

AD-A129 644

RADAR BRIDGE PATTERNS EXTRACTION AND RECOGNITION(U)
ARMY ENGINEER TOPOGRAPHIC LABS FORT BELVOIR VA
P CHEN ET AL. APR 83 ETL-0323

1/1

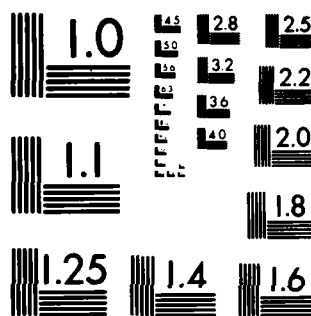
UNCLASSIFIED

F/G 17/9

NL

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

END
DATE
FILMED
7 83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA 129644

ETL-0323

Radar bridge patterns extraction and recognition

Pi-Fuay Chen
Neil D. Fox

April 1983

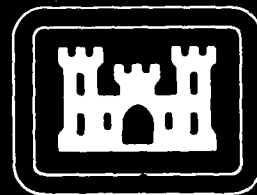
DTIC
ELECTE
JUN 22 1983
S D
A

DTIC FILE COPY

U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

83 06 22 019



E

T

L



Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

UNCLASSIFIED
 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0323	2. GOVT ACCESSION NO. AD-A129644	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RADAR BRIDGE PATTERNS EXTRACTION AND RECOGNITION		5. TYPE OF REPORT & PERIOD COVERED Research Note Oct 1981 - Sep 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Pi-Fuay Chen Neil D. Fox		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 6.11.02.A 4A161102B52C B 012
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, VA 22060		12. REPORT DATE April 1983
		13. NUMBER OF PAGES 42
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution Unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Hough Transform Image Edge Enhancement Radar Feature Extraction and Recognition		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A technique is described for detecting and recognizing various bridge patterns with different orientation angles from a set of radar imagery. The algorithm combines Hough transform, Sobel edge operator, image thresholding methods, and decision logics and was implemented as a software for an experimental hardware system consisting of a 32-element by 32-element solid-state array, a minicomputer, and a computer-controlled translational stage. The resolution of bridge orientation angle detection was determined to be approximately 22.5 degrees for the current system setup. This resolution can be improved by enlarging the input measurement array size.		

PREFACE

This study was conducted under DA Project 4A161102B52C, Task B, Work Unit 012, "Electronic Image Analysis for Feature Extraction."

The study was done under the supervision of Dr. F. Rohde, Team Leader, Center for Physical Sciences; and Mr. M. Crowell, Jr., Director, Research Institute.

COL Edward K. Wintz, CE, was Commander and Director and Mr. Robert P. Macchia was Technical Director of the Engineer Topographic Laboratories during the study period.



Project No.	
Task No.	
Work Unit No.	
Date	
Time	
Availability	
Part	
A	

CONTENTS

TITLE	PAGE
PREFACE	1
ILLUSTRATIONS	3
INTRODUCTION	4
SYSTEM DESCRIPTION	4
BRIDGE DETECTION AND RECOGNITION	6
SYSTEM TEST RESULTS	8
CONCLUSIONS	9
APPENDIX	31

ILLUSTRATIONS

FIGURE	TITLE	PAGE
1	System Block Diagram	5
2	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 0 Degree	10
3	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 22.5 Degrees	12
4	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 45 Degrees	14
5	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 67.5 Degrees	16
6	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 90 Degrees	18
7	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 112.5 Degrees	20
8	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 135 Degrees	22
9	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of Bridge Pattern at 157.5 Degrees	24
10	Pictorial Print of Processed Input Image, Recognition Result, and Hough Transform of an Entirely Black Area	26
11	Pictorial Print of Processed Input Image and Recognition Result of Bridge Pattern at 162 Degrees with Large Measurement Window	28
12	Pictorial Print of Processed Input Image and Recognition Result of Bridge Pattern at 172 Degrees with Large Measurement Window	29

RADAR BRIDGE PATTERNS EXTRACTION AND RECOGNITION

INTRODUCTION

The objective of this study is to devise a feature extractor for extracting various bridge patterns from radar imagery. This extractor can be used in conjunction with other feature extractors developed earlier, for example, the feature extractor using Walsh transforms for recognizing road intersections, line roads, and rectangular objects¹ and the feature extractor employing image statistics for classifying a set of imagery such as city, water, field and forest.²

The technique presented makes use of a combination of the Hough transform, the Sobel edge operator, image thresholding methods, and decision logics to detect and recognize bridge candidate patterns that are, in general, characterized by a pair of long lines that are relatively close to each other.

The hardware system consists of a 32-element by 32-element solid-state array to convert the bridge patterns into electronic signals, a minicomputer to process the electronic signals from the array using a controlled software, and a computer-controlled, two-dimensional stage as the image holder. The majority of the bridge detection and recognition routines have been implemented as software for the hardware system described.

The description of the system operation is followed by an explanation of the strategies of bridge detection and recognition and bridge orientation angle determination. The system test results are discussed and conclusions presented.

SYSTEM DESCRIPTION

The block diagram of the system is shown in figure 1. A 9- by 9-inch glass plate mounted with various 2- by 2-inch cut radar imagery is illuminated by a light source, and a section of the image containing a

¹P.F. Chen, F.W. Rohde, and W.W. Seemuller, Classification of Cartographic Features Through Walsh Transforms, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Va., ETL-0290, April 1982, AD-116 731.

²P.F. Chen, Preliminary Radar Feature Extraction and Recognition Using Texture Measurement, U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Va., Research note in progress.

bridge pattern is projected onto the surface of a Reticon 32-element by 32-element solid-state array through an imaging lens. The array converts the optical energy of the bridge pattern into a video signal. The video signal is quantized into 10 bits of digital signals and sent to the Hewlett-Packard 2117F minicomputer for processing. The computer first takes in a frame of 32 by 32 pixels of the digitized signal. Each pixel is sequentially compared to an automatically set threshold value. It is set to a "1" if it is greater than the threshold value and to a "0" otherwise.

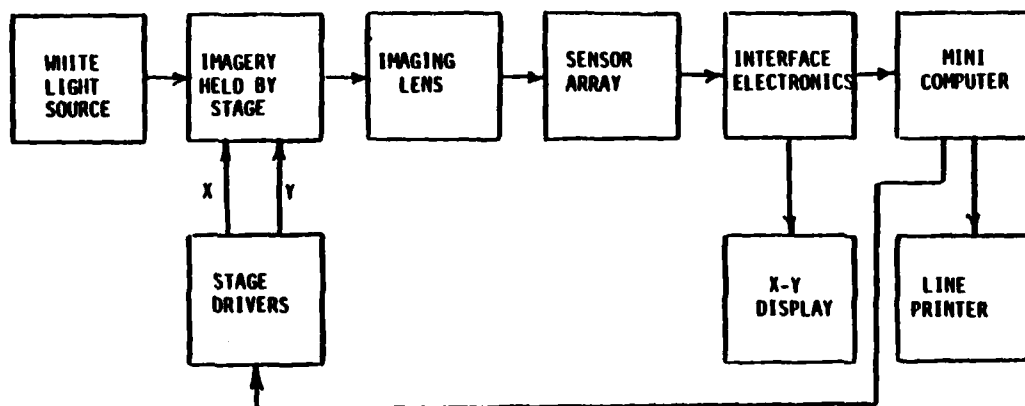


FIGURE 1. System Block Diagram.

The edges of the signal after thresholding are enhanced using the Sobel edge operator. The Hough transform of the edge-enhanced signal is then computed. The transformed matrix has a dimension of 64 by 32 and displays unique characteristics of a bridge. A recognition routine follows to screen out this unique characteristic and classifies the image as a bridge with its orientation angle against the X-axis of the measurement window (or the solid-state array). In case the unique characteristic is missing, the image is identified as "not a bridge." The recognition result is indicated on the CRT console. At the end of classification, the computer sends a signal to the translational stage controllers to move the stages in the predetermined x and y positions, and a new section of image is projected onto the surface of the solid-state array. The procedure described repeats until all preselected images are classified.

BRIDGE DETECTION AND RECOGNITION

As mentioned earlier, a candidate bridge pattern, in general, consists of a pair of long lines that are relatively close together. Thus, each possible bridge pattern is subjected to a set of tests that determine the pattern to be classified as either "a bridge" or "not a bridge."

The process involves at least the following four steps:

1. Preprocessing and thresholding to eliminate or reduce background and unwanted noise.
2. Edge detection to produce a pair of long, parallel lines.
3. Transformation of the long, parallel lines into a representation more easily identified by the computer.
4. Test to determine pattern as "a bridge" or "not a bridge."

To eliminate or reduce the background and unwanted noise, the incoming pixels were compared to an automatically set threshold value. The pixels that were greater than the threshold value were set equal to "1" and those that were less than or equal to the threshold value were set equal to "0." The threshold value was set equal to a half of the maximum pixel value within a given frame of pixels. With this method the unwanted background noise was eliminated in most of the cases.

The edge enhancement was accomplished by using a Sobel operator to produce a pair of long, parallel edge lines of the bridge pattern.³ The Sobel operator, according to our past experience, has proved to be the best for generating clear and faithful image edges.

Detection of the pair of long, parallel lines produced by the Sobel operation in the spatial domain was relatively complicated. Therefore, the lines were transformed into the ρ and θ domain using the Hough transform.⁴ The Hough transform maps a line in Cartesian coordinate space (x and y spatial domain) into a point in polar coordinate space (ρ and θ). A family of lines with different slopes, passing through a common point in the x - y domain, transforms into a connected set of ρ - θ points. For our case, each edge point produced by the Sobel operator in

³W.K. Pratt, Digital Image Processing, New York: John Wiley and sons, Inc., 1978.

⁴W.K. Pratt, Digital Image Processing, New York: John Wiley & Sons, Inc., 1978.

the x-y domain was transformed to a curve in the ρ - θ domain, which was quantized into cells. If an element of a curve fell in a cell, that particular cell was increased by one count. After all edge points were transformed, the ρ - θ cells were examined. Large cell counts correspond to colinear edge points that may be fitted by a straight line with the appropriate (ρ, θ) parameters.⁵ The size of the transform matrix was selected to be 64 by 32 to yield enough resolution for determining the orientation of bridge patterns. The maximum available matrix size of the Hough transform is directly related to the size of the measurement window (input sampling array).

To determine the orientation angle of the bridge patterns against the X-axis of the measurement window, the 10 largest numbers were sorted in a decreasing magnitude order together with their indices (the I's and J's specifying the location of the transform element holding the sorted numbers) from the transform matrix. These 10 numbers were compared to a second threshold value, "ITT," which was automatically set equal to 0.75 of the maximum element value of the Hough transform. Only the numbers greater than "ITT" were screened and saved from recognition and detection of bridge patterns and their orientation angles.

The determination of the bridge orientation is as follows:

(a) First, the row number, J, of the largest screened number is found. If there is at least one other screened number in the same row, or in other words, having the same J, then that J is used for the computation of the bridge orientation angle.

(b) If the condition (a) is not met, then look for a new row with two or more screened numbers. Use that row number, J, for the computation of the bridge orientation angle.

(c) If neither condition (a) nor condition (b) is present then just use the row number, J, in which the maximum screened number resides for the computation of the bridge orientation angle.

⁵W.K. Pratt, Digital Image Processing, New York: John Wiley & Sons, Inc., 1978.

