

ETL-0230

AD A088659

Correlation of noisy images

James A. Shine

Eugene A. Margerum

JUNE 1980

THIS DOCUMENT IS BEST QUALITY PRACTICABLE
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

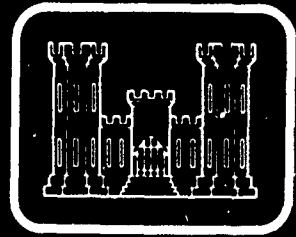
SEARCHED
SERIALIZED
A

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

DDC FILE COPY

APPROVED FOR PUBLIC RELEASE DISTRIBUTION UNLIMITED

80 9



ETL

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.

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

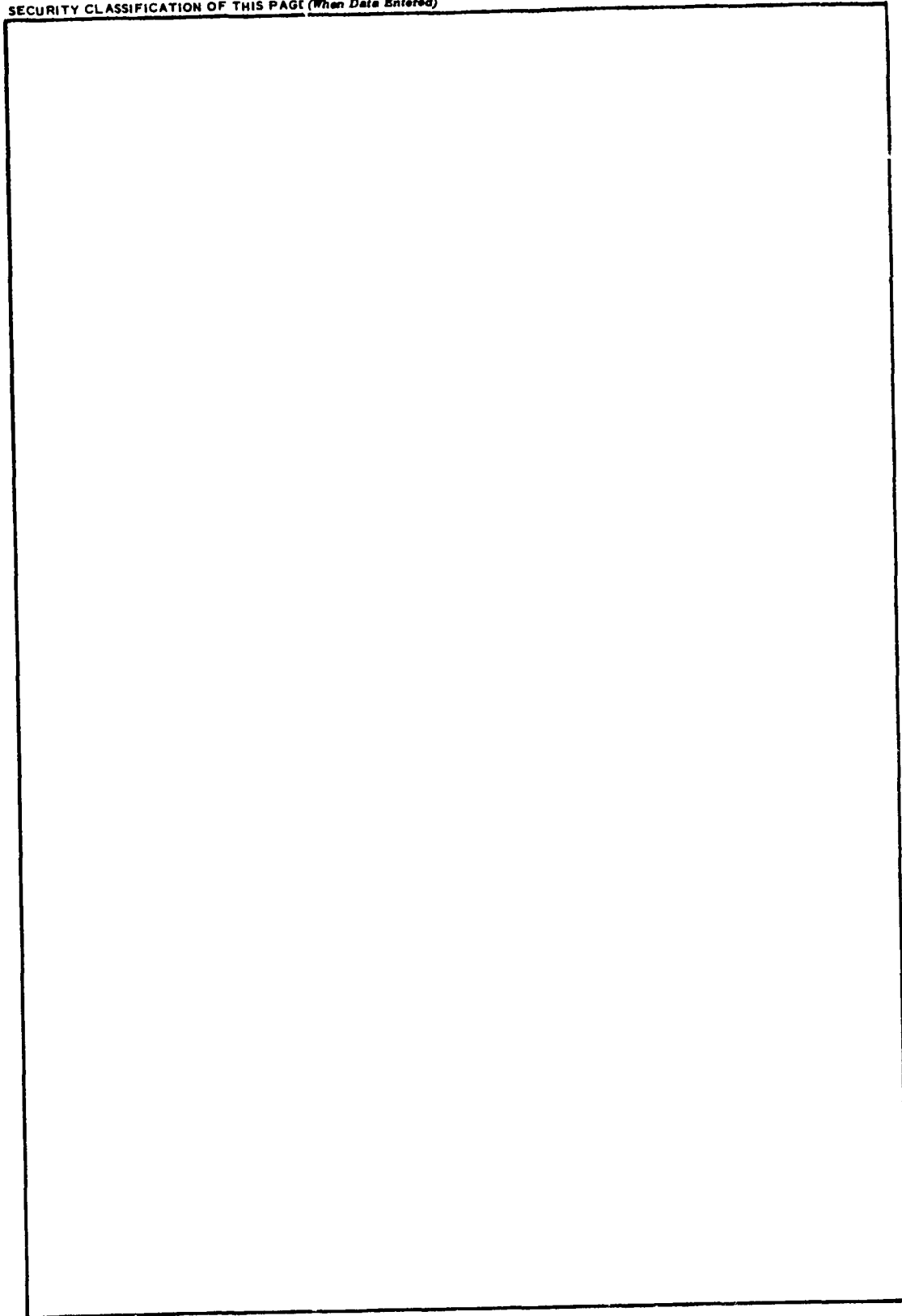
REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0230	2. GOVT ACCESSION NO. AD-A088659	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CORRELATION OF NOISY IMAGES	5. TYPE OF REPORT & PERIOD COVERED Research Note	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James A. Shine Eugene A. Margerum	8. CONTRACT OR GRANT NUMBER(s) 11 Jun 80	9. PERFORMING ORGANIZATION NAME AND ADDRESS U. S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060
10. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060	11. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 4A161102B52C	12. REPORT DATE June 1980
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	14. SECURITY CLASS. (of this report) Unclassified	15. NUMBER OF PAGES 51
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) Computer Simulation Correlation Filtering Fourier Transform Image Processing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A computer program can simulate star-shaped images and a correlation between two different images on 128 by 128 matrices. The Fourier transform, in a timesaving algorithm, is used to carry out the correlation. Different factors in the image (contrast, noise, size) are varied and the effects are observed. The results showed good correlation except when the contrast was low and the noise was high. Using a filtering function in the correlation process produced good results in improving poor correlation patterns.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

The work reported on was done under DA Project 4A161102B52C, Task S3, Work Unit 0009, under the supervision of Dr. Frederick W. Rohde, Team Leader, Center for Theoretical and Applied Physical Sciences; and Mr. Melvin Crowell, Jr., Director, Research Institute.

