background. For example, an aerial recognaissance image may require subdivision into areas of identical land usage. Image segmentation is defined as the process of partitioning an image into desired meaningful regions. Segmentation does not make any attempt to determine any relationship between individual segments of a segmentated image.

Segmentation may be conceived as a point-dependent or a region-dependent process. Point-dependent processes are concerned with examining an image using a pixel-by-pixel approach. In what manner a given pixel relates with its neighbor pixels is a region-dependent process. This report deals only with segmentation using a point-dependent process.

Consider an image f(x,y) that consists of a dark object imposed on a light background. If it is desired to separate the image into an object region and a background region, segmentation will create a new image that can be defined as follows:

$$g(x,y) = 1 \quad \text{if } f(x,y) \text{ is an object pixel}$$
$$= 0 \quad \text{if } f(x,y) \text{ is a background pixel}$$

The process of generating g(x,y) can be thought of as applying a threshold to the image's histogram and creating a new image from the results. A typical image histogram is illustrated in Figure 1. The upper gray levels represent the light background. Lower gray levels represent the darker object. Once the threshold has been determined, a segmented image is generated by assigning one gray level to pixels above the threshold and another to those

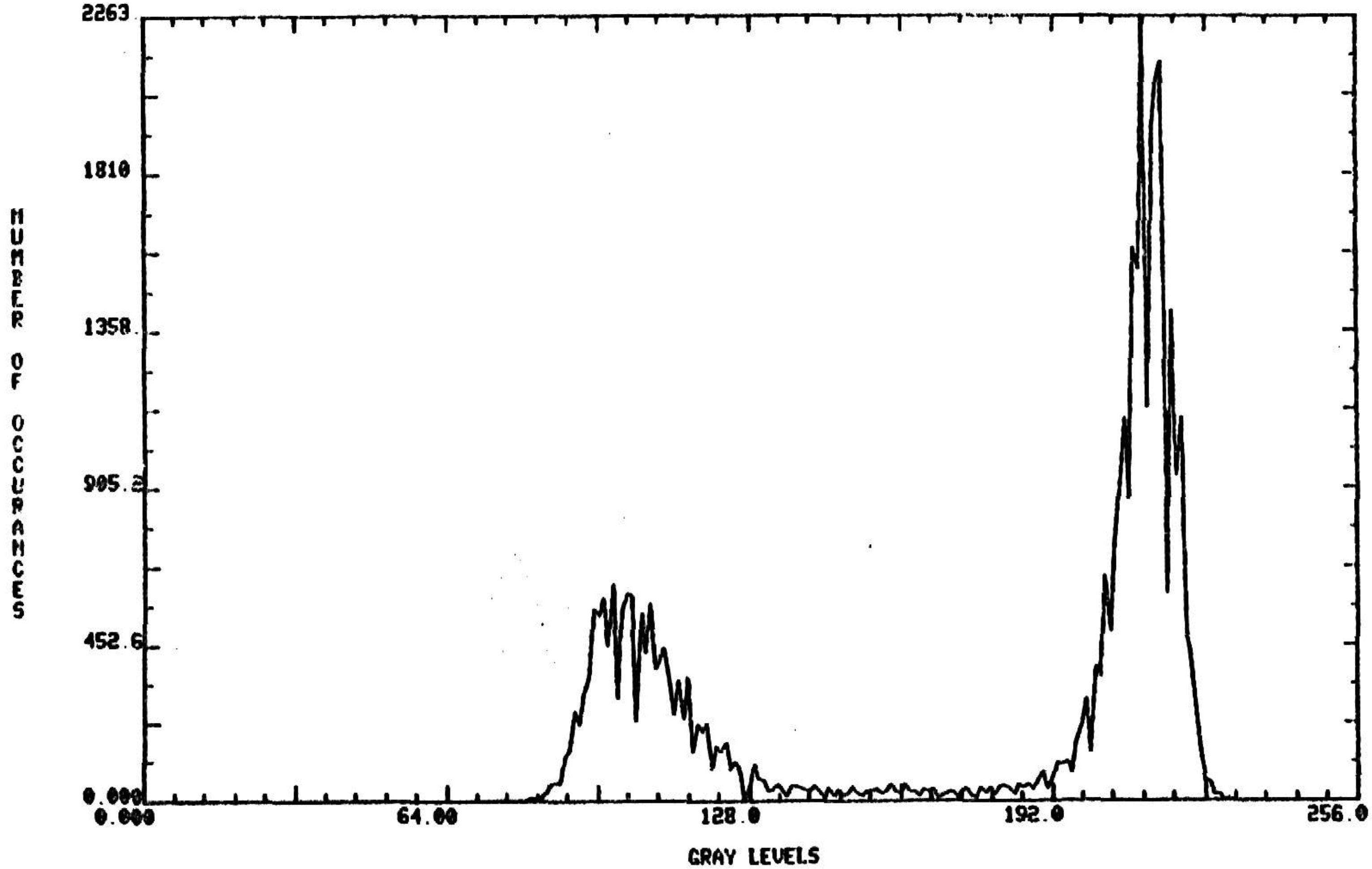

Figure 1. Typical image histogram.

pixels lying below the threshold. Those pixels on the threshold may be arbitrarily assigned either value as long as the same convention is observed thoughout the image.

Consider an image composed of 256 gray levels. Let any pixel greater than the threshold be assigned gray level 255. Any pixel less than or equal to the threshold will be assigned gray level 0. Thus g(x,y) can be redefined for this case to be

$$g(x,y) = 255 \qquad \text{threshold} < f(x,y)$$
$$= 0 \qquad 0 \leq f(x,y) \leq \text{threshold}.$$

The resulting segmented image will be composed of an all black object on an all white background.

IV. OPTIMAL THRESHOLD DETERMINATION [3]

Selection of a threshold value for the segmentation process is very important. A threshold value which is too low will result in some of the background pixels interpreted as object pixels. Too high of a threshold will result in some of the object pixels intrepreted as background pixels. In both cases, distortion is introduced in the segmented images.

Ideally, the selection of a threshold value should be determined by a minimum mean-squared error criterion. This would minimize the probability of assigning a pixel to the wrong region. Such a procedure can be implemented when considering an image on a pixel-by-pixel basis.

Let an image consist of only two brightness regions; i.e., a dark object and a light background. This image will produce a bimodal histogram (refer back to Figure 1). The area under one "hump" of the histogram will approximate the number of object or number of background pixels present in the image. Thus P_O and P_B can be defined as follows:

P_O = probability a pixel is an object pixel, and

P_B = probability a pixel is a background pixel.

Now let each "hump" of the histogram of the image approximate the probability density function of a pixel intensity for each region. If the probability density function is assumed to be Gaussian for each region, then we have:

$$p_O(x) = \frac{1}{\sqrt{2\pi}\sigma_O} \exp\left[-\frac{(x-\mu_O)^2}{2\sigma_O^2}\right] \quad (1a)$$

$$p_B(x) = \frac{1}{\sqrt{2\pi}\sigma_B} \exp\left[-\frac{(x-\mu_B)^2}{2\sigma_B^2}\right] \quad (1b)$$

where

μ_O, μ_B are the mean intensity levels

and

σ_O^2, σ_B^2 are the corresponding variance.

The overall density function becomes

$$p(x) = P_O\, p_O(x) + P_B\, p_B(x) \quad (2)$$

From the previous discussion, it follows that

$\mu_O < \mu_B$

where

μ_B is the mean intensity level of the background

and

μ_O is the mean intensity level of the object.

Figure 2 illustrates the pixel probability density functions modeled by the image histogram. Note the addition of a threshold denoted as T.

