AN EXAMINATION OF TWO ALGORITHMS FOR DIGITAL IMAGE REGISTRATION

by

Edward Joseph Cyran

Report submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF ENGINEERING
in
Civil Engineering

APPROVED:

S. D. Johnson, Chairman

R. B. McEwen

J. Campbell

May, 1983

Blacksburg, Virginia

AN EXAMINATION OF TWO ALGORITHMS FOR DIGITAL IMAGE REGISTRATION

by

Edward J. Cyran

(ABSTRACT)

Two digital image registration algorithms are tested and evaluated on a personal computer system. The two algorithms, correlation and sequential similarity detection, are tested and evaluated for speed of execution, accuracy, and optimum parameter determination.

The programs are written in BASIC and can be easily converted to FORTRAN or other high-level language.

Three different polynomial functions are tested and evaluated to improve the resolution of the correlation determination.

Based on the results of the tests, it was concluded that the testing of registration algorithms is feasible in small computer systems and that sequential similarity detection is faster. An optimum threshold setting can be determined for an individual image. Increasing the magnitude of the SSDA threshold parameter increases the execution time of the SSDA program. Also, the resolution of the correlation can be improved with a curve fitting technique.

ACKNOWLEDGEMENTS

First of all I would like to acknowledge the support provided to me by the people of the Defense Mapping Agency and the U.S. Geological Survey during the years I have been trying to complete the studies for this degree. In particular, I would like to thank of the Defense Mapping Agency/Hydrographic Topographic Center for providing me the digital image data so necessary to the success of the project. Next, I would like to thank my family, especially my wife , for their patience and understanding during the many hours I have spent with books and computers instead of them. I would like to thank for her excellent efforts typing this report while trying to decipher my handwriting. And last, but not least, Professor Steve Johnson, for his patience and help in completing this report.

TABLE OF CONTENTS

Section Page

ABSTRACT.. ii

ACKNOWLEDGEMENTS.. iii

LIST OF FIGURES... vi

LIST OF TABLES.. viii

1. INTRODUCTION... 1

1.1 PROBLEM IDENTIFICATION...................................... 1

1.2 OBJECTIVES.. 2

1.3 SCOPE AND ORGANIZATION OF REPORT............................ 3

2. BACKGROUND... 4

2.1 IMAGE CORRELATION... 4

2.2 SUB-PIXEL ACCURACIES.. 6

2.3 SEQUENTIAL SIMILARITY DETECTION............................. 8

3. APPROACH... 12

3.1 COMPUTER RESOURCES.. 12

3.1.1 Texas Instruments (TI) Model 59........................... 12

3.1.2 Texas Instruments 99/4A................................... 12

3.2 DIGITAL IMAGE DATA SET...................................... 13

3.3 IMAGE CORRELATION ALGORITHM................................. 15

3.3.1 Initial Correlation Algorithm............................. 15

3.3.2 An Advanced Algorithm..................................... 26

3.3.3 Sub-pixel Accuracies...................................... 30

3.4 SEQUENTIAL SIMILARITY DETECTION ALGORITHM................... 31

4. RESULTS.. 33

4.1 CORRELATION RESULTS AND ANALYSIS............................ 33

TABLE OF CONTENTS (cont.)

Section Page

4.1.1 Initial Correlation Algorithm.................... 33

4.1.2 Advanced Correlation Algorithm.................... 36

4.1.3 Sub-pixel Accuracies.............................. 42

4.2 SSDA RESULTS AND ANALYSIS.......................... 48

4.3 ALGORITHM EXECUTION TIME COMPARISON................. 68

5. CONCLUSIONS AND RECOMMENDATIONS...................... 72

5.1 CONCLUSIONS... 72

5.2 RECOMMENDATIONS..................................... 72

6. BIBLIOGRAPHY.. 74

7. APPENDIX.. 76

7.1 APPENDIX A.. 76

7.1.1 Program Documentation............................ 76

7.1.2 Program Listing................................... 77

7.1.3 Program Flow Chart................................ 80

7.2 APPENDIX B.. 82

7.2.1 Program Documentation............................ 82

7.2.2 Program Listing................................... 83

7.2.3 Program Flow Diagram.............................. 85

7.3 APPENDIX C.. 87

7.3.1 System.. 87

7.3.2 Description....................................... 87

7.3.3 Data Processing................................... 88

7.3.4 Programming Language.............................. 88

VITA... 90

LIST OF FIGURES

Figure Page

2.1 Search and Reference (Window) Scenes.......................... 7

2.2 Constant Threshold Process.................................... 11

3.1 Digitized Scene... 14

3.2 IMAGE_DAT0.. 16

3.3 IMAGE_DAT1.. 17

3.4 IMAGE_DAT2.. 18

3.5 IMAGE_DAT3.. 19

3.6 IMAGE_DAT4.. 20

3.7 IMAGE_DAT5.. 21

3.8 IMAGE_DAT6.. 22

3.9 IMAGE_DAT7.. 23

3.10 IMAGE_DAT8... 24

3.11 IMAGE_DAT9... 25

3.12 T-59 Correlation Program..................................... 27

3.13 T-59 Correlation Program..................................... 28

4.1 Correlation Array... 34

4.2 Correlation versus Column Number.............................. 35

4.3 Correlation Array IMAGE_DAT5.................................. 37

4.4 Correlation versus Column Number IMAGE_DAT5.................... 38

4.5 Correlation versus Column Number with Noise................... 39

4.6 Correlation Array IMAGE_DAT7.................................. 41

4.7 Fourth Order Polynomial Fit.................................... 43

4.8 Third Order Polynomial Fit..................................... 44

4.9 Second Order Polynomial Fit.................................... 45

LIST OF FIGURES (cont.)

Figure Page

4.10 SSDA Error versus Number of Compares........................ 49

4.11 SSDA Comparison Histogram IMAGE_DAT5........................ 50

4.12 SSDA Comparison Histogram IMAGE_DAT5........................ 51

4.13 SSDA Comparison Histogram IMAGE_DAT5........................ 52

4.14 SSDA Comparison Histogram IMAGE_DAT5........................ 53

4.15 SSDA Comparison Histogram IMAGE_DAT5........................ 54

4.16 SSDA Comparison Histogram IMAGE_DAT5........................ 55

4.17 SSDA Comparison Histogram IMAGE_DAT5........................ 56

4.18 SSDA Comparison Histogram IMAGE_DAT5........................ 57

4.19 SSDA Comparison Histogram IMAGE_DAT5........................ 58

4.20 SSDA Comparison Histogram IMAGE_DAT7........................ 59

4.21 SSDA Comparison Histogram IMAGE_DAT7........................ 60

4.22 SSDA Comparison Histogram IMAGE_DAT7........................ 61

4.23 SSDA Comparison Histogram IMAGE_DAT7........................ 62

4.24 SSDA Comparison Histogram IMAGE_DAT7........................ 63

4.25 SSDA Comparison Histogram IMAGE_DAT7........................ 64

4.26 SSDA Comparison Histogram IMAGE_DAT7........................ 65

4.27 SSDA Comparison Histogram IMAGE_DAT7........................ 66

4.28 SSDA Comparison Histogram IMAGE_DAT7........................ 67

4.29 Execution Time versus Threshold............................. 69

4.30 Termination Sum versus Column Number........................ 70

7.1 Block Diagram of 99/4A Console.............................. 89

LIST OF TABLES

Table Page

4.1 Data Set.. 46

4.2 Polynomial Coefficients........................... 47

1. INTRODUCTION

1.1 PROBLEM IDENTIFICATION

The process by which observations are made to facilitate the methods of analytical photogrammetry are still very human dependent. This process usually envolves viewing a stereo pair of the scene to be measured in a precision optical measuring instrument. The desired image is either directly measured or marked with a drill or heated die and then measured in a monocomparator. In either case, the desired image coordinates are obtained only after a person views the image and brings the measuring mark into coincidence with the image point and thereby makes the decision about where the image is located. The comparator's main function is to provide the coordinates of the image point in the comparator's coordinate system and then output them in the appropriate form. This process is tedious and requires experience in photo interpretation, good eye-hand coordination, and good stereo visual acuity.

Modern stereo comparators are connected to a minicomputer to greatly expand the capabilities available during the measuring process. The computer not only assists the operator with all the bookkeeping required for measuring a large strip or block of photos, but also provides for analytical and procedural processes too. The processes provide for multiple image measurements and the preprocessing of these coordinates to compensate for film and lens distortion. However, the process of making the measuring mark coincide with the image is still the function of the human operator.

Since the launching of the Landsat satellite and several interplanetary space probes with digital image camera systems, considerable technical effort has been expended on the processing of digital images. Image registration is one area where much has been accomplished. It has been used to register scenes taken at different times or in different spectral bands. It is also used in automatic stereo compilation machines to match subscenes from a stereo pair to obtain elevation data. Most early registration systems used combinations of analog electronic circuitry and optical devices to perform the required matching processes. With the arrival of the digital images, all of the flexibility and power of the digital computer could be applied in a variety of ways.

1.2 OBJECTIVES

The first objective of this project and report is to determine if it is feasible to execute registration algorithms in small computer systems. If this is possible, then tests and evaluations of the algorithms can be made to gain experience and knowledge of their performance. The two algorithms to be tested are normalized correlation and sequential similarity (sum of the absolute differences). These algorithms are designed to incrementally translate (one pixel distance) a selected subimage (reference scene) about a larger scene (search area) in two dimensions, computing a measure of the match between the reference scene and the subimage from the search area at each increment.

The time that a registration algorithm takes to search an image can significantly affect the performance of an automated image registration system. The objective of the time comparison portion of this project was to compare the execution times of the two algorithms by having each search the same scene with the same reference scene. The execution time of the Sequential Similarity Detection method, discussed in Chapter 2, is also affected by the magnitude of the apriori set threshold parameter. The objective of this part of the project was to measure this effect.

If the objective of executing a correlation algorithm in a small computer system is feasible, then perhaps a more advanced program could be written in a higher level language in a personal computer to test real digital images. The algorithm could be tested to determine such things as the sharpness of the correlation peak, if any false correlations exist, and what, if any, were the effects of noise. Given that the correlation peak was distinguishable, a method for obtaining subpixel accuracies would be developed and tested.

In Sequential Similarity Detection, a threshold parameter must be set before a search of an image can begin. An objective of testing this algorithm was to determine if an optimum value for this parameter could be determined by observing the effect that the magnitude of this parameter had on the distribution of pixel comparisons.

1.3 SCOPE AND ORGANIZATION OF REPORT

The report describes tests conducted on two methods to register digital images. The tests were conducted initially on a made-up digital image to determine a feasible algorithm. When this was accomplished, a more advanced algorithm was tested on actual digital image data sets. The report continues, describing the methods used, the computer resources, the origin of the data sets, and the algorithms used. In Chapter 4 the results of testing each algorithm are described and evaluated. Finally, a set of conclusions and recommendations are provided as a result of the tests.

2. BACKGROUND

2.1 IMAGE CORRELATION

The initial studies of the techniques by which digital images are registered evolved from research of the references listed in the bibliographic section of this report. The impression that most of these reports give is that a very large mainframe computer is required to study the application of image registration techniques. Pratt (15) in his paper indicates that approximately 10^5 multiplications are required to search a 32 x 32 pixel search area with an 8 x 8 window. However, it must be noted that most of these papers were written in the early 1970's when mainframe computers were the only systems suited for processing large arrays of data. Even today the size must be limited to insure reasonable processing times. These kind of facts tend to discourage any serious experimentation with image registration in small computers. To add to the difficulties, digital image data sets are not readily available to the average experimenter.

A digital image is a two-dimensional array of observations of the intensity of reflected light from an illuminated scene. Each observation is a number which represents the magnitude of the intensity of the light falling on the focal plane of the digital camera at a particular location defined by x,y coordinates. Since these observations are random in nature, they can be compared to other intensity observations using statistical methods. Registration is defined as "The act of superimposing two or more images or photographs so that equivalent geographic points coincide."(6)

Correlation is a measure of the degree of dependence between two random variables. If the intensities of the pixels in two images vary in magnitude and direction in the same way, there is a good chance that the images are of the same scene.

The correlation between two random variables is defined in the following way:

$$\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = E\left[\frac{(x-E(x))}{\sigma_x} \cdot \frac{(y-E(y))}{\sigma_y}\right] \tag{2.1}$$

where E is the expected value operator.

It can also be written in the following manner when comparing two orthogonal data sets:

$$r = \frac{\sum_{i=1}^{N} X_i Y_i}{\left(\sum_{i=1}^{N} X_i^2 \cdot \sum_{i=1}^{N} Y_i^2\right)^{\frac{1}{2}}} \tag{2.2}$$

where X_i, Y_i are the points in the sample space corrected for their respective means. That is,

$$X_1 = x_1-\bar{x} \, , \; Y_1 = y_1-\bar{y}$$

$$X_2 = x_2-\bar{x} \, , \; Y_2 = y_2-\bar{y}$$

$$X_N = x_n-\bar{x} \, , \; Y_N = y_n-\bar{y} \, .$$

The correlation r can assume any value between +1 and -1. A correlation of near zero is indicative of uncorrelated or completely unrelated data sets. A correlation equal to +1 is a desired condition for the registration of a reference scene with that of a search image.

A digital image is a two-dimensional array of intensity values. Typically these intensity values can have values of between 0 and 255.

The correlation between two digital image data sets can be determined as follows:

$$R(u,v) = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} \left[(f_1(i,j) - \bar{f}_1) \cdot (f_2(i,j) - \bar{f}_2)\right]}{\left[\sum_{i=1}^{I} \sum_{j=1}^{J} (f_1(i,j) - \bar{f}_1)^2\right]^{\frac{1}{2}} \cdot \left[\sum_{i=1}^{I} \sum_{j=1}^{J} (f_2(i,j) - \bar{f}_2)^2\right]^{\frac{1}{2}}} \qquad (2.3)$$

u,v = location of subimage in search area

$f_1(i,j)$ = reference scene array

$f_2(i,j)$ = corresponding subimage in search area at location u,v.

See figure 2.1 for a description of the reference scene and search scene.

The correlation function has been used extensively to register images in digital form. P. E. Anuta writes about a particular application used to register images from several different spectral bands (1). It is also discussed extensively in reference (9) by Michael Crombie of the Engineer Topographic Laboratories. Dr. William Pratt (14,15) describes a modified approach in which he applies a Wiener Filtering technique to improve the detection of the correlation peak in highly correlated images.

2.2 SUB-PIXEL ACCURACIES

Once the peak is detected, that is, the subimage and reference scene are registered, a means for improving the resolution of the registration is desirable. While no reference can be cited which describe a means for improving resolution, the following is a result of discussions with others interested in sub-pixel accuracies, at least as the technique in the case of correlation.

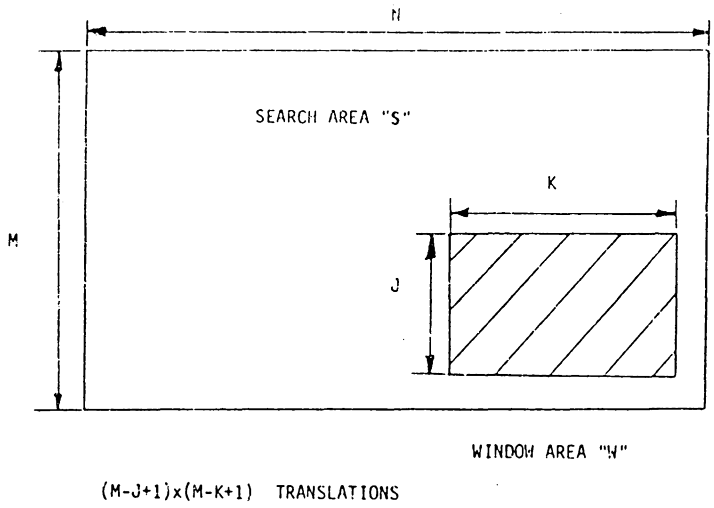


Figure 2.1 Search and Window Areas.

The pixels of a digital image can only approximate the continuously varying intensities of the actual scene. For example, the image data used in this experiment was digitized from an aerial photograph with a scale of 1 to 12,000. The pixel size as it was originally digitized was 50 x 50 micrometers. On the ground this represents a 0.6 x 0.6 meter square. After averaging the intensities of a 3 x 3 array of these pixels to compress the data, the new 150 x 150 micrometer pixel represents a 1.8 meter square on the ground. This places a resolution limit on the translational registration algorithms of 150 micrometers. This is in no way near the resolutions of 1 micrometer, least count, obtained in modern comparators. Finer digitization of the images does improve the resolution, but at the expense of much larger data sets. While 50 micrometer pixel sizes may not significantly degrade the accuracy of aerial photography, this size does have an effect on the ground resolution of satellite systems such as Landsat.

One possible technique that can be used to improve the resolution of translation registration is to use a curve fitting technique. The assumption made in using such a technique is that the values computed by the correlation algorithm are points along the continuous correlation response of the reference and subimage scenes.

2.3 SEQUENTIAL SIMILARITY DETECTION

The application of sequential similarity detection algorithms (SSDA) to the problem of translation image registration was first discussed in a paper by D. I. Barnes and H. F. Silverman (4). Their interest in this technique was driven by the need to develop an algorithm which could solve the problem of translational registration more quickly than the application of the mathematical correlation method could. A reduction of integer adds by 100 is claimed for SSDA registration.

The SSDA algorithm chosen for implementation in this project is called constant threshold technique. The search is carried out over all 144 subimages in the selected search image. For each window - subimage comparison, the pixel pairs are compared in a random order. The comparison error is defined as follows:

$$e(i,j,\ell,m) = \left| S_M^{i,j}(\ell,m) - W(\ell,m) \right| . \tag{2.4}$$

The error "e" is accumulated for all l,m at location i,j according to the following:

$$e(i,j) = \sum_{\ell=1}^{M} \sum_{m=1}^{M} \left| S_M^{i,j}(\ell,m) - W(\ell,m) \right| . \tag{2.5}$$

It is possible, therefore, for e(i,j) = 0 in the case of an ideal registration. As the error in equation (2.5) is accumulated, a test against the apriori threshold T is made. When the magnitude of e is greater than T, the search is stopped for the window - subimage location and the algorithm then starts to compare at a new location. The number of comparisons is stored, and at the end, the locations with the largest number of comparisons are considered points of similarity or registration. The value chosen for T can have a significant effect on the amount of time spent in finding points of similarity. A small T value can reduce execution time on noise-free images. However, too small a value for T could cause the rejection of all locations on real images because of image dissimilarities. Figure 2.2 describes the constant threshold process.

The search area "S" in which the window "W" is to be translated to find the best match is defined in much the same way as in direct correlation. Figure 2.1 describes the situation. Normally the area "S" has been determined to contain the replica of the window to reduce the amount of searching required. The search area is an L x L array of digital picture elements or pixels. In the case of this experiment, any one of the 10 digital image data sets could be used equally well as the search area. The dimension L for this experiment was limited to 19 and the dimension M to 8. Therefore, 19-8+1 or 12 subimages in each direction would need to be compared to detect the match. Barnea and Silverman define translational registration as a search over some subset of the allowed array of pixels to find a point which is the most similar to the given window.

The basic concept of translational registration is that there are M^2 points to be compared between a particular subimage S_{i},j and a window "W". Using the correlation function previously discussed, $M^2(L-M+1)^2$ pixel pairs must be compared before the decision about a match can be made. For a typical data set in this project, that meant $8^2(19-8+1)^2 = 9{,}216$ pixel comparisons would have to be made. In SSDA, the comparison process can be stopped before all M^2 comparisons are made for each subimage - window location. This will reduce significantly the total number of pixels compared (for example to 207 in this experiment).

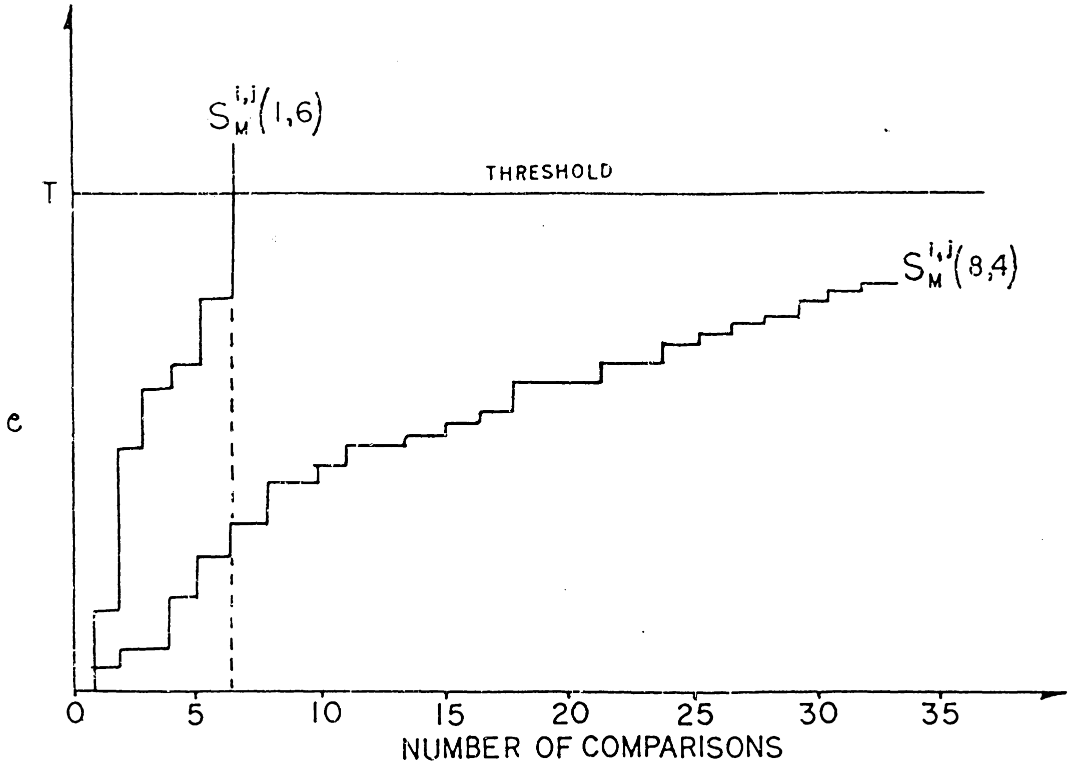


Figure 2.2 Constant Threshold Process.

3. APPROACH

3.1 COMPUTER RESOURCES

3.1.1 Texas Instruments (TI) Model 59. The TI 59 programmable calculator first appeared on the consumer market in the late 1970's. It was the successor to the model SR-52 which was the first programmable calculator offered by T.I. It has 100 data storage registers and can store programs up to 960 steps long. A magnetic card reader/writer is built into the unit. The magnetic cards can be used to store both data and programs. The capabilities of the calculator are expanded by the use of a plug-in module containing 25 programs. Included is a program which can invert a matrix 9 by 9 in size within 12 minutes. A program which proved useful for determining the maximum value of a polynomial is called "Zeros of Functions". This calculator, along with a printer, was used to test the feasibility of writing and executing an algorithm to register images using correlation.