COL Daniel L. Lycan, CE was the Commander and Director and Mr. Robert P. Macchia was Technical Director of the Engineer Topographic Laboratories during the study and report preparation.

The authors are grateful to Robert Pazak, Robert Matos, and Barbara Brooke for their assistance in reviewing the computer programs and to Michael McDonnell for his helpful comments on the report.

A 23
CP

TABLES OF CONTENTS

	PAGE
PREFACE	1
TABLE	2
ILLUSTRATIONS	2
INTRODUCTION	3
THEORY	4
RESULTS	7
CONCLUSION	9
BIBLIOGRAPHY	9
APPENDIXES	
A. List of Variables	46
B. Listing of Computer Programs	47

TABLE

NUMBER	TITLE	PAGE
1	Values for Correlation Figures	8

ILLUSTRATIONS

NUMBER	TITLE	PAGE
1-18	Correlation Functions	10 - 45

CORRELATION OF NOISY IMAGES

INTRODUCTION

Purpose • The purpose of this study was to simulate star-shaped patterns and to observe whether any effects on correlation occurred between an "ideal" pattern and a "real" pattern when different factors affecting the pattern were varied.

Background • Real-world guidance systems (i.e. self-guided missiles) often determine their course of motion by taking continuous images of their surroundings, correlating these images with a "target" image contained within the system, and altering the course of motion (if necessary) towards the region that produces the strongest correlation, or correlation peak. The system's target image is constant and noise-free, but the incoming images often contain extraneous signals (noise) that can disrupt the correlation pattern and produce false peaks.

A correlation process was simulated by forming two matrices representing two separate images. One matrix had random noise scattered throughout, and the other matrix had no such noise; otherwise, the matrices were identical. Four different factors were then tested for their effects on correlation:

1. Amount of contrast between the pattern and its background.
2. Amount of random noise added to pattern and background.
3. Relative size of the pattern.
4. Application of a "filtering" function (of variable strength) to the pattern and background during the correlation process.

The correlation was expected to be good except where there was very little contrast between the pattern and the background and a lot of noise. It was also expected that the filtering function would help improve correlation in those cases when it was poor.

THEORY

Photographic images can be simulated on computers by using a two-dimensional matrix, or array, in which each position is given by a numerical value proportional to the lightness or darkness of the corresponding small square area of the photograph. This program simulates star-shaped figures by first mapping out a boundary around a central region, and then assigning greater values to inner positions than to those outside. The boundary was measured from the center of the array, and the center-to-boundary distance was computed as

$$L = R + A \cos (Nx) \quad (1)$$

where R (radius), A (arm length), and N (number of arms) were input into the program, and x was the angle between a line out to some boundary point and a horizontal reference line. Both the inner and outer values were subject to random variations ranging up to a certain value that was also input into the program, thus simulating photographic "noise." These variations could be either positive or negative.

To simulate image correlations, two separate arrays of the same pattern were generated, one with noise and the other with no noise. (This is analogous to comparing a photograph of an object with a perfect image of that object and seeing how well they match up.) A correlation array h can be computed directly from the two image arrays, f and g , by the formula

$$h_{jk} = (1/N) \sum_{n=1}^N \sum_{m=1}^N f_{mn} g_{m-j, n-k} \quad (2)$$

(Note: f , g , and h are all assumed periodic with period N , so $f_{mn} = f_{m+N, n} = f_{m, n+N}$ and similarly for g and h).

To save computer time, a similar transformation was used, the convolution

$$h_{jk} = \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^N f_{mn} g_{j-m, k-n} = \frac{1}{N} \sum_{r=1}^N \sum_{m=1}^N f_{mn} \hat{g}_{m-j, n-k} \quad (3)$$

($g_{a,b} = \hat{g}_{-a,-b}$). Since the arrays are two-dimensional, rotating one array 180° and taking a convolution is the same as taking a correlation. The convolution has the useful property that its discrete Fourier transform

$$H_{jk} = \frac{1}{N} \sum_{b=1}^N \sum_{a=1}^N e^{-\frac{2\pi i(a_j - b_k)}{N}} h_{ab} \quad (4)$$

is a simple product of the Fourier transforms of the two image arrays,

$$H_{jk} = \frac{1}{N} F_{jk} G_{jk} \quad (5)$$

By using the Fast Fourier Transform (FFT), the Fourier transforms of any of the arrays can be computed in substantially fewer steps than a direct correlation, as in equation 2. Because of the resulting savings of computer time, the correlation was accomplished by the following steps:

1. Flip one of the image arrays.
2. Take the Fourier transforms of both arrays using the FFT.
3. Multiply the transformed arrays together as in equation 5 to produce the correlation/convolution transform matrix H.
4. Take the inverse Fourier transform of H to produce the correlation matrix h.

The values in h were then normalized on a scale of 0 to 10 and truncated, their integer parts being printed out with odd numbers left blank for better contrast.