(d) The formulas for the computation of bridge angle, R_1 , are

$$R_1 = 22.5 (32-J) \text{ degrees; if } 20 < J < 32 \quad (1)$$

$$R_1 = 180 - 2.5 J \text{ degrees; if } 1 \leq J < 20 \quad (2)$$

The test for deciding whether a pattern belongs to the category of a bridge is as follows: Since a bridge pattern always displays some width in its appearance in radar imagery, each pixel, after first thresholding in the measurement window, is tested for coexistence with its neighbors. As indicated before, the measurement window after first thresholding contains pixels of only two values, either a "1" or a "0". Thus, if a pixel itself is a "1" and its left-hand neighbor is also a "1", a row count will be made. Similarly, if a pixel is a "1" and the pixel just below it is also a "1", then a column count will be made. The row and column counts are examined after the tests are made for all pixels in the measurement window. A flag is set if any one of these two counts is greater than eight. This flag is designated as "ANS." This flag, together with the existence of one of the first three conditions stated in the previous paragraph, constitutes the absolute requirement for recognizing and detecting bridge patterns and their orientation angles.

SYSTEM TEST RESULTS

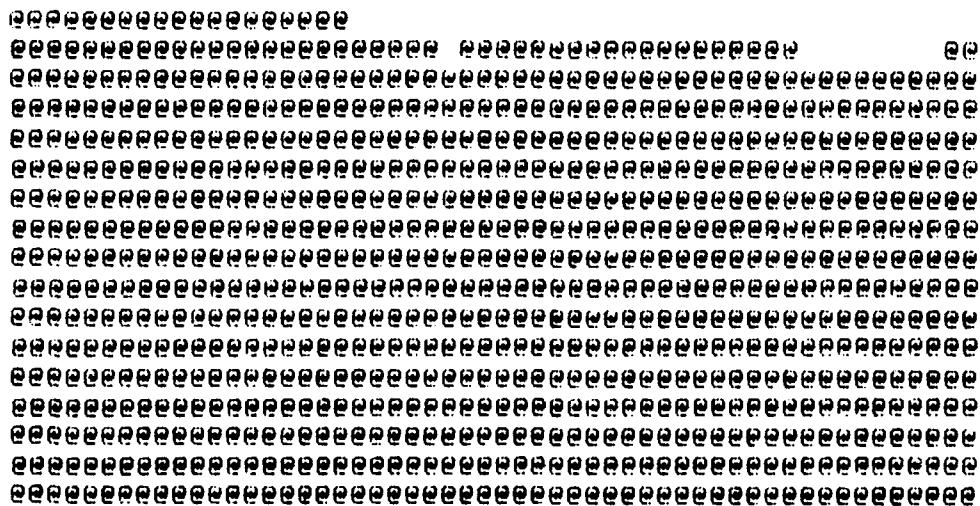
A set of high quality, scale of 1 to 100,000, X-band synthetic aperture radar imagery from the Hudson River, New York State, area consisting of bridge images (transportation DLMS category number 503, FIC 260) was used for this experimentation.

Figures 2(a) to 9(a) show the line printer output of bridge patterns with different orientation angles. Each pattern is printed in 16 gray shades by line printing. Figures 2(b) through 9(b) show the recognition results of bridges at various orientation angles. The Hough transform for these bridge patterns is illustrated in figures 2(c) to 9(c). It is seen that each bridge pattern was recognized properly and its orientation angle identified correctly. Figures 10(a), 10(b), and 10(c) demonstrate the input image, the recognition result, and the Hough transform when the measurement window sampled an entirely black area. As expected, the system recognized this image as "not a bridge." The resolution of the bridge detection with present system setup is approximately 22.5 degrees. This resolution can be increased by using a larger size measurement window. Figures 11 and 12 illustrate the bridge-recognition results for the case of using 64- by 64- pixels measurement window. As expected, the bridge-detection resolution was increased to approximately 10 degrees.

It was discovered that the bridge orientation angle determination was less accurate for bridge patterns oriented at 45 degrees and 135 degrees than for other orientations. This is due to the quantization error of the small-size measurement window (32 by 32 pixels for our case) that distorts the straight bridge-edge lines into looking like staircase patterns at 45 degrees and 135 degrees, especially when the patterns shift their position from one corner of the measurement window to the other. This error can be reduced if the size of the measurement window is enlarged at the cost of increased processing time. The software listing for this study is included as an appendix.

CONCLUSIONS

1. The Hough transform, together with the Sobel edge operator, image-thresholding methods, and decision logics, provides an effective means for detecting and recognizing various bridge patterns and their orientation angles from a set of radar imagery of the Hudson River, New York State, area.
2. The entire set of the selected radar bridge image patterns was recognized correctly with this scheme even if the bridges were not centered in the measurement window.
3. The resolution of bridge orientation angle determination with a 32- by 32-pixels measurement window is approximately 22.5 degrees. The resolution can be increased by enlarging the measurement window size as demonstrated in the report.
4. The quantization error of the measurement window can also be reduced by using a larger window size at the cost of increasing the processing time.



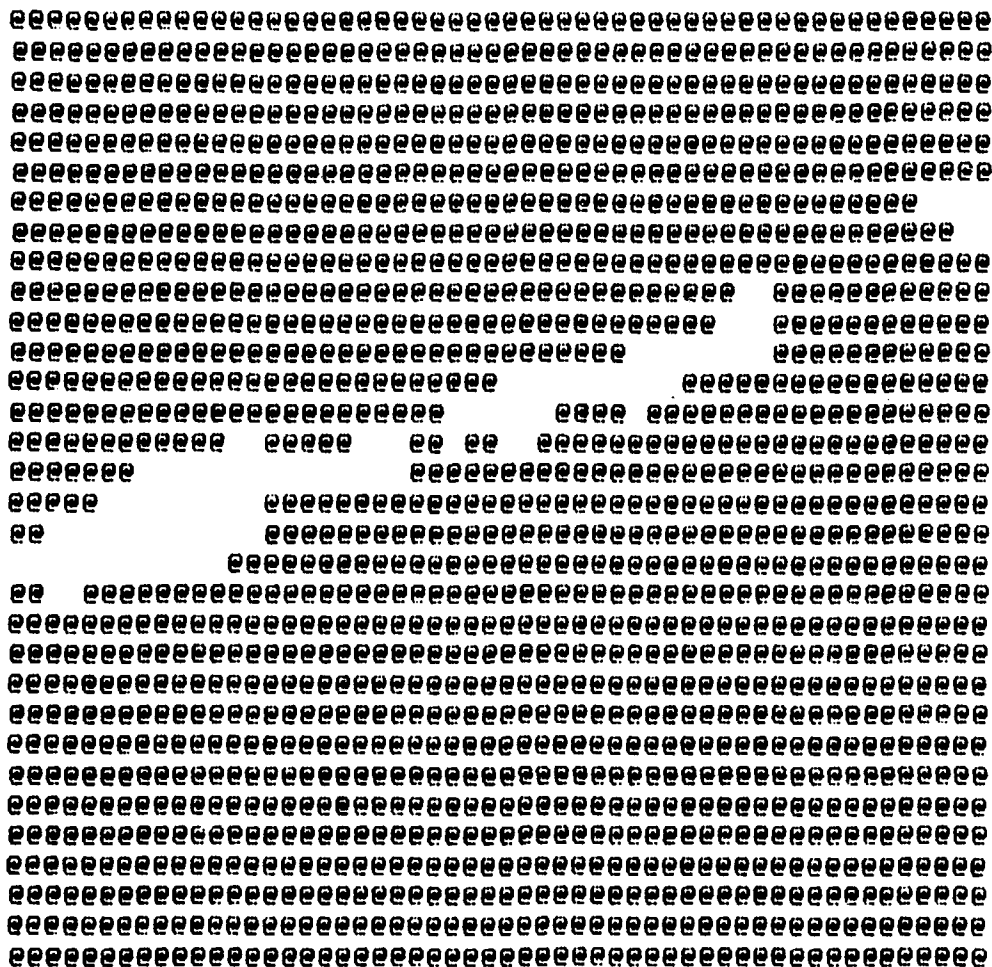
```

IROW = 30
ICOL = 3
ANS = 1
ITT = 22.
J = 32
RI = 0.0
Bridge Detected at 0.0 Degrees with X-Axis.

```

10

[illegible]

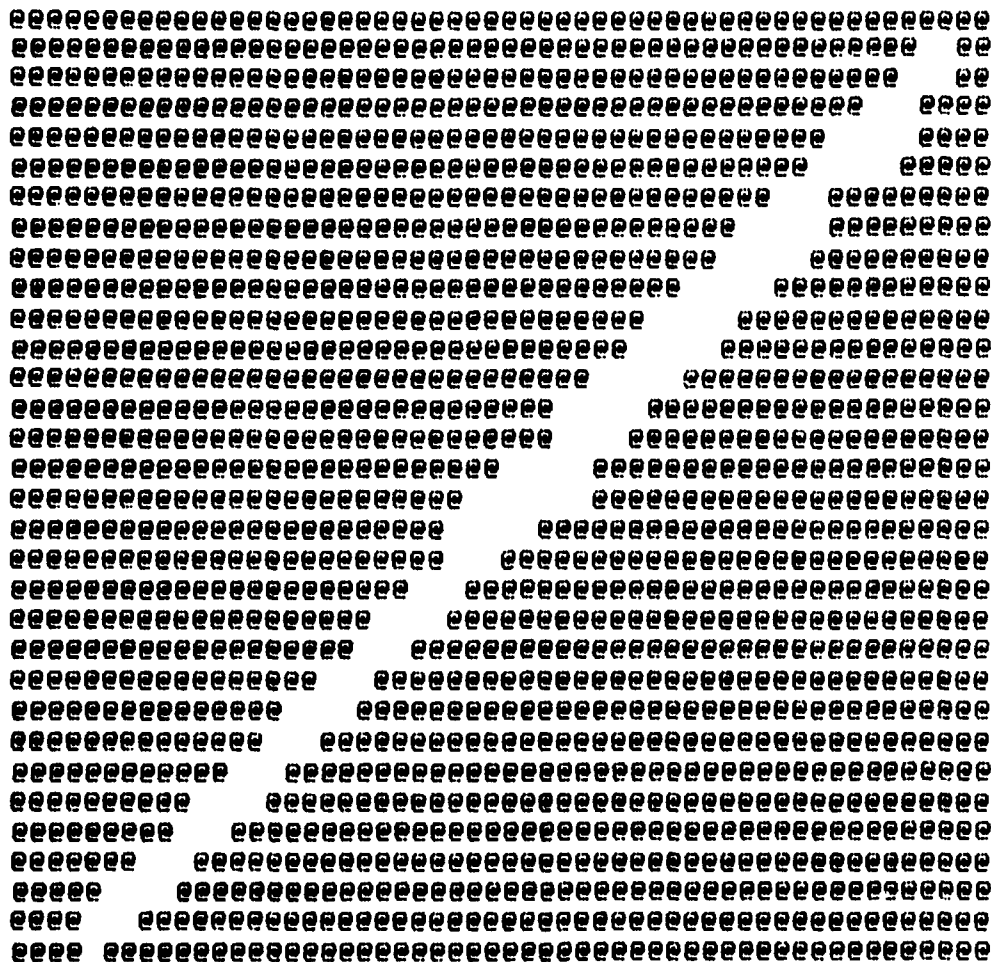


(a) Pictorial Print of Processed Input Image.