3.1.2 Texas Instruments 99/4A. The Texas Instruments 99/4 series computers have all the elements of a true computer. That is, it will store data and programs, and the programs can be written in a higher-level language (BASIC). The computer uses a 16-bit microprocessor (TMS-9900) and has a 16-kilobyte random access memory which can be expanded to 48 kilobytes. Numeric constants are maintained internally in seven radix-100 digits (18). This is equivalent to 13 or 14 decimal digits. The range of numeric constants is from $-9.9999999999999 \times 10^{127}$ to -1×10^{-128}, 0, and 1×10^{-128} to $9.9999999999999 \times 10^{127}$. All of the programs written for this project were written in the BASIC language. A more detailed description of this computer is contained in Appendix C of this report.

3.2 DIGITAL IMAGE DATA SET

The digital image data sets used in this experiment were originally digitized for a similar experiment at the Defense Mapping Agency (DMA). The original aerial photographs were stereo pairs taken over Fort Belvoir, Virginia, at a scale of 1:12,000. These photographs are used often for training and demonstration purposes and have a large number of identifiable control points to observe. The photographs have standard 9- x 9-inch format.

When the images were digitized on a Perkin-Elmer 1010G microdensitometer, some compromises were made to minimize the size of the resulting data sets. If the full image was digitized at the pixel size of 50 micrometers, the data set would have contained well over 5×10^8 pixels. To process such a large set would have proved too costly, especially since much of the information contained in the image was not needed for the experiment.

To significantly reduce the size of the data set, the digitizing was limited to a 3- x 3-inch area on each photo. This still resulted in arrays of size 1,536 x 1,536 pixels.

Unfortunately, the conjugate image of the one used in this experiment was not available in time to conduct this test using stereo pairs. Figure 3.1 shows the location of each control point in the digitized scene. In the DMA experiment, 19 x 19 pixel arrays were selected with a control point centered in them. These 19 x 19 arrays were used as search area images in both the correlation and SSDA experiments.

Each pixel represents a square of 50 x 50 microns on the film. The intensity of the light passing through the film in this square is

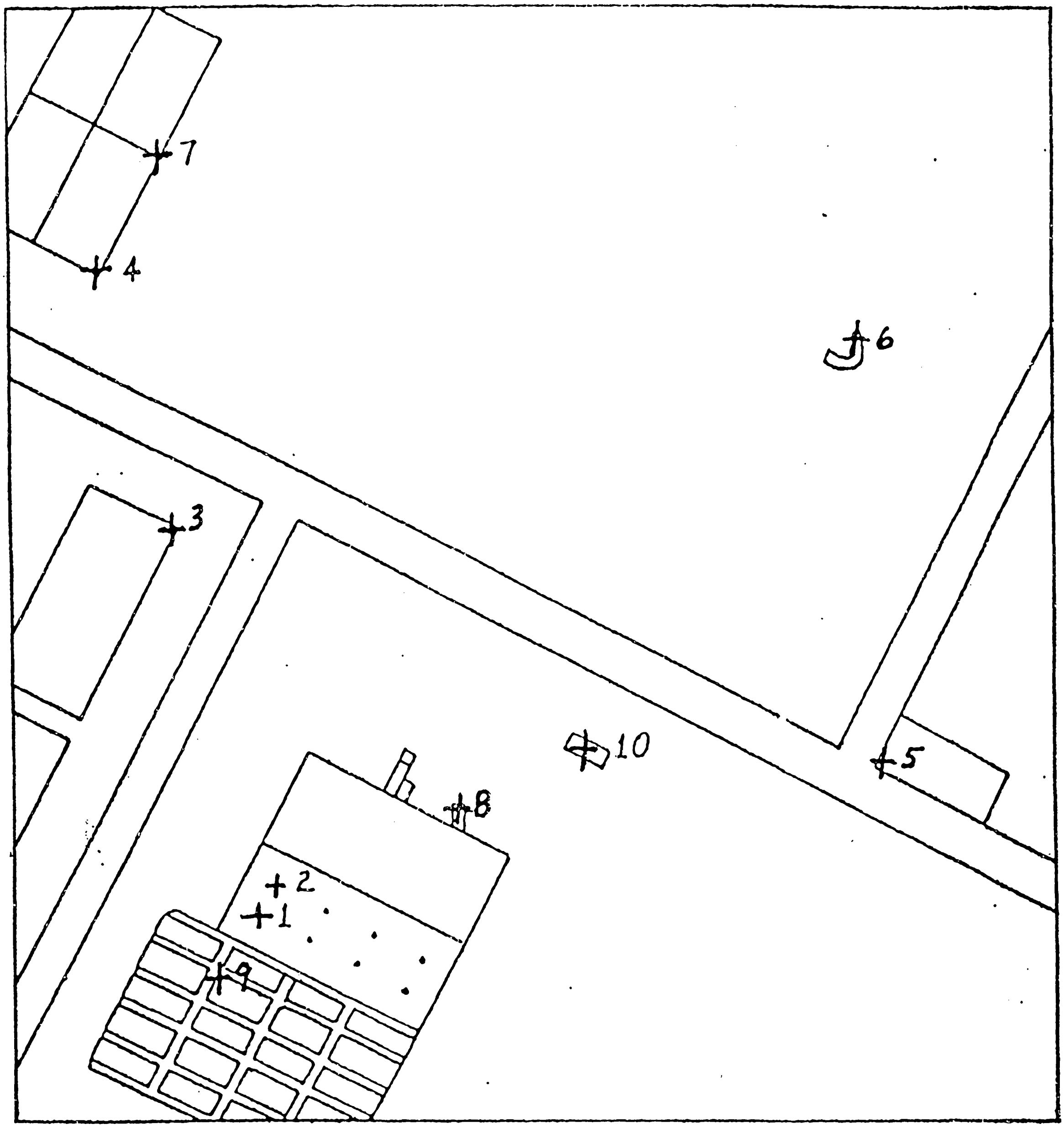


Figure 3.1 Digitized Scene.

digitized by assigning a value from 0 to 255 to it. A value of 255 represents a low density area on the film. Since an image of this size (2,359,296 pixels) still represents a formidable amount of data to be processed, a further reduction in array size was made by processing in the computer. The processing, in effect, increased the pixel size to 150 x 150 microns. The intensity value assigned was the average value of the 3 x 3 array of pixels the larger pixel replaced. This reduced the image size to a more manageable 512 x 512 array. For the purposes of this experiment, 19 x 19 were selected.

The following pages in this section list the gray levels for all 10 images selected. Also listed for each are the mean, minimum, maximum, and standard deviation of the pixel values. All 10 images are stored on a single 5 1/4-inch floppy disk. The image data files were named IMAGE_DAT0 through 9. When either program (SSDA or IMAGE_CORR) is running, the file to be used is assigned interactively. A pixel size of 150 x 150 microns represent approximately a two-meter square on the ground. At such a low resolution and large scale, these subimages are, at best, coarse or blocky in appearance when viewed on a CRT. However, for the purposes of this experiment, they are a realistic data source at an excellent price (free). The data is available to anyone else who might have use for it on punched cards.

3.3 IMAGE CORRELATION ALGORITHM

3.3.1 Initial Correlation Algorithm. Since the first objective was to demonstrate that writing a correlation algorithm was feasible for a small computer system, it was necessary to write a program for the TI-59

```
PATCH CENTER   X = 143   Y = 408
 213 211 225 224 222 222 217 226 225 227 226 221 227 230 217 214 223 225 238
 217 215 215 218 223 223 216 221 216 216 227 221 221 223 210 227 230 227 216
 223 223 218 221 218 220 218 221 213 218 225 226 222 223 221 223 230 229 221
 217 218 208 221 221 217 215 223 222 217 215 223 224 222 223 221 224 223 214
 222 225 228 224 224 224 223 224 220 217 220 229 222 223 224 221 222 214 220
 202 214 229 224 217 223 221 231 218 221 209 222 220 218 217 222 224 218 225
 217 220 227 210 224 218 223 232 207 137 132 211 222 222 222 225 221 215 221
 224 227 220 222 221 214 216 225 220 185 109 188 217 222 224 224 224 227 221
 222 214 221 218 222 221 211 218 213 188 183 202 210 223 223 216 223 212 220
 215 210 218 215 222 217 218 216 218 222 222 227 217 225 220 218 220 224 217
 228 223 220 215 218 222 210 224 220 218 224 222 224 220 227 221 225 223 228
 218 220 224 220 215 216 228 218 216 217 220 227 221 223 217 220 223 218 220
 217 224 225 211 220 214 216 223 220 222 218 217 217 223 215 228 218 221 223
 225 231 224 213 214 218 223 228 220 223 225 228 225 218 217 216 221 223 223
 222 214 217 221 220 213 224 222 223 214 218 217 215 218 216 210 215 216 211
 221 224 215 217 224 222 215 229 221 223 221 221 217 225 213 224 216 220 211
 228 230 213 221 217 222 231 222 216 223 227 227 216 218 223 224 215 221 220
 217 222 214 223 224 220 229 221 216 225 214 221 223 217 222 215 214 222 208
 217 215 225 227 220 221 220 226 223 224 215 225 228 218 214 213 221 221 211
 MAXIMUM INTENSITY IN PATCH WAS    239
 MINIMUM INTENSITY IN PATCH WAS    109
 MEAN WAS       219.233
 STANDARD DEVIATION IS    10.616608
 RMS VALUE IS   219.488886
```

Figure 3.2 IMAGE_DAT0.

```
PATCH CENTER   X •  167   Y •  376
 222 224 221 216 220 228 218 221 213 225 223 220 218 220 218 214 214 211 222
 217 221 210 215 214 222 216 220 225 229 220 223 220 221 220 216 220 221 224
 220 210 209 213 217 221 216 215 221 228 214 223 211 213 217 221 223 222 221
 228 218 221 217 222 225 218 218 214 221 224 217 227 217 217 223 220 223 223
 217 223 215 214 224 227 223 223 213 203 220 223 223 215 220 214 220 220 220
 220 223 216 221 224 224 230 223 215 214 224 224 222 216 218 222 223 218 223
 222 223 221 203 210 222 230 229 194 210 224 218 224 220 221 223 217 211 225
 230 216 229 224 223 225 231 221 178 182 157 218 220 228 218 224 215 207 215
 218 224 228 214 223 223 236 232 197 109 135 210 216 220 216 217 221 209 221
 222 222 222 221 223 221 229 208 204 209 208 214 215 220 224 218 223 213 223
 218 218 224 218 216 224 225 220 224 214 218 217 225 215 220 218 223 216 223
 220 218 217 223 229 222 224 223 222 220 227 218 227 218 214 217 217 211 216
 215 224 215 229 223 220 225 220 223 230 223 214 223 218 224 220 216 214 224
 223 225 227 221 223 216 222 224 230 227 218 223 225 215 220 218 215 217 216
 234 225 221 221 227 228 221 227 223 224 223 223 222 222 227 218 223 214 217
 227 222 220 220 224 232 222 230 221 221 227 222 225 225 218 230 225 221 223
 215 218 217 223 223 221 223 229 224 218 218 223 227 218 222 224 228 222 221
 216 220 230 220 217 227 227 224 227 221 221 225 222 222 230 230 229 223 224
 218 225 224 221 227 235 221 221 227 223 223 216 223 217 220 224 218 218 214
MAXIMUM INTENSITY IN PATCH UAS    236
MINIMUM INTENSITY IN PATCH UAS    109
MEAN UAS       219.463
STANDARD DEUIATION IS    11.262492
RMS UALUE IS   219.750602
```

Figure 3.3 IMAGE_DAT1·

```
PATCH CENTER   X =  135   Y =  247
  55  16  26  29  23  10  61 139 188 192 188 186 186 203 158 133 148 186 163
  37  47  46  22  27  32  19  25  58 149 188 194 196 189 193 133 168 139 181
 213 109 118 105 108 126 110 103 114 144 200 222 206 214 218 185 183 188 185
 199 220  87   0  16  15  17  32  29  23  39 130 171 179 176 162 129 125 158
  86 186 224 116  12  18  39  36  31  32  30  36  40 154 185 183 145 136 157
  86 101 143 216 181  36  36  44  19  18  26  22  38  39 143 194 167 140 144
  82  80  94 109 168 220 129  27  15  24  40  41  39  31  48  65 164 161 138
  97 100  90  29  28 135 220 181  34  27  27  36  41  37  37  41  45 108 183
  96  86  94  97  76  80 108 195 214  79  29  17  27  31  46  38  30  29  50
  86  26  72  89  87  91  96  86 172 227 133  29  43  37  20  26  32  24  41
  82  76  78  61  85  91  93 104 178 180  67  27  46  36  31  41  37  13  31
  74  78  62  69  87  79  78 126 188  78  30  31  46  51  53  52  29  24  39
  88  60  82  73  86  69 107 192 102   8  37  41  36  40  53  38  18   4  54
  65  69  65  79  81  87 156 171  12   6  29  43  45  33  52  46  82  37 118
  76  95  91  82  91 133 196  59  17  18  38  45  50  55  66  52  16 103 137
  74  72  71  81  98 188 102  11   9   6  30  31  22  40  37  16  43 130 139
  72  74  60  82 172 158   6  15  13  38  26  54  62  57  51  37 117 158 167
  66  62  76 125 196  65  23  31  32  29  27  39  38  41  26  91 125 159 143
 116 119 153 202 104  51  22  38  34  41  18  44  44  50  83 130 152 165 165
MAXIMUM INTENSITY IN PATCH WAS     227
MINIMUM INTENSITY IN PATCH WAS       0
MEAN WAS         86.133
STANDARD DEVIATION IS    59.837649
RMS VALUE IS   104.830823
```

Figure 3.4 IMAGE_DAT2

```
PATCH CENTER   X =   96   Y =  131
 216 219 209 217 223 218 206 215 211 215 209 204 188 186 197  54   0   6  11
 213 218 213 218 215 220 210 211 222 211 203 209 168 204 144  84   0   0   0
 189 208 222 218 213 216 218 214 213 214 204 186 202 173  48  11  65  33   1
 183 209 206 211 210 213 207 207 213 206 187 187 197  71  85  57  74  85  84
 128 160 183 202 210 216 211 209 214 194 166 202 111  18   0  13  86  78 103
 115 130 165 189 190 203 216 210 203 179 203 168  45   1  83   0   0  62  58
 149 123 147 175 183 201 186 200 186 200 209  91  80  12  39  29  19   0   9
 152 156 153 122 133 180 182 178 178 201  90   3  36  36  41  46  53  41  10
 151 154 137 136 125 123 128 170 204 178  17  12   4  29  47  80  50  50  17
 165 159 153 130 128 114  94 121 175 125 107 111 115  94  93  94  95  86  38
 152 150 140 137 110  95 100 105 107 101 116 117 135 110 129 183 165 114 131
 154 136 150 119 111 104 102 109 115 121 111 136 140 135 144 144 151 163 167
 138 142 151 126 140 131 118 119 111 124 129 133 138 158 146 149 149 160 179
 156 156 151 139 161 145 116 129 107 129 136 149 131 152 167 160 160 173 183
 149 149 156 137 147 146 122 130 145 144 144 144 147 160 159 168 167 183 167
 175 160 150 160 145 147 139 137 163 136 139 156 163 167 164 170 176 186 181
 166 158 160 163 159 166 154 149 150 161 160 150 160 160 181 179 175 179 186
 179 173 157 176 167 160 151 149 147 159 160 173 163 158 185 176 183 189 183
 176 185 178 206 165 163 158 167 171 170 175 185 174 192 192 185 176 189 194
MAXIMUM INTENSITY IN PATCH WAS    223
MINIMUM INTENSITY IN PATCH WAS      0
MEAN WAS     142.274
STANDARD DEVIATION IS    58.811097
RMS VALUE IS   153.843114
```

Figure 3.5 IMAGE_DAT3.

```
PATCH CENTER   X =  412   Y =  380
 243 239 242 236 223 221 232 213 206 181 176 164 174 189 181 207 203 211 200
 243 239 233 227 217 220 214 210 203 174 173 182 176 178 209 206 203 200 202
 245 239 232 232 215 236 217 188 181 178 173 180 166 179 209 213 208 207 203
 239 231 230 229 229 236 232 207 173 174 186 198 206 218 214 217 218 218 216
 236 236 223 231 232 228 215 216 175 173 178 178 210 208 211 207 202 214 210
 232 226 225 229 224 218 195 186 174 176 179 204 207 207 208 203 210 218 207
 235 230 220 221 214 211 183 181 174 187 196 222 206 211 211 221 218 215 214
 230 232 225 216 197 207 190 178 187 196 215 214 211 217 215 214 206 213 214
 232 227 220 211 206 210 208 190 197 190 215 223 218 211 214 209 209 207 215
 231 225 227 216 214 199 186 197 206 194 183 211 214 209 217 218 211 216 211
 225 227 234 229 215 196 186 178 208 199 181 178 203 211 217 206 207 209 204
 214 221 225 238 210 206 213 206 193 200 190 173 176 189 210 208 208 218 204
 214 209 218 228 222 208 201 209 206 197 178 175 185 178 183 204 204 213 204
 214 210 214 207 216 220 207 199 201 211 185 160 171 170 167 167 208 208 206
 221 216 209 201 208 213 222 220 201 206 209 200 172 172 176 196 185 185 200
 210 209 213 207 204 197 197 220 224 203 199 203 204 192 189 178 182 180 176
 209 216 209 207 200 209 202 199 211 222 213 210 203 204 193 185 182 198 193
 215 214 216 209 202 215 210 209 208 203 213 213 204 204 204 193 190 182 203
 216 227 213 206 200 204 216 209 210 197 199 211 220 206 201 195 194 178 193
MAXIMUM INTENSITY IN PATCH WAS    245
MINIMUM INTENSITY IN PATCH WAS    156
MEAN WAS        205.695
STANDARD DEVIATION IS    17.408428
RMS VALUE IS   206.423653
```

Figure 3.6 IMAGE_DAT4.

```
PATCH CENTER    X =  423    Y =  247
  31  65 102 110 132  91  97  94  93  88 139 189 108 119 136 181  85 138 118
  95 113  64  68  54  37  81  82 102 109 130  94  85 108 130  97 108  88  79
 169 157 144 123 112  98 129 147 168 159 173 156 164 178 175 138 173 158 150
 123  94  88  78  69  37  57  61  54  79 103 103 159 158 156 163 154 149 130
 129 110 125 130 158 123 121 149 139 145 122 119 137 181 183 180 165 139 145
  85 130 130 133 165 154 115 164 160 150 125 144 133 188 195 173 133 139 128
 118 105 106 144 195 190 129  97 133 145 172 156 136 181 200 190 159 154 166
 144  85  85 103 154 179 153 158 145 143 171 178 128 107 193 203 167 171 187
 131  83  74 103 189 190 176 153 150 133 140 158 156 110 165 196 183 163 153
  89  93 108 165 167 170 165 142 131 136 163 186 152 128 164 206 181 150 143
  60 142 186 180 150 147 154 147 138 158 163 176 158 118 147 186 173 153 163
  67 116 157 156 144 138 147 144 151 173 190 170 183 168 123 145 158 152  90
  66  81  94 121 137 124 136 136 157 178 189 164 161 161 128 102 152 150 116
  60  66  64  54 101  80 125 144 153 146 154 161 163 160 156 101  85 168 139
  39  51  46   8  60  62 100 105 117 131 138 149 168 165 143  78  18 129 119
  91  98 108  98  95 181 203 187 158 144 160 167 202 189 195 190 107 185 180
  71  61  79  74  89 154 185 168 157 137 128 121 153 140 150 117  83 128 142
  73  86 128 129 123 151 163 167 202 207 158 112 115 123 118  87  72 128 128
  89  91 142 159 152 156 181 179 186 192 160 138 116 111  86  85 104 121 189
MAXIMUM INTENSITY IN PATCH WAS    207
MINIMUM INTENSITY IN PATCH WAS      8
MEAN WAS        132.896
STANDARD DEVIATION IS     38.434360
RMS VALUE IS    138.328133
```

Figure 3.7 IMAGE_DAT5.

```
PATCH CENTER   X =  134   Y =   76
 210 217 220 217 217 208 214 209 203 193 203 208 206 182 176  57   0   0  10
 197 202 221 218 213 209 216 204 208 203 202 204 186 175 100   6   4   0   4
 178 204 213 221 207 206 214 211 204 207 194 178 183 137  16   0   0  11   0
 181 187 203 204 208 210 207 206 204 214 190 178 168  54   0   2   0   6   0
 190 192 187 188 196 211 202 210 206 211 192 178 100   4   3   2   0   0   5
 206 199 174 176 178 188 199 206 208 196 178 139  18   5   0   0   0   5   2
 214 204 189 192 166 172 183 189 183 176 168  37   0   0   0   0   5   0   0
 214 213 216 198 181 188 183 168 190 176  88   0   0   0   0   0  16  10   1
 217 220 211 214 202 194 181 180 185 133   6   0   0   0   0   0  11  25   0
 218 221 215 218 214 209 187 188 168  27   0   2   0   0   9   0   8   3   0
 210 213 214 216 211 214 202 196 108   2   0   0   0   1   0   4  11   0   6
 218 217 214 213 210 203 199 109   1  19   9   0   0   0   0   0   0   4   5
 214 213 214 208 202 197 143  47  17   3  17  39   0   0   0   3   0   6  29
 210 214 206 206 203 176  55   0  10   2  62 114 103  86  12   0   1  30 111
 203 206 208 208 189 110   6   0   0   9  53 117  96  24  15   0  11 102 149
 203 211 194 187 142   2   0   0   0   0   0  18  46   9  15   0  58 143 144
 213 208 183 156  38   0   3   0   0   0   0   0   0   6   5  37 109 145 138
 203 183 176  76  10   0   0   0   0   6   0   0   0  13  15 109 140 143 181
 183 185 124   0   0   0  11   0   0   3   5   0   0   6  79 145 150 153 135
MAXIMUM INTENSITY IN PATCH WAS    221
MINIMUM INTENSITY IN PATCH WAS      0
MEAN WAS       104.336
STANDARD DEVIATION IS    92.882340
RMS VALUE IS   139.603287
```

Figure 3.8 IMAGE_DAT6.

```
PATCH CENTER   X = 279   Y = 345
 125 144 147 133 126 158 152 165 175 173 152  97  81  87  88 174 186 168 180
 130 140 138 137 146 137 146 174 173 164 158 153 135 149 158 173 181 175 187
 113 149 157 146 151 149 163 187 168 147 153 176 143 152 167 158 159 166 187
 104 139 135 167 152 170 166 164 150 161 152 140 146 160 178 164 164 160 157
  44 115 145 153 158 165 179 158 153 130 159 160 144 163 159 166 173 181 154
   8  23  57 163 168 159 202 175 140 145 150 156 154 146 163 157 185 159 170
   0  27  31  71 151 182 235 232 213 144  91 139 150 130 151 142 163 161 158
  15  36  32  23  30 100 236 242 238 128  87 107 148 149 144 143 142 160 183
  24  34  32  26  32  88 236 242 232  76  22 109 136 131 151 136 145 147 157
  18  37  52  45  44  38 196 241 227  48  31 125 125 132 181 130 123 149 163
  22  41  55  53  40  37  93 185 179  11  37 139 115 126 123 118 140 149 150
  34  47  59  33  46  82 128  93  38   8   3  54 111 112 121 110 131 123 123
 158  73  62  68  46  75 133 121  54  12  25  16  50 102 140 126 131 117 130
 196 123  86  40  33  22  44  46  32   0  12  24   8  24  83 112 126 110 125
 206 214 197 116  41  89   9  10  10   4  16   5   9   6  17  41 165 122 133
 216 221 223 201 140  34  10  16  10   6  23  24  18  16   9   5  19 111 144
 218 221 216 214 206 182  98  11   4   4  18  30  13  13  24   6  17  36  75
 215 211 211 218 211 204 208 182  76  15  13  38  17   9  26  10   6  37   9
 215 208 208 214 209 209 215 210 187 102  20  53  36  24  16   6  16  12  12
 MAXIMUM INTENSITY IN PATCH UAS    242
 MINIMUM INTENSITY IN PATCH UAS      0
 MEAN UAS      112.737
 STANDARD DEUIATION IS    68.652264
 RMS UALUE IS   131.945720
```