Some of the noisy correlations were subjected to a filter function, the sharpness of which was dependent on an input value W. If an array size of 128 x 128 is used, then W can characterize the width of the filtering function

$$S_\ell = \frac{1}{1-e^{-a^2}} \left[e^{-\left(\frac{\ell-1}{\sigma}\right)^2} + e^{-\left(\frac{129-\ell}{\sigma}\right)^2} \right]; \quad 1 \leq \ell \leq 128 \quad (6)$$

where $\sigma = 128 W/E$ and $a = 128/\sigma = E/W$

with E being the length of an edge of the array. (A small value of W produced a very sharp filter, and a large value resulted in a wider, less effective one). If the Fourier transform of a matrix being correlated is given by F_{jk} , the corresponding filter function is obtained by

$$\tilde{F}_{jk} = S_j S_k F_{jk}; \quad 1 \leq j, k \leq 128 \quad (7)$$

where the filtering has been performed over the first period. The filtering function will tend to suppress the higher frequency components of the image, while leaving the lower frequency components alone. Since the noise is largely in the high frequency range, it will be greatly suppressed relative to the essential image information contained in the low frequency range.

If considered to be periodic, the filtering function is symmetrical with respect to the origin. This was the primary reason for using this function, which is a combination of two Gaussian functions (one at each end of the period). A single Gaussian centered at the origin would not be symmetrical over the period, and one placed at the middle of the period ($j = 64$ or $j = 65$) would introduce phase terms into the correlation results.

The problem of aliasing (false correlation signals resulting from overlap of images with one another while performing the correlation) was considered in this work, but the signal level in the overlap region was so low that it was insignificant in comparison to the correlation values; thus it could be neglected.

RESULTS

When correlation was done with a noiseless image, the correlation pattern was very regular, with a sharp central peak fading quickly into a uniform background. This was the case whether the contrast between the star and its background was quite noticeable (as in figure 1), somewhat noticeable (as in figure 2), or only slightly noticeable (as in figure 3). Adding noise to a high-contrast figure broke up the edges of the correlation pattern, but the central peak was still quite sharp and visible. Noise in a low-contrast figure disrupted the entire pattern, although there was still a slight indication of a central peak area (see figures 4 and 5).

Large amounts of noise are needed to cause this pattern disruption; smaller amounts do proportionally less damage to the correlation pattern. The progression of the pattern degradation can be seen for both high-contrast and low-contrast figures (see figures 6-13).

The smallness of the star also affected pattern disruption; the larger stars maintained their central peak much better than the smaller ones under the deluge of heavy noise (compare figures 5, 13, and 14).

Finally, our filter function did an excellent job of restoring badly disrupted patterns. The "sharpness" of the filter could be varied; a very sharp filter (small value of W) restored the worst patterns to smoothness; however, wider filters (large values of W) did not work as well.

Table 1 contains a complete listing of the numerical values used in the programs to produce each figure. Figure 18 contains a sample printout of one of the stars.

TABLE 1. Values for Correlation Figures

Figure	R	A	VIN,VIN2	VOUT,VOUT2	RNIN2	RNOUT2	W(where used)
1	20	10	8	2	0	0	---
2	20	10	6	5	0	0	---
3	20	10	6	5.5	0	0	---
4	20	10	8	2	5	5	---
5	20	10	6	5.5	5	5	---
6	10	5	8	2	0	0	---
7	10	5	8	2	1	1	---
8	10	5	8	2	3	3	---
9	10	5	8	2	5	5	---
10	10	5	6	5.5	0	0	---
11	10	5	6	5.5	1	1	---
12	10	5	6	5.5	2	2	---
13	10	5	6	5.5	5	5	---
14	7	3.5	6	5.5	5	5	---
15	7	3.5	6	5.5	5	5	3
16	7	3.5	6	5.5	5	5	10
17	7	3.5	6	5.5	5	5	100

RNIN = RNOUT = 0 for all 17 figures; NARMS = 6 for all 17 figures

CONCLUSIONS

1. The simulation produced good, centralized correlation patterns in cases where little or no noise was involved.
2. Correlation became poor only when contrast between the figure and its background was low (less than 10 percent), and the noise levels were high (greater than 90 percent of peak values).
3. The filtering function did an excellent job in restoring poor correlations, which should make it useful for future applications.

BIBLIOGRAPHY

Andrews, Harry C., *Computer Techniques in Image Processing*, Academic Press, 1970.

Cochran, William T., et al., "What is the Fast Fourier Transform?," *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-15, No. 2, June 1965, pp. 45-55.

Cooley, James W. and Tukey, John W., "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, Vol. 19, April 1965, pp. 297-301.