IROW = 9
 ICOL = 7
 ANS = 1
 ITT = 22.
 J = 31
 R1 = 22.5
 Bridge Detected at 22.5 Degrees with X-Axis.

FIGURE 3. (b) Recognition Result of Bridge Pattern at 22.5 Degrees.

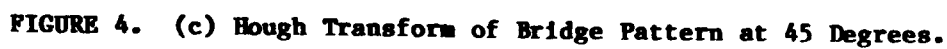
FIGURE 3. (c) Hough Transform of Bridge Pattern at 22.5 Degrees.

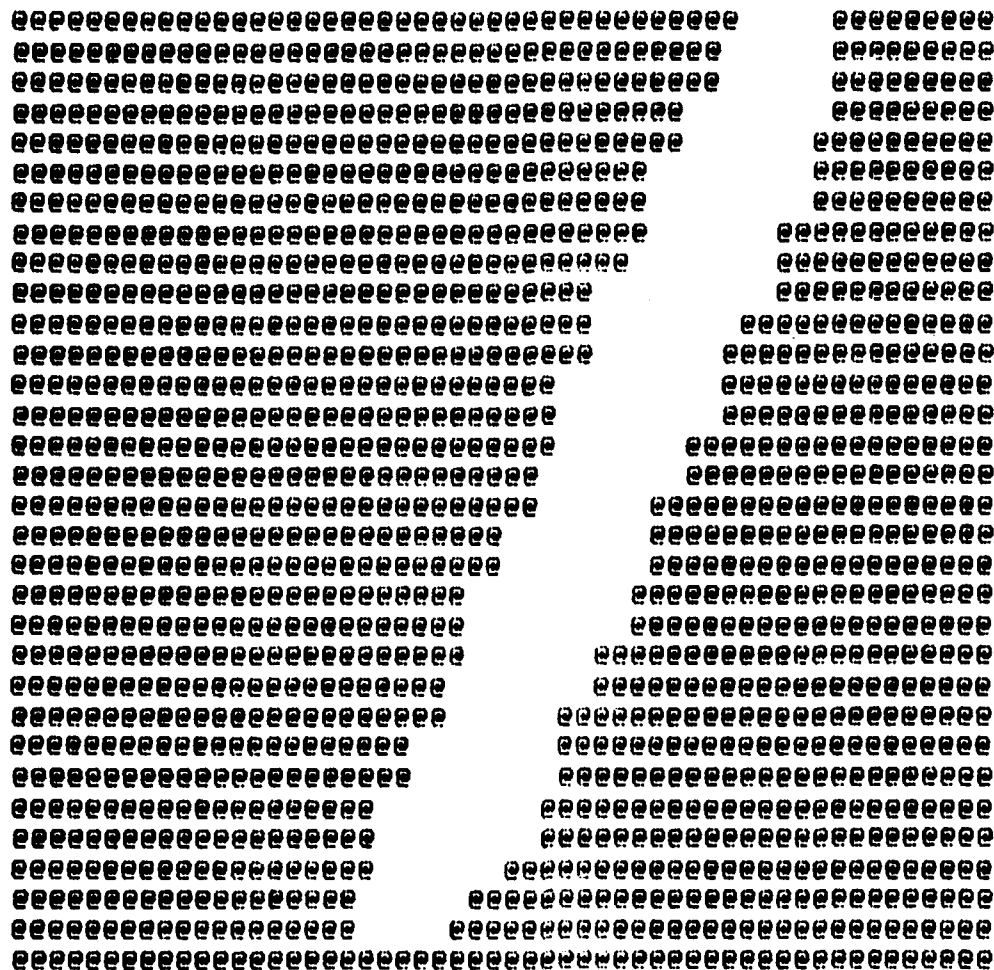


(a) Pictorial Print of Processed Input Image.

IROW = 14
 ICOL = 12
 ANS = 1
 ITT = 22.
 J = 30
 R1 = 45.0
 Bridge Detected at 45.0 Degrees with X-Axis.

FIGURE 4. (b) Recognition Result of Bridge Pattern at 45 Degrees.





(a) Pictorial Print of Processed Input Image.

IROW = 15
 ICOL = 30
 ANS = 1
 ITT = 22.
 J = 29
 R1 = 67.5
 Bridge Detected at 67.5 Degrees with X-Axis.

FIGURE 5. (b) Recognition Result of Bridge Pattern at 67.5 Degrees.

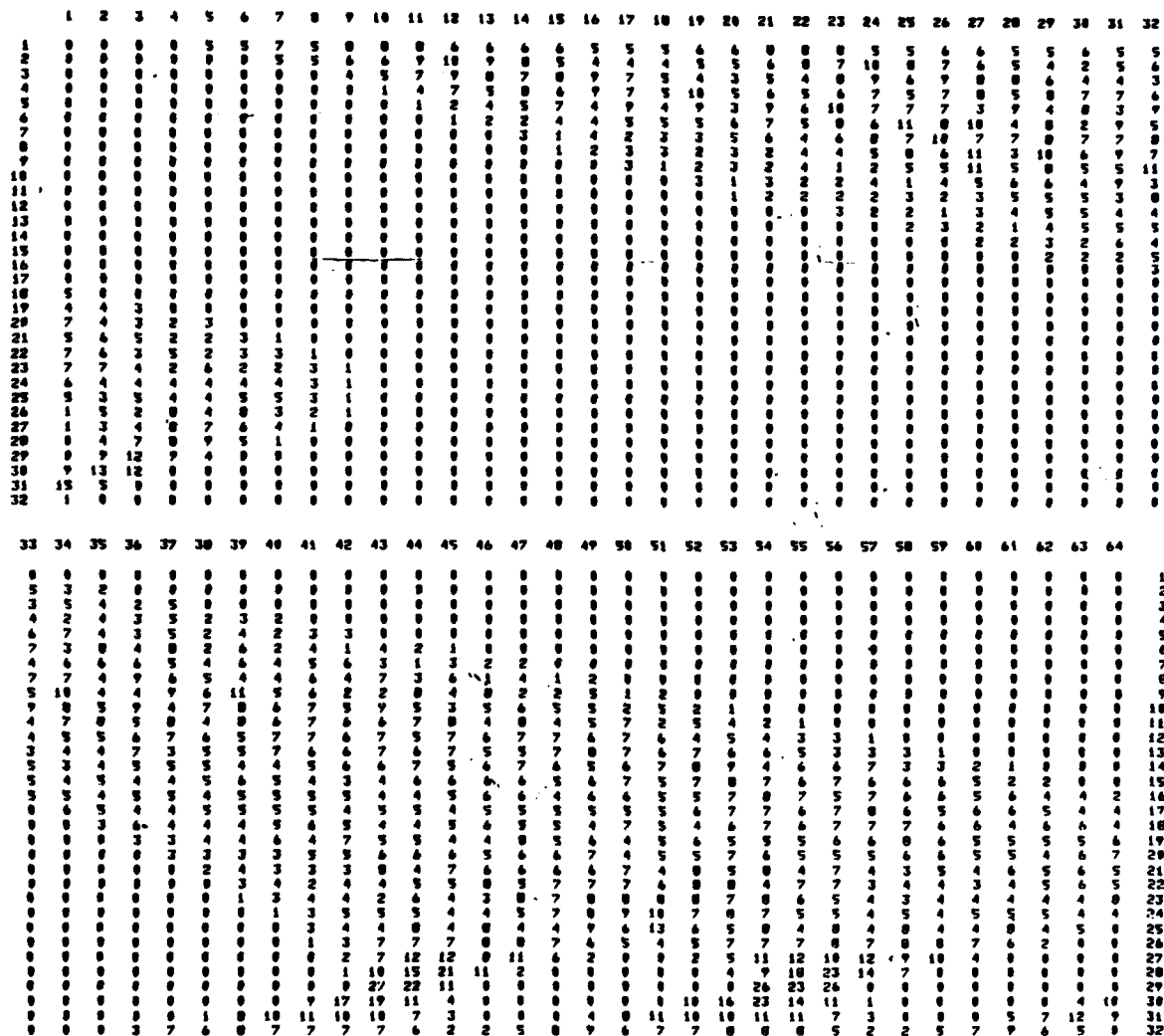
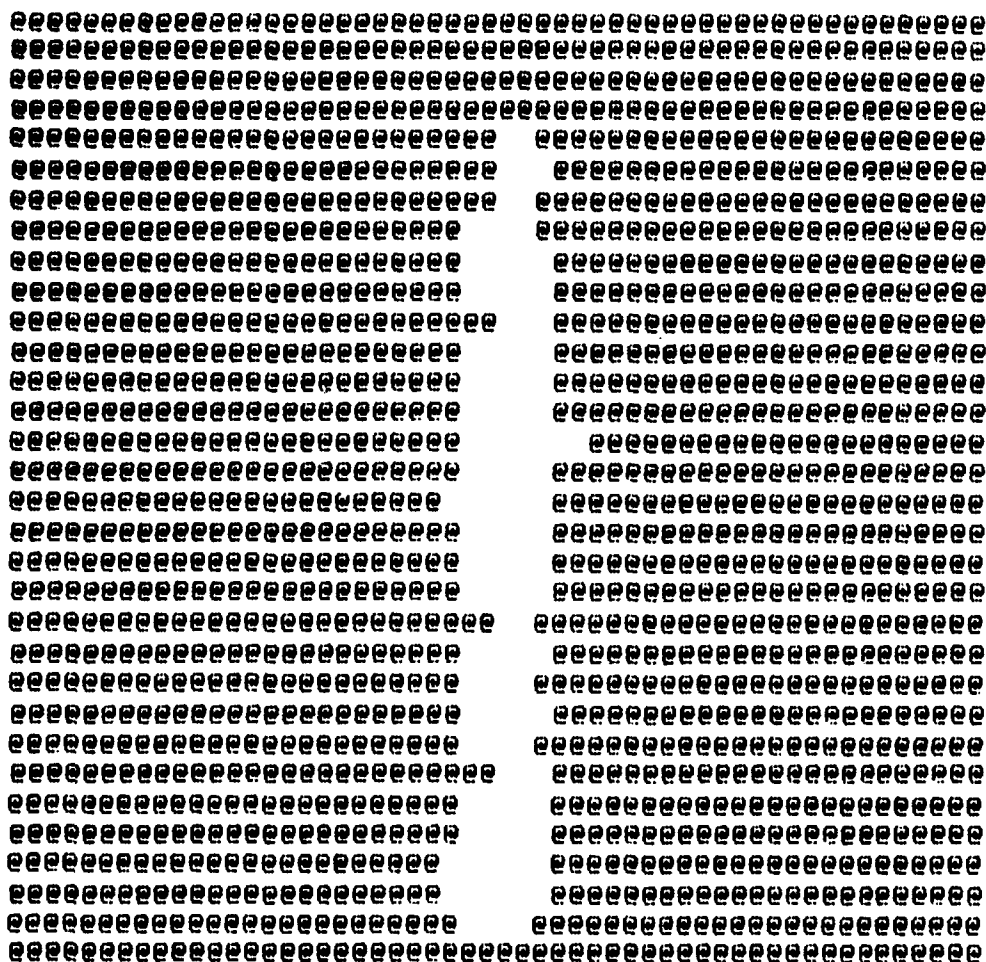


FIGURE 5. (c) Hough Transform of Bridge Pattern at 67.5 Degree.