Figure 3.9 IMAGE_DAT7.

```
PATCH CENTER    X =  113    Y =  435
 172 215 209 222 228 232 231 231 229 230 224 222 231 228 228 227 230 231 227
 170 203 225 221 214 229 231 227 228 225 229 234 226 224 224 231 237 231 230
  61  94 200 228 231 228 230 229 229 231 228 224 230 231 235 230 225 230 236
  79 176 178 103 116 173 222 226 222 230 221 221 230 232 234 224 224 236 230
  96 220 206 118  96 100 199 224 218 222 228 223 234 222 227 232 232 234 235
  73 135 183 186 176 179 218 224 224 222 222 227 226 228 230 228 229 222 229
 138  78 109 179 223 227 231 227 222 225 227 230 228 225 228 226 234 229 228
 100 103  89 154 227 223 228 234 194 181 206 231 228 228 220 231 234 228 226
 197 157 203 199 190 203 209 223 135  62 103 208 227 223 224 223 235 228 224
 235 196 214 229 230 224 228 153 102  59  65  85 156 196 210 222 230 224 216
 222 224 200 222 228 238 227 215 145 119  75  81  81 107 200 197 208 189 147
 218 223 217 200 228 236 192 136 153 133 119  76  86  62 181 213 227 218 193
 224 223 215 228 221 189 110 148 110 150 132 128  79 167  62  85 171 221 160
 216 221 224 218 227 185 157  83  96 187 139  91 171 117  86  60  58 136 218
 232 236 241 236 237 241 241 227 225 229 206 131 187 179 151 167 128 125 161
 215 223 218 225 222 223 229 232 225 236 175  69  60 132 137  98  86  59  54
 224 223 217 221 228 230 223 226 231 215 107 114  51  65  90 122 108  91  50
 232 223 143  91 130 208 230 229 229 215 188 140 119 114  67 118 124 129 119
 216 190 111  75  68  68 147 223 231 221 216 214 186 119 117  73  72 123 116
MAXIMUM INTENSITY IN PATCH WAS    241
MINIMUM INTENSITY IN PATCH WAS     50
MEAN WAS      185.551
STANDARD DEVIATION IS    56.065817
RMS VALUE IS   193.814173
```

Figure 3.10 IMAGE_DAT8.

```
PATCH CENTER   X •  321   Y •  346
 227 222 227 224 229 223 232 227 228 229 224 224 216 210 206 201 215 221 210
 222 225 228 228 234 238 231 229 228 232 234 224 231 218 213 206 199 214 216
 222 222 224 238 232 237 235 237 236 239 237 228 232 237 222 222 208 200 214
 203 217 227 234 201 236 234 230 230 232 232 231 231 232 232 228 216 215 208
 207 208 218 235 231 237 231 214 218 201 211 228 223 237 230 238 223 216 225
 233 207 214 229 230 235 225 185 223 175 181 181 197 218 231 232 230 230 222
 189 195 200 187 206 222 218 225 220 199 192 176 187 192 197 215 231 231 232
 201 186 185 193 190 182 175 182 195 192 180 176 178 188 189 206 215 220 228
 181 183 195 178 217 182 145 181 165 157 183 187 196 182 188 187 193 217 230
 157 178 178 209 201 110  68 108 157 140 184 172 175 194 174 178 188 186 218
 170 174 178 220 152 133 138 112  44  82 181 159 182 182 187 175 175 194 189
 166 174 168 181 126 174 165 114  83  37  54  96 183 179 171 176 187 172 165
 182 181 185 190 167 164 158 128 131 121  78  57 105 160 183 176 190 183 178
 158 158 152 182 186 152 142 109 138 112  91  65  39  68 153 183 178 172 176
 181 175 150 154 160 168 161 164 157  95 100  33  45  97 178 170 183 185 178
 170 152 163 154 168 170 170 151 144 156 152 110 149 157 168 175 159 161 181
 189 188 163 165 170 172 175 170 157 187 187 195 188 179 168 172 164 160 165
 181 190 193 156 178 168 168 171 183 195 195 186 176 171 181 154 168 166 185
 179 174 182 187 183 185 166 167 192 194 196 187 172 164 167 171 183 160 164
MAXIMUM INTENSITY IN PATCH WAS    239
MINIMUM INTENSITY IN PATCH WAS     33
MEAN WAS       184.233
STANDARD DEVIATION IS     40.307997
RMS VALUE IS   188.578676
```

Figure 3.11 IMAGE_DAT9.

to gain some practical experience. The first problem was to design a digital image suitable for processing by the algorithm. An array of dimensions 8 by 8 was chosen. The values inserted in the array were arbitrarily chosen and never greater in magnitude than eight. The numbers in column five were all set equal to five to simulate a linear feature.

The algorithm to search this scene is shown in Figures 3.12 and 3.13. The program scans the search array from left to right and from top to bottom. The correlation is computed and printed for each location in the search area. When the correlation is completed and printed for the last location, the program execution is stopped. The program used two loops to perform the horizontal and vertical scanning action. The image data must have been stored in the memory prior to program execution. Both the data and the program were stored on magnetic cards for ease of entry. The TI-59 program was written in 191 steps and used 86 data registers. It computed a matrix of correlation values in less than 15 minutes. Correlations were computed at (m-n+1)*(j-k+1) locations; where m and j were the row and column dimensions of the search area, and n,k the row and column dimensions of the reference scene.

3.3.2 <u>An Advanced Algorithm</u>. Since the initial tests using the TI-59 were generally successful, an advanced algorithm was written for the TI-99/4A. The 99/4A uses a higher-level language called BASIC. The algorithm for executing image registration is described and listed in Appendix A. Figure 2.1 describes the search space $\underline{S}$ in which the window $\underline{W}$ is to be translated to determine the address of the matching scene. Note that there is an allowable range of addresses u,v that the

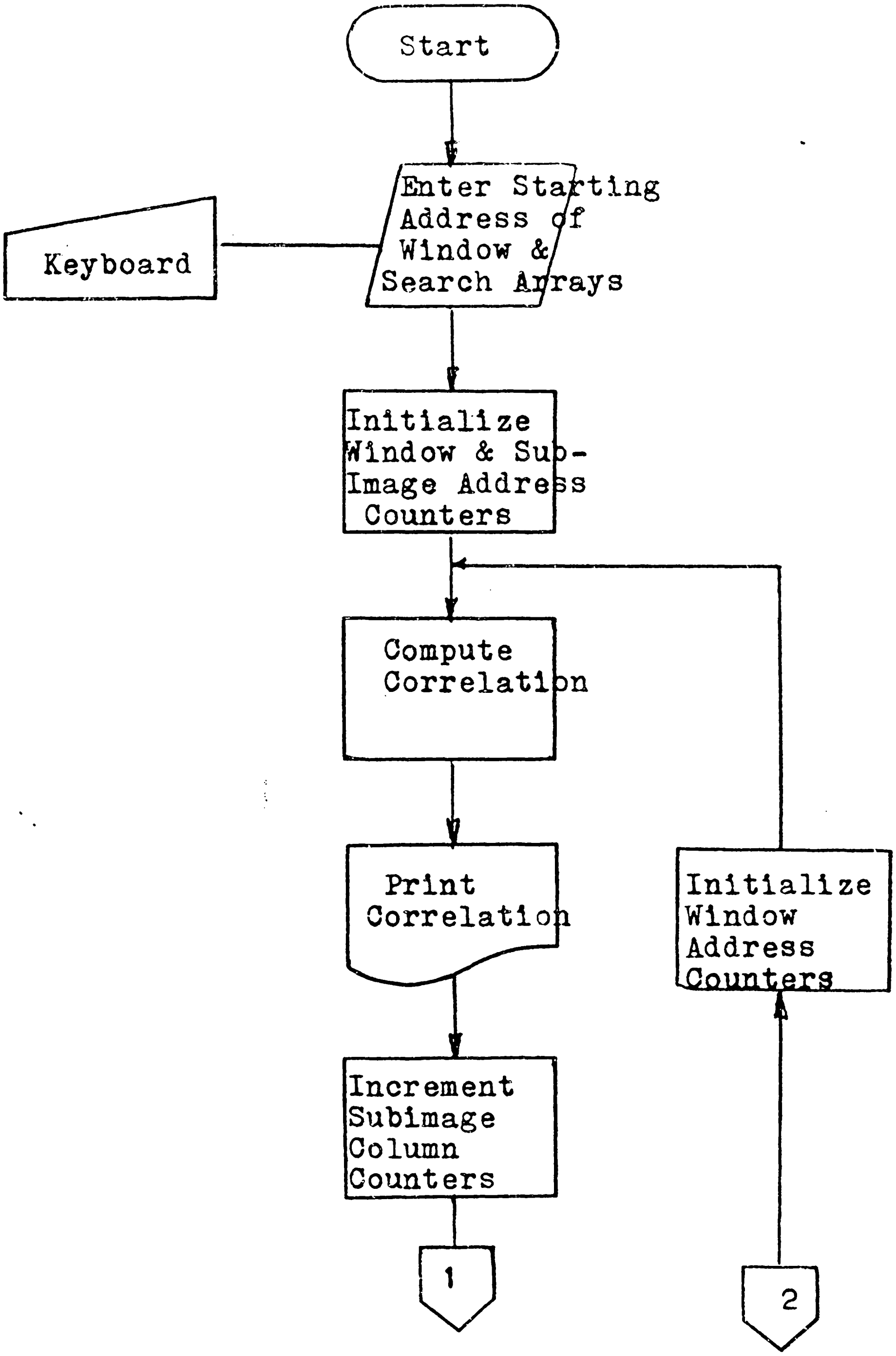


Figure 3.12 TI-59 Correlation Program.

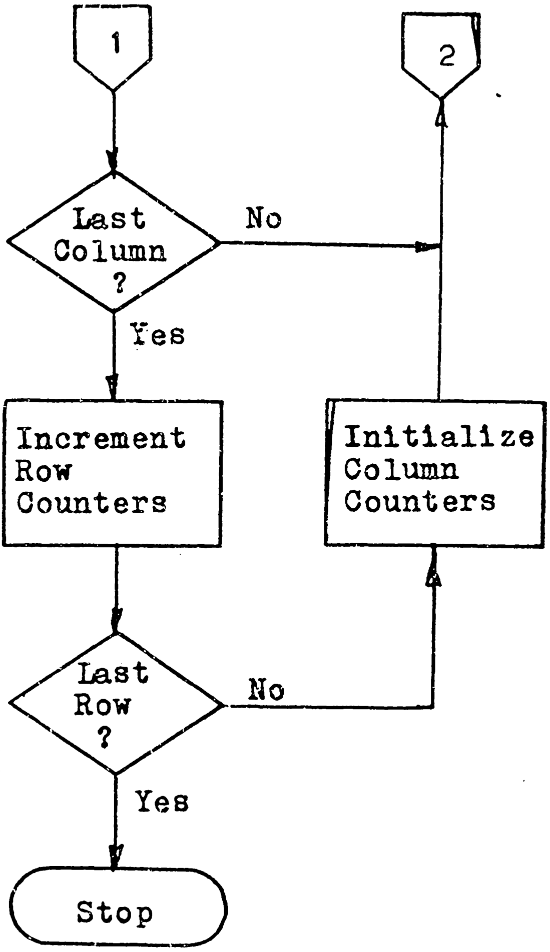


Figure 3.13 TI-59 Correlation Program.

window is allowed to translate through in order to stay within the search area S. The dimensions of the window or reference scene are M x M. Those of the search scene are L x L. As a result, there are $(L-M+1)^2$ subimages for which the correlation must be computed. If the search area S is 512 x 512 pixels in size and the window W is 32 x 32 pixels, then the correlation function must be computed 231,361 times. For the size of data sets used in this experiment (S(19,19), W(8,8)), the correlation was computed 144 times for each run. Needless to say, correlation computation is expensive for large data sets. Dr. William Pratt (15) has shown that for windows 32 x 32 pixels in a search area of 128 x 128 pixels that $>10^7$ multiplications are required in the computation of a statistical correlation measure.

The program is designed to first load a 19 x 19 digital image array into core for later processing. The processing includes computing an image mean pixel value, normalizing the image to have a mean of zero, selecting a reference window and then computing the correlation for all translations in the search area.

Since the time a correlation algorithm takes to scan a search image is dependent on the size of the search area and not on the scene, only one image (IMAGE_DAT5) was used to determine the execution time of the correlation algorithm. The execution time is relatively long, making it possible to time with a stopwatch. The timing was started after the initialization of the program for such things as name of digital image file, search area dimensions, reference dimensions, and the location of the reference scene in the search area. Timing was stopped when the algorithm completed execution as indicated on the CRT. The correlation at each subimage location was printed by the system printer.

To test the image for the effects of noise, a randomly-generated value was added to the pixel values. The randomly-generated values had a mean of zero and a standard deviation of six. The digital image used was IMAGE_DAT5. This image was selected because of distinct linear features visible in the displayed image.

The correlation algorithm was also executed using IMAGE_DAT7. This image was selected because of distinct linear features visible in the displayed image.

3.3.3 Sub-Pixel Accuracies. The following method was used to demonstrate some of the polynomials that could be used and to evaluate their advantages and disadvantages.

The mathematical method used in this part of this experiment to fit a curve to the correlation data is that of simultaneous equations. A set of simultaneous equations can be solved in matrix form in the following way:

$$X\ A = B \tag{3.1}$$

where X is an n x n matrix of the row correlation locations, A is an n x 1 column vector of unknown curve coefficients with n equal to the number of coefficients to be solved, and B is an n x 1 column vector of observed correlation values. The solution is

$$A = X^{-1}B \tag{3.2}$$

where X^{-1} is the inverse of X.

Three different polynomials were fit to the correlation data of IMAGE_DAT5. They were the following:

$$R(x,y) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 , \tag{3.3}$$

$$R(x,y) = a_0 + a_1x + a_2x^2 + a_3x^3 , \tag{3.4}$$

$$R(x,y) = a_0 + a_1x + a_2x^2 . \tag{3.5}$$

One item of great concern is the location of the maximum value of the fitted curve to that of the true maximum. To determine this, the first derivative of each curve was taken and set equal to zero as follows:

$$dR(x,y)/dx = a_1 + 2a_2x + 3a_3x^2 + 4a_4x^3 = 0 \tag{3.6}$$

$$dR(x,y)/dx = a_1 + 2a_2x + 3a_3x^2 = 0 \tag{3.7}$$

$$dR(x,y)/dx = a_1 + 2a_2x = 0. \tag{3.8}$$

Equations (3.6) and (3.7) were solved using a Master Library function of the TI-59 calculator called "Zeros of Functions". The program uses the graphical bisection method to determine the roots of a function, defined by the user, within a predetermined set of limits. The program is especially useful for solving equations like (3.6) and (3.7) where the region of the maximum is approximately known.

The data set used in this experiment was from row 9 of the correlation matrix shown in Figure 4.5. Similarly, a curve could be fitted to column 5 to determine a continuous correlation function in that direction.

3.4 SEQUENTIAL SIMILARITY DETECTION ALGORITHM

The program used to test the SSDA technique is shown in Appendix B. The program operates in much the same way that the correlation algorithm does in that it asks for the name of an image data file. It then asks for a window address which it uses to fill the reference window array. It then generates a non-repeating sequence of random numbers. The algorithm uses this set to compare pixels at random when comparing the subimages of the search area to the reference scene. The program searches the entire array only displaying those sums that are less than the threshold. The program also asks the user to set the threshold. At the completion of the search, "Search complete" is displayed and a listing is provided of the number of compares made at each subimage location.

The time comparison test was conducted in much the same manner as in the correlation experiment. The image used was the same, IMAGE_DAT5. The address of the reference scene used was the same, (6,6). The stopwatch was started at the display "SEARCHING!" on the CRT. The stopwatch was stopped at the display "SEARCH COMPLETE" on the CRT.

After the completion of the search, the program prints the frequency distribution of comparisons. This was repeated for nine different threshold settings, (10 through 250) and on two images, IMAGE_DAT5 and IMAGE_DAT7.

To demonstrate the error, "e" accumulates as in equation (2.5) and Figure 2.2. The program was modified slightly to display the accumulated error after each pixel pair comparison. This data was then used to generate the graph.

Finally, to determine the effect that the threshold setting had on the execution time of the algorithm, the threshold was varied from a setting of 10 to 250 while the execution time was checked with a stopwatch. The experiment tested two images, IMAGE_DAT5 and IMAGE_DAT7.

4. RESULTS

4.1 CORRELATION RESULTS AND ANALYSIS

4.1.1 Initial Correlation Algorithm. The correlation data output by the initial correlation algorithm is listed in Figure 4.1. The search area was an 8 x 8 array of pixels and the reference scene, a 3 x 3 array, was selected from the 8 x 8 array. The number of correlations computed were 36 $(8-3+1)^2 = 36)$. The reference scene starting address was 1,4. That is, the first pixel of the 3 x 3 array came from address 1,4 in the search area. The last pixel came from address 3,6. The TI-59 program completed the search in just under 15 minutes. The results of row 1 of the correlation matrix are shown in Figure 4.2.

The algorithm for searching an image was able to detect the match or register the reference scene in the search area. A perfect correlation of +1 is present at location 1,4 of the correlation array. However, unexpectedly a high correlation of +0.96 was found at location 4,4. A review of the digital image data set indicated a very similar 3 x 3 matrix of pixel magnitude was present at that location. The result indicates that a high correlation is not necessarily a sufficient condition for registration.

One of the first concerns was how sharp the correlation peak would be and therefore how clearly the point of registration would be determined. Figure 4.2 shows a plot of the correlation as a function of column number for row 1 of the correlation matrix. The correlation drops to 0.68 at column 3 and 0.11 at column 5. This provides a peak to off-peak ratio of at least 1.5 to 1. However, the same condition is not true for column 4, refer to Figure 4.1. At location 4,4 a second peak

-0.04	0.12	0.68	1.0	0.11	-0.37
-0.67	-0.6	0.51	0.89	0.12	-0.36
-0.59	-0.57	0.66	0.88	0.18	-0.26
-0.66	-0.44	0.58	0.96	0.14	-0.27
-0.58	-0.58	0.54	0.79	-0.02	-0.31
-0.68	-0.68	0.42	0.66	0.04	-0.05

Figure 4.1 Correlation Array.

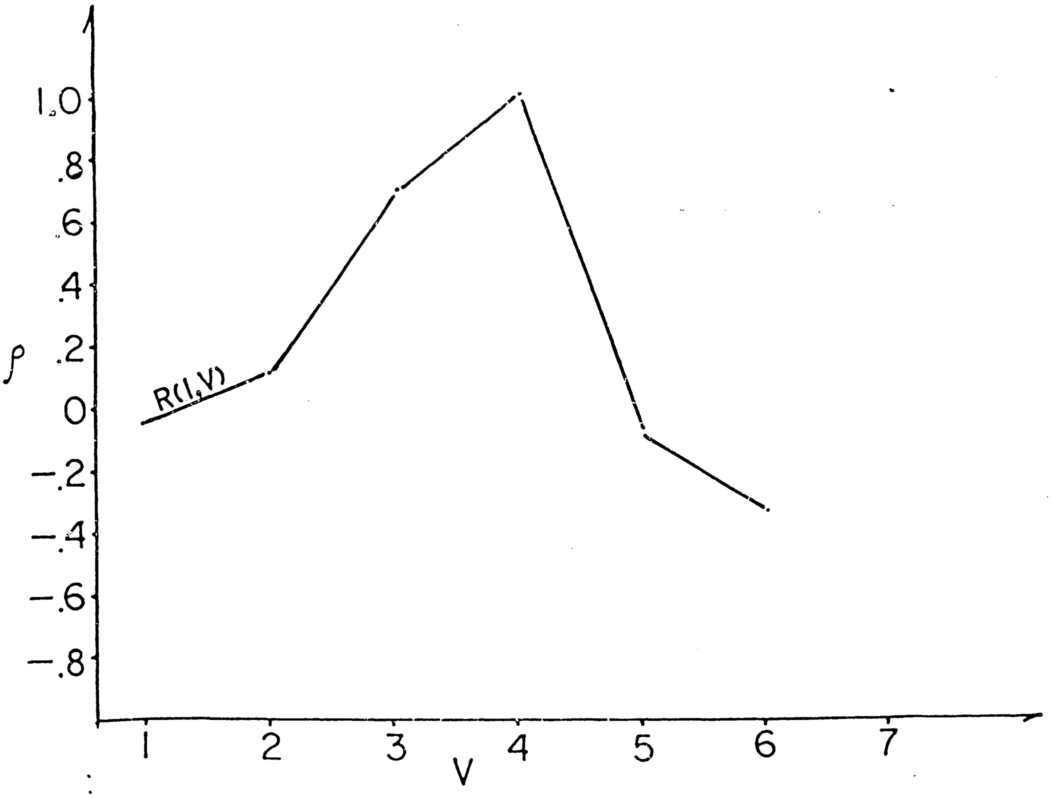


Figure 4.2 Correlation versus Column Number.

(0.96) is found indicating another point of possible registration. Of course, location 1,4 is still the point of highest registration or correlation; and since it was defined that way, there is a tendency to believe that this will not be a problem. However, real images have noise and several ground features in an image may appear to be the same. Therefore, the correlation peak will be much harder to distinguish then it was in this experiment. Add to this the possibility of several image features of similar shape and reflectivity, and it becomes quite possible to select the wrong features as a match. For example, if the control point image was in the top of a storage tank which was part of a tank farm containing a number of tanks of the same size, the number of high correlations would equal the number of tanks.

The initial experiment with the TI-59 provided the experience with algorithm design and analysis. There are two major parts of the program to perform image registration. They are the reference scene scanner and the correlation computation routine. The scanner was designed to move the reference scene both horizontally and vertically through the search area. It starts the scan in the upper left corner and stops it in the lower right corner. At each location, the correlation routine computes the correlation using the overlapping pixel data.

4.1.2 Advanced Correlation Algorithm. The correlation array from a search of the image "IMAGE_DAT5" is shown in Figure 4.3. The reference scene matched exactly. The correlation peak is sharp and clearly shows the correct location. Again the reference scene was chosen from the search array and demonstrates that the program is working correctly. Plots of the correlations for the row containing the match and surrounding

-.06	.18	.28	.33	.28	.23	.24	.18	.05	.04	.01	-.01
-.02	.23	.3	.26	.2	.15	.12	.17	.06	-.04	-.14	-.19
.07	.21	.23	.18	.17	.14	.02	.03	.07	.03	-.16	-.26
.18	.19	.19	.11	.03	.18	.11	-.02	-.14	-.07	-.09	-.2
.24	.23	.24	.27	.06	.04	.05	-.07	-.28	-.36	-.22	-.11
.29	.28	.27	.36	.35	.17	-.05	-.19	-.25	-.33	-.36	-.24
.12	.3	.39	.37	.33	.31	.15	-.19	-.28	-.22	-.29	-.32
.01	.09	.33	.69	.53	.26	.24	.15	-.12	-.29	-.3	-.21
.05	.03	.03	.54	$\boxed{1}$	.62	.3	.28	.3	.03	-.23	-.25
-.23	.08	.13	.14	.49	.78	.66	.46	.32	.29	.11	-.13
-.33	-.27	.11	.27	.27	.35	.6	.71	.54	.33	.27	.19
-.1	-.26	-.28	-.02	.27	.35	.3	.52	.73	.6	.39	.27

Figure 4.3 Correlation Array IMAGE_DAT5.