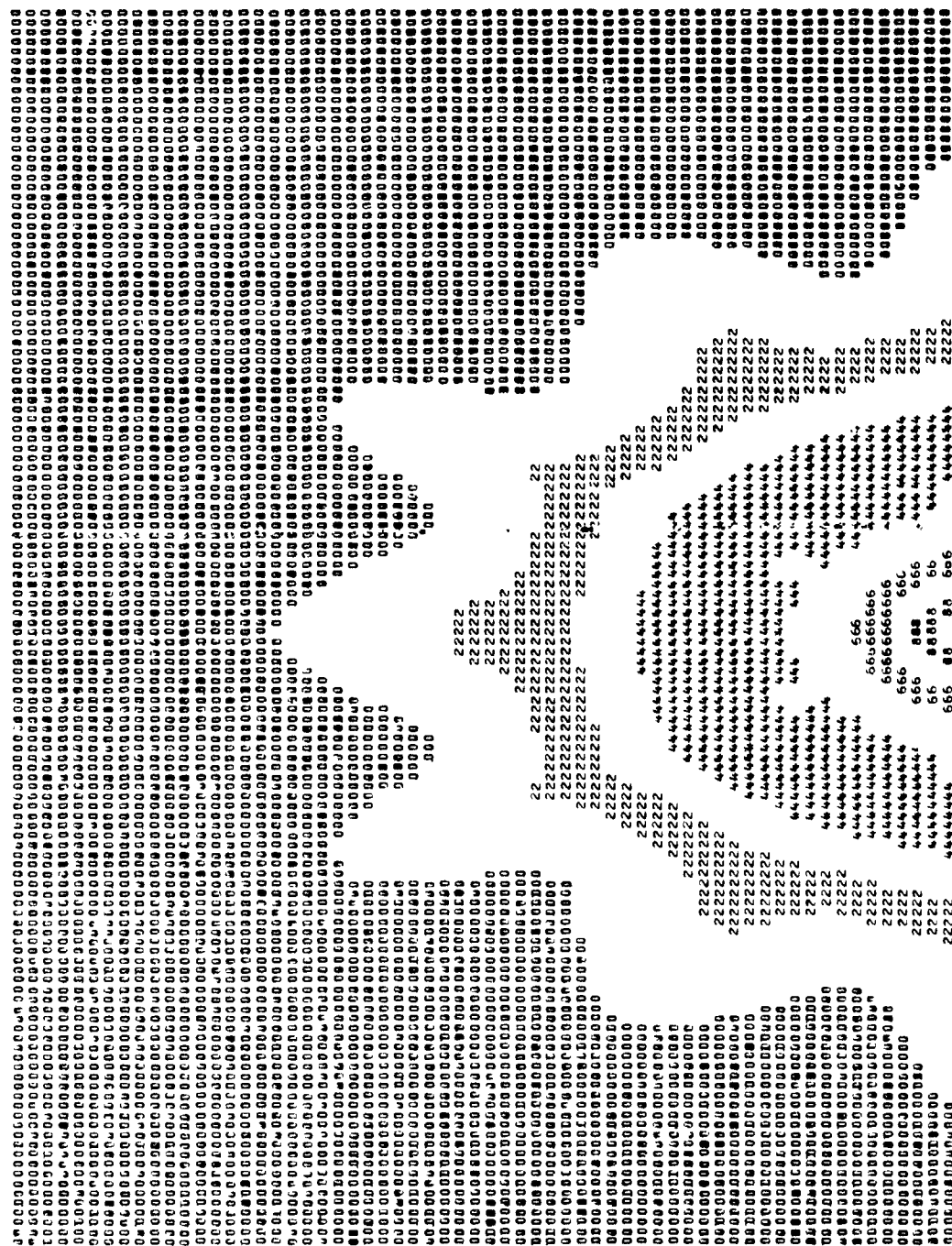


FIGURE 1. R-20, A-10, VIN,VIN2-8, VOUT,VOUT2-2, RNIN2-0, RNOUT2-0, W-N/A.