Conditional errors of assigning a pixel to the wrong region are defined as

$$P_{x|O}\,(E/O) = \int_T^{\infty} p_O(x)\; dx \quad (3a)$$

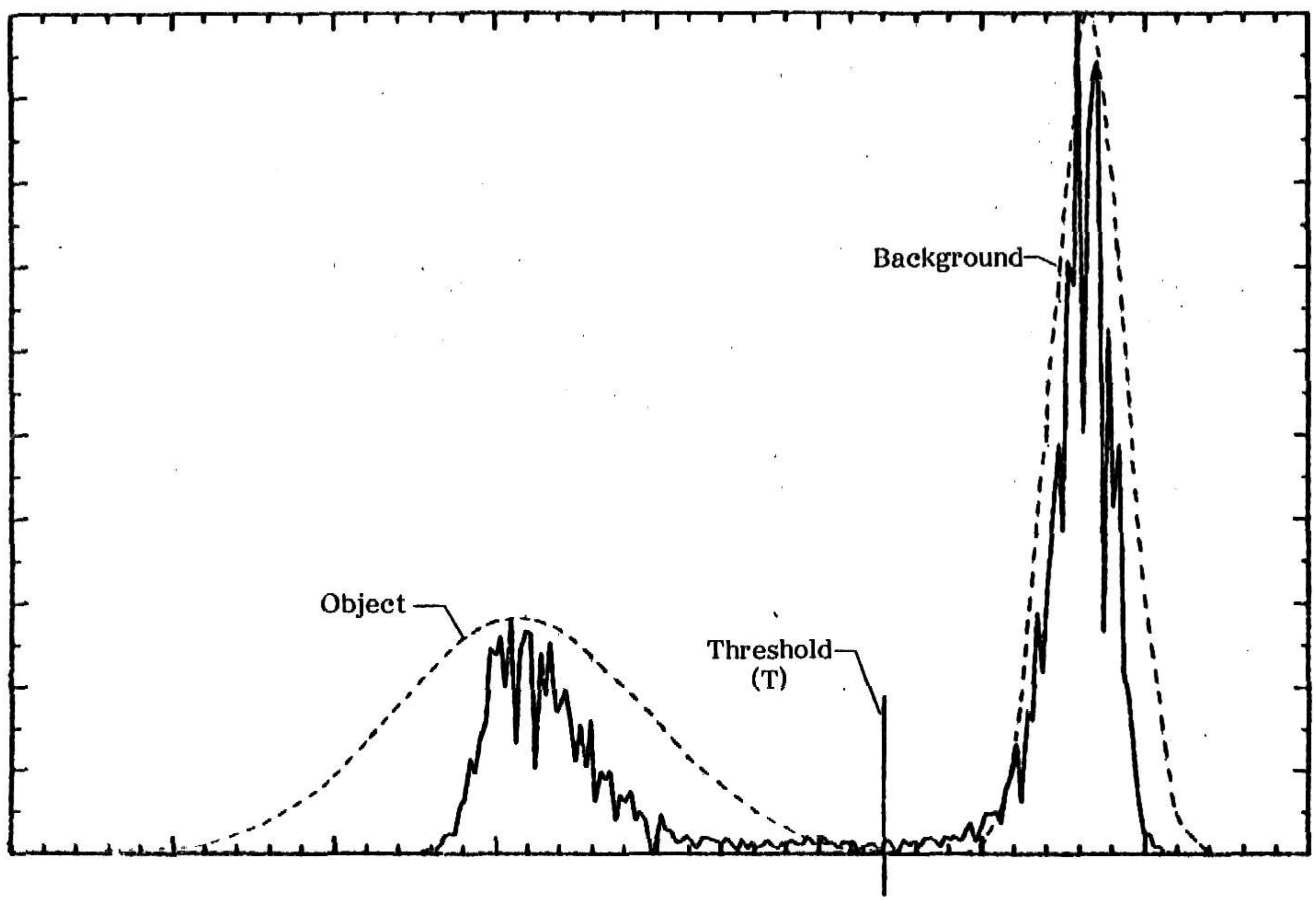

Figure 2. Pixel density functions modeled by image histogram.

and

$$P_{x|B}\ (E/B) = \int_{-\infty}^{T} p_B(x)\ dx. \qquad (3b)$$

Since the probabilities of individual errors will be a function of the threshold value T, it is convenient to redefine the errors as

$$P_O(T) = P_{x|O}\ (E/O)$$

and

$$P_B(T) = P_{x|B}\ (E/B). \qquad (4)$$

Therefore, the total probability of error P(T) is

$$P(T) = P_O\ P_O(T) + P_B\ P_B(T) \qquad (5)$$

where P_O and P_B are the priori probabilities of a pixel representing an object or background area.

To minimize the error, the first derivative with respect to T of the total error function P(T) is calculated and then set to 0. The resulting equation is solved for T. To obtain the deratives required, we recall Leibniz's Rule states that

$$\frac{d}{dx} \int_{a(x)}^{b(x)} g(z)\ dz = g[b(x)]\ b'(x) - g[a(x)]\ a'(x). \qquad (6)$$

Applying the process described above, the following results:

$$\frac{d}{dT}\ P(T) = \frac{d}{dT}\ \{P_O P_O(T) + P_B P_B(T)\} = 0. \qquad (7)$$

Substituting Equations (3a) and (3b) into (5) gives these results

$$P_O\ \frac{d}{dT} \left\{ \int_{T}^{\infty} p_O(x)\ dx \right\} + P_B\ \frac{d}{dT} \left\{ \int_{-\infty}^{T} p_B(x)\ dx \right\} = 0. \qquad (8)$$

Using Liebnitz's Rule to solve Equation (8) we obtain

$$P_O\ p_O(T) = P_B\ p_B(T) \qquad (9)$$

Substituting T into Equations (1a) and (1b) and subsequently solving for T gives an equation for the threshold in terms of P_O, P_B, μ_O, μ_B, σ_O^2, and σ_B^2. This yields

$$T^2 (\sigma_B^2 - \sigma_O^2) = T [2(\sigma_O^2 \mu_B - \sigma_B^2 \mu_O)] +$$

$$\sigma_B^2 \mu_O^2 - \sigma_O^2 \mu_B^2 - 2\sigma_O^2 \sigma_B^2 [\ln \frac{P_O \sigma_B}{P_B \sigma_O}] = 0. \qquad (10)$$

Defining constants, Equation (10) becomes

$$XT^2 + YT + Z = 0 \qquad (11)$$

where

$$X = \sigma_B^2 - \sigma_O^2 ,$$

$$Y = 2(\sigma_O^2 \mu_B - \sigma_B^2 \mu_O) ,$$

and

$$Z = \sigma_B^2 \mu_O^2 - \sigma_O^2 \mu_B^2 - 2 \sigma_O^2 \sigma_B^2 [\ln \frac{P_O \sigma_B}{P_B \sigma_O}]$$

Use of the quadratic equation is required to find the actual roots to Equation (11).

Special cases of the probability density function parameters result in simplified expressions for T. Consider the special case when the variance of the object pixel distribution is equal to the variance of the background pixel distribution, i.e.,

$$\sigma_O^2 = \sigma_B^2 = \sigma^2 \qquad (12)$$

Then X, Y, and Z coefficients of Equation (11) simplify to yield

$$T = \frac{\mu_B + \mu_O}{2} + \frac{\sigma 2}{\mu_B - \mu_O} \ln \frac{P_O}{P_B} \qquad (13)$$

An even simpler expression for T may be obtained. In addition to the assumption of Equation (12), let each pixel have equal probability of representing an object or background region. Then

$$P_O = P_B \qquad (14)$$

and the expression for T now becomes

$$T = \frac{\mu_B + \mu_O}{2} \quad . \tag{15}$$

For regions having equal areas (results in equal probabilities) and equal variances for each distribution, Equation (15) indicates the minimum error will result when the threshold is set midway between the two means.