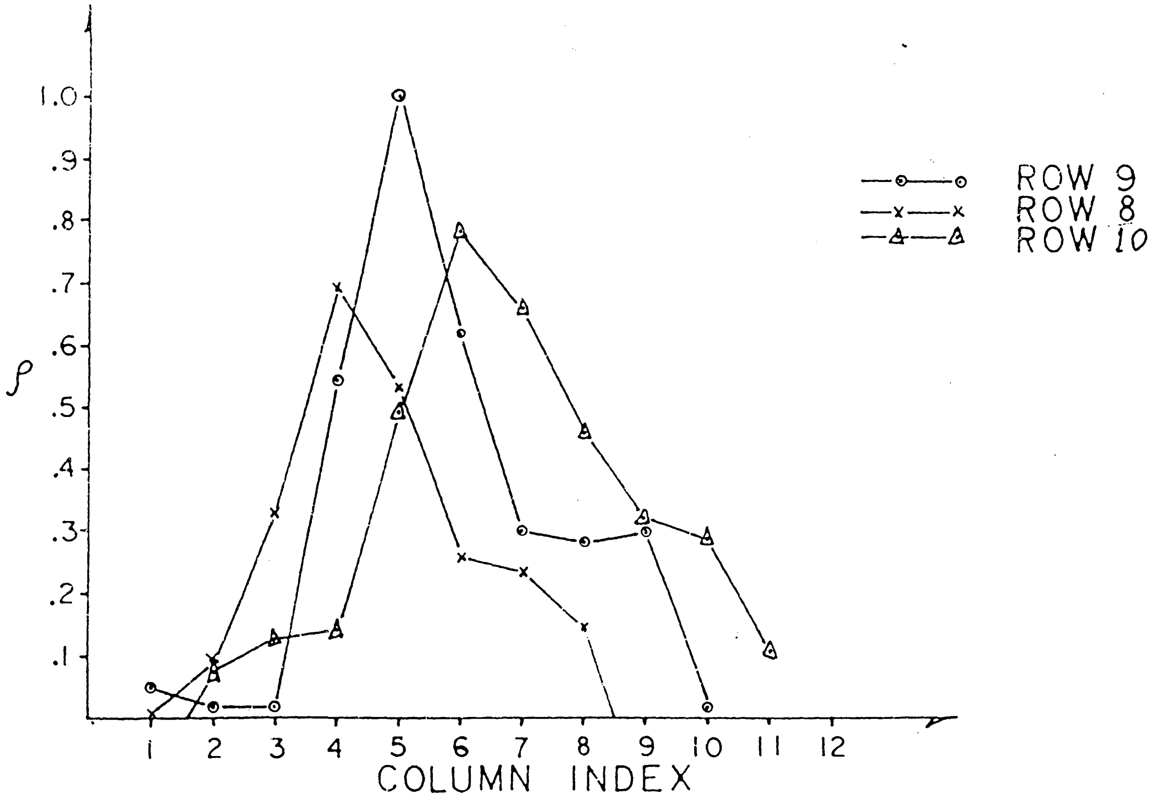


Figure 4.4 Correlation versus Column Number IMAGE_DAT5.

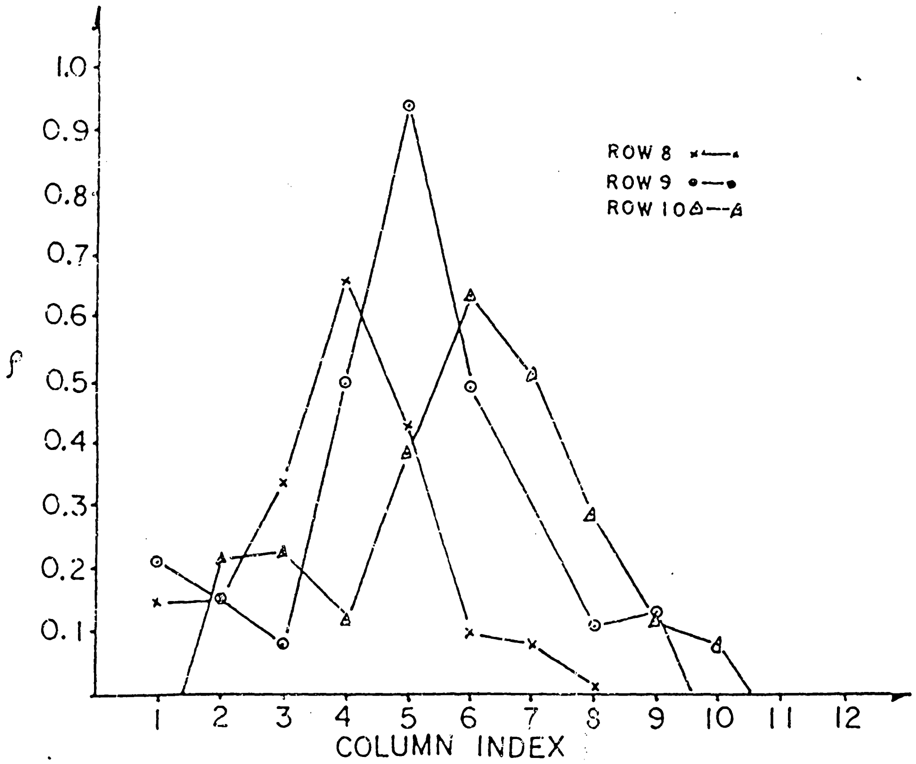


Figure 4.5 Correlation versus Column Index with Noise.

rows are shown in Figure 4.4. The plot contained in Figure 4.5 shows the effect of noise in the reference scene on the correlation peak. The noise was simulated by summing random integers from a number generator to the pixel values in the reference scene. The peak is still clearly discernible from the surrounding measures. The main effect appears to be an overall reduction in the maximum correlation values.

The correlation array for scene "IMAGE_DAT7" is shown in Figure 4.6. By chance, the reference scene was chosen from location 6,6 in the search area. When the program was executed, the match was found as expected (value surrounded by a square). However, several other near matches were found also. These values follow a straight line running diagonally from top-right to bottom-left. The chance selection of a reference scene from a linear feature in the image demonstrated again the problem of the correlation measure not necessarily determining the correct peak in the presence of similar features in the search area and that the correlation measure can be used to extract a feature from the image. The presence of this linear feature was confirmed by viewing the image on a CRT.

The inclusion of real image data sets and a higher level language program contributed greatly towards a more realistic experiment. The program in Appendix A could still be optimized to decrease its execution time. It is a part of the design of BASIC interpreters to always start at the top of the program to find a line number called for in a GOTO or GOSUB statement and then count up. Therefore, frequently called subroutines should be at the top of a program; those routines called once and initialization statements should be at the bottom of the program. Shorter variable names also speed execution.

.06	.06	.06	.08	.17	.32	.53	.75	.96	.94	.71	.46
.06	.07	.07	.14	.27	.46	.68	.9	.97	.79	.52	.31
.06	.07	.11	.22	.39	.6	.83	.98	.86	.6	.37	.19
.05	.08	.16	.32	.51	.75	.96	.93	.69	.44	.24	.1
.06	.13	.25	.44	.67	.91	.97	.77	.52	.31	.14	.03
.1	.21	.37	.59	.84	1	.85	.59	.37	.19	.06	0
.19	.32	.52	.75	.95	.89	.64	.39	.2	.07	.02	-.3
.29	.46	.68	.91	.92	.69	.42	.2	.07	.01	-.01	-.03
.42	.62	.86	.97	.77	.48	.25	.08	.01	0	-.01	-.07
.56	.8	.97	.88	.6	.32	.13	.02	0	0	-.04	-.15
.73	.95	.94	.72	.44	.19	.05	-.01	-.01	-.04	-.12	-.28
.9	.97	.79	.53	.28	.08	0	-.02	-.03	-.09	-.24	-.45

Figure 4.6 Correlation Array IMAGE_DAT7.

Figure 4.5 shows the effect of noise on the correlation peak. The random values added to the pixel values were less than 10 in magnitude. There is little effect on the determination of the registration location with noise added of this magnitude.

In the first part of this experiment, it was noted that more than one location of high correlation can occur. This is especially true of scenes containing repetitive features. Figure 4.6 demonstrates this very graphically. In this case, the algorithm is a very capable feature extractor.

4.1.3 Sub-pixel Accuracies. The third part of the correlation experiment demonstrates a method by which a continuous correlation function can be determined using the discrete correlation observations and thereby improve the resolution of the registration. Plots comparing the observed correlation values to continuous function values are shown in Figures 4.7, 4.8, and 4.9. The data set used in this problem is shown in Table 4.1. The coefficients determined for all of the polynomial fits are shown in Table 4.2.

The local maximums for the three functions were determined by first differentiating equations 3.3, 3.4, and 3.5, and then setting the result to zero. The locations for the maximums for the fourth, third, and second order polynomials respectively were as follows:

$$X = 5.0244$$

$$X = 5.1875$$

$$X = 5.046$$

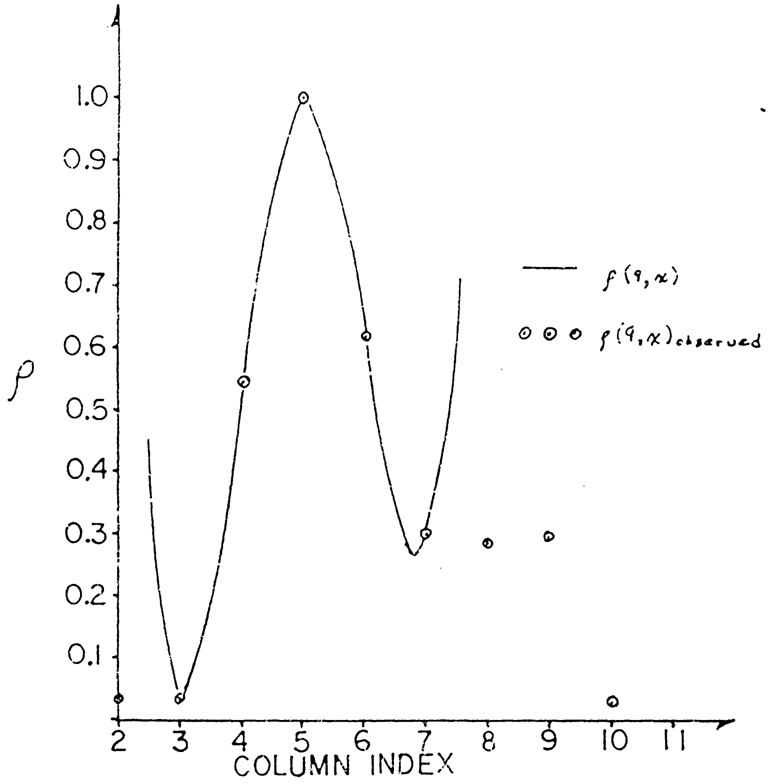


Figure 4.7 Fourth Order Polynomial Fit.

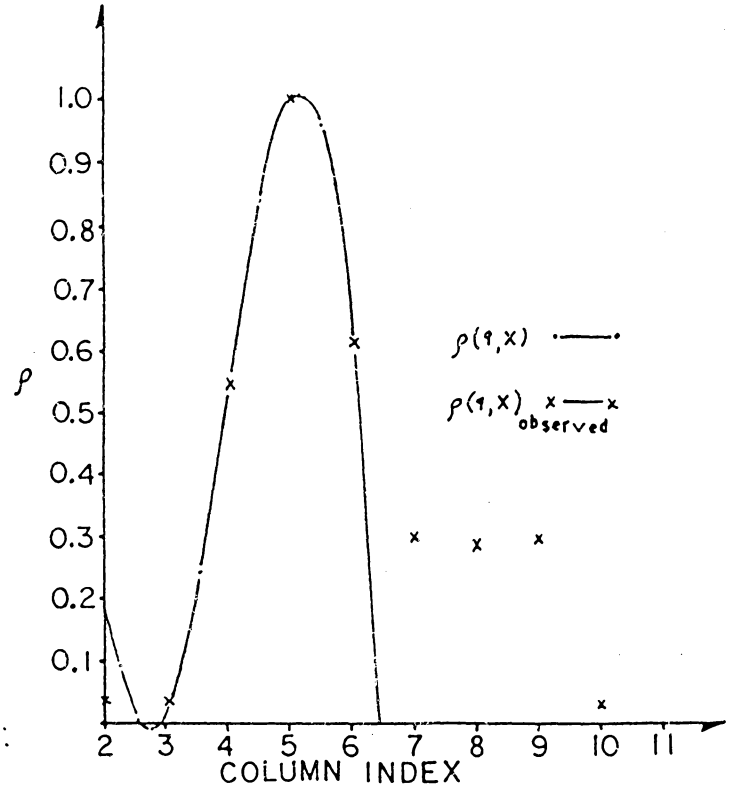


Figure 4.8 Third Order Polynomial Fit.

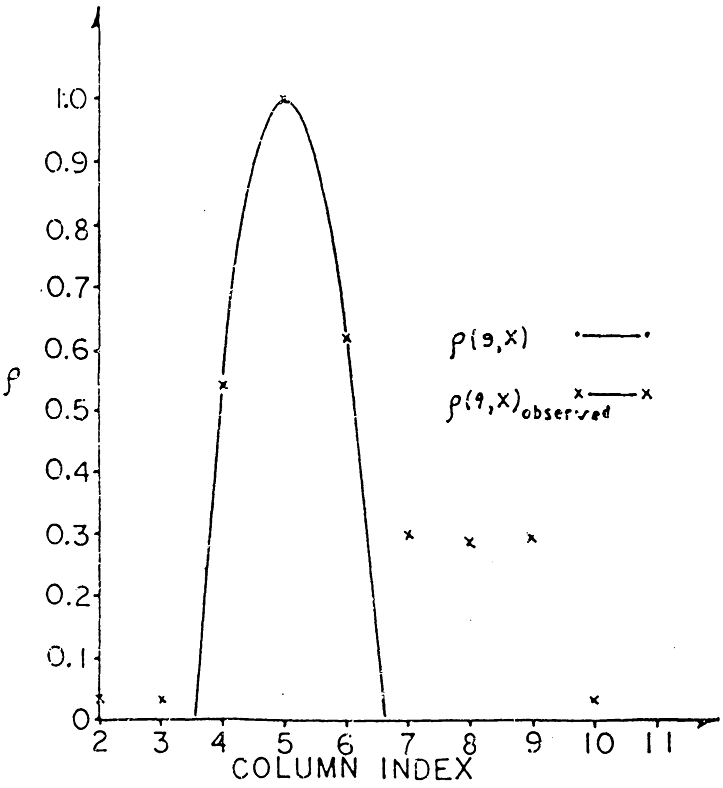


Figure 4.9 Second Order Polynomial Fit.

Table 4.1 Data Set.

Column Index	R(9,x)
1	0.0452
2	0.0341
3	0.0313
4	0.5423
5	1.0
6	0.619
7	0.2999
8	0.28
9	0.2987
10	0.0269
11	-0.2284
12	-0.2477

Table 4.2 Polynomial Coefficients.

1. Fourth Order Polynomial

$$A = \begin{bmatrix} 31.3225 \\ -29.48025 \\ 9.9039 \\ -1.3954 \\ 0.07025 \end{bmatrix}$$

2. Third Order Polynomial

$$A = \begin{bmatrix} 6.0325 \\ -5.45475 \\ 1.54415 \\ -0.1309 \end{bmatrix}$$

3. Second Order Polynomial

$$A = \begin{bmatrix} -9.6755 \\ 4.2319 \\ -0.4194 \end{bmatrix}$$

Since the location of the true maxima was 5 in all three cases, an error was introduced by the fit. If the differences (0.024, 0.188, 0.046) are translated into shift in micrometers, then the error from using the fourth order function would be the least, 3.6 micrometers. The error using the third order function, 28.2 micrometers; the second order function, 6.9 micrometers. Obviously some care must be exercised in using this method since the offset may vary from fit to fit. When designing an algorithm to use any of these computed functions, consideration must be given to the fact that the function is not valid outside the range of data used.

4.2 SSDA RESULTS AND ANALYSIS

The SSDA algorithm completes the search of the search area in much less time than correlation can. SSDA completes the search of IMAGE_DAT5 in 85 seconds when the threshold is set to 50. The error accumulation for a typical SSDA run is shown in Figure 4.10. The error accumulation for two subimage locations (7,8 and 7,5) are plotted with the threshold set at 50. The algorithm requires that the error must be greater than 50 before the comparison process is stopped. Therefore, the dotted line for location 7,5 did not exceed the threshold until four comparisons were made. The comparisons were terminated with the error e=145 for locations 7,5 and e=162 for locations 7,8. These plots are representative of most of the comparisons in this image.

The selection of the threshold affects the execution time of the algorithm and the number of pixels that are compared at each subimage location. Figures 4.11 through 4.28 show the frequency of occurrence,

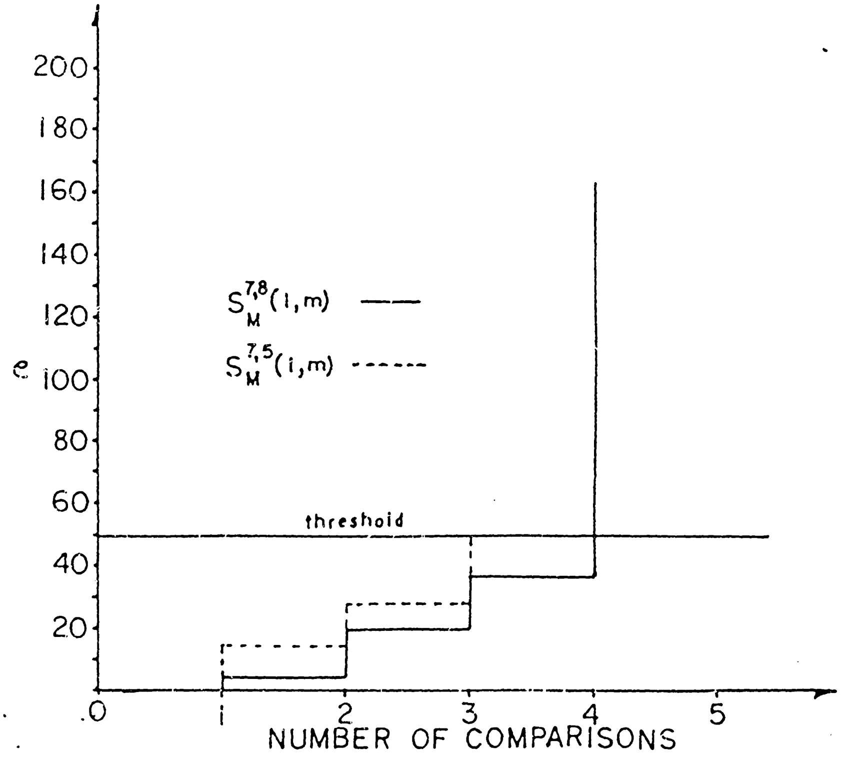


Figure 4.10 SSDA Error versus Number of Comparisons.

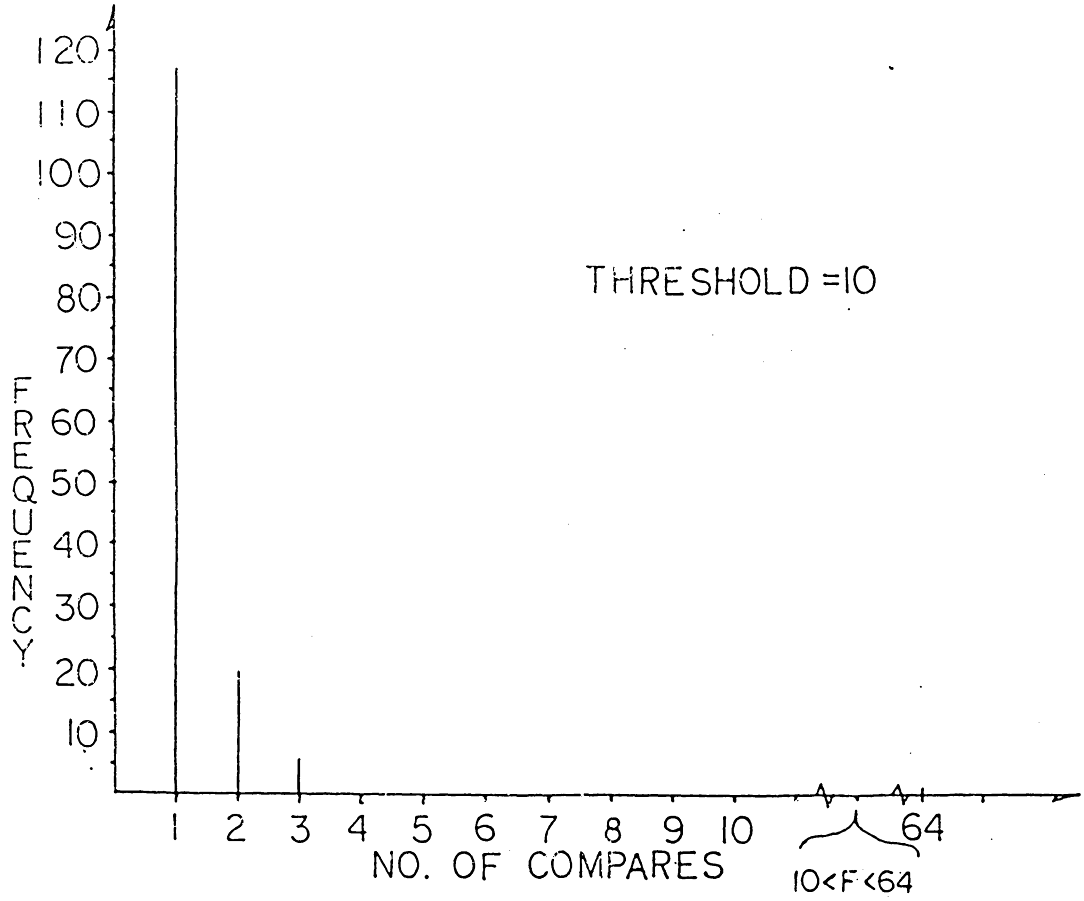


Figure 4.11 SSDA Comparison Histogram - IMAGE_DAT5.