FIGURE 1. Continued

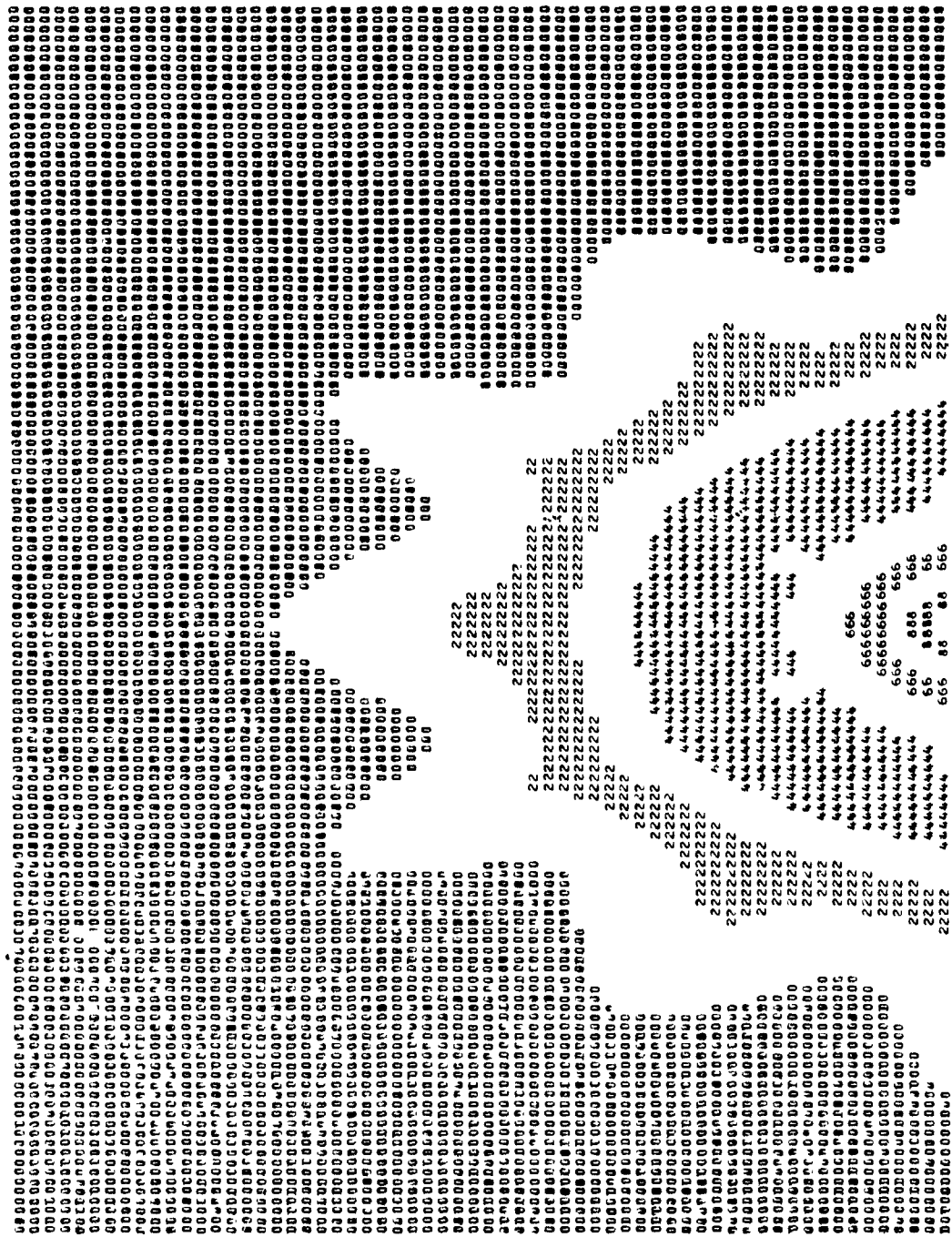


FIGURE 2. R-20, A-10, VIN,VIN2-6, VOUT,VOUT2-5, RNIN2-0, RNOOUT2-0, W-N/A.

FIGURE 2. Continued

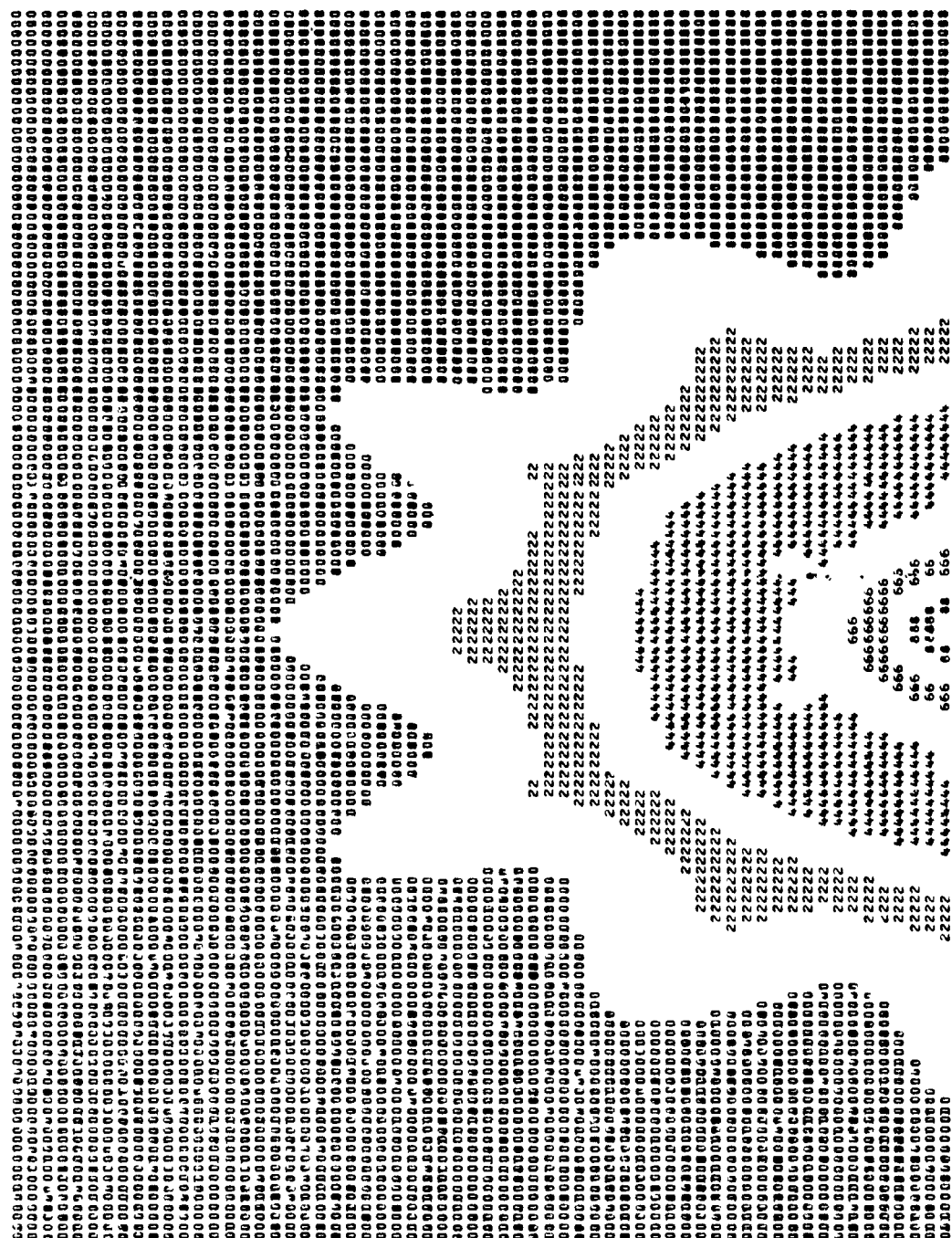
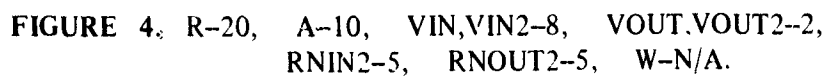


FIGURE 3. R-20, A-10, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-0, RNOUT2-0, W-N/A.