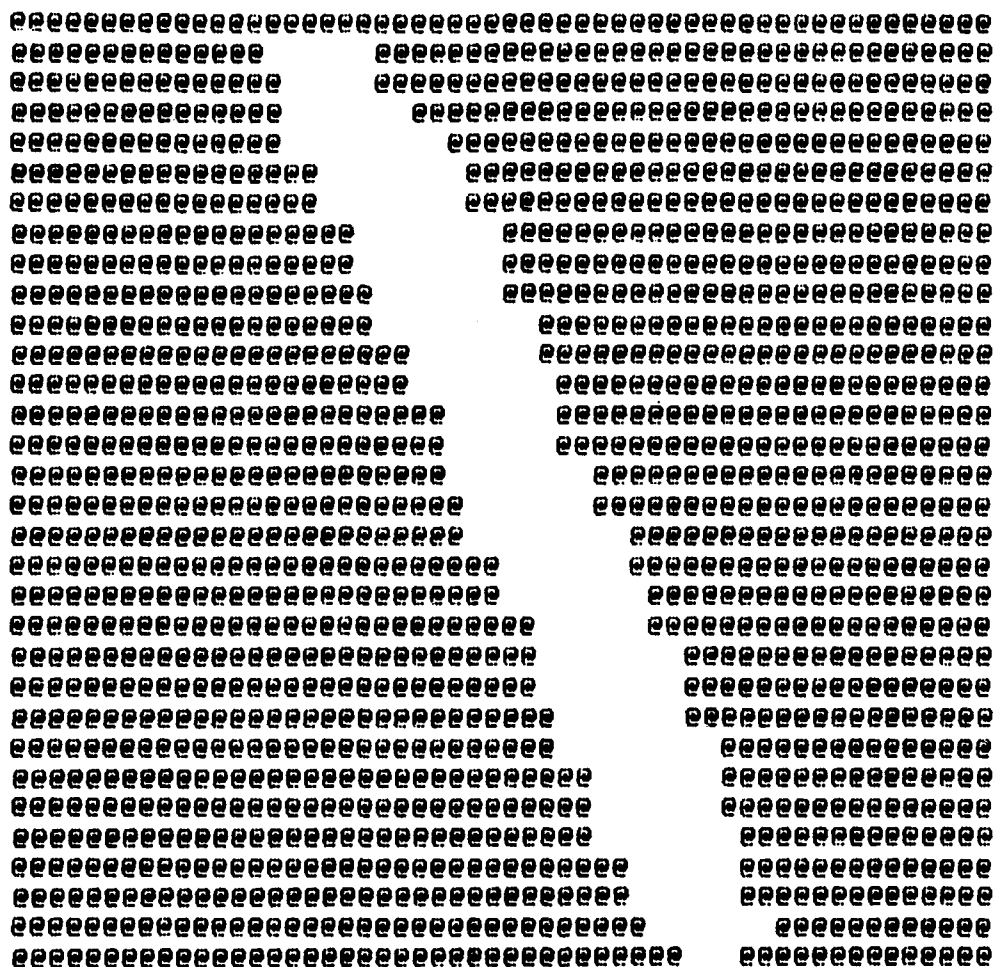


(a) Pictorial Print of Processed Input Image.

IROW = 2
 ICOL = 13
 ANS = 1
 ITT = 22.
 J = 28
 R1 = 90.0
 Bridge Detected at 90.0 Degrees with X-Axis.

FIGURE 6. (b) Recognition Result of Bridge Pattern at 90 Degrees.

FIGURE 6. (c) Hough Transform of Bridge Pattern at 90 Degrees.



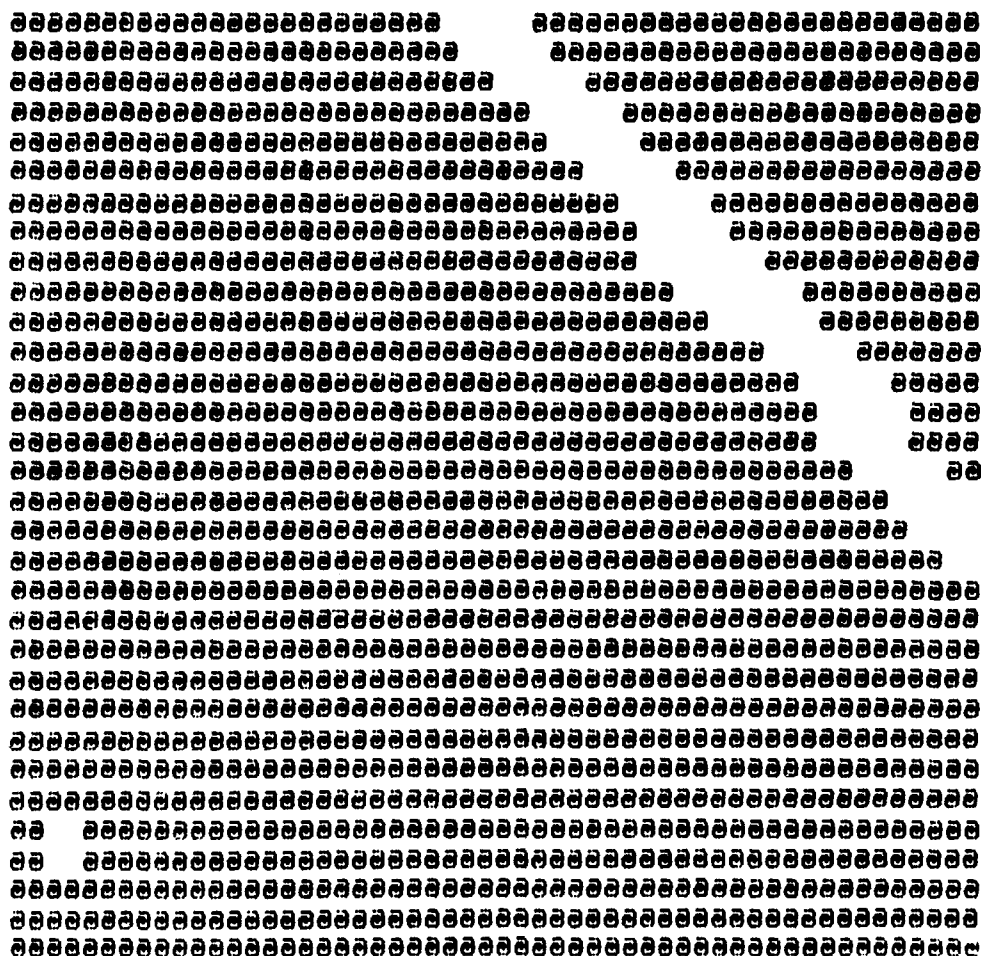
(a) Pictorial Print of Processed Input Image.

IROW = 15
 ICOL = 30
 ANS = 1
 ITT = 29.
 J = 27
 R1 = 12.5
 Bridge Detected at 112.5 Degrees with X-Axis.

FIGURE 7. (b) Recognition Result of Bridge Pattern at 112.5 Degrees.

[illegible]

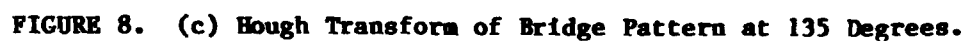
FIGURE 7. (c) Hough Transform of Bridge Pattern at 112.5 Degrees.

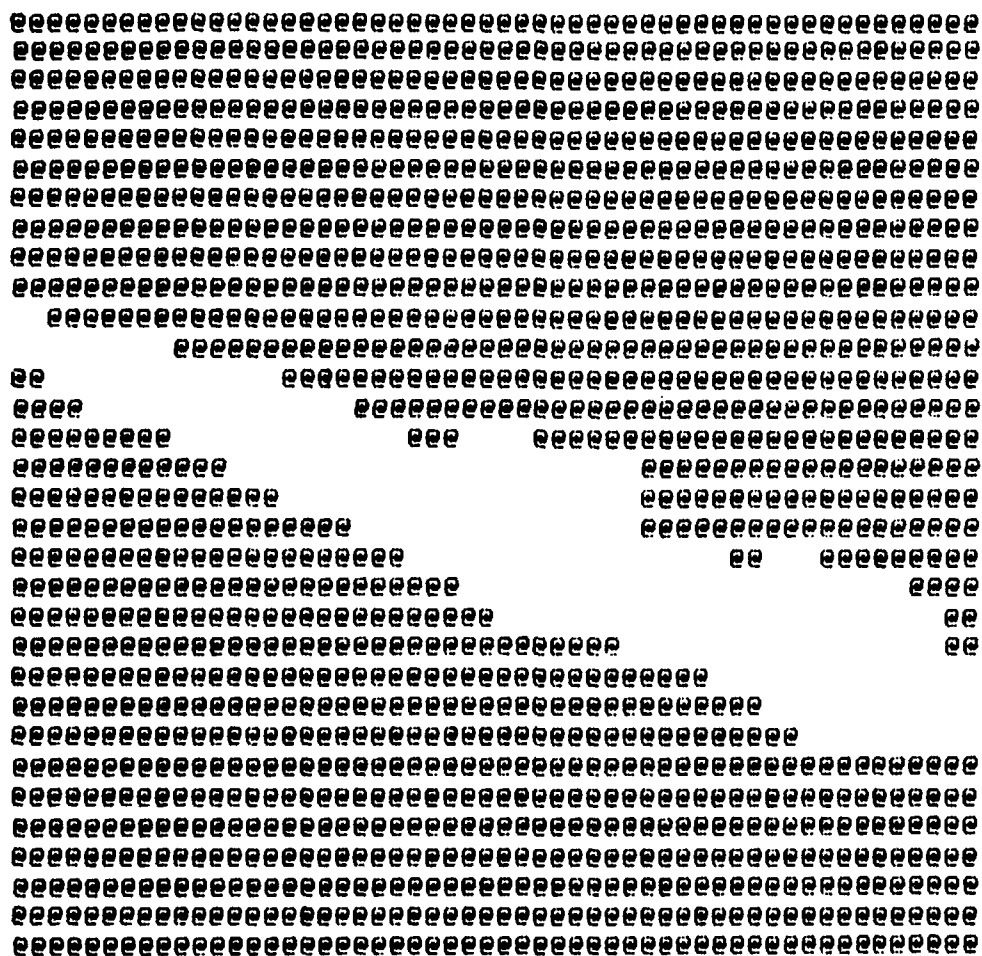


(a) Pictorial Print of Processed Input Image.

IROW = 14
 ICOL = 15
 ANS = 1
 ITT = 18.
 J = 26
 RI = 135.0
 Bridge Detected at 135.0 Degrees with X-Axis.

FIGURE 8. (b) Recognition Result of Bridge Pattern at 135 Degrees.





(a) Pictorial Print of Processed Input Image.

IROW = 29
 ICOL = 14
 ANS = 1
 ITT = 14.
 J = 25
 R1 = 157.5
 Bridge Detected at 157.5 Degrees with X-Axis.

FIGURE 9. (b) Recognition Result of Bridge Pattern at 157.5 Degrees.