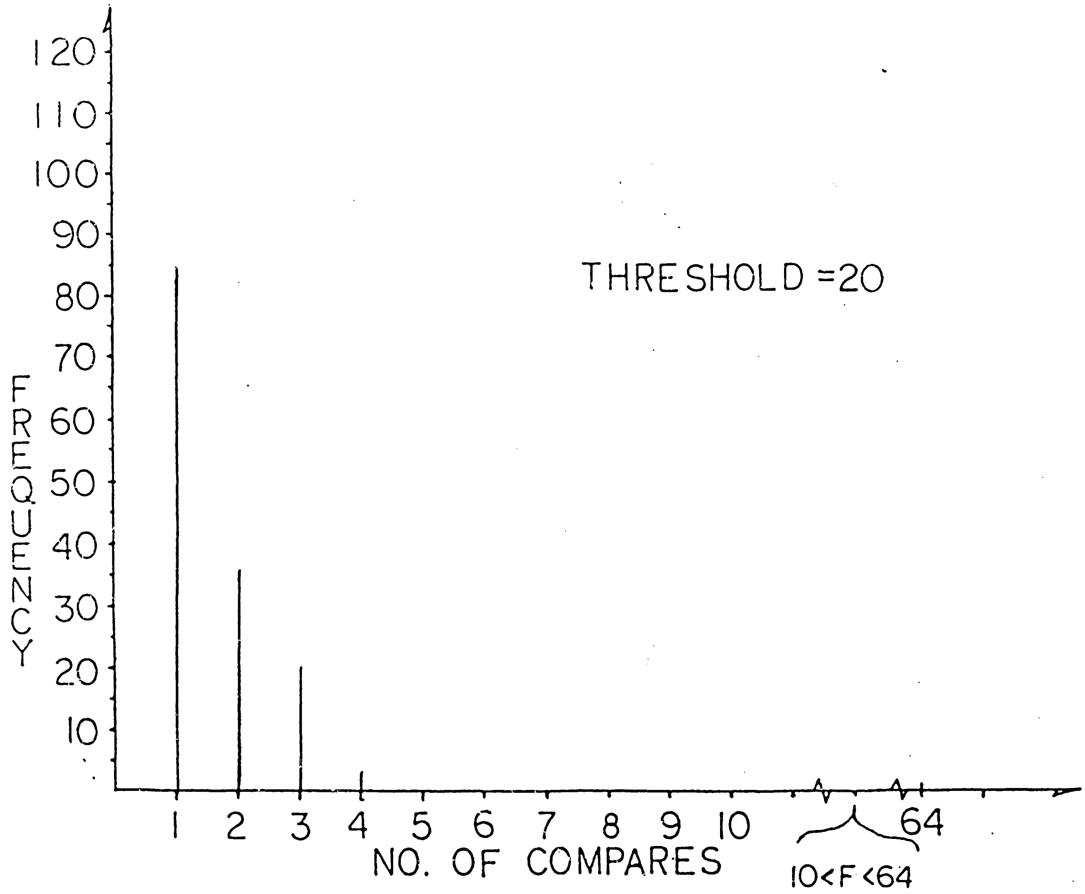


Figure 4.12 SSDA Comparison Histogram - IMAGE_DAT5.

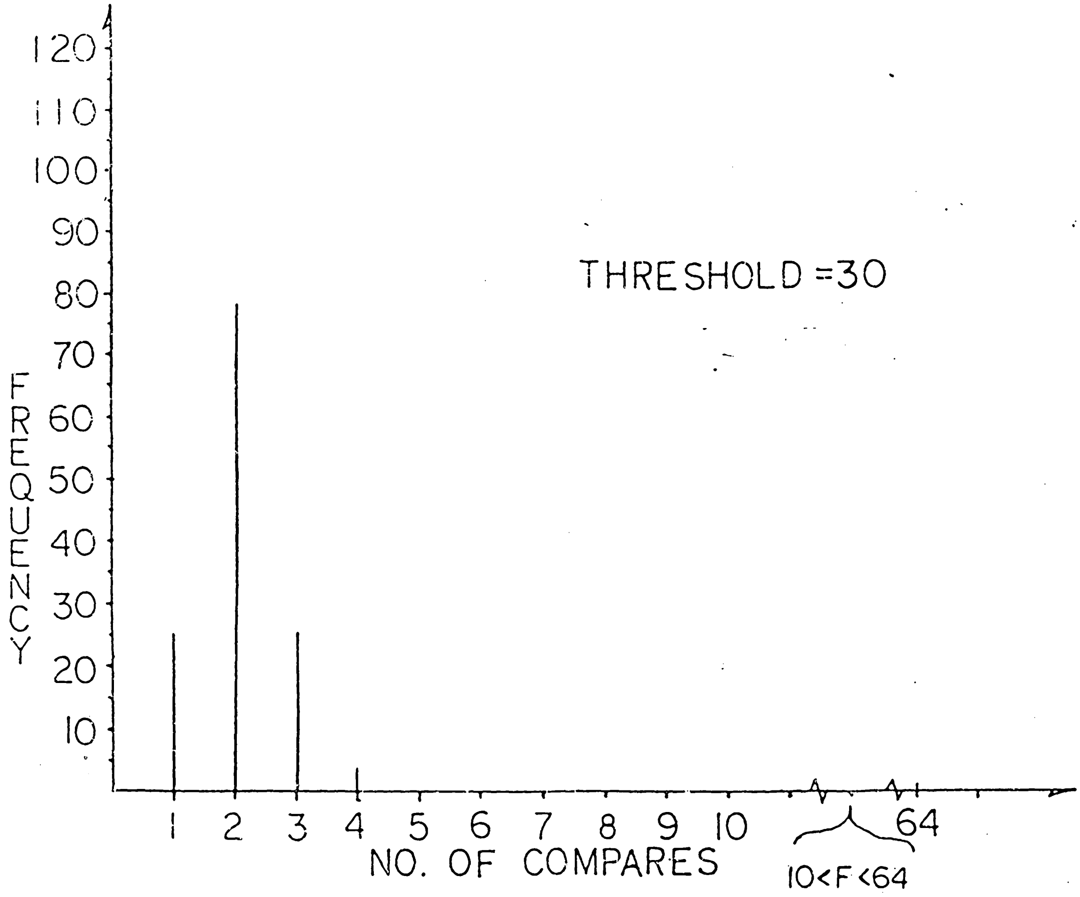


Figure 4.13 SSDA Comparison Histogram - IMAGE_DAT5.

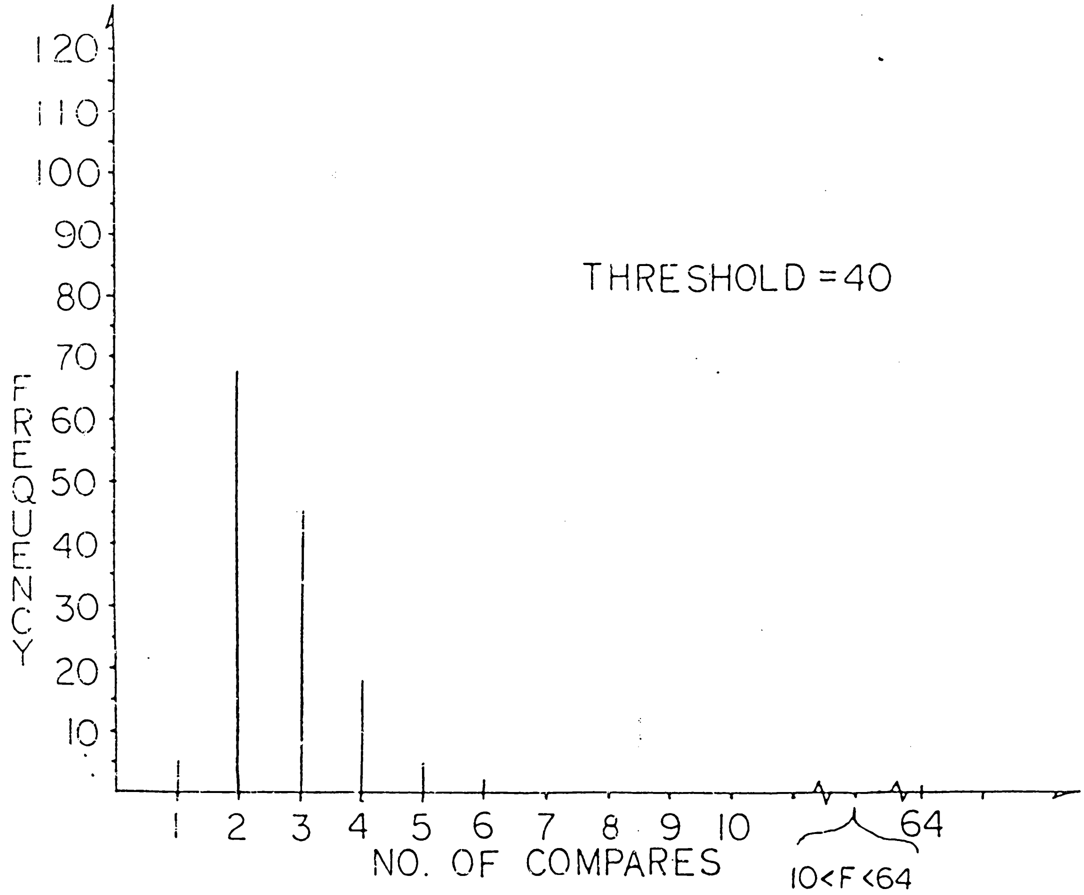


Figure 4.14 SSDA Comparison Histogram - IMAGE_DAT5.

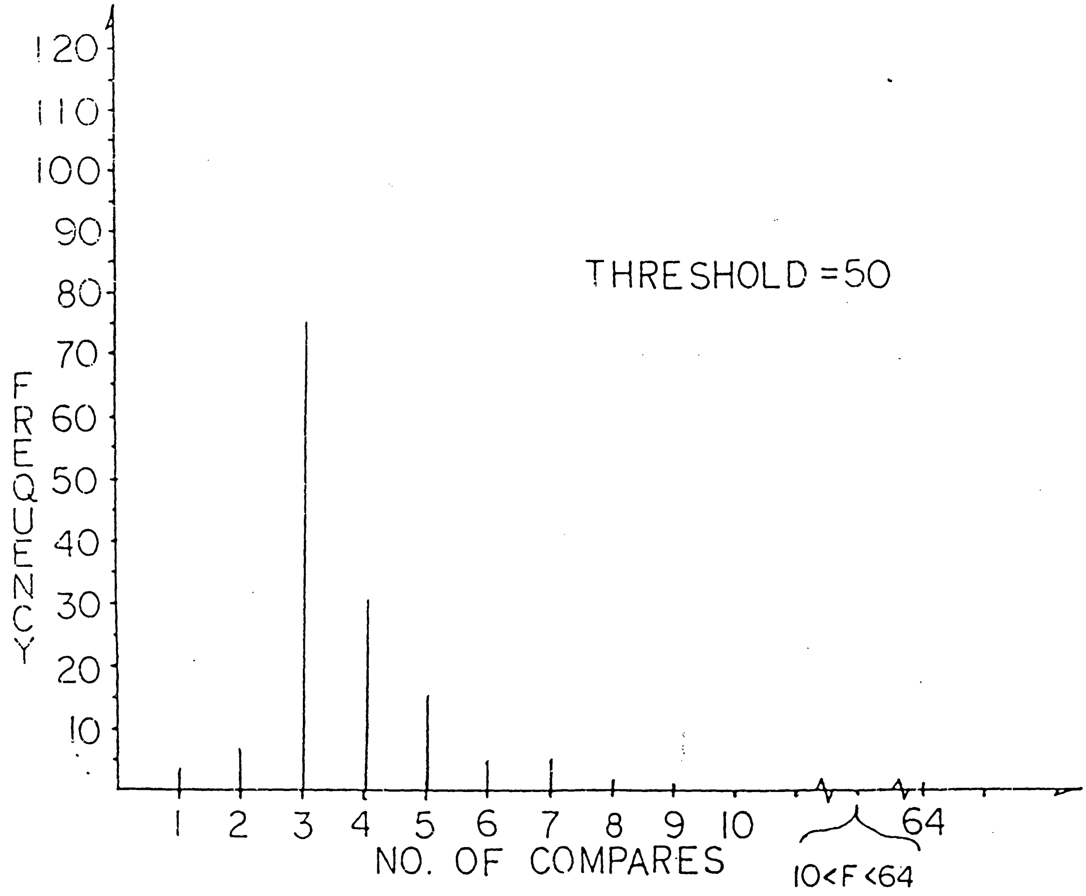


Figure 4.15 SSDA Comparison Histogram - IMAGE_DAT5.

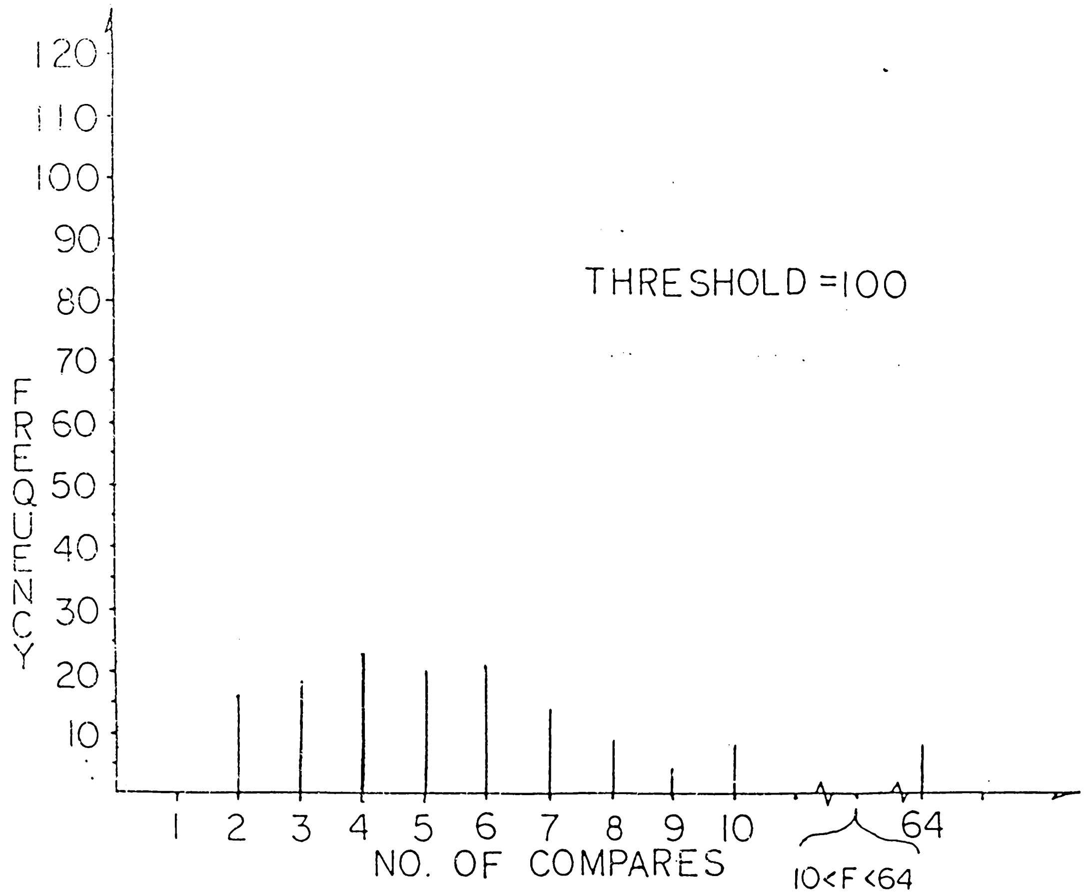


Figure 4.16 SSDA Comparison Histogram - IMAGE_DAT5.

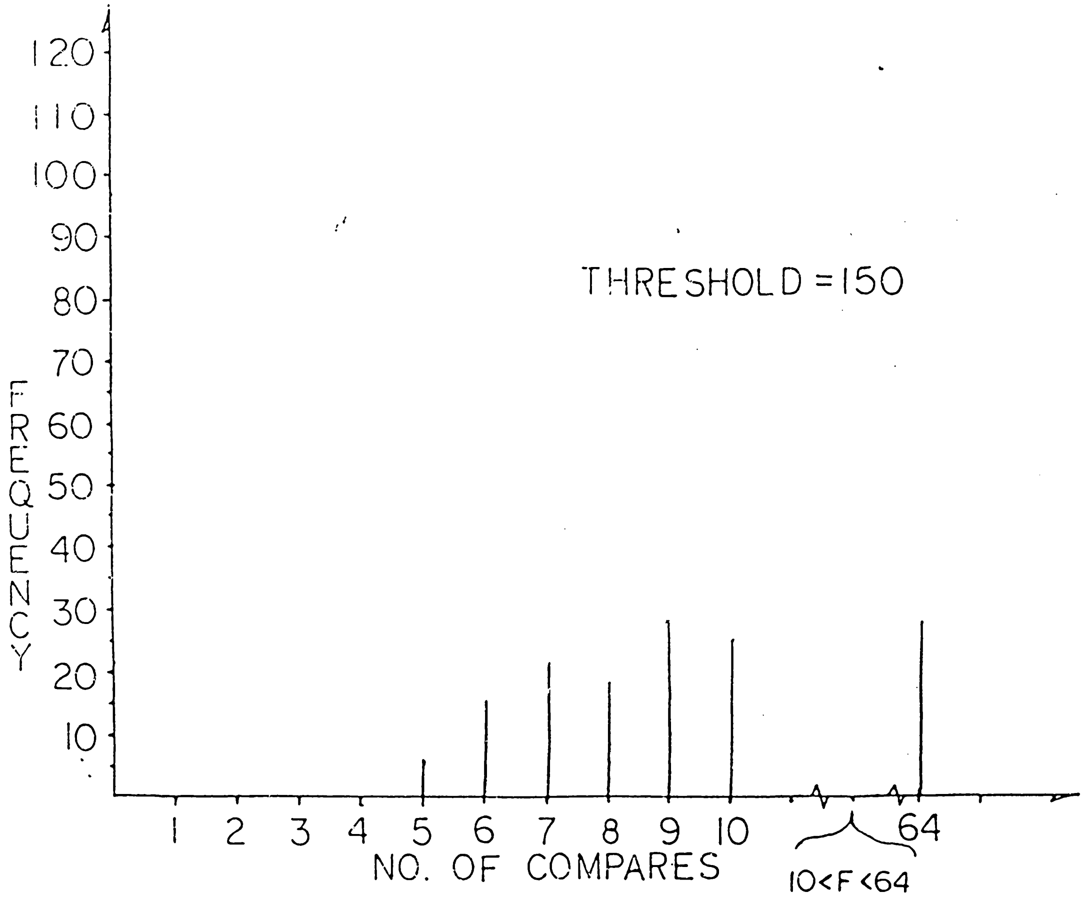


Figure 4.17 SSDA Comparison Histogram - IMAGE_DAT5.

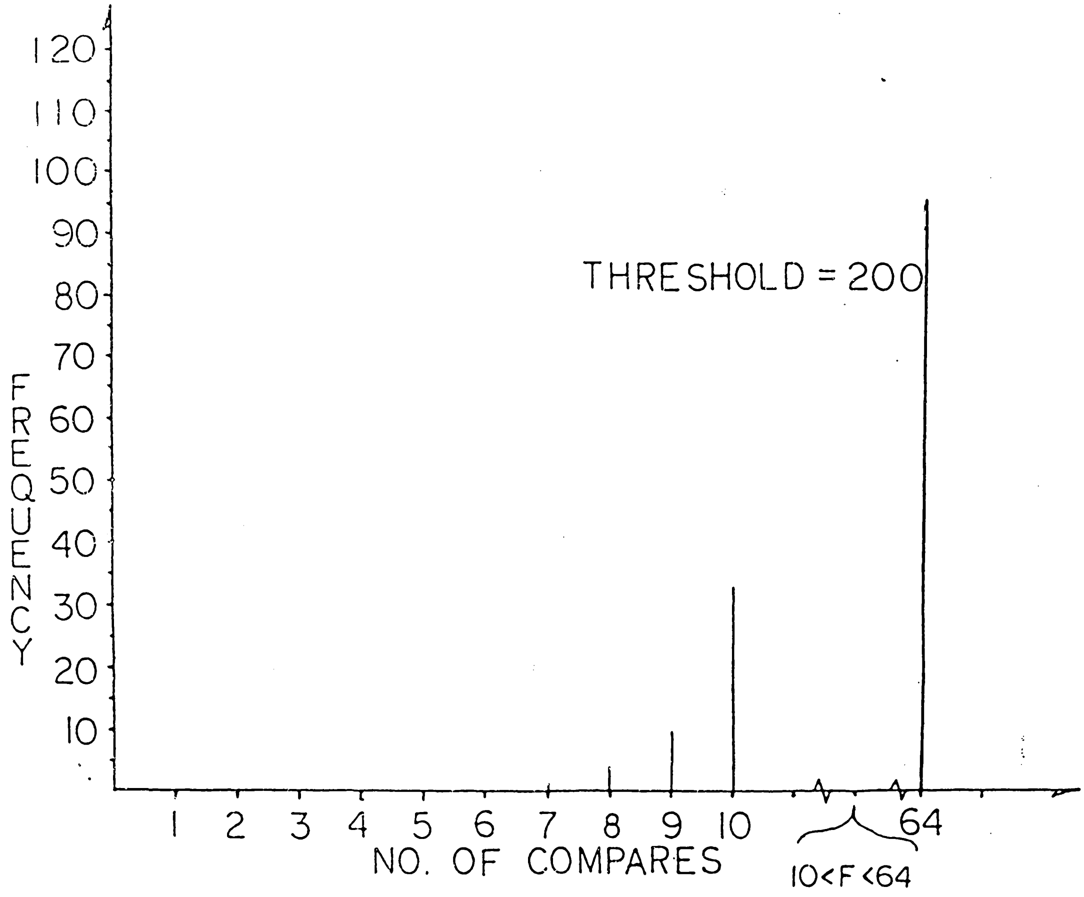


Figure 4.18 SSDA Comparison Histogram - IMAGE_DAT5.

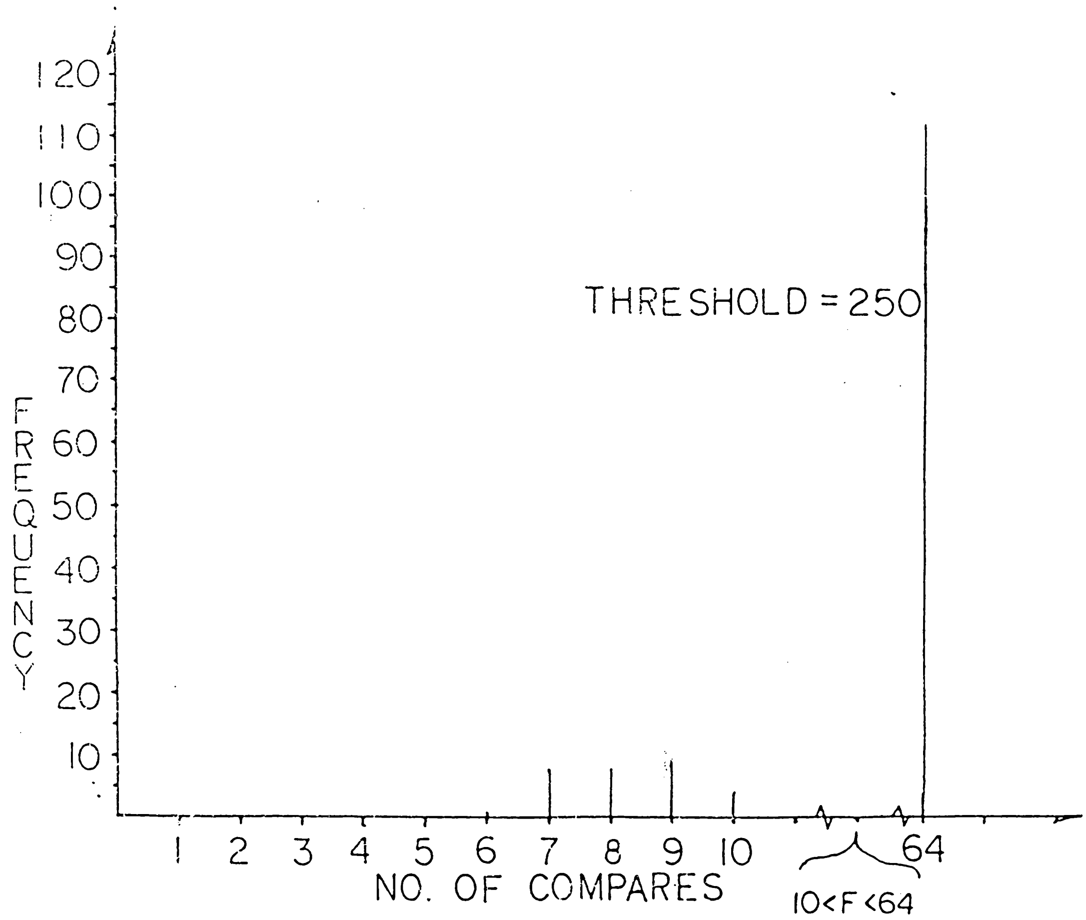


Figure 4.19 SSDA Comparison Histogram - IMAGE_DAT5.