For cases when neither of the assumptions considered in Equations (12) and (14) are valid, the calculation of T may become difficult. A priori knowledge of the pixel density distribution function's parameters as well as the probability of a pixel being from a background or object area is required. Some if not all of these parameters may not be readily available.

A simpler solution to threshold determination is now considered, keeping in mind the following questions:

> Could a terminal operator view an image histogram and from the histogram determine a threshold that will create a segmented image that has few visible errors?
>
> How will a segmented image appear when the selected threshold is below or above the optimal level?

These two questions are addressed in the following section.

V. EXPERIMENTAL RESULTS [4]

Early in the development of the optimal threshold determination derivation, a condition was imposed requiring the image to be such that it possessed a histogram of a biomodal nature. This requires a high degree of gray level difference as well as distinct boundaries between the object and background.

To accomplish this criterion, an arrow was constructed from typical black construction paper and glued to a background of white construction paper (See Figure 3).

The digital image was created by first taking an analog picture using a standard television camera. Various combinations of overhead lighting, spot lights, and camera adjustments were tried in creating a series of data files, details related to which are given in Appendix B. The image file selected for use as data was named ARROW5.IM. This image displayed the largest separation between major peaks in its histogram.

Digitization of the analog video signal is accomplished within the Grinnell System. Experimentation with different image sizes and their resulting histograms indicated a 256 row by 256 column image to be a convenient size to work with. This size was selected rather than a 512 row by 512 column image due to the sharpness of the histogram that could be obtained. The smaller images also exhibit an advantage in total program operation time. The resulting digitized images were stored on the DP 0 disk pack.

Five programs were developed for analysis and image file creation. Three programs developed exist for the purpose of generating histograms of an operator specified image. THP is a general purpose program that will calculate and plot on the graphics terminal the total histogram of an image. The histogram of a specific line within an image is calculated and plotted by the use of LHP. The histogram of a certain rectangular block within an image can be calculated using SHP. The program draws the original image on the Grinnell display and indicates the operator entered subsection of the image by drawing a box surrounding the region. The resulting histogram is plotted on the graphics terminal. Typical output from programs THP, LHP, and SHP are illustrated in Figures 4, 5, and 6.

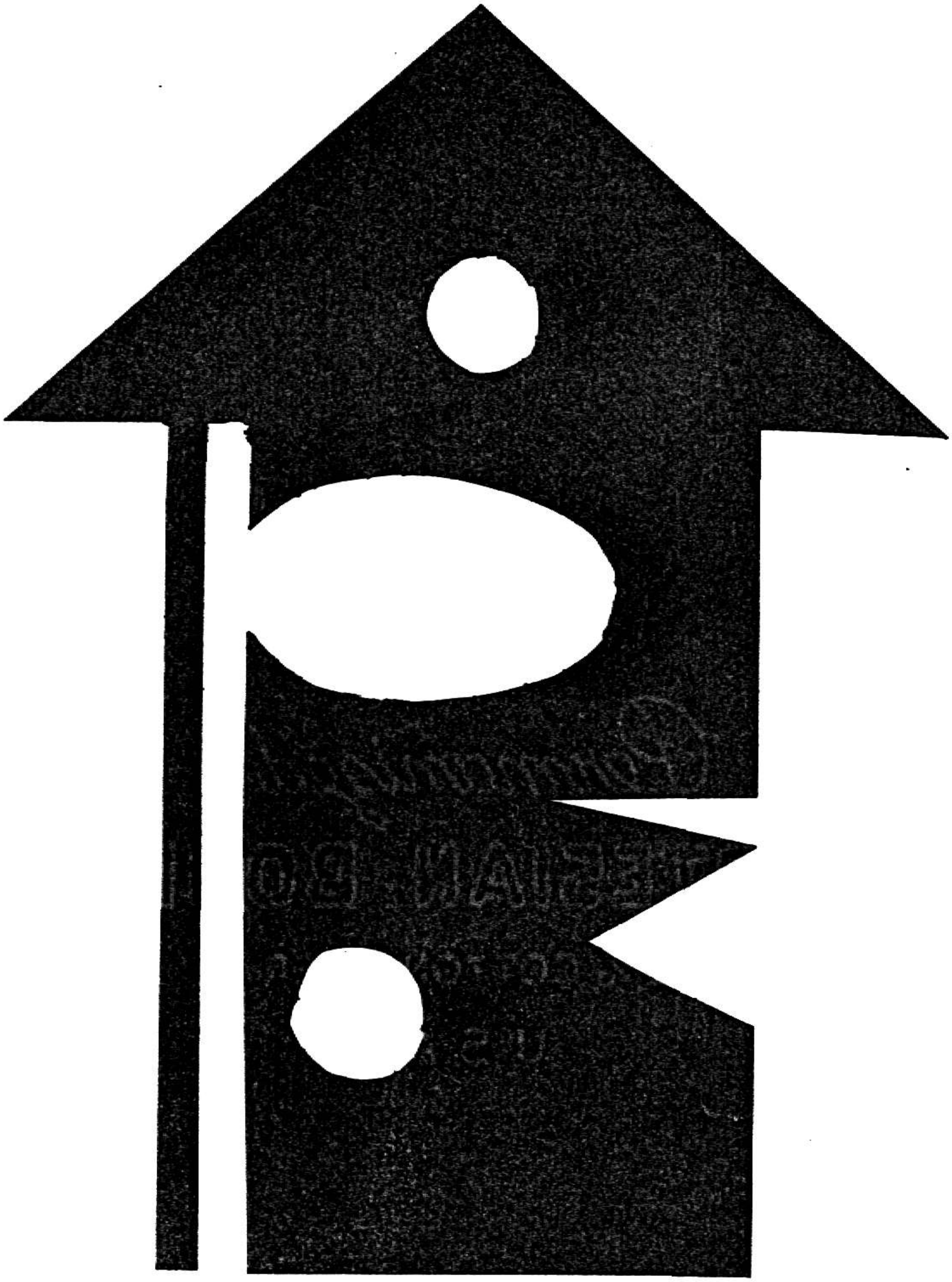

Figure 3. Original construction paper figure.