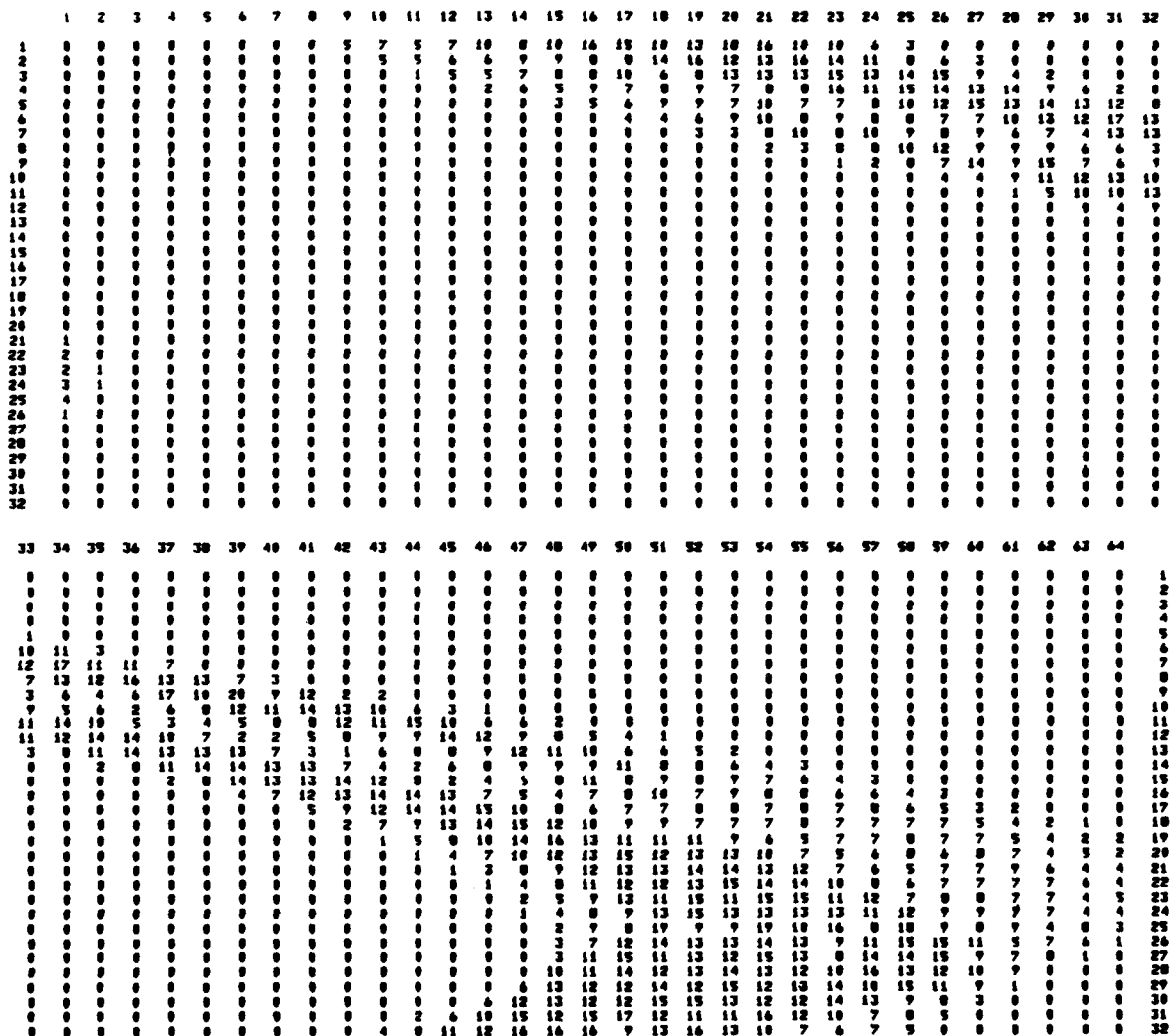
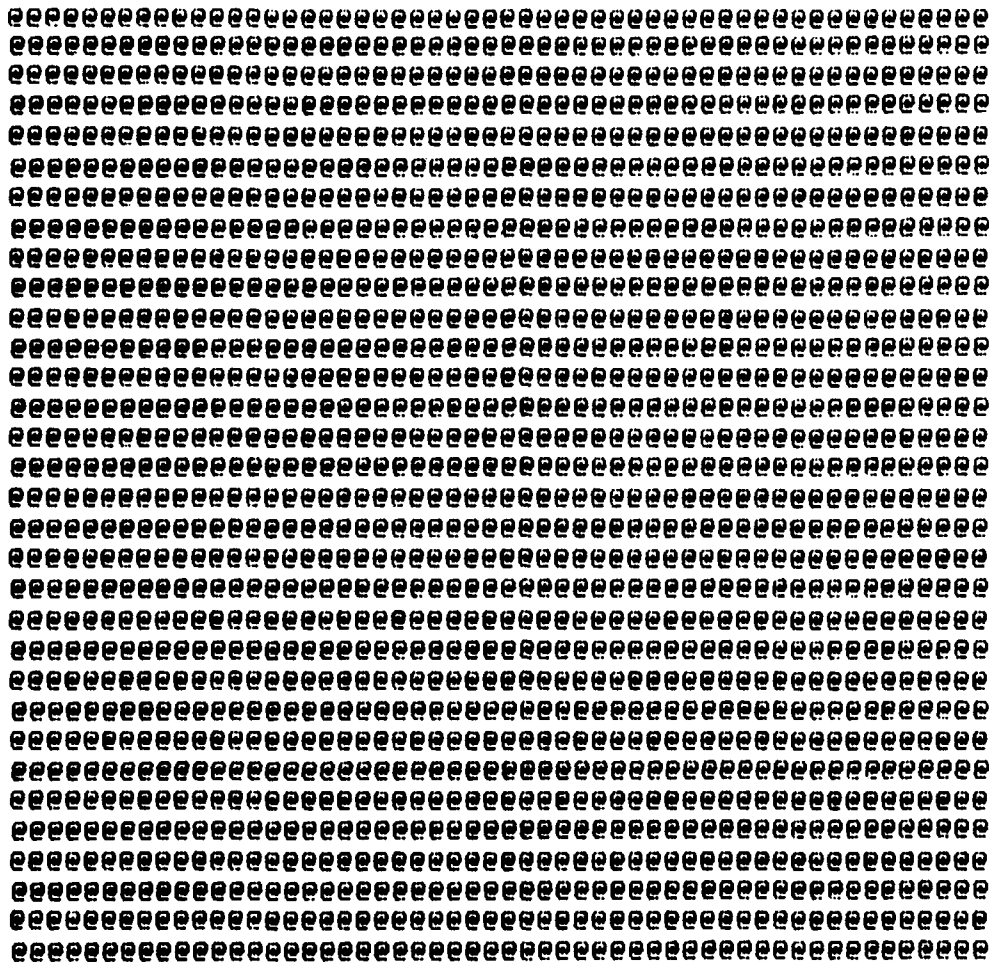


FIGURE 9. (c) Hough Transform of Bridge Pattern at 157.5 Degrees.



(a) Pictorial Print of Processed Input Image.

IROW = 0
ICOL = 0
ANS = 0
ITT - 47.
J = 25
R1 = 157.5
"not a bridge."

FIGURE 10. (b) Recognition Result of an Entirely Black Area.

FIGURE 10. (c) Hough Transform of an Entirely Black Area.

WRTICE DELETED AT 162.00 DEGREES WITH X AXIS

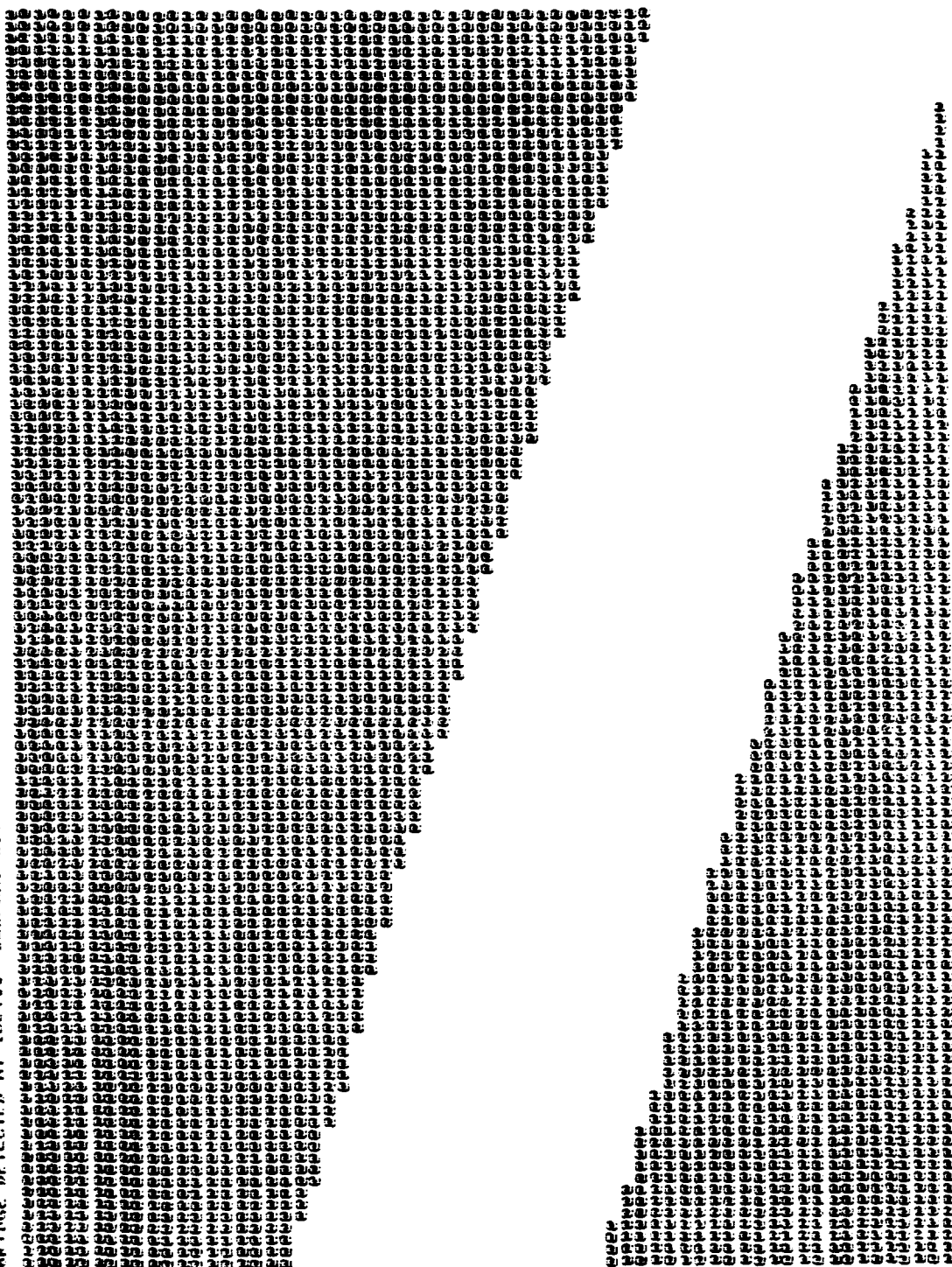


FIGURE 11. Pictorial Print of Processed Input Image, and Recognition Result of Bridge Pattern at 162 Degrees with Large Measurement Window.