FIGURE 3. Continued



17

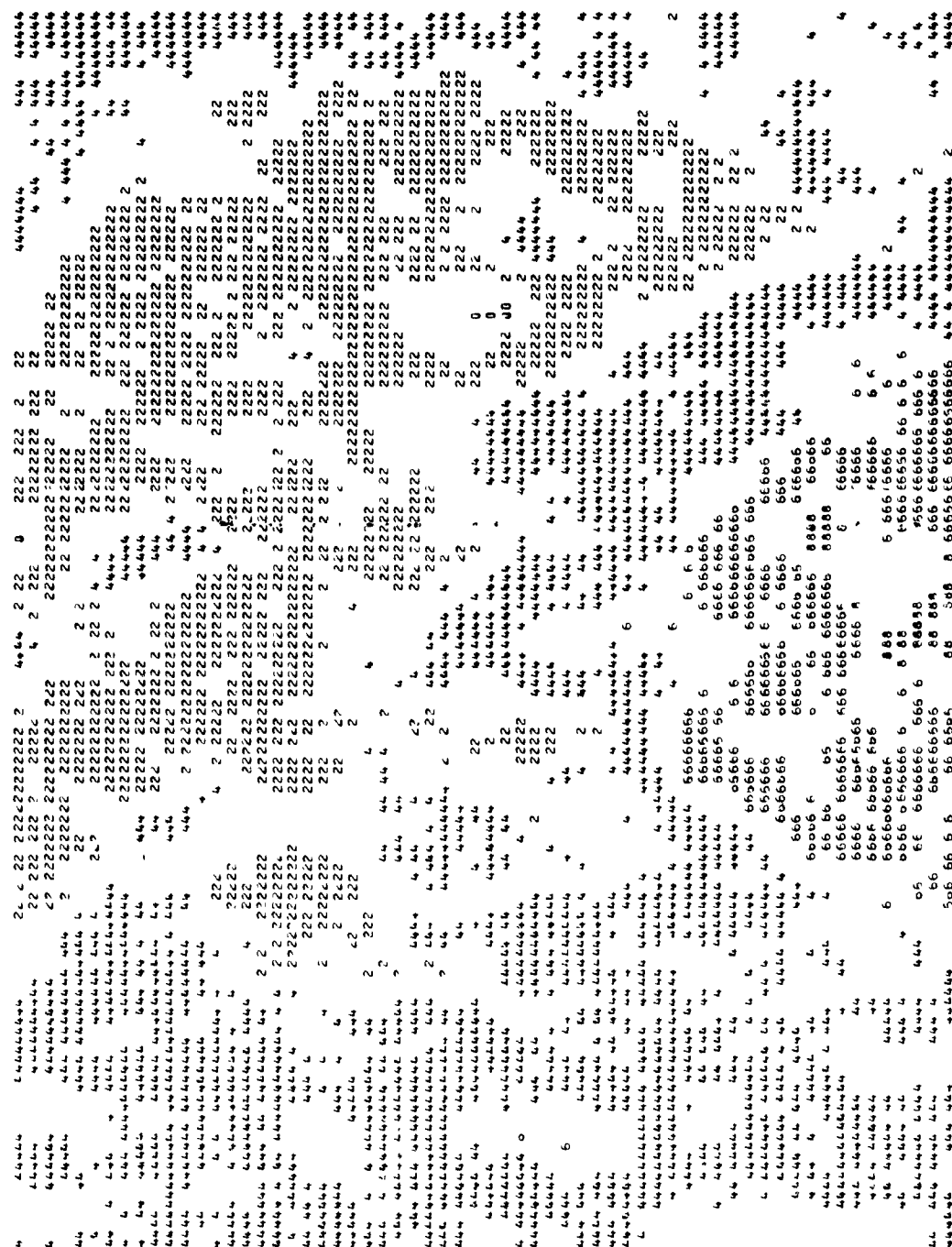
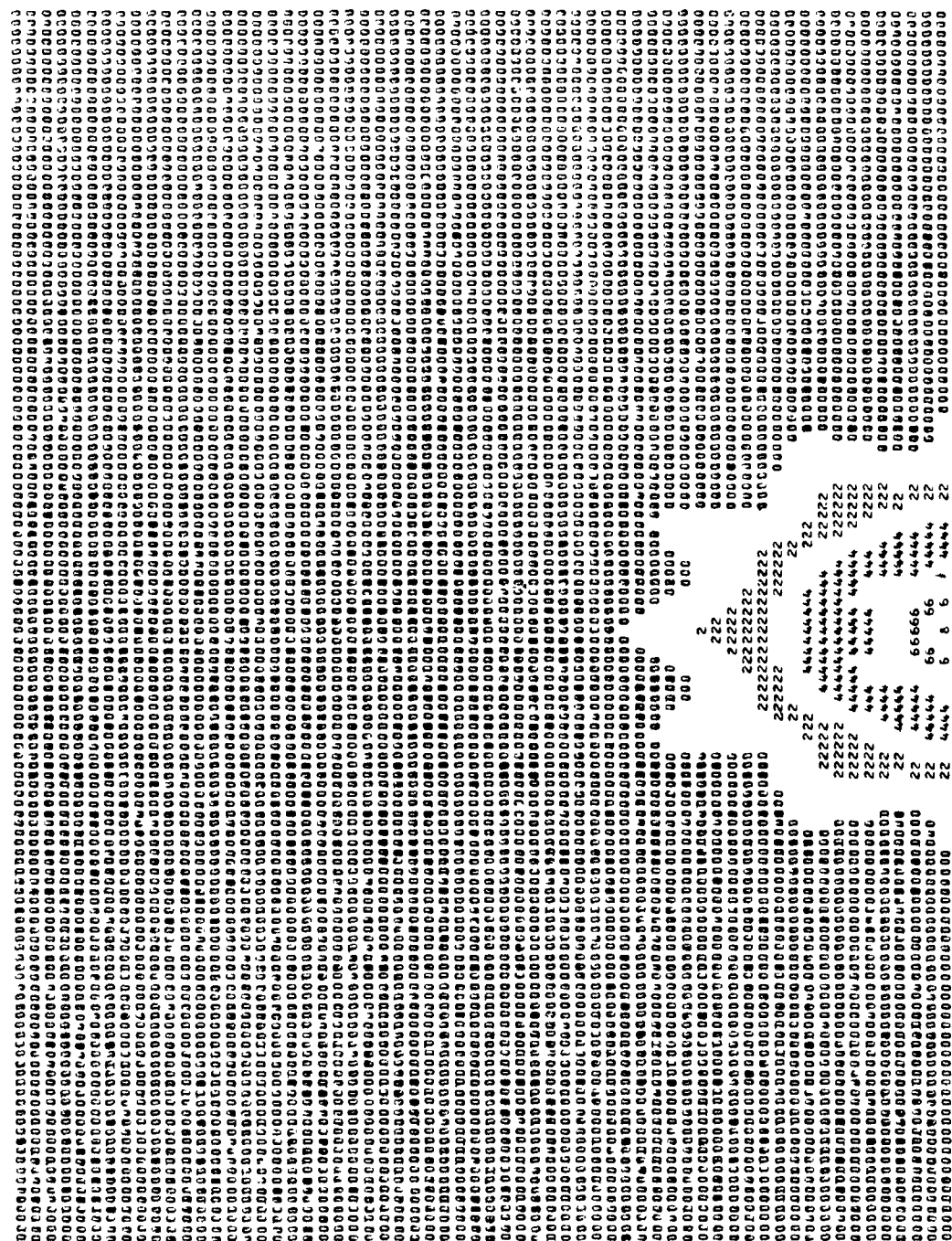