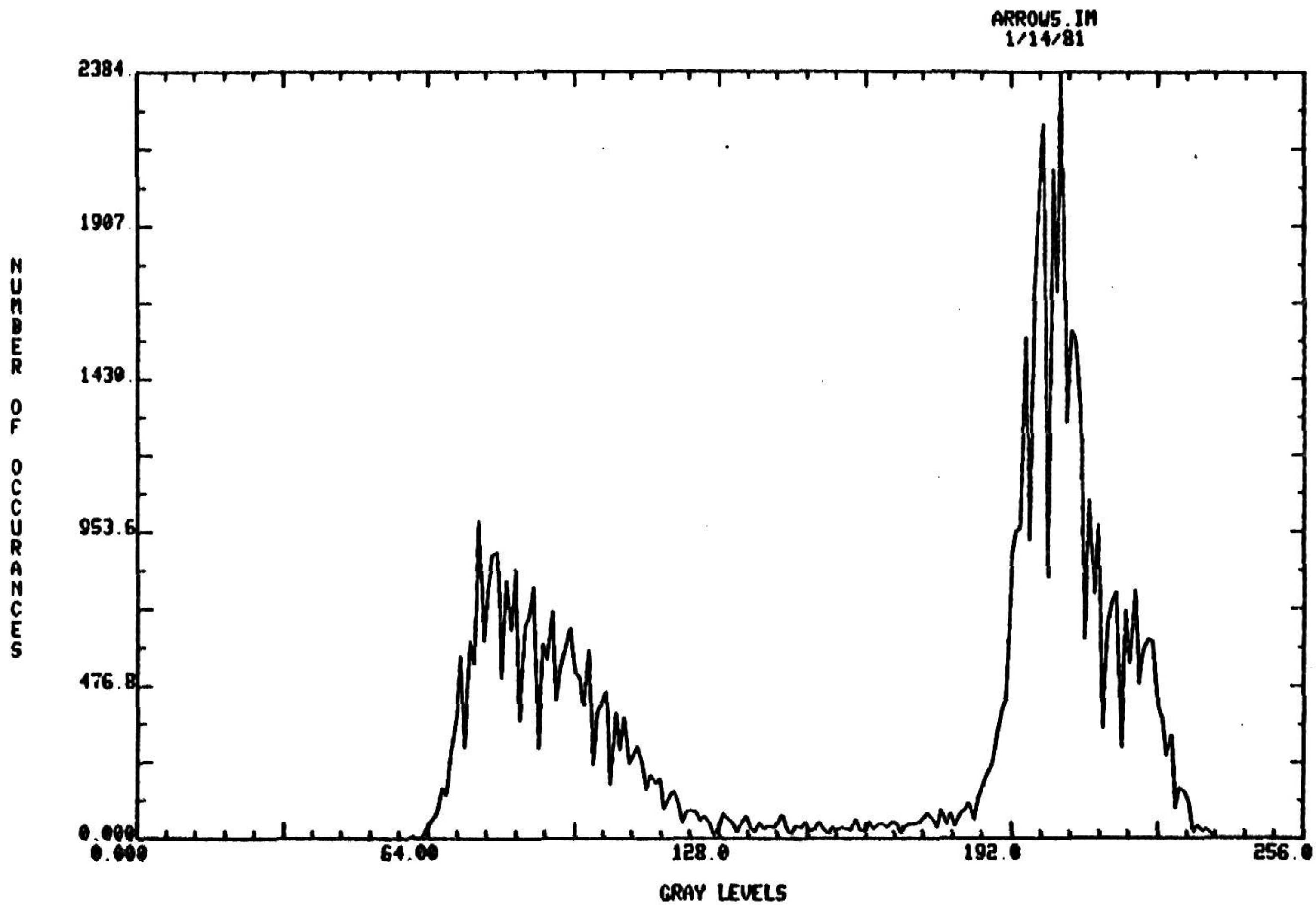

Figure 4. Typical graphics terminal output using THP.

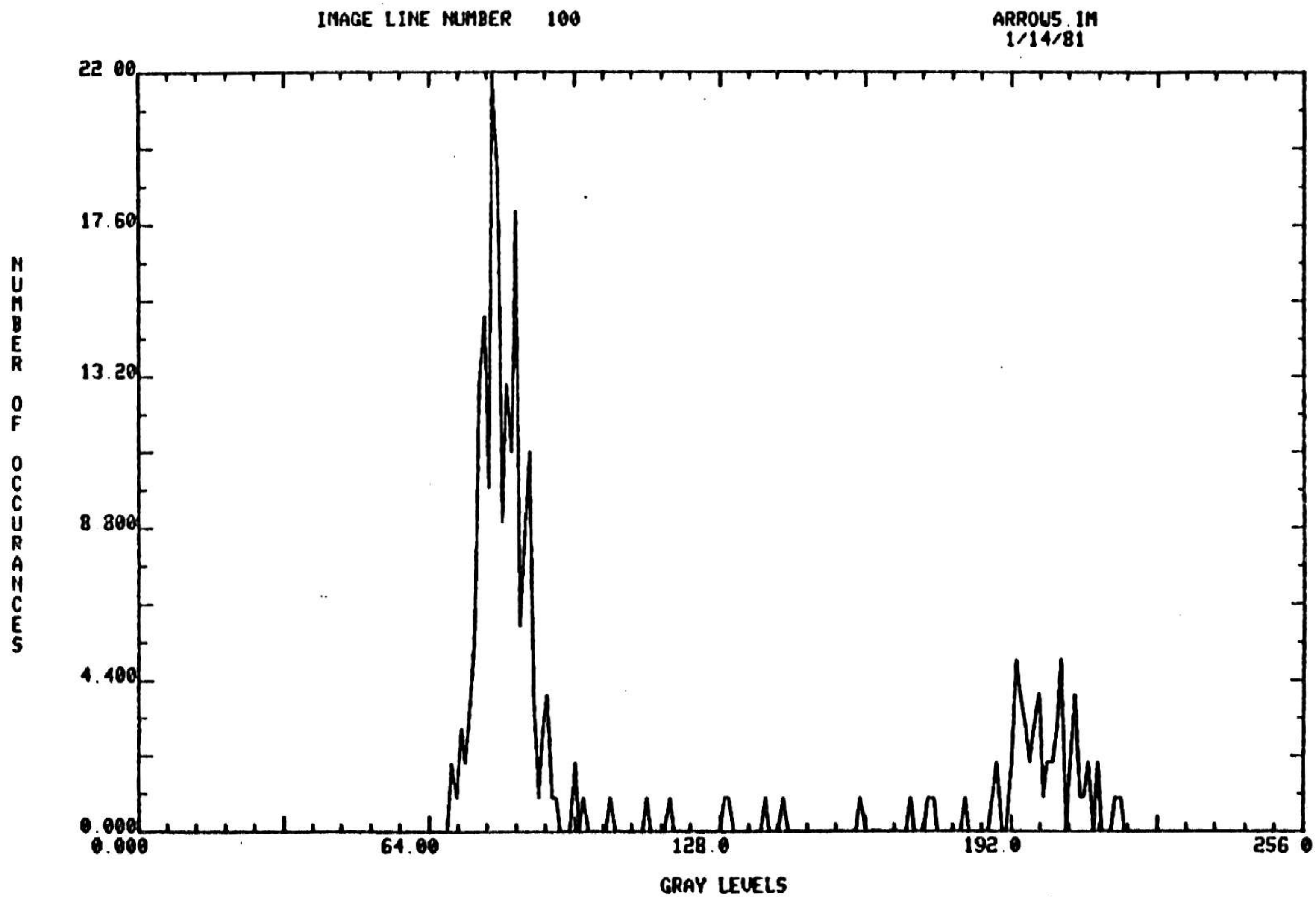

Figure 5. Typical graphics terminal output using LHP.