BRIDGE DETECTED AT 172.00 DEGREES WITH X AXIS

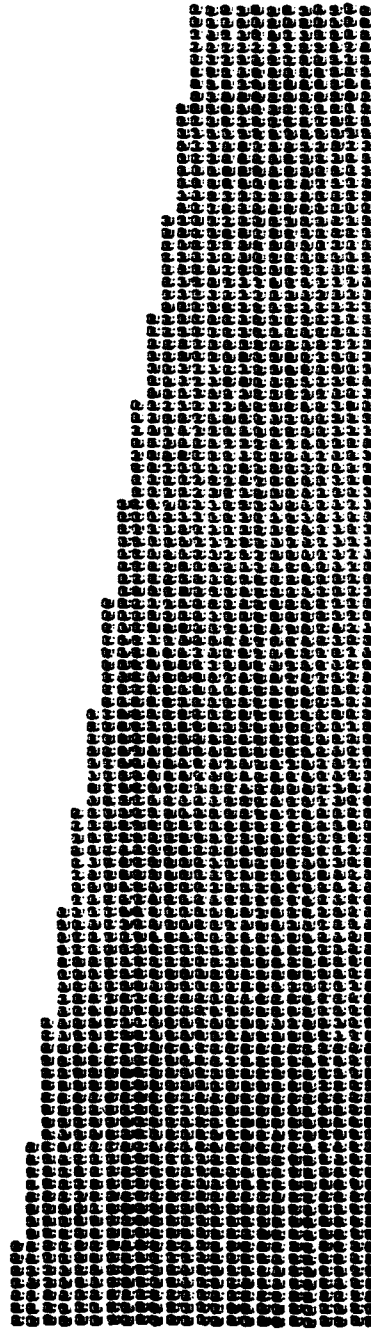
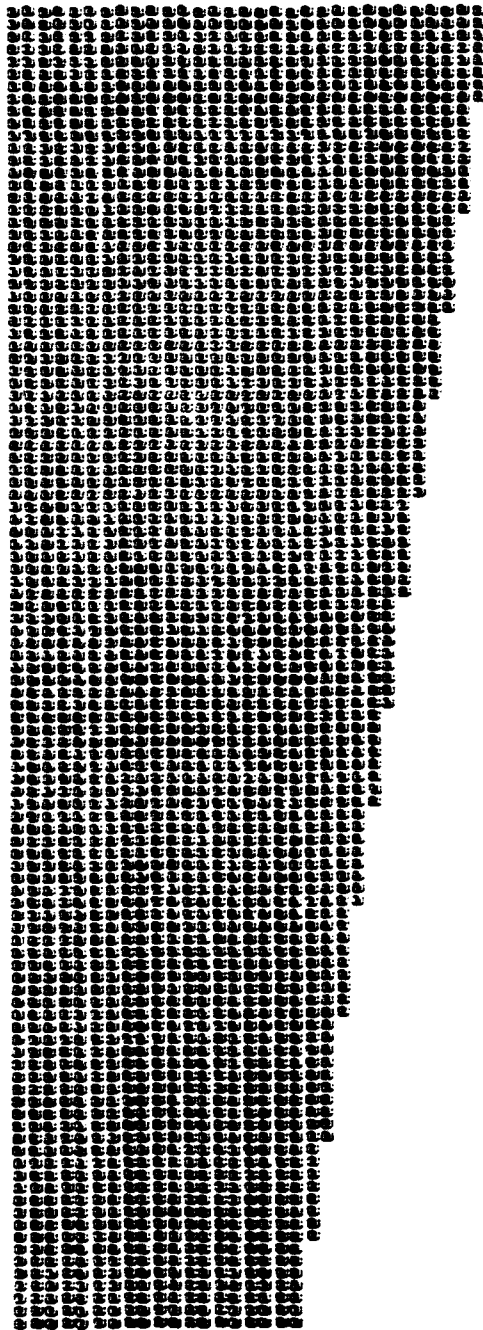


FIGURE 12. Pictorial Print of Processed Input Image, and Recognition Result of Bridge Pattern at 172 Degrees with Large Measurement Window.

**APPENDIX
SOFTWARE LISTING**

ARDS 1-00004 IS ON CK00009 USING 00071 BLKS R=0000

```
0001 FIN4.L
0002 C*****PROGRAM "RDS"-----REV 05/14/82*****
0003 C THIS PROGRAM RECOGNIZES A BRIDGE OVER WATER USING THE Hough
0004 C TRANSFORM WHICH CONVERTS A LINE INTO A POINT. THE EDGES ARE
0005 C DETECTED USING A SOBEL OPERATION (SUBROUTINE "SOBEL").
0006 C MODIFIED TO RUN IN THE RTE OP. SYS.
0007 C
0008 C
0009 C
0010 C
0011 PROGRAM RDS
0012 COMMON IXPOS,IYPOS,IRUN,IMAGE,IA1
0013 COMMON LEXLU
0014 DIMENSION IA3(1026)
0015 EQUIVALENCE(IA1,IA3(3))
0016 DIMENSION IA1(1024),IA2(1024)
0017 DIMENSION IEDGE(1024),IDCB(144),IXBUF(49),IYBUF(49)
0018 DIMENSION IXFRM(64,32),IA(64,32),MAX(11),JMAX(11)
0019 DIMENSION JMAX(11)
0020 DIMENSION IDARK(512),INPUT(43),LABIN(10),LABPT(10),LABEG(10)
0021 DIMENSION NAME(3),LARMU(10),LABA(10)
0022 DIMENSION IWRD(7),IWR2(7)
0023 DATA INPUT/2HIS,2HIN,2HPS,2HPI,2HJP,2HTR,2HSR,2HI1,2HFV,2HIH,2HFD,
0024 12HOR,2HLU,2HXY,2HRS,2HNI,2HDK,2HNA,2HIM,2HRU,2HRA,2HRI,2HGR,2HPA,
0025 22HGL,2HMA,2HMN,2HTF,2HST,2HNG,2HHS,2HAP,2HSP,2HUE,2HPE,2HHU,2HPH,
0026 32HTI,2HRD,2HTS,2HLX,2HFA,2HFA/
0027 DATA LABIN/2HIN,2HPU,2HT ,7*2H /
0028 DATA LABPT/2HIN,2HPU,2HT ,2HSC,2HAL,2HED,4*2H /
0029 DATA IYES/2HYES/
0030 DATA LABEG/2HED,2HGE,8*2H /
0031 C
0032 C SET VARIABLES TO DEFAULT VALUES
0033 C
0034 IC=0
0035 CALL RMPAR(IA1)
0036 IF (IA1.EQ.0)IA1=1
0037 LU=IA1
0038 LUOT=LU
0039 CALL LUNT(75B,LUWAT,I)
0040 ITI=0
0041 3 WRITE(LU,2)
0042 2 FORMAT("STEP FILE NAME?")
0043 READ(LU,30)NAME
0044 4 CALL OPEN(IDCB,I,NAME)
0045 CALL FEROR(I,4,LU)
0046 IF (.LT.0)GO TO 3
0047 CALL READF(IDCB,IERR,IXBUF,49,IMAGS)
0048 CALL READF(IDCB,IERR,IYBUF,49,IMAGS)
0049 CALL CLOSE(IDCB)
0050 IMAGS=IMAGS+1
0051 WRITE(LU,1)IMAGS
0052 1 FORMAT("NO. OF IMAGES=",I2)
0053 NAV=5
0054 IXPOS=0
0055 IYPOS=0
0056 IMAGE=1
0057 IRUN=0
0058 NGRAY=16
```

```

0059      IHH=0
0060      ISS=1
0061      IDIR=1
0062      IDR=2
0063      III=0
0064      MAX=0
0065      MIN=0
0066      NCHRS=72
0067      IDIM=32
0068      IA3(1)=MAX
0069      IA3(2)=MIN
0070      IR=-1
0071 C
0072 C      INPUT COMMAND LOOP
0073 C
0074 C      DK--INPUT ARRAY DARK LEVELS
0075 C      LU--INPUT LIST LU
0076 C      IN--INPUT ARRAY DATA
0077 C      IT--INPUT THRESHOLD CONSTANT FOR INPUT IMAGE
0078 C      PI--PRINT THE INPUT ARRAY
0079 C      TR--TERMINATE PROGRAM
0080 C      TH--THRESHOLD FOR BACKGROUND
0081 C      SP--SOBEL OPERATION
0082 C      OE--OUTPUT RESULT OF SOBEL OPERATION
0083 C      HU--HOUGH TRANSFORM
0084 C      PH--OUTPUT HOUGH TRANSFORM MATRIX
0085 C      TT--THRESHOLD VALUE FOR DETECTING AN EDGE
0086 C      BD--BRIDGE DETECTION
0087 C      TS--INPUT THRESHOLD VALUE BEFORE SOBEL OPERATION
0088 C      LX--DISPLAY RESULT OF BRIDGE DETECTION ROUTINE ON LEXIDATA
0089 C      PE--DISPLAY HARD COPY OF INPUT IMAGE
0090 C      XY--SLEW TO AN IMAGE
0091 C      IS--INPUT STAGE STEPS FOR NEXT IMAGE
0092 C      OR--INPUT ORIENTATION VALUE
0093 C      RS--RESET TO FIRST IMAGE
0094 C      NI--SLEW TO NEXT IMAGE IN SEQUENCE
0095 C      NA--INPUT NUMBER OF FRAMES AVERAGED
0096 C      IM--SLEW TO SELECTED IMAGE
0097 C      RU--AUTOMATIC RUN THROUGH IMAGES
0098 C      RA--RUN AGAIN STARTING AT STATEMENT 3
0099 C      RI--REMOVE SELECTED IMAGE
0100 C
0101 10      WRITE(LU,20)
0102 20      FORMAT("??")
0103      READ(LU,30)ICMND
0104 30      FORMAT(3A2)
0105      IF (ICMND.EQ.INPUT(1))GO TO 100
0106      IF (ICMND.EQ.INPUT(2))GO TO 200
0107      IF (ICMND.EQ.INPUT(8))READ(LU,*)I11
0108      IF (ICMND.EQ.INPUT(4))GO TO 300
0109      IF (ICMND.EQ.INPUT(10))READ(LU,*)ITH
0110      IF (ICMND.EQ.INPUT(11))GO TO 6000
0111      IF (ICMND.EQ.INPUT(20))GO TO 1300
0112      IF (ICMND.EQ.INPUT(15))CALL RESET
0113      IF (ICMND.EQ.INPUT(16))GO TO 1000
0114      IF (ICMND.EQ.INPUT(18))READ(LU,*)NAV
0115      IF (ICMND.EQ.INPUT(19))GO TO 1200
0116      IF (ICMND.EQ.INPUT(21))GO TO 2100
0117      IF (ICMND.EQ.INPUT(22))GO TO 2300
0118      IF (ICMND.EQ.INPUT(17))GO TO 1100