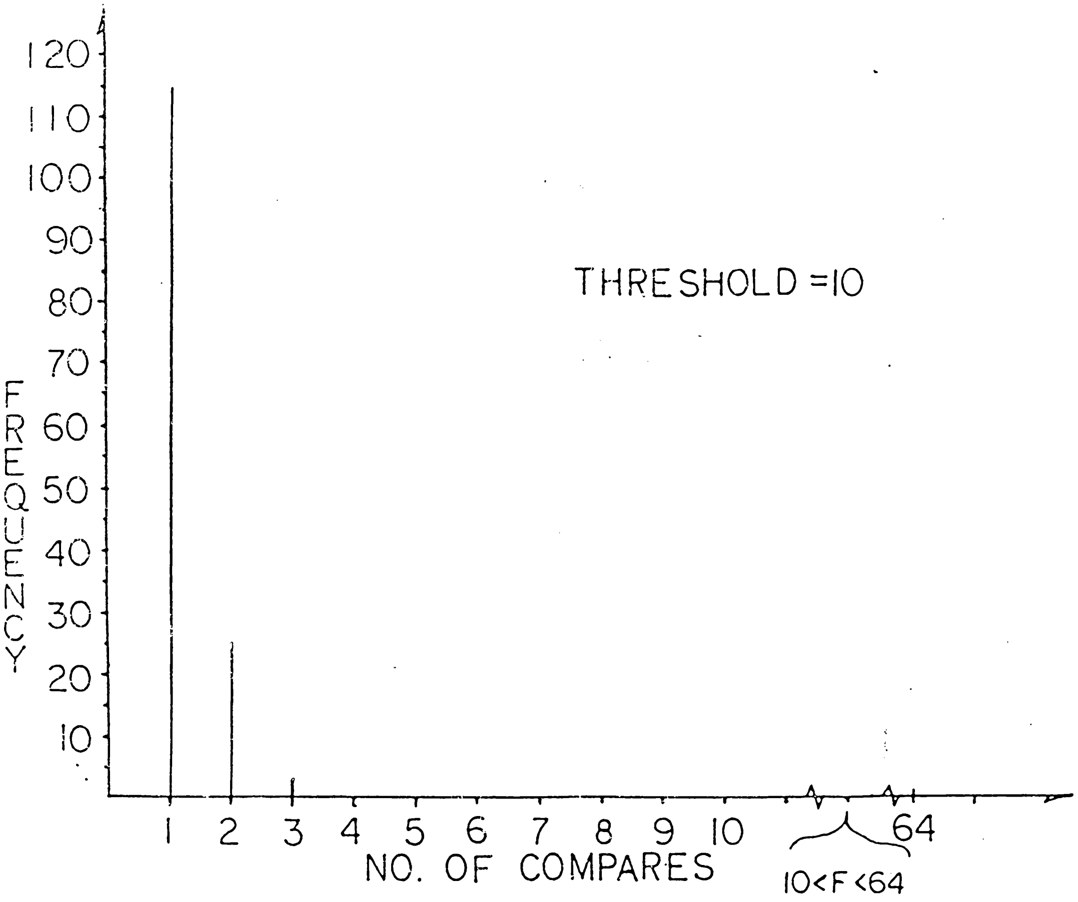


Figure 4.20 SSDA Comparison Histogram - IMAGE_DAT7.

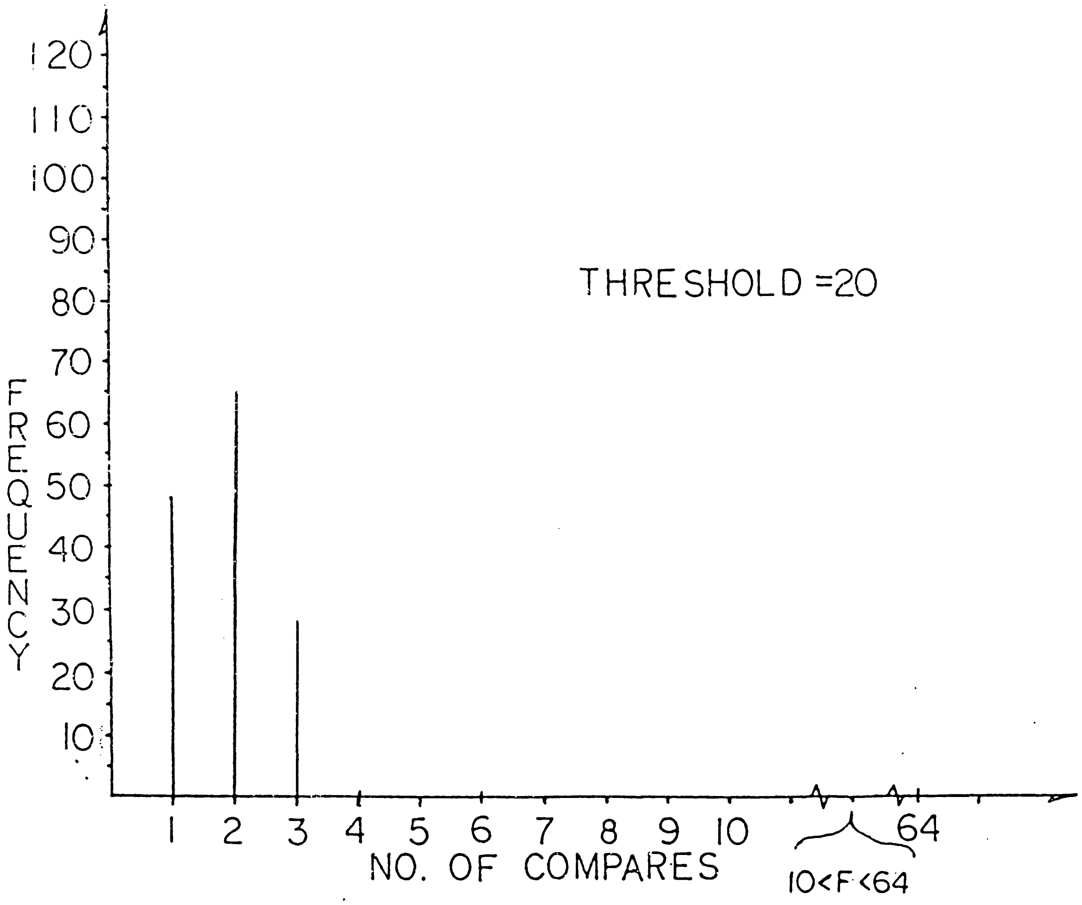


Figure 4.21 SSDA Comparison Histogram - IMAGE_DAT7.

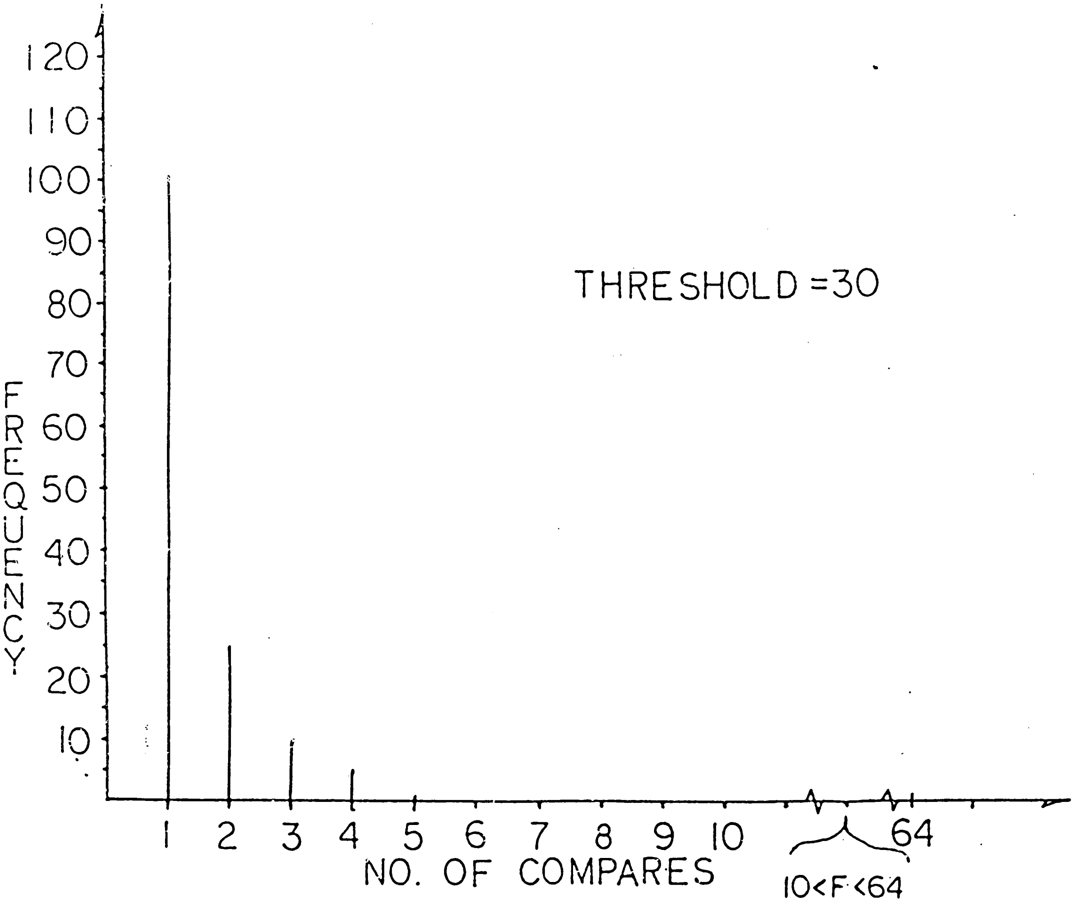


Figure 4.22 SSDA Comparison Histogram - IMAGE_DAT7·

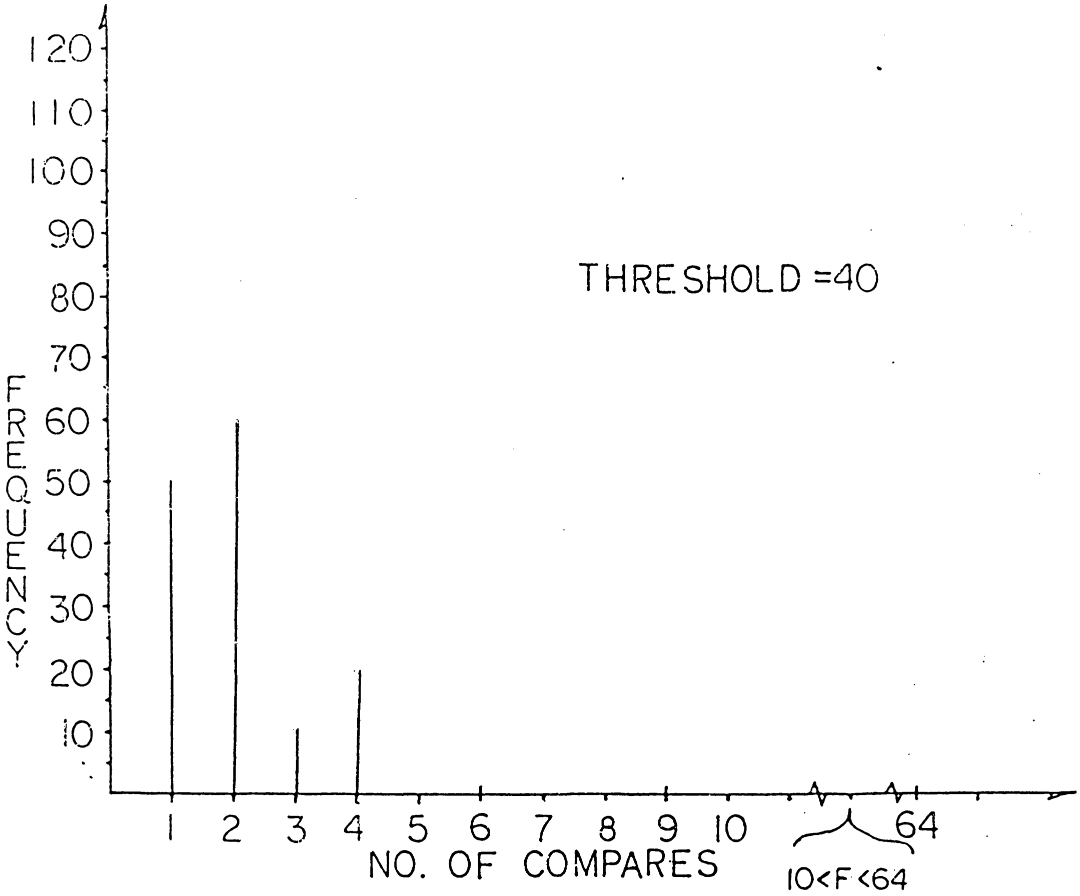


Figure 4.23 SSDA Comparison Histogram - IMAGE_DAT7.

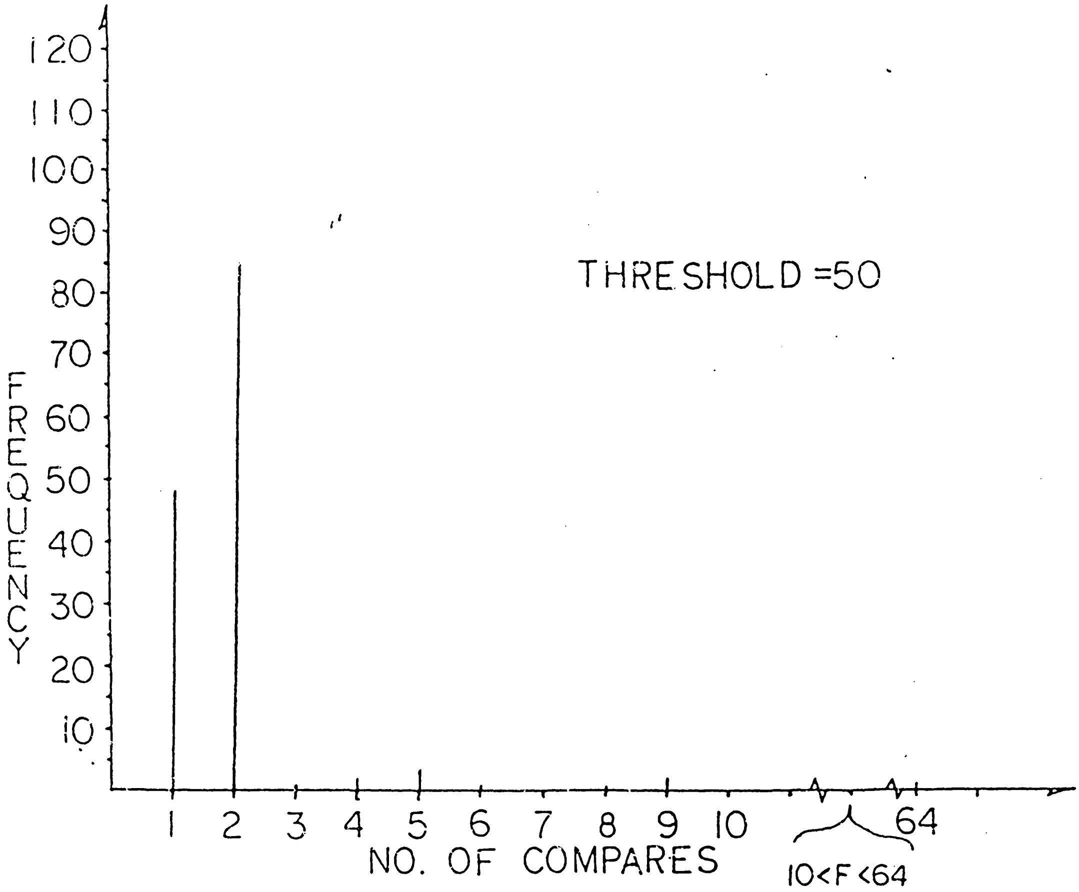


Figure 4.24 SSDA Comparison Histogram - IMAGE_DAT7.

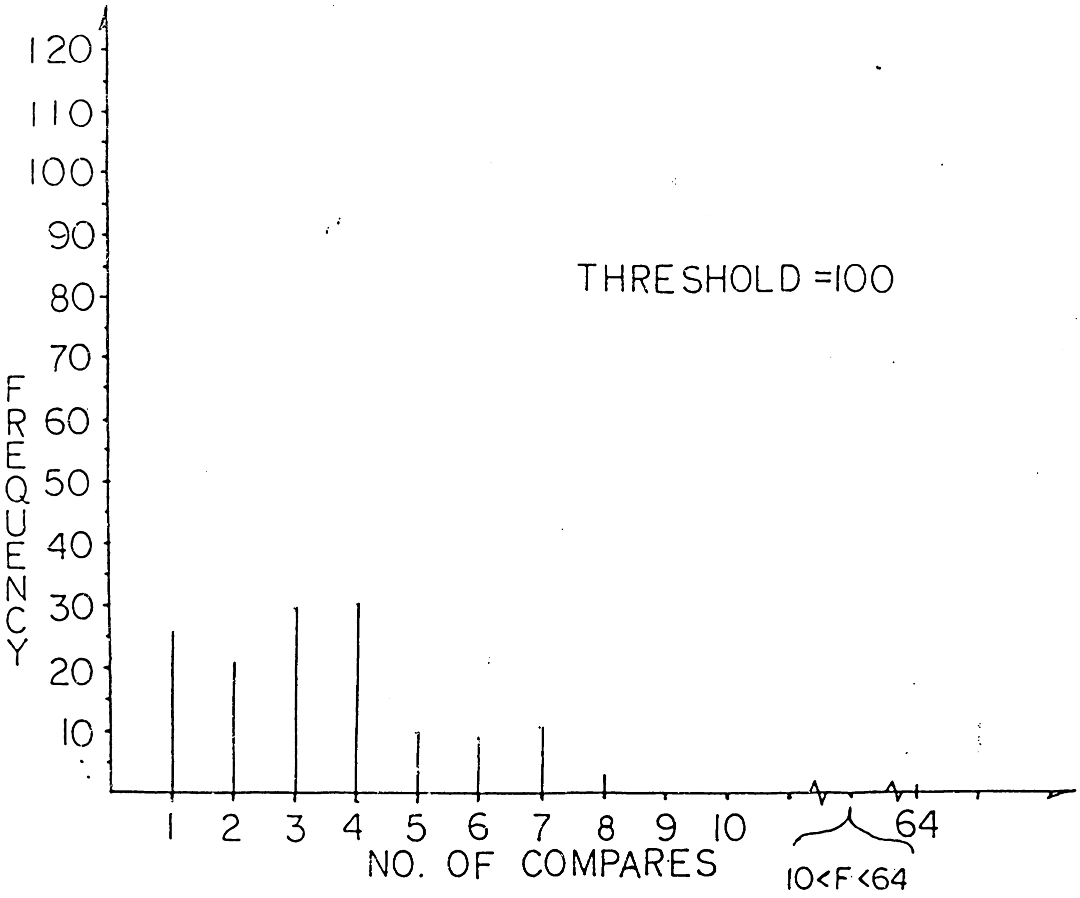


Figure 4.25 SSDA Comparison Histogram - IMAGE_DAT7.

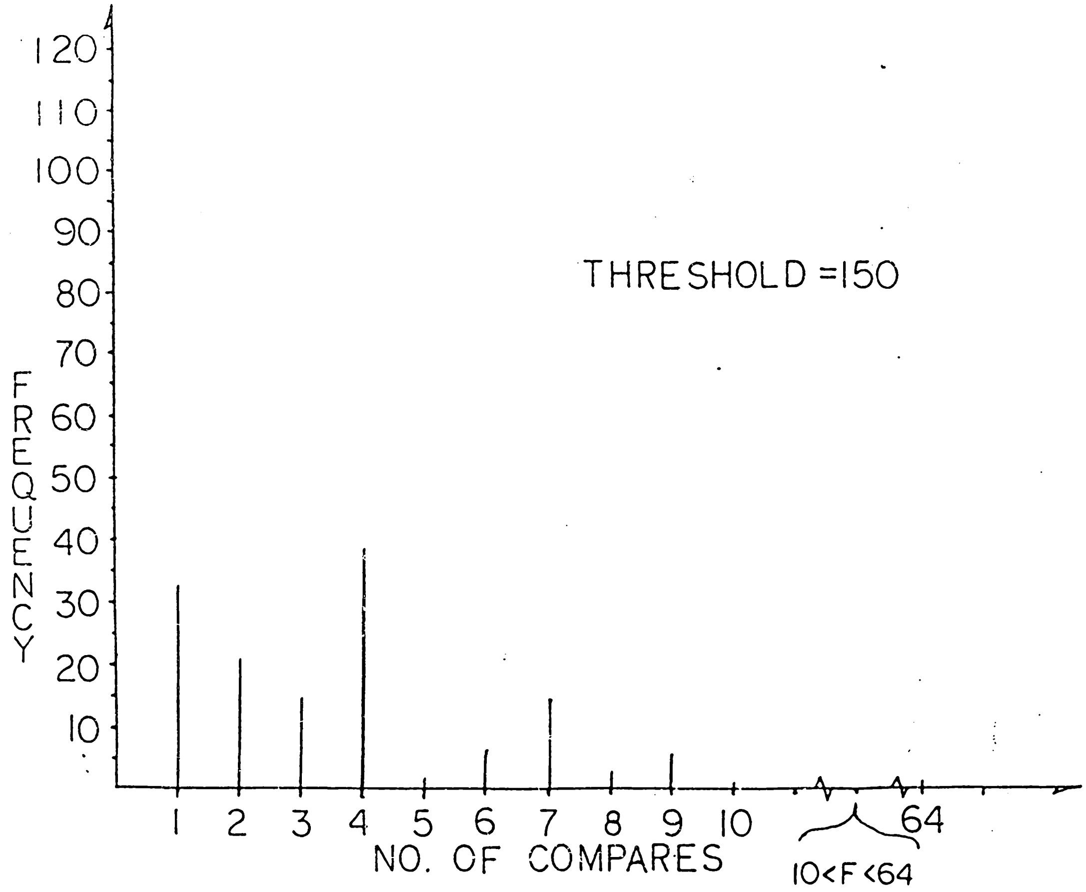


Figure 4.26 SSDA Comparison Histogram - IMAGE_DAT7.