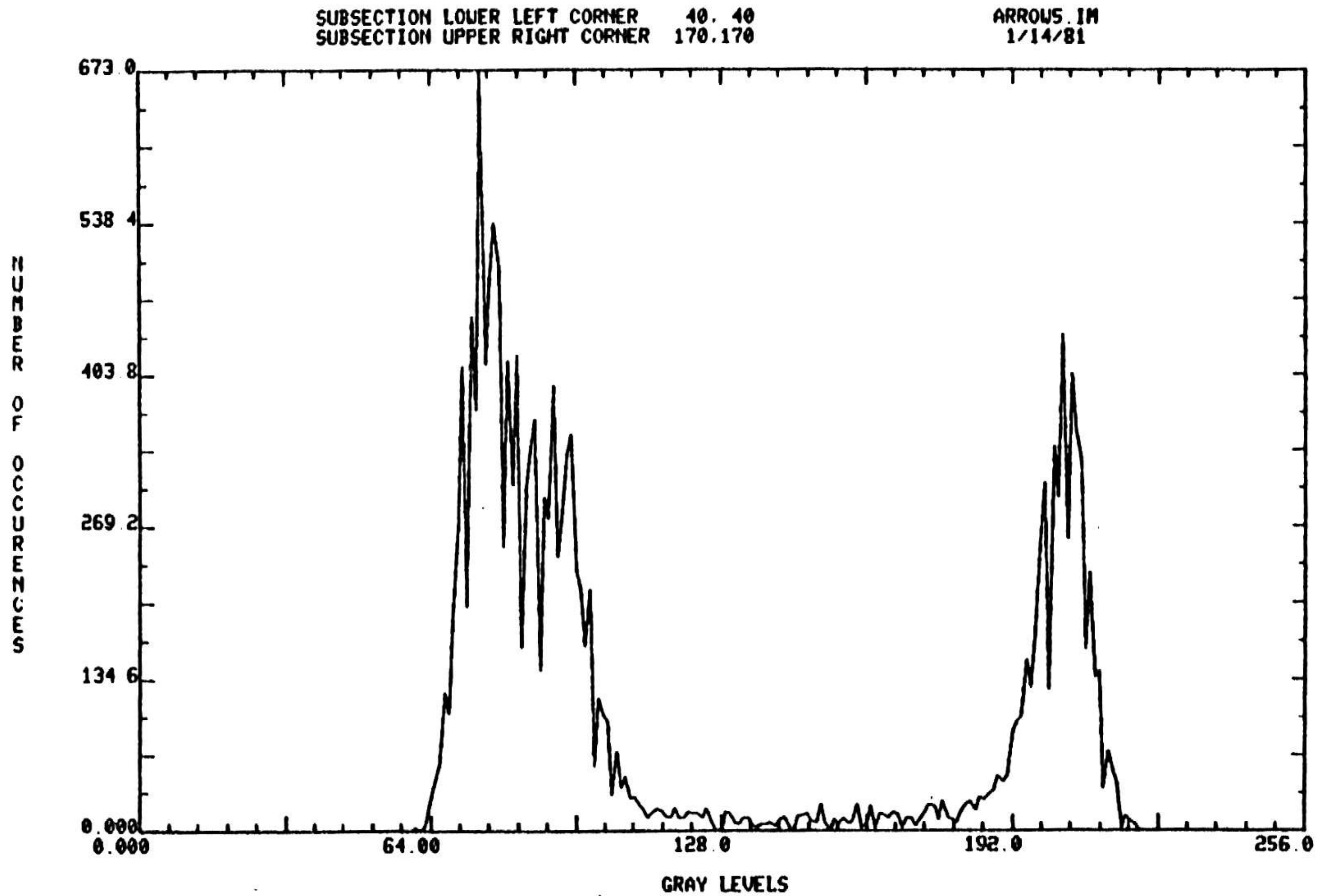

Figure 6a. Typical graphics terminal output using SIIP.

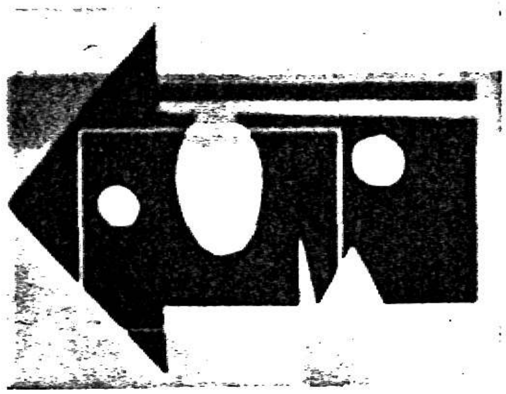

Figure 6b. Typical Grinnell display image using SHP.

Two programs were developed to create segmented images from a given image. PIXANL uses an operator entered threshold and applies it to a given image, creating a segmented image. The resulting image is stored on disk memory. A rectangular update draw program is required to draw the segmented image on the Grinnell display. MPIX is a self-contained version of PIXANL in the respect that it creates as well as draws the segmented image. The segmented image created by PIXANL and MPIX will be the same if the same image and threshold are used.

Creating a segmented image using an operator determined threshold will require the use of two programs. First, the histogram of the image is calculated and displayed on the graphics terminal using THP. Observing the histogram, the operator can select an appropriate value for the threshold. Second, the threshold is entered in the program MPIX. The segmented image is calculated and drawn on the Grinnell display. From the display the operator can visually check the results of the threshold. Image pixels assigned to the background region or background pixels assigned to the object region indicate an incorrect threshold.

Using THP, the histogram of ARROW5.IM is illustrated in Figure 4. Changing the threshold input to MPIX from 75 to 225 in steps of 25 illustrates in Figure 7 how changing the threshold affects the resulting segmented image. Figures 7b and 7c depict a threshold value that is too low. A threshold that is too high produces a segmented image seen in Figures 7g and 7h.

From Figure 4, it can be estimated that the range between which the threshold can be selected is gray levels 130 to 180. Figures 7e and 7f support this conclusion by displaying few visible errors. These results support the concept of allowing a terminal operator to choose a threshold to be used in creating a segmented image. Large errors in the choice of the threshold are

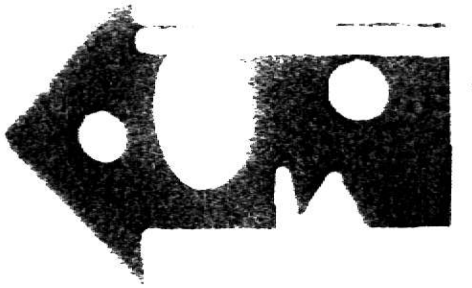

Figure 7a. Original image.

Figure 7b. T = 75.

Figure 7c. T = 100.

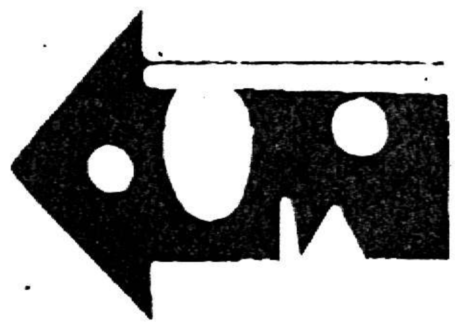

Figure 7d. T = 125.

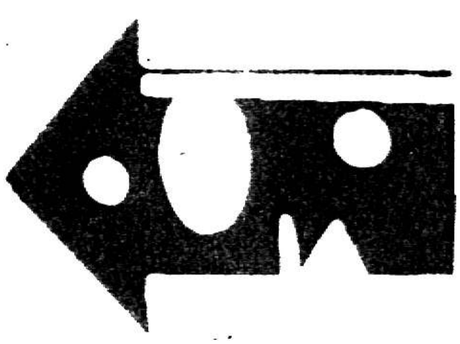

Figure 7e. T = 150.

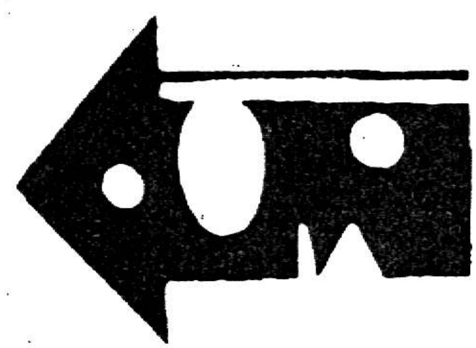

Figure 7f. T = 175.