```

```

0119      IF (ICMND.EQ.INPUT(13))GO TO 500
0120      IF (ICMND.EQ.INPUT(6))GO TO 1400
0121      IF (ICMND.EQ.INPUT(33)) GOTO 4000
0122      IF (ICMND.EQ.INPUT(35)) GOTO 4200
0123      IF (ICMND.EQ.INPUT(37)) GOTO 4600
0124      IF (ICMND.EQ.INPUT(34)) GOTO 4100
0125      IF (ICMND.EQ.INPUT(36)) GOTO 4500
0126      IF (ICMND.EQ.INPUT(38))READ(LU,*)ITT
0127      IF (ICMND.EQ.INPUT(14))GOTO 600
0128      IF (ICMND.EQ.INPUT(40))READ(LU,*)IT2
0129      IF (ICMND.EQ.INPUT(41)) GOTO 987
0130      IF (ICMND.EQ.INPUT(24)) GO TO 6800
0131 31    FORMAT("1")
0132      GO TO 10
0133 C
0134 C      STORE STAGE STEPS FOR NEXT IMAGE OR INSERTED IMAGE
0135 C
0136 C      SLEW TO STARTING IMAGE, 0 INPUT ABORTS
0137 C      SECTION.
0138 C
0139 C
0140 100    WRITE(LU,101)
0141 101    FORMAT("START AT IMAGE 0?")
0142      READ(LU,*)I
0143      IF (I.GT.IMAGS.OR.I.LT.1)GO TO 10
0144      IRUN=2
0145      CALL STEPS(IXBUF,IYBUF,IMAGE,I,IXSTP,IYSTP)
0146      IXPOS=IXPOS+IXSTP
0147      IYPOS=IYPOS+IYSTP
0148      IMAGE=I
0149      CALL SSLEW(-IXSTP,-IYSTP,1)
0150 C
0151 C      INPUT ADD OR INSERT IMAGE FLAG, "ICMND", AND MOVE TO
0152 C      NEW IMAGE WITH KBRD CONTROL.
0153 C
0154      WRITE(LU,120)
0155 120    FORMAT("ADD(1) OR INSERT(2) IMAGES?")
0156      READ(LU,*)ICMND
0157 107    WRITE(LU,102)
0158 102    FORMAT("MOVE STAGES WITH KBRD UNTIL DESIRED IMAGE IS")
0159      WRITE(LU,103)
0160 103    FORMAT("DISPLAYED. ENTER 1 TO STORE NEW IMAGE POSITION")
0161      WRITE(LU,109)
0162 109    FORMAT("ENTER 2 FOR FEATURE DETECTION")
0163      IXD=0
0164      IYD=0
0165 105    READ(LU,*)IXSTP,IYSTP
0166      IF (IXSTP.EQ.1)GO TO 104
0167      IF (IXSTP.EQ.2)GO TO 200
0168      CALL SSLEW(-IXSTP,-IYSTP,1)
0169      IXD=IXD+IXSTP
0170      IYD=IYD+IYSTP
0171      GO TO 105
0172 C
0173 C      COMPUTE NEW STAGE POSITIONS RELATIVE TO STARTING
0174 C      IMAGE AND BRANCH TO ADD INSERT SECTION.
0175 C
0176 104    IXPOS=IXPOS+IXD
0177      IYPOS=IYPOS+IYD
0178      IF (ICMND.EQ.2)GO TO 111

```

```

0179 C
0180 C ADD IMAGE SECTION: STORE NEXT IMAGE STEPS
0181 C
0182 IXRUF(IMAGE)=IXD
0183 IYRUF(IMAGE)=IYD
0184 GO TO 114
0185 C
0186 C INSERT IMAGE SECTION: INSERT NEW STEPS INTO CURRENT
0187 C IMAGE INDEX, MOVE IXBUF AND IYBUF DOWN, COMPUTE STEPS
0188 C FROM NEW IMAGE TO OLD NEXT IMAGE AND STORE IN INDEX+1
0189 C
0190 111 IXSTP=IXRUF(IMAGE)-IXD
0191 IYSTP=IYRUF(IMAGE)-IYD
0192 IXBUF(IMAGE)=IXD
0193 IYRUF(IMAGE)=IYD
0194 IF(IMAGS-IMAGE.LT.2)GO TO 115
0195 DO 113 I=1,IMAGS-IMAGE-1
0196 IXBUF(IMAGS-I+1)=IXBUF(IMAGS-I)
0197 113 IYBUF(IMAGS-I+1)=IYRUF(IMAGS-I)
0198 115 IXBUF(IMAGE+1)=IXSTP
0199 IYRUF(IMAGE+1)=IYSTP
0200 IMAGS=IMAGS+1
0201 C
0202 C ASK IF TO CONTINUE AND IF NOT STORE STEP FILE
0203 C IN DESIGNATED NAME
0204 C
0205 114 IMAGE=IMAGE+1
0206 IF(IMAGE.EQ.50.OR.IMAGS.EQ.50)GO TO 108
0207 WRITE(LU,106)
0208 106 FORMAT("CONTINUE?")
0209 READ(LU,30)I
0210 IF(I.EQ.IYES)GO TO 107
0211 108 IF(ICMND.EQ.1)IMAGS=IMAGE
0212 112 WRITE(LU,2)
0213 READ(LU,30)NAME
0214 CALL OPEN(IDC8,I,NAME)
0215 IF(I.EQ.-6)CALL CREAT(IDC8,I,NAME,1,3,0,9)
0216 CALL WRITE(IDC8,I,IXBUF,IMAGS-1)
0217 CALL WRITE(IDC8,I,IYBUF,IMAGS-1)
0218 CALL CLOSE(IDC8)
0219 CALL RESET
0220 GO TO 10
0221 C
0222 C INPUT AND AVERAGE ARRAY FRAMES, SUBTRACT DARK
0223 C LEVELS, THRESHOLD AND ORIENT
0224 C
0225 200 CALL AVGIN(IA1,IA2,1024,NAV,IDARK)
0226 C
0227 C THE FOLLOWING STATEMENTS FIX AN APPARENT HARDWARE ERROR BY SORTING
0228 C SUCH THAT THE FIRST ROW IN "IA1" BECOMES THE LAST ROW.
0229 C
0230 DO 203 I=1,1024
0231 203 IA2(I)=IA1(I)
0232 DO 201 I=1,992
0233 201 IA1(I)=IA1(I+32)
0234 DO 202 I=993,1024
0235 202 IA1(I)=IA2(I-992)
0236 CALL IORNT(IA1,32,IOR)
0237 C SET THRESHOLD FOR INPUT ARRAY
0238 C

```

```

0239      CALL MATRI(IA1,ITH)
0240      DO 220 I=1,1024
0241      IF(IA1(I).GE.ITH)IA1(I)=1
0242 220    CONTINUE
0243      IF(IRUN.GT.0)4000,10
0244      C
0245      C      PRINT INPUT IMAGE
0246      C
0247 300    CALL PDATE(LUOT)
0248      CALL MTOUT(IA1,LABIN,32,32,3,LUOT,-NCHRS)
0249      GO TO 10
0250      C
0251      C      CHANGE PARAMETERS
0252      C
0253      C      INPUT LIST LU
0254      C
0255 500    READ(LU,*)LUOT
0256      IF (LUOT.EQ.6)NCHRS=132
0257      IF (LUOT.EQ.LU)NCHRS=72
0258      GO TO 10
0259      C
0260      C      INPUT AND PACK DARK LEVELS
0261      C
0262 1100   CALL AVGIN(IA1,IA2,1024,NAV)
0263      GOTO 10
0264      C
0265 1400   STOP
0266      C
0267      C      SLEW TO NEXT IMAGE
0268      C
0269 1000   IF(IFBRK(I).LT.0)GO TO 1010
0270      IF(IMAGE.EQ.IMAGS)GO TO 1010
0271      CALL SSLEW(-IXBUF(IMAGE),-IYBUF(IMAGE),1)
0272      IXPOS=IXPOS+IXBUF(IMAGE)
0273      IYPOS=IYPOS+IYBUF(IMAGE)
0274      IMAGE=IMAGE+1
0275      IF(IRUN.EQ.1)200,10
0276 1010   CALL RESET
0277      GO TO 10
0278      C
0279      C      SLEW TO SELECTED IMAGE
0280      C
0281 1200   READ(LU,*)I
0282      IF(I.GT.IMAGS.OR.I.LT.1)GO TO 10
0283      CALL STEPS(IXBUF,IYBUF,IMAGE,I,IXSTP,IYSTP)
0284      IXPOS=IXPOS+IXSTP
0285      IYPOS=IYPOS+IYSTP
0286      IMAGE=I
0287      CALL SSLEW(-IXSTP,-IYSTP,1)
0288      GO TO 10
0289      C
0290 1300   CALL RESET
0291      IRUN=1
0292      GO TO 200
0293      C
0294      C
0295 2300   WRITE(LU,2310)
0296 2310   FORMAT("IMAGE 0 TO BE REMOVED?")
0297      READ(LU,*)I
0298      IF(I.GT.IMAGS.OR.I.LT.1)GO TO 10

```

```

0299      IF(I.EQ.IMAGS)GO TO 2320
0300      CALL STEPS(IXBUF,IYBUF,I-1,I+1,IXD,IYD)
0301      IXBUF(I-1)=IXD
0302      IYBUF(I-1)=IYD
0303      DO 2330 IXD=1,IMAGS-2
0304      IXBUF(IXD)=IXBUF(IXD+1)
0305 2330      IYBUF(IXD)=IYBUF(IXD+1)
0306 2320      IMAGS=IMAGS-1
0307      WRITE(LU,2)
0308      READ(LU,30)NAME
0309 2350      CALL OPEN(IDCB,IXD,NAME)
0310      IF(IXD.EQ.-6)CALL CREAT(IDCB,IXD,NAME,1,3,0,9)
0311      CALL WRITE(IDCB,IXD,IXBUF,I)
0312      CALL WRITE(IDCB,IXD,IYBUF,I)
0313      CALL CLOSE(IDCB)
0314      GO TO 10
0315 C      SOBEL OPERATION : USED TO DETECT EDGES
0316 C
0317 4000      CALL SOBEL(IA1,IEDGE,32,32)
0318 C
0319 C      SET THRESHOLD FOR SOBEL ARRAY "IEDGE"
0320 C
0321      CALL MATR2(IEDGE,IT2)
0322      IDIN=32
0323      IF(IRUN.GT.0)4500,10
0324 C
0325 C      OUTPUT RESULT OF SOBEL OPERATION TO LINE PRINTER
0326 C
0327 4100      CALL MOUT(IEDGE,LABEL,IDIN,IDIN,3,LUOT,NCHRS)
0328      GOTO 10
0329 C
0330 C      PICTORIALY PRINT OUTPUT OF INPUT ARRAY
0331 C
0332 4200      CALL LPIN(IA1,32)
0333      GOTO 10
0334 C
0335 C      HOUGH TRANSFORM : USED TO CONVERT LINE INTO POINT
0336 C
0337 4500      I=1
0338 4501      CALL HOUGH(IEDGE,32,32,IT2,IXFM,64,32)
0339 C
0340      IF(IRUN.GT.0)6000,10
0341 C
0342 C      PRINT HOUGH TRANSFORM MATRIX ON LINE PRINTER
0343 C
0344 4600      CALL MOUT(IXFM,LABEL,64,32,3,LUOT,-NCHRS)
0345      GOTO 10
0346 C
0347 C      SLEW TO AN IMAGE
0348 C
0349 600      READ(LU,*)IXSTP,IYSTP
0350      IF(IXSTP.EQ.1)GO TO 10
0351      CALL SBLEW(-IXSTP,-IYSTP,1)
0352      GOTO 600
0353 907      IC=IC+1
0354      IR=IR+1
0355 C
0356 C      DISPLAY THE INPUT ARRAY "IA1" ON THE LEXIDATA. THE SUBROUTINE IS
0357 C      CAPABLE OF DISPLAYING 12 ARRAYS,(4 COLUMNS,3 ROWS). IT DISPLAYS
0358 C      THEM SEQUENTIALLY STARTING AT TOP LEFT CORNER AND ENDING AT THE