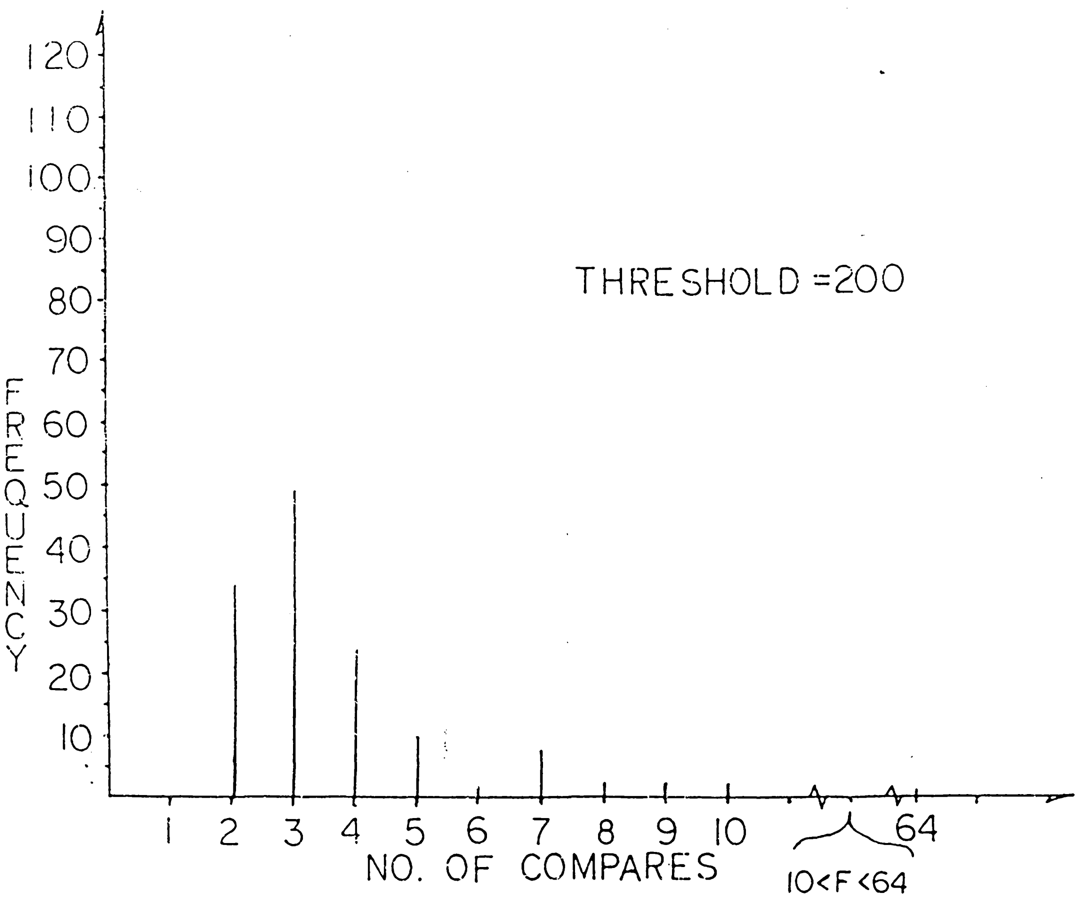


Figure 4.27 SSDA Comparison Histogram - IMAGE_DAT7.

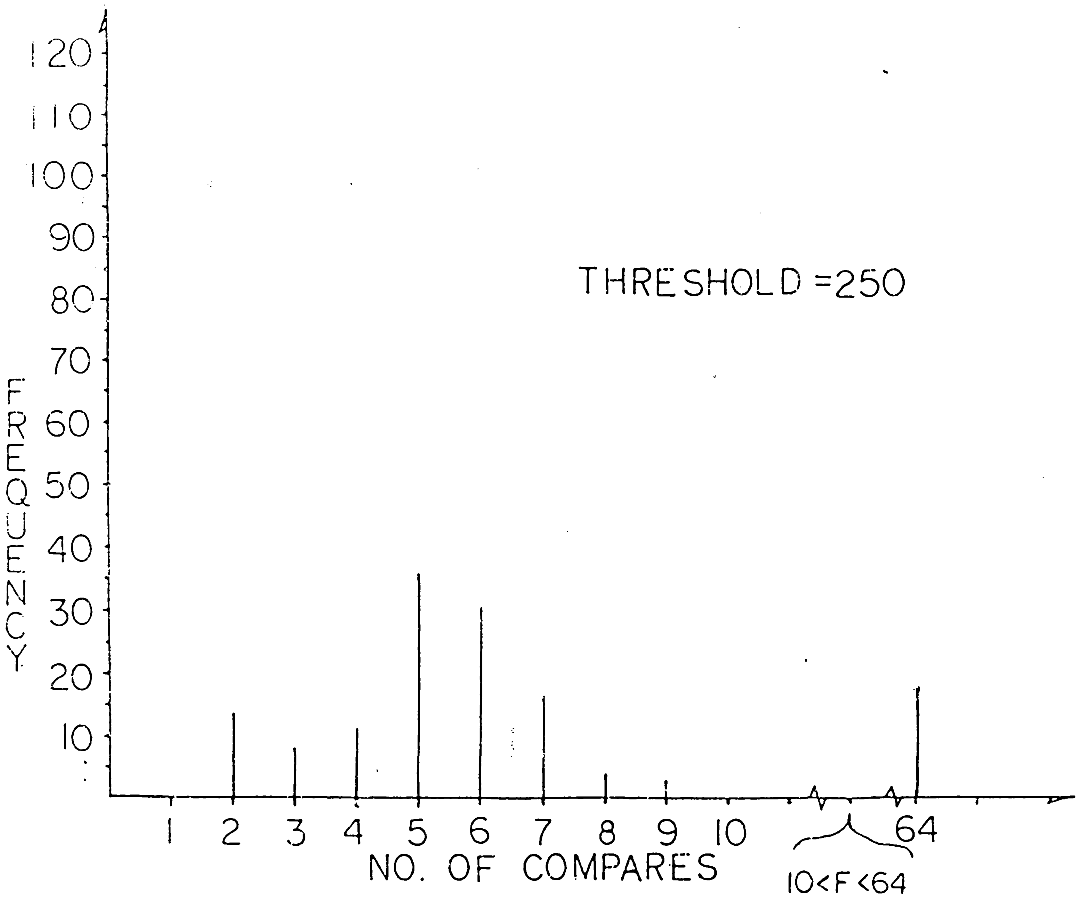


Figure 4.28 SSDA Comparison Histogram - IMAGE_DAT7.

for a particular threshold setting, of the number of compares. Two extremes are shown in Figures 4.11 and 4.19 for IMAGE_DAT5; 4.20 and 4.28 for IMAGE_DAT7.

With the threshold set at 10, most of the subimages are rejected after only one comparison. This is not a desirable condition because not enough of the image is compared at each location. On the other hand, a threshold of 250 causes most of the subimages to be compared by testing all of the pixels at each location. In this case the optimum choice lies somewhere in between. A threshold of 50 for #5 and 100 for #7 would be a reasonable choice.

The effect of threshold setting on execution time is plotted in Figure 4.29. High thresholds can cause wide variations in execution time, making performance of a measurement system dependent on what was imaged on the photography. It is not possible to determine whether SSDA is more accurate than correlation from the limited data available to this experiment. A stereo pair of images would have been needed to do this. Obviously, SSDA is faster than correlation. However, there is no simple means of selecting an optimum threshold without a lot of testing.

The termination sum versus the column number for a typical SSDA search is shown in Figure 4.30. At the point of a match, (6,6), the error is equal to zero. This is because the reference scene is the same as the subimage. It can be concluded from this graph that for the particular threshold setting (40) used, the match is clearly distinguishable from the surrounding images.

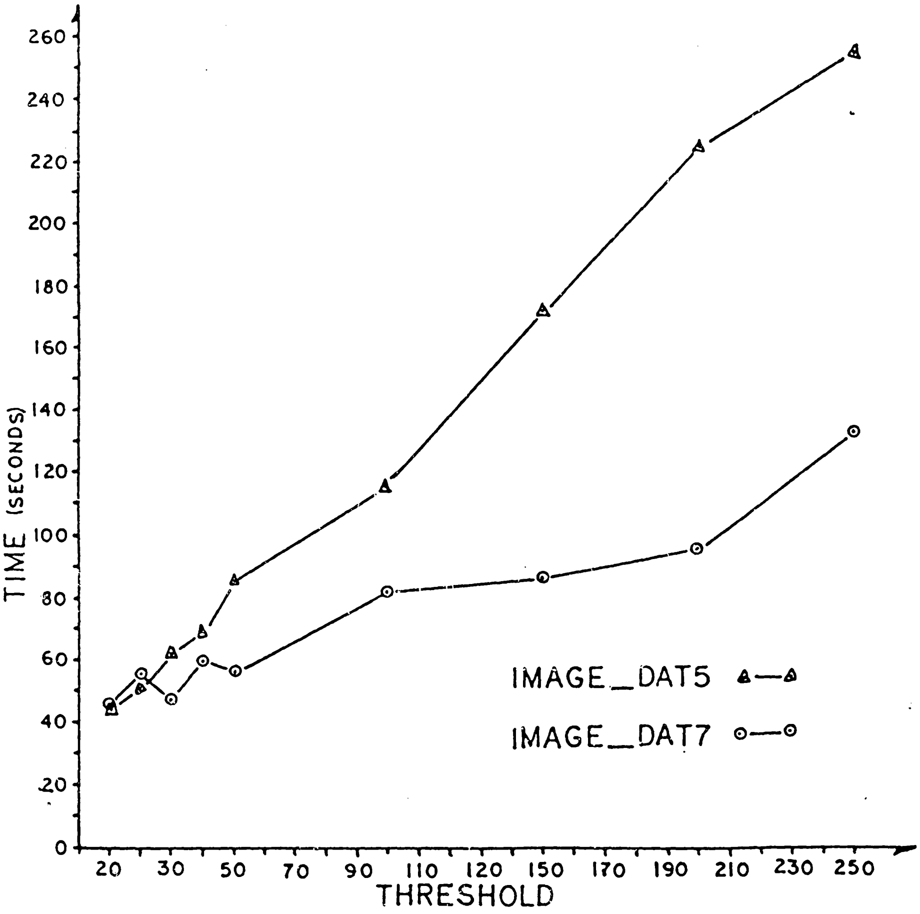


Figure 4.29 Execution Time versus Threshold.

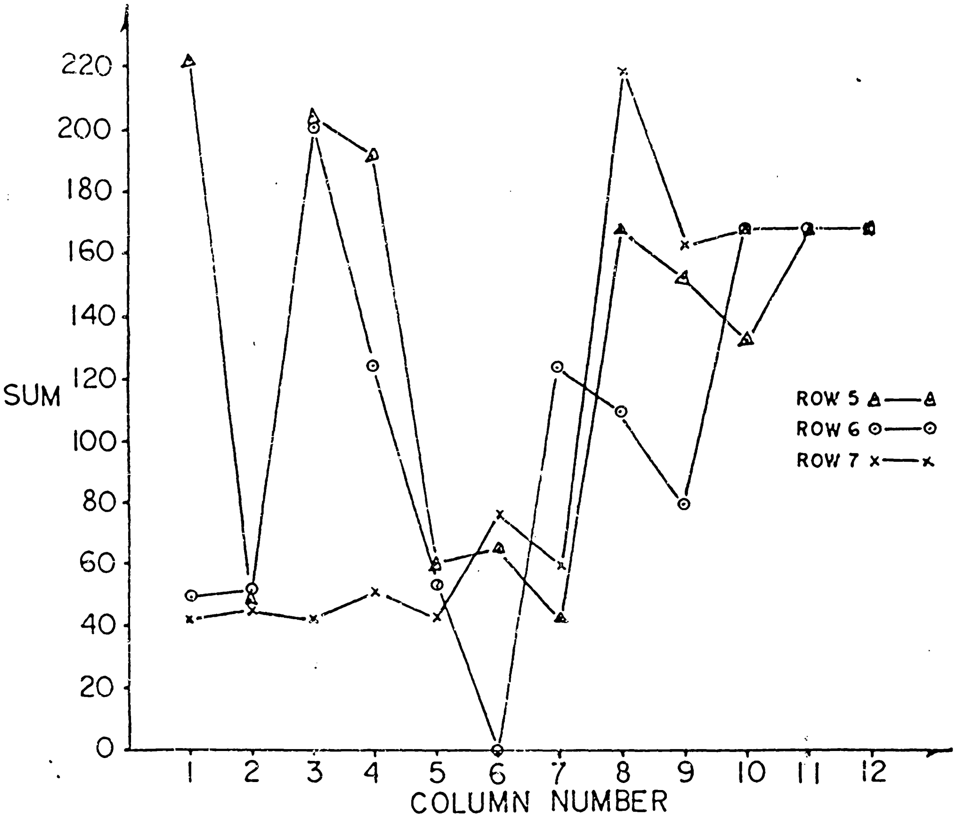


Figure 4.30 Termination Sum versus Column Number.

4.3 ALGORITHM EXECUTION TIME COMPARISON

The execution of the correlation function in the computer is time consuming because the number of multiplications increases rapidly as the size of the reference scene increases. W. K. Pratt indicates that the number of scalar multiples is as follows:

$$U^2/2 + (1+2V)U + V,$$

where $U = JxK$ (the number of pixels in the array) and $V = (M-J+1) \times (N-K+1)$ (the number of possible registration points). For a search of the scenes used in the project, the number of multiplications would equal 20,688. The correlation program used in this experiment completed a search in 37 minutes. However, SSDA is much faster. It completed the same search in 85 seconds. This is directly due to its simpler nature, a summation of absolute differences. SSDA does suffer, though, from the lack of certainty about what threshold setting to use.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

From the results of the experiment it can be concluded that the testing of registration algorithms in small computer systems is feasible if the data sets are limited in size and that the algorithms work correctly. It was also concluded that

1. Sequential Similarity Detection Algorithm (SSDA) completes a search of an image in significantly less time than the correlation algorithm does.

2. There can be more than one location where the reference scene will correlate well with subimages in the search area.

3. The correlation peak is clearly detectable.

4. The time with which the SSDA executes increases as the magnitude of the threshold parameter increases and can cause wide variation in execution speed from image to image.

5. An optimum setting for the SSDA threshold parameter can be determined by noting the distribution pixel comparisons.

6. Curve fitting techniques can be used to increase the precision of a correlation registration. However, care must be taken to choose the right order polynomial to minimize offset error.

5.2 RECOMMENDATIONS

Based on the results of the tests and evaluations, the following recommendations are made for future experimentation in image registration.

1. That the testing and evaluation of the algorithms be continued using stereo pairs.

2. That the accuracy of the two methods be compared by using the two algorithms to determine image coordinates of known topographic features.

3. Continue to test the possibilities of using correlation as a feature extractor.

4. That the execution speed of the algorithms be improved.

5. That the tests of the effects of noise on the registration algorithms be continued.

6. BIBLIOGRAPHY

1. Anuta, P.E., "Digital Registration of Multispectral Video Imagery," Society of Photo-optical Instrumentation Engineers J., Vol. 7, Sept. 1969, pp. 168-175.

2. Anuta, P.E., "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transfer Techniques," IEE Transactions on Geoscience Electronics, Vol. GE-8, No. 4, Oct. 1970, pp. 353-368.

3. Arcese, A, Mengert, P.H., "Image Detection Through Bipolar Correlation," IEEE Transactions on Information Theory, Vol. IT-16, No. 5, Sept. 1970, pp. 534-541.

4. Barnea, D.I., et al, "A Class of Algorithms for Fast Digital Image Registration," an article from Digital Image Processing for Remote Sensing," IEEE Press, 1978, pp. 139-145.

5. Bernstien, R., et al, "Digital Techniques for Earth Resource Image Data Processing," an article from Digital Image Processing for Remote Sensing, IEEE Press, 1978, pp. 109-111.

6. Bernstien, R., editor, "Digital Image Processing for Remote Sensing," IEEE Press, 1978, p. 453.

7. Colvocoresses, A.P., "The Relationship of Acquisition Systems to Automated Stereo Correlation," Photogrammetric Engineering and Remote Sensing, Vol. 49, No. 4, April 1983, pp. 539-544.

8. Crombie, M.A., "Coordination of Stereo Image Registration and Pixel Classification," Photogrammetric Engineering and Remote Sensing, Vol. 49, No. 4, April 1983, pp. 529-532.

9. Crombie, M.A., "Single Photo Analysis of Sampled Aerial Imagery," Research Note ETL-RN-74-10, August 1974, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia.

10. Gonzalez, R., et al, "Digital Image Processing," Addison-Wesley, 1977, Reading, Mass., Section 7.4, pp. 382-386.

11. Hall, E.L., "Computer Image Processing and Recognition," Academic Press Inc., New York, N.Y., 1979, Section 8.1.2, pp. 480-487.

12. Horn, B.K.P., "Non-Correlation Methods for Stereo Matching," Photogrammetric Engineering and Remote Sensing, Vol. 49, No. 4, April 1983, pp. 533-534.

13. Opitz, B.K., "Advanced Stereo Correlation Research," Photogrammetric Engineering and Remote Sensing, Vol. 49, No. 4, April 1983, pp. 533-534.

14. Pratt, W.K., "Generalized Wiener Filtering Computation Techniques," IEEE Transactions on Computers, Vol. C-21, No. 7, July 1972, pp. 636-641.

15. Pratt, W.K., "Correlation Techniques of Image Registration," IEEE Transactions on Aerospace and Electronics Systems, May 1974, pp. 465-470.

16. "User's Reference Guide," Texas Instruments, Inc., 1981, p. II-9.

17. Welch, R., "Measurement from Linear Array Camera Images," Photogrammetric Engineering and Remote Sensing, Vol. 46, No. 3, March 1980, pp. 315-318.

18. Wood, G.A., "Realities of Automatic Correlation Problems," Photogrammetric Engineering and Remote Sensing, Vol. 49, No. 4, April 1983, pp. 537-538.

7. APPENDIX

7.1 APPENDIX A

7.1.1 Program Documentation.

Program Name. Image Correlation Program

Programmer. Edward J. Cyran

Date. 4/10/82

Purpose. Determine the location of a subimage with the greatest correlation with a reference subimage in a designated search area.

Language. Texas Instruments BASIC

Computer. TI 99/4A

Description. The program searches a predetermined search array of pixels and computes a correlation measure at each subimage location within the area. The program normalizes the pixel data to have a mean of zero. The correlation value at each location is displayed on the monitor and printed. The search is completed when all subimage locations are compared. All image data is stored in core to speed program execution.

7.1.2 Program Listing

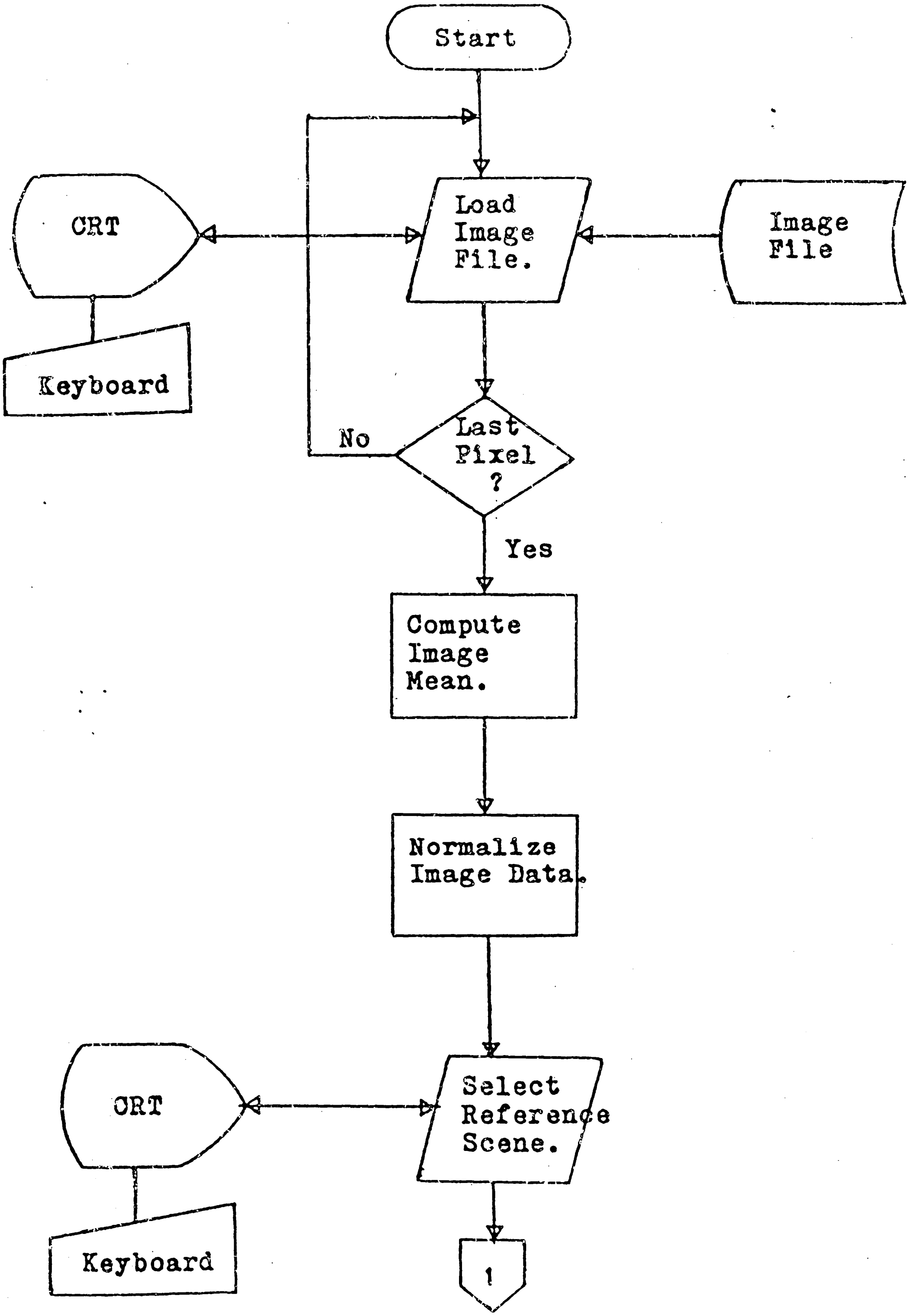
```
100 REM IMAGE CORRELATION PROGRAM.
110 MAIN PROGRAM
115 OPTION BASE 1
120 DIM IMAGE1(16,16),IMAGE2(16,16),R(13,13),B(19,19)
125 OPEN #2:"RS232/1.BA=600.DA=7.LF."
126 CALL CLEAR
130 INPUT "WINDOW ROW DIMENSION?":N
140 INPUT "SEARCH ROW DIMENSION?":N1
145 INPUT "PRINTER OUTPUT Y?N?":C$
150 INPUT "WHICH IMAGE DATA FILE?":A$
160 OPEN #1:A$,RELATIVE 361,INTERNAL, INPUT, FIXED 9
170 FOR I5=1 TO N1
180 FOR I6=1 TO N1
190 INPUT #1:B(I5,I6)
200 NEXT I6
210 NEXT I5
220 REM COMPUTE IMAGE MEAN
230 GOSUB 3000
240 REM NORMALIZE IMAGE TO HAVE MEAN OF ZERO.
250 GOSUB 4000
260 INPUT "INITIAL ADDRESS OF REFERENCE SCENE?":N2,N3
270 A1=N2+N-1
280 A2=N3+N-1
290 A4=1
300 A5=1
310 FOR A3=N2 TO A1
320 FOR A6=N3 TO A2
330 IMAGE1(A4,A5)=B(A3,A6)
340 A5=A5+1
350 NEXT A6
360 A5=1
370 A4=A4+1
380 NEXT A3
390 REM SEARCH LOOPS
400 LIMIT1=N1-N+1
410 FOR L1=1 TO LIMIT1
420 FOR L2=1 TO LIMIT1
430 L3=1
440 L4=1
450 L5=L1+N-1
460 L6=L2+N-1
470 FOR L7=L1 TO L5
480 FOR L8=L2 TO L6
490 IMAGE2(L3,L4)=B(L7,L8)
500 L4=L4+1
510 NEXT L8
520 L4=1
530 L3=L3+1
540 NEXT L7
```