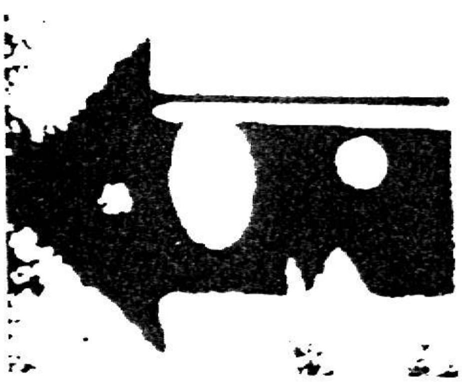

Figure 7g. T = 200.

Figure 7h. T = 225.

easily detected on the Grinnell display. For such a simple high contrast image, a large range of acceptable threshold values may be used.

VI. CONCLUSION

From the above results, it is seen that for simple, high contrast images that exhibit a bimodal histogram with well separated peaks, thresholds are easily determined and implemented. Data presented indicated the ease with which incorrect thresholds may be observed. Note is made of the wide range for which an acceptable threshold can be selected.

However, as the image becomes more complicated, more of the range of gray levels are required for total scene description. This effectively will reduce the peaks of the histogram and fill in the valleys between the peaks. The histogram may be reduced from the bimodal Gaussian model of Figure 2 to a uniform distribution model by an image with a high amount of information content. For such a case, the operator may experience a large amount of difficulty in selecting a threshold value that will give an acceptable degree of performance.

This report provides a foundation for continued research in two distinct areas. The histogram concepts may be further pursued via the implementation of histogram equalization techniques commonly used for image enhancement. The creation of segmented images is the first step in the development of a pattern recognition system or scene analysis process.

REFERENCES

[1] A. Rosenfeld, Picture Processing by Computer, Academic Press, 1969, Chapter 8.

[2] W.K. Pratt, Digital Image Processing, John Wiley and Sons, 1978, Chapter 18.

[3] R.C. Gonzalez and P. Wintz, Digital Image Processing, Addison-Wesley Publishing Company, Inc., 1977, Chapter 7.

[4] "Image Processing Display System GMR 270 User's Manual GSC-102370", Grinnell Systems Corporation.

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank all those people who assisted and provided encouragement in various ways thoughout my academic career. Appreciation is due Dr. Donald R. Hummels and Dr. Leonard E. Fuller as members of my graduate committee. Special appreciation is given to Dr. Nasir Ahmed for his patience and assistance in the capacity of my major advisor. I would also like to thank the Department of Electrical Engineering for their financial support.