```

```

0359 C      BOTTOM RIGHT CORNER. THE WORD "BRIDGE" APPEARS IF A BRIDGE OVER
0360 C      WATER IS DETECTED. ALSO, THE ANGLE THE BRIDGE MAKES WITH THE POS-
0361 C      ITIVE X-AXIS IS SHOWN ON THE LEXIDATA.
0362 C
0363      IF (ANS.EQ.0)GOTO 10
0364      IDEG=R1
0365      CALL LXRDI(IA1, IDEG, JC, JB, ANS)
0366      GO TO 10
0367 C
0368 C      CLASSIFICATION OF BRIDGES
0369 C
0370 6000 CALL BRIDGE(IA1,ANS,ICOL,IROW)
0371      WRITE(LUOT,225)IROW
0372 225  FORMAT(IX,"IROW= ",I3)
0373      WRITE(LUOT,226)ICOL
0374 226  FORMAT(IX,"ICOL= ",I3)
0375      WRITE(LUOT,227)ANS
0376 227  FORMAT(IX,"ANS= ",I4)
0377 C      SET THRESHOLD FOR HOUGH TRANSFORM ARRAY "IXFRM"
0378      CALL MATR3(IXFRM,ITT)
0379      WRITE(LUOT,4505)ITT
0380 4505  FORMAT(IX,"ITT= ",F3.1)
0381      DO 6050 J=1,32
0382      DO 6050 I=1,64
0383      IA(I,J)=IXFRM(I,J)
0384 6050  CONTINUE
0385      DO 6700 K=1,10
0386      MX=0
0387      IMX=0
0388      JMX=0
0389      DO 6600 J=1,32
0390      DO 6600 I=1,64
0391      IF (IA(I,J).LT.MX)GO TO 6600
0392      MX=IA(I,J)
0393      IMX=I
0394      JMX=J
0395 6600  CONTINUE
0396      MAX(K)=MX
0397      JMAX(K)=IMX
0398      JMAX(K)=JMX
0399      IA(IMX,JMX)=0
0400 6700  CONTINUE
0401 6900  M=0
0402      DO 6910 K=1,10
0403      IF (JMAX(K).NE.JMAX(1))GO TO 6910
0404      IF (MAX(K).LE.ITT)GO TO 6910
0405      M=M+1
0406      IF (M.LT.2)GO TO 6910
0407      J=JMAX(1)
0408      GO TO 6100
0409 6910  CONTINUE
0410      DO 6925 I=2,10
0411      M=0
0412      DO 6920 K=2,10
0413      IF (JMAX(K).NE.JMAX(I))GO TO 6920
0414      IF (MAX(K).LE.ITT)GO TO 6920
0415      M=M+1
0416      IF (M.LT.2)GO TO 6920
0417      J=JMAX(I)
0418      GO TO 6100

```

```

0419 6920 CONTINUE
0420 6925 CONTINUE
0421 J=JMAX(1)
0422 GO TO 6100
0423 C
0424 6800 CALL M1001(JA,LABA,64,32,3,LUOT,-NCHRB)
0425 6100 WRITE(LUOT,6150)J
0426 6150 FORMAT(IX,"J= ",I3)
0427 R1=(32-J)*22.5
0428 IF(J.LT.20)R1=180-J*2.5
0429 WRITE(LUOT,6160)R1
0430 6160 FORMAT(IX,"R1= ",F5.1)
0431 IF(ANS.EQ.1.0)WRITE(LUOT,6170)R1
0432 IF(ANS.EQ.0.0)WRITE(LUOT,6180)
0433 6170 FORMAT(IX,"BRIDGE DETECTED AT ",F5.1," DEGREES WITH X-AXIS")
0434 6180 FORMAT(IX,"THIS IS NOT A BRIDGE")
0435 IF(IRUN.EQ.1)GO TO 6190
0436 IF(IRUN.EQ.2)105,10
0437 6190 CALL EXEC(1,LUOT,150,1)
0438 GO TO 1000
0439 C RESET THE MAIN PROGRAM AND INITIAL VALUES
0440 C
0441 2100 CALL RESET
0442 GO TO 3
0443 END
0444 C
0445 C THIS IS A SUBROUTINE TO RESET THE MAIN PROGRAM "RD" TO ITS
0446 C INITIAL STAGES
0447 C
0448 SUBROUTINE RESET
0449 COMMON IXPOS,IYPOS,IRUN,IMAGE
0450 IMAGE=1
0451 IRUN=0
0452 CALL SBLEW(IXPOS,IYPOS,1)
0453 IXPOS=0
0454 IYPOS=0
0455 RETURN
0456 END
0457 C
0458 C SUBROUTINE TO CALCULATE THE NUMBER OF STEPS BETWEEN
0459 C IMAGES IM1 AND IM2 FROM THE SUCCESSIVE DISTANCES
0460 C STORED IN "IXBUF" AND "IYBUF".
0461 C
0462 SUBROUTINE STEPS(IXBUF,IYBUF,IM1,IM2,IXSTP,IYSTP)
0463 DIMENSION IXBUF(1),IYBUF(1)
0464 IXSTP=0
0465 IYSTP=0
0466 IF(IM1.EQ.IM2)RETURN
0467 M=IM1
0468 N=IM2
0469 IF(IM1.LT.IM2)GO TO 20
0470 M=IM2
0471 N=IM1
0472 20 DO 30 I=M,N-1
0473 IXSTP=IXBUF(I)+IXSTP
0474 IYSTP=IYBUF(I)+IYSTP
0475 30 IF(IM1.LT.IM2)RETURN
0476 IXSTP=-IXSTP
0477 IYSTP=-IYSTP
0478 RETURN

```

```

0479      END
0480 C
0481 C      AUTOMATIC THRESHOLD FOR INPUT IMAGE. THE ARRAY IS CALLED "IA1".
0482 C
0483      SUBROUTINE MATR1(IA1,ITH)
0484      DIMENSION IA1(1024)
0485      SUM=0.
0486      IMAX=IA1(1)
0487      DO 28 I=2,1024
0488      IF(IA1(I).GT.IMAX)IMAX=IA1(I)
0489 28      CONTINUE
0490      ITH=.50*FLOAT(IMAX)
0491      DO 38 I=1,1024
0492      IF(ITH.GT.IA1(I)) IA1(I)=0
0493 38      CONTINUE
0494      RETURN
0495      END
0496 C
0497 C      SET THRESHOLD FOR SOBEL OPERATION. USES ARRAY "IEDGE".
0498 C
0499      SUBROUTINE MATR2(IEDGE,IT2)
0500      DIMENSION IEDGE(1024)
0501      SUM=0.
0502      IMAX=IEDGE(1)
0503      DO 28 I=2,1024
0504      IF(IEDGE(I).GT.IMAX)IMAX=IEDGE(I)
0505 28      CONTINUE
0506      IT2=FLOAT(IMAX)*.55
0507      DO 38 I=1,1024
0508      IF(IT2.GT.IEDGE(I))IEDGE(I)=0
0509 38      CONTINUE
0510      RETURN
0511      END
0512 C
0513 C      SETS THRESHOLD FOR THE HOUGH TRANSFORM. ARRAY NAME:"IXFRM".
0514 C
0515      SUBROUTINE MATR3(IXFRM,ITT)
0516      DIMENSION IXFRM(64,32)
0517      IMAX=IXFRM(1,1)
0518      DO 28 I=1,64
0519      DO 28 J=1,32
0520      IF(IXFRM(I,J).GT.IMAX)IMAX=IXFRM(I,J)
0521 28      CONTINUE
0522      ITT=IMAX*.75
0523      RETURN
0524      END
0525      SUBROUTINE LXRD(IA1,IDEG,IC,IB,ANS)
0526      *****
0527      *****THIS DISPLAYS A 32X32 ARRAY AND THE RESULT OF A BRIDGE DETECTION
0528      *****ROUTINE*****
0529      *****
0530      COMMON LEXLU
0531      DIMENSION IA1(1024),INUM(2),IWD(3)
0532      DATA IWD/2HRR,2HID,2HGE/
0533      IONES=MOD(IDEG,10)
0534      IDEG=(IDEG-IONES)/10
0535      ITENS=MOD(IDEG,10)
0536      IDEG=(IDEG-ITENS)/10
0537      IHUN=MOD(IDEG,10)
0538      INUM(1)=254*(IHUN+48)+(ITENS+48)

```

```

0539      INUM(2)=256*(IONES+48)
0540      IR=IC-1
0541      IL=0
0542      IR=IC-1
0543      IF(IC.GT.4)IL=1
0544      IF(IC.GT.8)IL=2
0545      IF(IR.EQ.4)IR=0
0546      IF(IR.EQ.5)IR=1
0547      IF(IR.EQ.6)IR=2
0548      IF(IR.EQ.7)IR=3
0549      IF(IR.EQ.8)IR=0
0550      IF(IR.EQ.9)IR=1
0551      IF(IR.EQ.10)IR=2
0552      IF(IR.EQ.11)IR=3
0553      IF(IR.EQ.12)IC=0
0554      IF(IR.EQ.12)IL=0
0555      IF(IR.EQ.12)IR=0
0556      IF(IC.NE.1) GOTO 10
0557      IF(IR.NE.0) GOTO 10
0558      IF(ANS.EQ.0.0) GOTO 25
0559      C
0560      C      INITIALIZE THE LEXIDATA
0561      C
0562      CALL LINIT
0563      CALL DSMRG(0,0,539,639,0)
0564      CALL DSCHN(-1,-1,-1)
0565      C
0566      C      JUMP TO 10 IF INITIALIZATION HAS BEEN ALREADY COMPLETED
0567      C
0568      10  CALL DSLIM(38+60*IR,8+IL*70,69+60*IR,39+IL*70)
0569      CALL DSPUT(IA1,1024)
0570      CALL DSZOM(1,1,2)
0571      CALL DSSAO(31+60*IR,45+IL*70,1000B,0,1)
0572      CALL DSTXT(6,IWD)
0573      CALL DSSAO(40+60*IR,56+IL*70,1000R,0,1)
0574      CALL DSTXT(3,INUM)
0575      25  CONTINUE
0576      RETURN
0577      END
0578      C
0579      C      THIS SUBROUTINE DISTINGUISHES BETWEEN A BRIDGE OVER WATER AND
0580      C      ANY OTHER TYPE OF GEOGRAPHY. IF A BRIDGE IS DETECTED, ANS=1 IF
0581      C      IT IS NOT, ANS=0.
0582      C
0583      SUBROUTINE BRDGE(IA1,ANS,ICOL,IROW)
0584      DIMENSION TX(32,32),IA1(32,32)
0585      N=0
0586      ANS=0
0587      N1=0
0588      ICOL=0
0589      IROW=0
0590      DO 10 I=1,32
0591      DO 10 J=1,32
0592      TX(I,J)=IA1(I,J)
0593      10  CONTINUE
0594      DO 20 J=1,32
0595      TX(1,J)=5
0596      TX(32,J)=5
0597      TX(J,1)=5
0598      TX(J,32)=5

```

```

0599 20  CONTINUE
0600      DO 40 J=2,31
0601      DO 30 I=2,31
0602          IF (TX(I,J)+TX(I-1,J).EQ.2)N=N+1
0603          IF (TX(J,I)+TX(J,I-1).EQ.2)N1=N1+1
0604      30  CONTINUE
0605          IF (N.LT.18.AND.N.GE.2)ICOL=ICOL+1
0606          IF (N1.LT.18.AND.N1.GE.2)IROW=IROW+1
0607      N=0
0608      N1=0
0609  40  CONTINUE
0610      IF (ICOL.GE.8)ANS=1
0611      IF (IROW.GE.8)ANS=1
0612      RETURN
0613      END
0614      FND$

```

ATE
LMED
-8