FIGURE 5. R-20, A-10, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-5, RNOUT2-5, W-N/A.



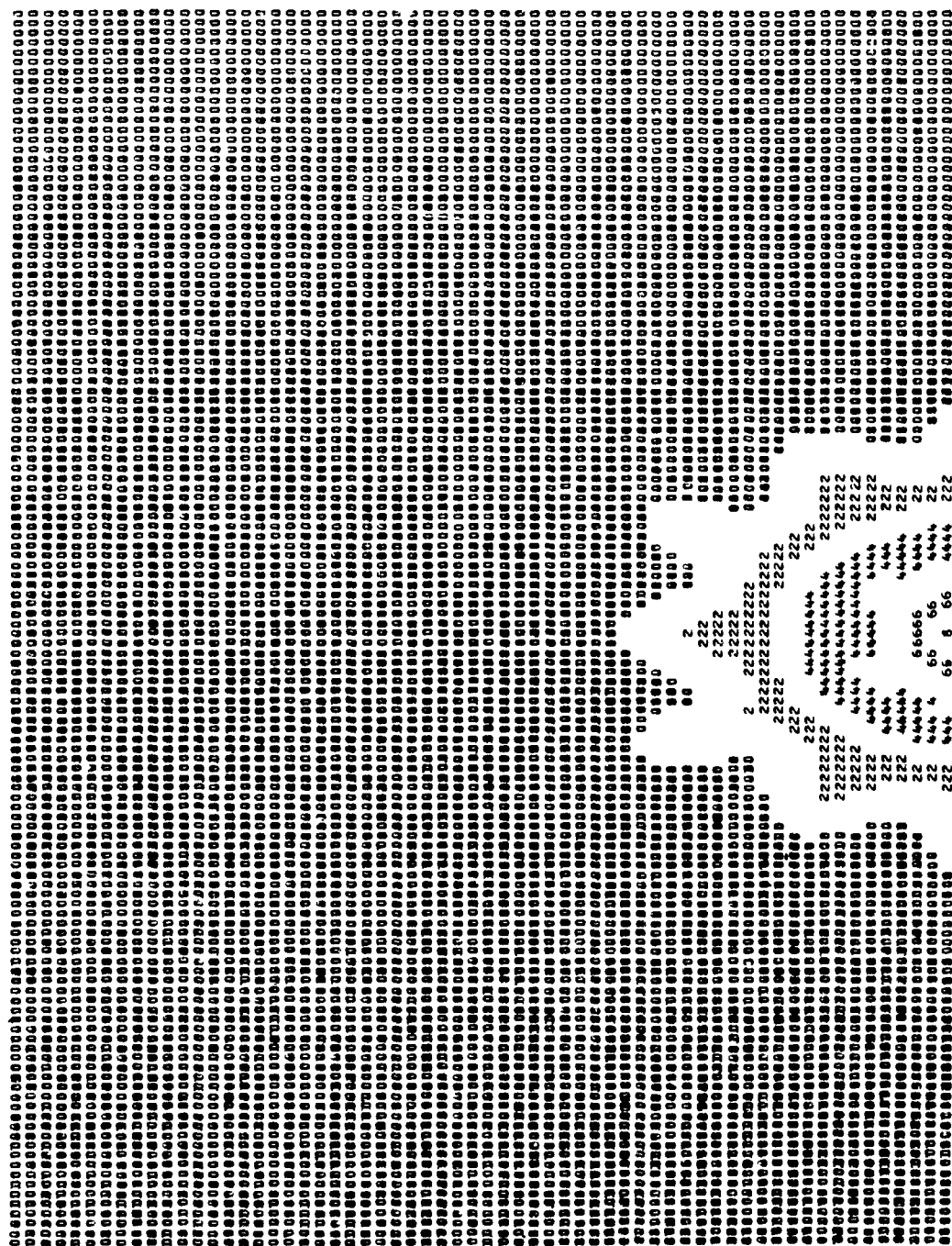


FIGURE 7. R-10, A-5, VIN,VIN2-8, VOUT,VOUT2-2, RNIN2-1, RNOUT2-1, W-N/A.