APPENDIX A

```
C*************************************************************************
C
C       LINE HISTOGRAM PLOT
C
C       DG FORTRAN 5 SOURCE FILENAME:             LHP.FR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION         DATE                     PROGRAMMER
C       --------         ----                     ----------
C       00.0             JAN 18, 1981             TROY K. MOORE
C       01.0             JAN 24, 1981             TROY K. MOORE
C
C*************************************************************************
C
C       PURPOSE
C
C               THIS PROGRAM DETERMINES THE HISTOGRAM OF AN INDIVIDUAL
C               LINE IN AN IMAGE AND PLOTS THE RESULT.  THE LINE NUMBER
C               REFERS TO THE LOCATION OF THE DESIRED LINE IN REFERENCE
C               TO IT'S POSITION IN THE IMAGE, NOT THE LINE ON THE
C               DISPLAY CRT.
C
C               NOTE:  PROGRAM REQUIRES USING THE GRAPHICS TERMINIAL
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               GRAPH   AMODE
C               OPENR   CURSOR
C               ERASE   GMODE
C               QUERY   OUT
C               READR   FGDAY
C               RESET   CHECK
C
C*************************************************************************
C
C       SET UP PROGRAM PARAMETERS
        COMMON  /OPN/  IUFD(18),NAME(13)
        PARAMETER ITTO=10, ITTI=11
  15    FORMAT(I3)
  20    FORMAT(I2'/'I2'/'I2)
        DIMENSION X(257)
        INTEGER IBUFF(256)
C       ENTER PARAMETERS OF IMAGE FILE
 500    ACCEPT 'NUMBER OF ELEMENTS PER LINE IN THE IMAGE ? ',NELEM
        ACCEPT 'NUMBER OF LINES IN THE IMAGE ? ',NLINE
        XBYTE=FLOAT(NELEM)*FLOAT(NLINE)
        CALL OPENR(0,'IMAGE INPUT FILENAME ? '.NELEM,FSIZE)
        FSIZE=ABS(FSIZE-XBYTE)
        IF (FSIZE.GT.1.0) TYPE'IMAGE FILE SIZE DIFFERS FROM DEFINED SIZE'
        IF (FSIZE.GT.1.0) GO TO 500
C       SET TO ZERO VECTOR FOR HISTOGRAM CALCULATION
C       ODD SIZE OF ARRAY DUE TO SUBROUTINE GRAPH CHARACTERISTICS
 230    DO 50 I=1,257
                X(I)=0.
  50            CONTINUE
C       ENTER LINE NUMBER
 130    ACCEPT 'LINE NUMBER FOR WHICH HISTOGRAM DESIRED?  ', LINE
        IF (LINE-NLINE) 100,100,120
 120    TYPE 'DESIRED LINE NUMBER LIES OUTSIDE IMAGE'
        GO TO 130
C       READ IN THE DESIRED LINE
 100    CALL READR(0,LINE,IBUFF,1,ICNT,IER1)
        CALL CHECK(IER1)
```

```
C       UNPACK PACKED IMAGE DATA
        NWORD=NELEM/2
        DO 110 I=1,NWORD
                JH=ISHFT(IBUFF(I),-8)+1
                JL=IAND(IBUFF(I),000377K)+1
C       INCREMENT THE CORRECT COUNT
                X(JH)=X(JH)+1.0
                X(JL)=X(JL)+1.0
 110            CONTINUE
C       GRAPH HISTOGRAM
        CALL ERASE
        CALL GRAPH('GRAY LEVELS','NUMBER OF OCCURANCES',.FALSE.,1,1,257,1,1,
     1  256.0,0.0,X)
C       LABEL GRAPH WITH FILE NAME, LINE NUMBER, AND DATE
        CALL OUT(NAME,798,620,.FALSE.)
        CALL GMODE(.FALSE.,300,620)
        CALL AMODE
        TYPE 'IMAGE LINE NUMBER'
        CALL GMODE(.FALSE.,300,620)
        CALL AMODE
        CALL CURSOR(20,1)
        WRITE(10,15) LINE
        CALL GMODE(.FALSE.,798,620)
        CALL AMODE
        CALL CURSOR(1,2)
        CALL FGDAY(IMONTH,IDAY,IYEAR)
        WRITE(10,20) IMONTH,IDAY,IYEAR
        CALL GMODE(.FALSE.,0,0)
        CALL AMODE
C       RE-EXECUTE PROGRAM?
        CALL QUERY('RE-EXECUTE PROGRAM USING SAME IMAGE?  ',IANS)
        IF (IANS.EQ.1) GO TO 230
        CALL RESET
        TYPE 'NORMAL TERMINATION'
        STOP
        END
```

```
C*****************************************************************************
C
C       MODIFIED PIXEL ANALYSIS
C
C       DG FORTRAN 5 SOURCE FILENAME:             MPIX.FR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER
C       --------        ----                     ----------
C       00.0            DEC 04, 1980             TROY MOORE
C       01.0            JAN 29, 1981             TROY MOORE
C       02.0            MARCH 25, 1981           TROY MOORE
C
C*****************************************************************************
C
C       PURPOSE
C
C       PROGRAM EXAMINES ON A PIXEL BY PIXEL BASIS, EACH ELEMENT OF
C       AN IMAGE FILE.  EACH PIXEL INTENSITY IS COMPARED TO AN OPERATOR
C       ENTERED THRESHOLD VALUE.  ANY PIXEL WITH AN INTENSITY GREATER THAN
C       OR EQUAL TO THE THRESHOLD WILL BE ASSIGNED A GRAY LEVEL INTENSITY OF
C       255 IN A NEW IMAGE FILE.  INTENSITY VALUES BELOW THE THRESHOLD ARE
C       ASSIGNED GRAY LEVEL OF 000 INTENSITY.  THE PROGRAM WILL DRAW THE
C       SEGMENTED IMAGE FILE.
C
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               OPENR   CHECK
C               OPENW
C               READR
C               RESET
C               WBGNL
C               WRITR
C
C*****************************************************************************
C
C
C       SET UP PROGRAM PARAMETERS
C
        INTEGER IA(520)
        INTEGER IB(520)
        INTEGER IXIM(256)
        INTEGER IDATA(256)
C
C       ENTER PROGRAM PARAMETERS
C
   30   ACCEPT 'NUMBER OF ELEMENTS PER LINE IN INPUT IMAGE?      ', NELEM
        ACCEPT 'NUMBER OF LINES IN INPUT IMAGE?      ', NLINES
        AIMSIZ=FLOAT(NELEM)*FLOAT(NLINES)
        CALL OPENR(0,'INPUT IMAGE FILENAME?      ',NELEM,FSIZ1)
        XBYTE=ABS(AIMSIZ-FSIZ1)
        IF (XBYTE.GT.1.0) TYPE 'IMAGE FILE SIZE DIFFERENT THAN ENTERED SIZE'
        IF (XBYTE.GT.1.0) GO TO 30
   35   ACCEPT 'DESIRED THRESHOLD VALUE?      ',NTHRES
   50   CALL QUERY('SAVE GENERATED SEGMENTED IMAGE FILE ON DISK?  ',IANS)
        IF (IANS) 50,60,70
   70   CALL OPENW(1,'SEGMENTED IMAGE OUTPUT FILENAME?      ',NELEM,FSIZE)
   60   IF (FSIZE.EQ.-1.0) TYPE 'FILE NOT OPEN'
C
C       SET UP GRINNELL FOR DRAWING IMAGE (RECTAGULAR UPDATE)
C
        LMAR=(512-NELEM)/2
```

```
      TMAR=(512-NLINES)/2
      IDATA(1)=120000K
      IDATA(2)=100001K
      IDATA(3)=010377K
      IDATA(4)=024400K
      IDATA(5)=026001K
      IDATA(6)=044000K+LMAR
      IDATA(7)=064000K+TMAR
      IDATA(8)=050000K+NELEM-1
      IDATA(9)=070000K+NLINES-1
      IDATA(10)=030000K
      CALL WBGNL(IDATA,10,IER1)
      IF (IER1.NE.1) GO TO 65
C
C     UNPACK AND EXAMINE DATA
C
      NT=NELEM/2
      DO 100 K=1,NLINES
              IK=NLINES-K+1
              CALL READR (0,IK,IXIM,1,ICNT,IER2)
              CALL CHECK(IER2)
              DO 200 I=1,NT
                      IDATA(I)=000000K
                      L=2*I-1
                      J=2*I
                      IA(L)=ISHIFT(IXIM(I),-8)
                      IA(J)=IAND(IXIM(I),000377K)
                      IF (IA(L)-NTHRES.GT.0) IDATA(I)=177400K
                      IF (IA(J)-NTHRES.GT.0) IDATA(I)=IDATA(I)+000377K
                      IB(L)=000000K
                      IB(J)=000000K
                      IF (IA(L)-NTHRES.GT.0) IB(L)=000377K
                      IF (IA(J)-NTHRES.GT.0) IB(J)=000377K
 200                  CONTINUE
C
C     CREATE SEGEMENTED IMAGE FILE, DATA STORED IN PACKED FORMAT
C
              IF (IANS.EQ.1) CALL WRITR(1,K,IDATA,1,IER3)
              IF (IANS.EQ.1) CALL CHECK(IER3)
C
C     DRAW SEGMENTED IMAGE FILE
C
              CALL WBGNL(IB,NELEM,IER4)
              IF (IER4.NE.1) GO TO 65
 100          CONTINUE
  55  CALL QUERY('RE-EXECUTE MPIX?  ',IANS)
      IF (IANS) 55,65,75
  75  CALL QUERY('USING SAME IMAGE INPUT FILE?  ',IANS)
      IF (IANS) 75,80,85
  65  CALL RESET
      TYPE 'NORMAL TERMINATION'
      IF (IER1.NE.1) TYPE 'WBGNL ERROR:  GRINNELL SET UP'
      IF (IER4.NE.1) TYPE 'WBGNL ERROR'
      STOP
      END
```

```
C*****************************************************************************
C
C       PIXEL ANALYSIS
C
C       DG FORTRAN 5 SOURCE FILENAME:            PIXANL.FR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING    KANSAS STATE UNIVERSITY
C
C       REVISION        DATE                     PROGRAMMER