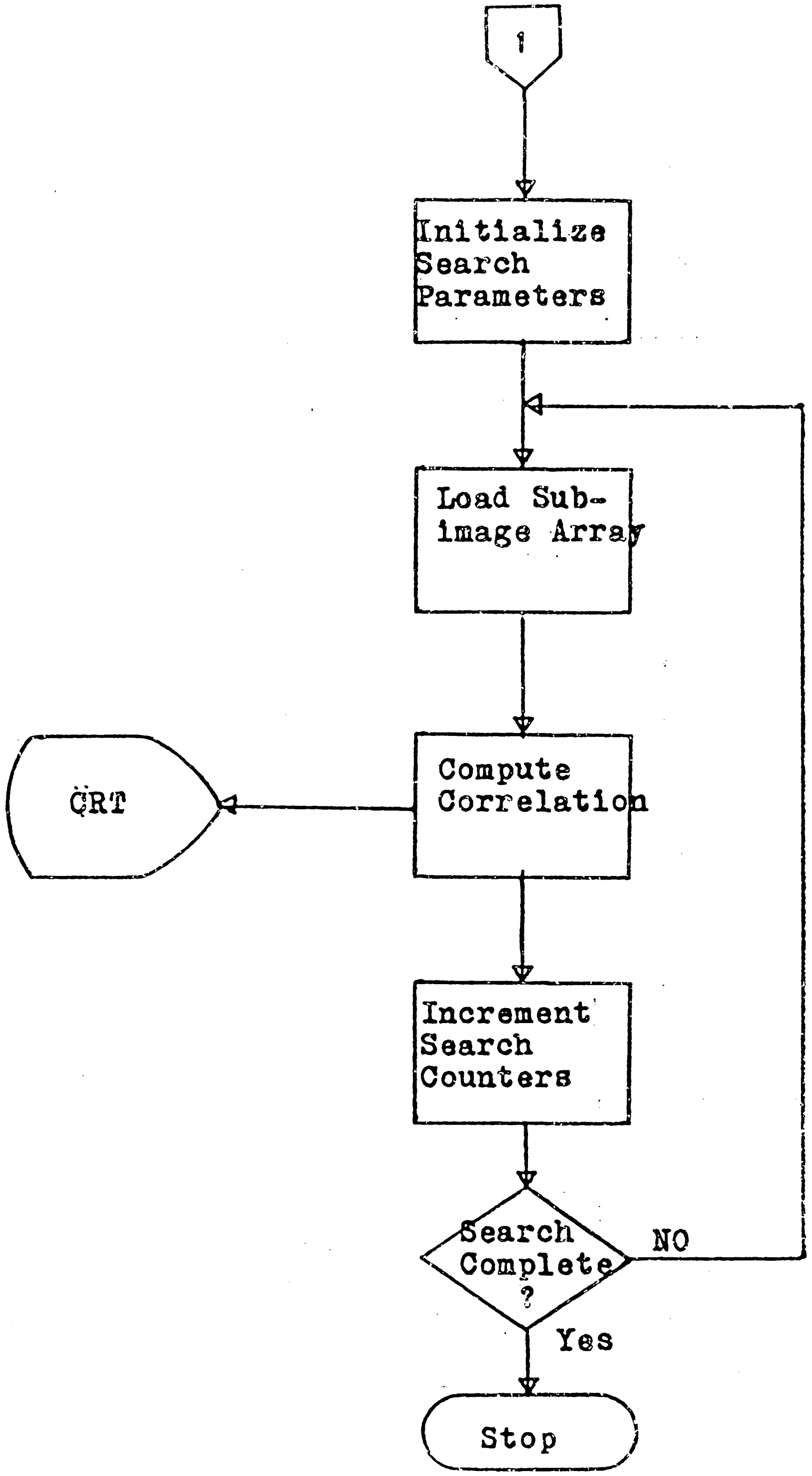
```
550 REM COMPUTE CORRELATION COEFFICIENT
560 U=L1
570 V=L2
580 GOSUB 1010
585 IF C$="Y" THEN 586 ELSE 590
586 PRINT #2: "R(";U;", ";V;")=";R3
590 NEXT L2
600 NEXT L1
605 CLOSE #1
606 CALL SOUND(500,300,2,600,2,900,2,-2,2)
610 STOP
1000 REM COMPUTE THE CORRELATION COEFFICIENT FOR LOCATION U,V OF
     IMAGES 1 AND 2.
1010 SUM1=0
1020 SUM2=0
1030 SUM3=0
1040 FOR I=1 TO N
1050 FOR J=1 TO N
1060 SUM1=SUM1+IMAGE1(I,J)*IMAGE2(I,J)
1070 SUM2=SUM2+(IMAGE1(I,J)**2)
1080 SUM3=SUM3+(IMAGE2(I,J)**2)
1090 NEXT J
1100 NEXT I
1110 DENOM2=SQR(SUM2)
1120 DENOM3=SQR(SUM3)
1130 R(U,V)=SUM1/(DENOM2*DENOM3)
1134 RT=R(U,V)
1135 GOSUB 2010
1140 DISPLAY "R(";U;",";V;")=";R3
1150 RETURN
2000 REM FIX 2 DECIMAL PLACE ROUTINE
2010 R1=RT*100
2020 R2=INT(R1)
2030 R3=R2/100
2031 R4=RT-R3
2032 R5=INT(R4*1000)
2033 IF R5>=5 THEN 2037
2036 GOTO 2040
2037 R2=R2+1
2038 R3=R2/100
2040 RETURN
3000 REM COMPUTE IMAGE MEAN ROUTINE
3010 XSUM=0
3020 FOR X1=1 TO N1
3030 FOR X2=1 to N1
3040 XSUM=XSUM+B(X1,X2)
3050 NEXT X2
3060 NEXT X1
3070 XBAR=XSUM/(N1*N1)
3080 RETURN
4000 REM MEAN OF ZERO
```

```
4010 FOR Y1=1 TO N1
4020 FOR Y2=1 TO N1
4030 B(Y1,Y2)=B(Y1,Y2)-XBAR
4040 NEXT Y2
4050 NEXT Y1
4060 RETURN
4070 END
```

End of listing

7.1.3 Program Flow Chart




1
Initialize Search Parameters
Load Sub-image Array
Compute Correlation
CRT
Increment Search Counters
Search Complete ?
NO
Yes
Stop

7.2 APPENDIX B

7.2.1 Program Documentation

Program Name: Sequential Similarity Detection Algorithm

Programmer: Edward J. Cyran

Date: 5/8/82

Purpose: Determine the location of a subimage with greatest similarity to a reference subimage in a designated area.

Language: Texas Instruments BASIC

Computer: TI-99/4A

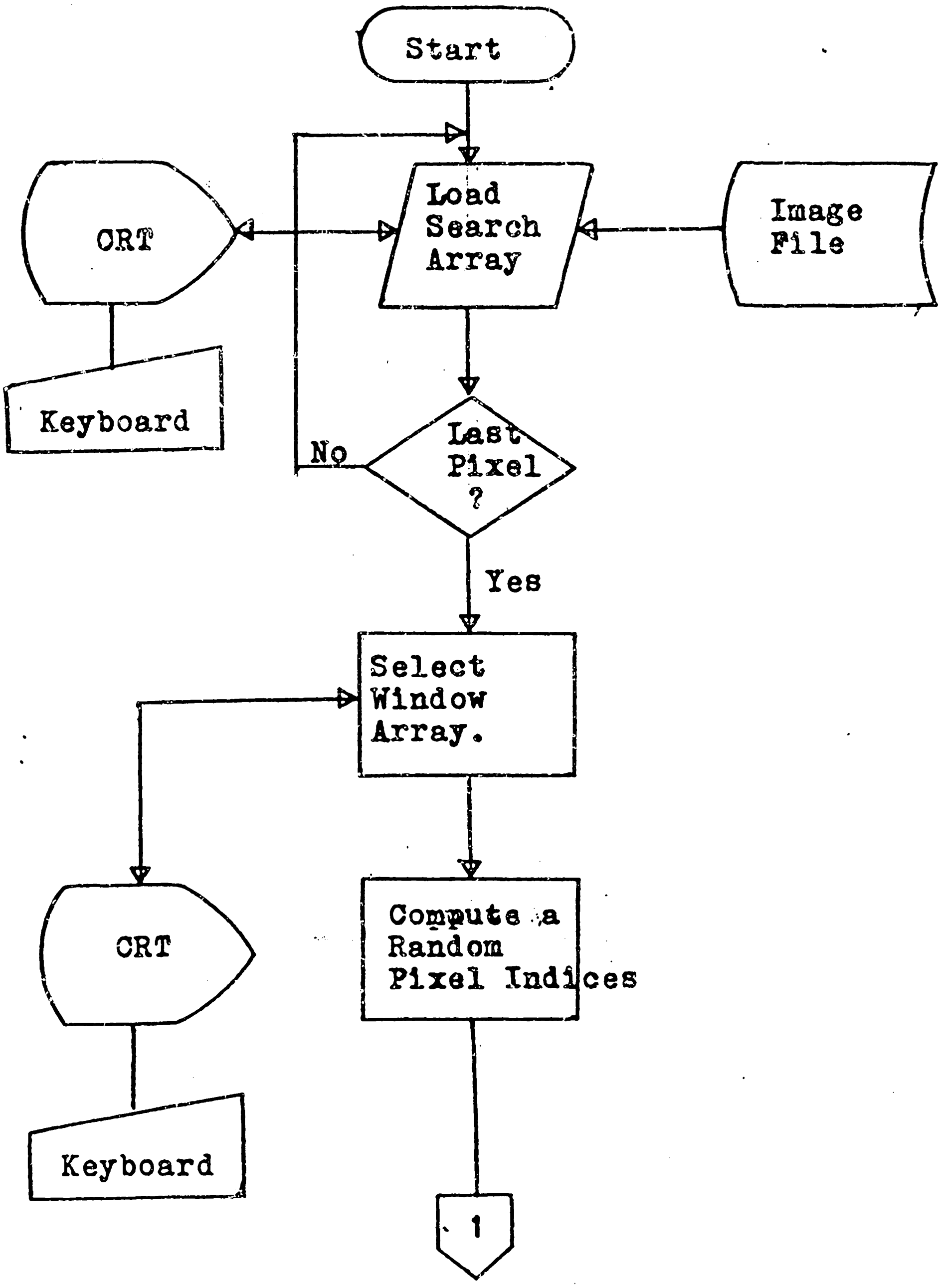
Description: This program searches a predetermined search array of pixels and computes an error sum by comparing matching pixel locations in a random order. The random order is computed using a BASIC language function called "RND". To insure that this function computes an order differently each time, another function called "RANDOMIZE" is called each time the program is executed. The search proceeds from left to right and from top to bottom of the search array. Upon completion of the search, the number of compares at each subimage location is displayed. The location of greatest similarity is displayed when it is determined. All image data is stored in core to speed program execution.

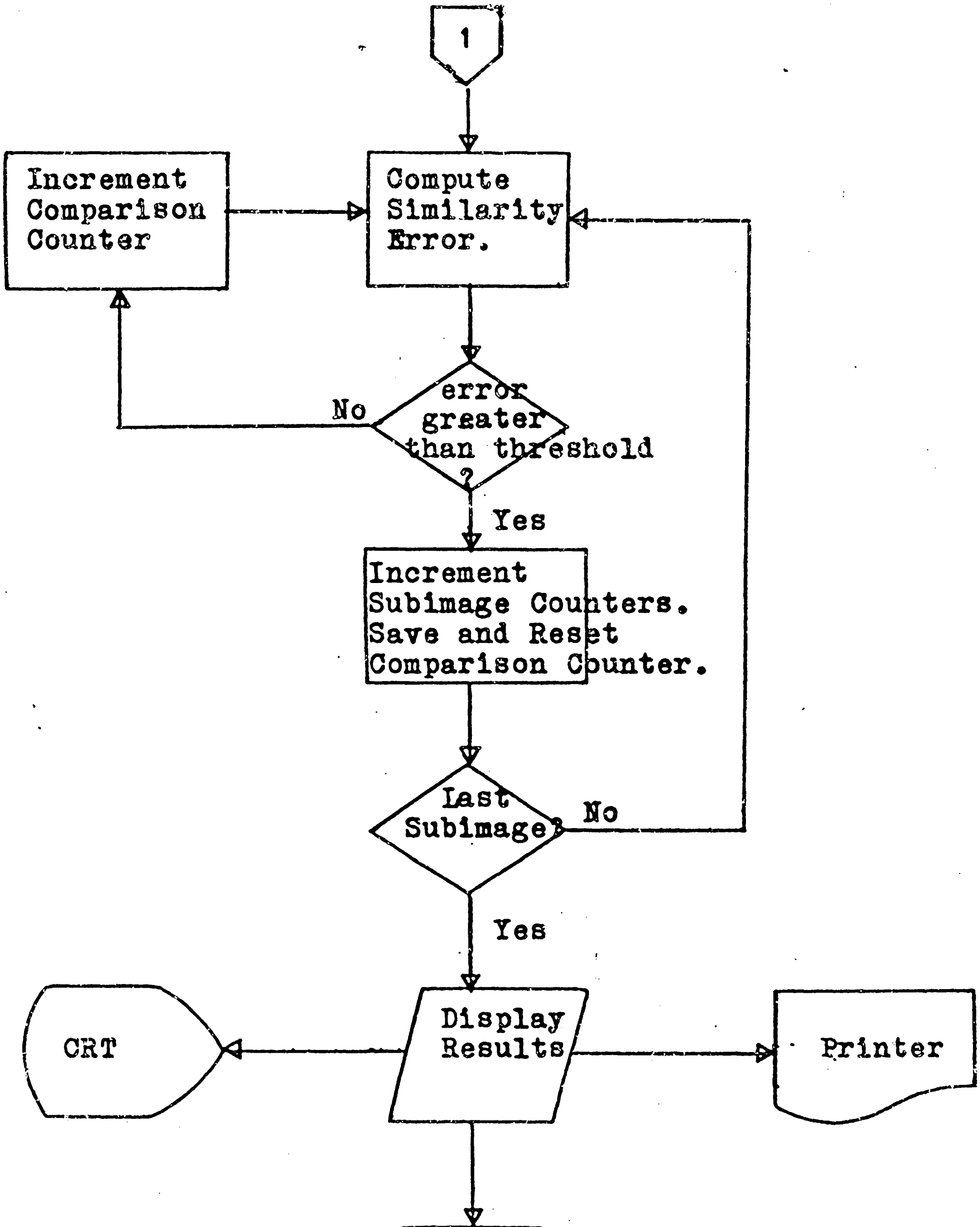
7.2.2 Listing

```
100 REM SEQUENTIAL SIMILARITY ALGORITHM
     Edward J. Cyran, 5/8/82
110 DIM IMAGE(19,19),IMAGE1(8,8),I(12,12),A(8)
120 CALL CLEAR
130 DISPLAY "Sequential Simularity Algorithm"
140 INPUT "Image data file?":A$
150 OPEN #1:A$,RELATIVE 361,INTERNAL,INPUT,FIXED 9
160 INPUT "Set threshold ":T
170 FOR I1=1 TO 19
180 FOR I2=1 TO 19
190 INPUT #1:IMAGE(I1,I2)
200 NEXT I2
210 NEXT I1
220 INPUT "Search Window Address":N1,N2
230 A1=N1+8-1
240 A2=N2+8-1
250 A3=1
260 A4=1
270 FOR I3=N1 TO A1
280 FOR I4=N2 TO A2
290 IMAGE1(A3,A4)=IMAGE(I3,I4)
300 A4=A4+1
310 NEXT I4
320 A4=1
330 A3=A3+1
340 NEXT I3
350 GOSUB 770
360 DISPLAY "SEARCHING!"
370 REM SEARCH LOOPS
380 LIMIT1=19-8+1
390 FOR L1=1 TO LIMIT1
400 FOR L2=1 TO LIMIT1
410 T1=0
420 SUM=0
430 L3=1
440 L4=1
450 L5=L1+8-1
460 L6=L2+8-1
470 FOR L7=L1 TO L5
480 FOR L8=L2 TO L6
490 C1=A(L3)
500 C2=A(L4)
510 C3=A(L3)+L1-1
520 C4=A(L4)+L2-1
530 SUM=SUM+ABS(IMAGE1(C1,C2)-IMAGE(C3,C4))
540 T1=T1+1
550 IF SUM T THEN 630
560 L4=L4+1
570 NEXT L8
```

```
580 L4=1
590 L3=L3+1
600 NEXT L7
610 DISPLAY "SUM;U ;V"
620 DISPLAY SUM; L1;L2
630 I(L1,L2)=T1
640 NEXT L2
650 NEXT L1
660 DISPLAY "SEARCH COMPLETE"
670 DISPLAY "TEST ARRAY"
680 FOR J1=1 TO 12
690 FOR J2=1 TO 12
700 DISPLAY I(J1,J2)
710 NEXT J2
720 NEXT J1
730 STOP
740 REM Subroutine Random Integers
750 REM Compute Random Sequence of Non-repeating Integers
760 REM
770 RANDOMIZE
780 FOR I5=1 TO 8
790 B=INT(10*RND)
800 IF B 9 THEN 810 ELSE 790
810 FOR I6=1 TO 8
820 IF B=A(I6)THEN 790
830 NEXT I6
840 A(I5)=B
850 NEXT I5
860 RETURN
870 END
```

7.2.3 Program Flow Diagram




1
Compute Similarity Error.
Increment Comparison Counter
error greater than threshold ?
No
Yes
Increment Subimage Counters. Save and Reset Comparison Counter.
Last Subimage?
No
Yes
Display Results
CRT
Printer
End

7.3 APPENDIX C

Description of the Texas Instruments 99/4A computing system.

7.3.1 System. The TI computer used in this project consists of the following:

a. A console, including the cpu, 16K byte random access memory, keyboard, and interfaces to external devices

b. A disk memory controller and Shugart 5-1/4-inch floppy disk drive

c. RS232 interface

d. Radio Shack printer

e. 10-inch color monitor

f. Cassette tape recorder

7.3.2 Description. The TI model 99/4A computer system is designed around the TI TMS 9900 16 bit microprocessor chip. It operates at a clock rate of 3 mhz. The memory is capable of expansion by another 32K bytes to a total of 48K bytes. Figure 7.3.1 is a block diagram of the computer console. In its present configuration, all random access memory, color display controller, complex sound generator, etc. are a part of the console. The disk memory controller and the RS232 interface are separate boxes plugged into the side of the console. The RS232 interface has two ports and can send and receive data at baud rates of from 110 to 9,600. The interface is set up under software control to satisfy a variety of external device requirements such as baud rate, number of stop bits, and data bits.

7.3.3 Data Processing. All of the applications programs and image data are stored on the floppy disk. The applications programs are also stored on cassette tape as a means of mass storage backup.

7.3.4 Programming Language. All of the software is written in Texas Instruments version of BASIC. This BASIC operates as an interpreter converting each program line to machine code before execution. It is a fairly up-to-date version of BASIC that includes IF-THEN-ELSE, FOR-NEXT, GOSUB, color graphics, and sound functions. All of the software has been written to make use of its interactive capabilities and maximize ease of operation.

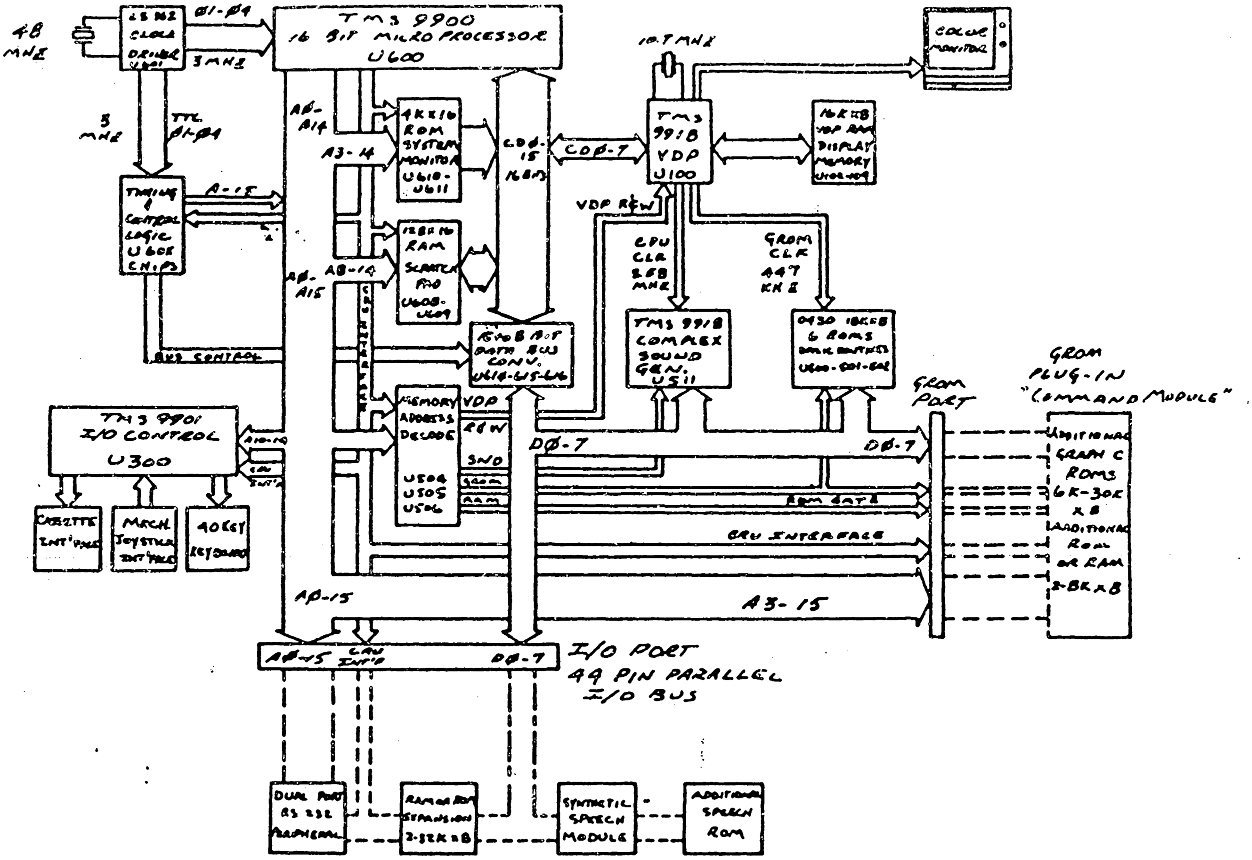


Figure 7.1 Block Diagram of 99/4A Console.