C       --------        ----                     ----------
C       00.0            NOV 04, 1980             TROY MOORE
C
C*****************************************************************************
C
C       PURPOSE
C
C       PROGRAM EXAMINES ON A PIXEL BY PIXEL BASIS, EACH ELEMENT OF
C       AN IMAGE FILE.  EACH PIXEL INTENSITY IS COMPARED TO AN OPERATOR
C       ENTERED THRESHOLD VALUE.  ANY PIXEL WITH AN INTENSITY GREATER THAN
C       OR EQUAL TO THE THRESHOLD WILL BE ASSIGNED A GRAY LEVEL INTENSITY OF
C       255 IN A NEW IMAGE FILE.  INTENSITY VALUES BELOW THE THRESHOLD ARE
C       ASSIGNED GRAY LEVEL OF 000 INTENSITY.
C
C*****************************************************************************
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               OPENR   CHECK
C               OPENW
C               WRITR
C               READR
C               RESET
C
C*****************************************************************************
C
        INTEGER IXIM(256)
        INTEGER IDATA(256)
C
C       ENTER PROGRAM PARAMETERS
C
 120    ACCEPT 'NUMBER OF ELEMENTS PER LINE IN INPUT IMAGE?      ', NELEM
        ACCEPT 'NUMBER OF LINES IN INPUT IMAGE?      ', NLINES
        ACCEPT 'DESIRED THRESHOLD VALUE?      ', NTHRES
        AIMSIZ=FLOAT(NELEM)*FLOAT(NLINES)
        CALL OPENR(0,'INPUT IMAGE FILENAME?      ',NELEM,FSIZ1)
        XBYTE=ABS(AIMSIZ-FSIZ1)
        IF (XBYTE.GT.1.0) TYPE 'IMAGE FILE SIZE DIFFERENT THAN ENTERED SIZE'
        IF (XBYTE.GT.1.0) GO TO 120
        CALL OPENW(1,'SEGMENTED IMAGE OUTPUT FILENAME?      ',NELEM,FSIZE)
        IF (FSIZE.EQ.-1.0) TYPE 'FILE NOT OPEN'
C
C       UNPACK AND EXAMINE DATA
C
        NELEM=NELEM/2
        DO 100 K=1,NLINES
                CALL READR (0,K,IXIM,1,ICNT,IER1)
                CALL CHECK(IER1)
                DO 200 I=1,NELEM
                        IDATA(I)=000000K
                        IH=ISHIFT(IXIM(I),-8)
                        IL=IAND(IXIM(I),000377K)
                        IF (IH-NTHRES.GT.0) IDATA(I)=177400K
 220                    IF (IL-NTHRES.GT.0) IDATA(I)=IDATA(I)+000377K
 200                    CONTINUE
```

```
C
C       CREATE SEGEMENTED IMAGE FILE
C
                CALL WRITR(1,K,IDATA,1,IER2)
                CALL CHECK(IER2)
 100            CONTINUE
        CALL RESET
        TYPE 'NORMAL TERMINATION'
        STOP
        END
```

```
C*****************************************************************************
C
C       SUBSECTION HISTOGRAM AND PLOT
C
C       DG FORTRAN 5 SOURCE FILENAME:               SHP.FR
C
C       DEPARTMENT OF ELECTRICAL ENGINEERING     KANSAS STATE UNIVERSITY
C
C       REVISION          DATE                      PROGRAMMER
C       --------          ----                      ----------
C       00.0              JAN 27, 1981              TROY K. MOORE
C       01.0              MARCH 25, 1981            TROY K. MOORE
C
C*****************************************************************************
C
C       PURPOSE
C
C               THIS PROGRAM ALLOWS THE OPERATOR TO DETERMINE THE HISTOGRAM
C               OF A SPECIFIED SUBSECTION OF AN IMAGE.  THE IMAGE IS FIRST
C               DRAWN ON THE GRINNELL.  THE OPERATOR ENTERS THE COORDINATES
C               OF THE LOWER LEFT AND UPPER RIGHT CORNER OF THE SUBSECTION.
C               A BLOCK IS DRAWN BY THE PROGRAM AROUND THE SUBSECTION FOR A
C               VISUAL CHECK OF THE REGION TO BE USED.  THE HISTOGRAM OF THE
C               SUBSECTION IS THEN CALCULATED AND DISPLAYED ON THE GRAPHICS
C               TERMINAL.
C
C       ROUTINE(S) CALLED BY THIS ROUTINE
C
C               OPENR   QUERY   GMODE
C               WBGNL   ERASE   AMODE
C               READR   GRAPH   FGDAY
C               CHECK   OUT     CURSOR
C
C*****************************************************************************
C
C               NOTES:  1)  THE BOTTOM LINE OF THE IMAGE WILL BE CONSIDERED
C                           LINE 1.  THE FAR LEFT COLUMN OF THE IMAGE WILL
C                           BE CONSIDERED COLUMN 1.  THE TOP LINE WILL BE
C                           INDEXED BY THE NUMBER OF LINES IN THE IMAGE.
C                           THE NUMBER OF COLUMNS WILL POINT TO THE FAR
C                           RIGHT COLUMN.
C                       2)  POSITION OF THE SUBSECTION WILL BE IN REFERENCE
C                           TO THE LINE AND COLUMN DESCRIBED IN 1) ABOVE.
C                           THIS GIVES THE SUBSECTION BLOCK COORDINATES X1,
C                           Y1, X2, AND Y2 THE FOLLOWING RANGE:
C                                       0 < A < # OF ELEMENTS + 1
C                                       0 < B < # OF LINES + 1
C                               WHERE  A -- X1 OR X2
C                                      B -- Y1 OR Y2
C                       3)  THE COORDINATES OF THE LOWER LEFT CORNER OF THE
C                           SUBSECTION WILL BE ENTERED FIRST FOLLOWED BY THE
C                           UPPER RIGHT CORNER.  THIS ORDER DICTATES THE
C                           FOLLOWING:
C                                       X2 > X1        Y2 > Y1
C                       4)  THE BOX DRAWN ON THE IMAGE ENCLOSES THE SUBSECTION
C                           AND IS NOT INCLUDED IN THE SUBSECTION HISTOGRAM
C                           CALCULATION.
C                       5)  THIS  PROGRAM REQUIRES THE USE OF THE GRAPHICS
C                           TERMINAL, GRINNELL IMAGE PROCESSOR, AND THE
C                           GRINNELL IMAGE DISPLAY.
C
C*****************************************************************************
C
C       SET UP THE PARAMETERS
```

```
      COMMON  /OPN/  IUFD(18),NAME(13)
      PARAMETER ITTO=10,ITTI=11
20    FORMAT (I2'/'I2'/'I2)
25    FORMAT ('SUBSECTION LOWER LEFT CORNER   'I3','I3)
30    FORMAT ('SUBSECTION UPPER RIGHT CORNER  'I3','I3)
      DIMENSION IBUF(300)
      DIMENSION IUPACK(512)
      DIMENSION ICX(2)
      DIMENSION ICY(2)
      DIMENSION X(257)
C     ENTER IMAGE FILE PARAMETERS
100   ACCEPT 'NUMBER OF ELEMENTS PER LINE IN THE IMAGE ?  ',NELEM
      ACCEPT 'NUMBER OF LINES IN THE IMAGE?  ',NLINE
      CALL OPENR(0,'IMAGE INPUT FILENAME?  ',NELEM,FS)
      DECSIZ=FLOAT(NELEM)*FLOAT(NLINE)
      SIZDIF=ABS(DECSIZ-FS)
      IF (SIZDIF.GT.1.0) TYPE 'ENTERED FILE SIZE DIFFERENT'
      IF (SIZDIF.GT.1.0) GO TO 100
C     SET UP TO DRAW IMAGE USING RECTANGULAR UPDATE
400   ILM=(512-NELEM)/2
      IBM=(512-NLINE)/2
      IBUF(1)=120000K
      IBUF(2)=100001K
      IBUF(3)=010377K
      IBUF(4)=024400K
      IBUF(5)=026001K
      IBUF(6)=044000K+ILM
      IBUF(7)=064000K+IBM
      IBUF(8)=050000K+NELEM-1
      IBUF(9)=070000K+NLINE-1
      IBUF(10)=030000K
      CALL WBGNL(IBUF,10,IER1)
      IF (IER1.NE.1) GO TO 900
C     DRAW IMAGE
      NWORD=NELEM/2
      DO 110 I=1,NLINE
              IK=NLINE-I+1
              CALL READR(0,IK,IBUF,1,ICNT,IER2)
              CALL CHECK(IER2)
              DO 120 J=1,NWORD
                      L=2*J-1
                      K=2*J
                      IUPACK(L)=ISHIFT(IBUF(J),-8)
                      IUPACK(K)=IAND(IBUF(J),000377K)
120                   CONTINUE
              CALL WBGNL(IUPACK,NELEM,IER3)
              IF (IER3.NE.1) GO TO 900
110           CONTINUE
C     ENTER LOWER LEFT AND UPPER RIGHT CORNER OF SUBSECTION BLOCK
145   IS=0
150   ACCEPT 'ENTER LOWER LEFT CORNER OF SUBSECTION BLOCK:  X1,Y1  ',
     1 ICX(1),ICY(1)
155   IF (IS.NE.1)
     1 ACCEPT 'ENTER UPPER RIGHT CORNER OF SUBSECTION BLOCK:  X2,Y2  ',
     2 ICX(2),ICY(2)
      IS=1
C     CHECK TO SEE IF SUBSECTION BLOCK LIES WITHIN IMAGE
      ICOUNT=1
      IEFLG=0
      DO 130 K=1,2
              IF (ICX(K).LT.1.OR.ICX(K).GT.NELEM) IEFLG=1
              IF (ICY(K).LT.1.OR.ICY(K).GT.NLINE) IEFLG=1
              IF (IEFLG.EQ.1) GO TO 140
              ICOUNT=ICOUNT+1
```