FIGURE 8. R-10, A-5, VIN,VIN2-8, VOUT,VOUT2-2, RNIN2-3, RNOUT2-3, W-N/A.

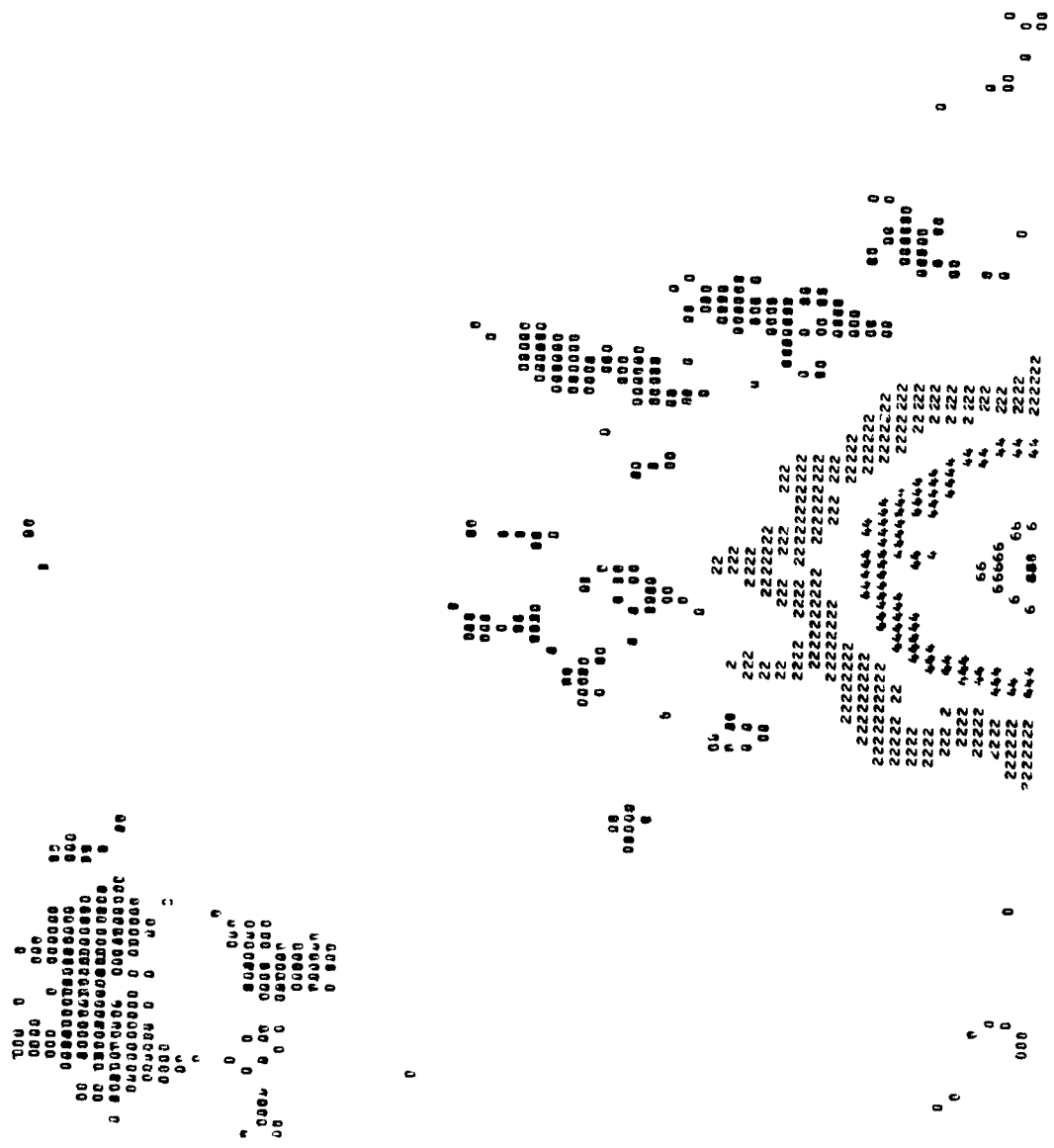


FIGURE 9. R-10, A-5, VIN,VIN2-8, VOUT,VOUT2-2,
RNIN2-5, RNOUT2-5, W-N/A.

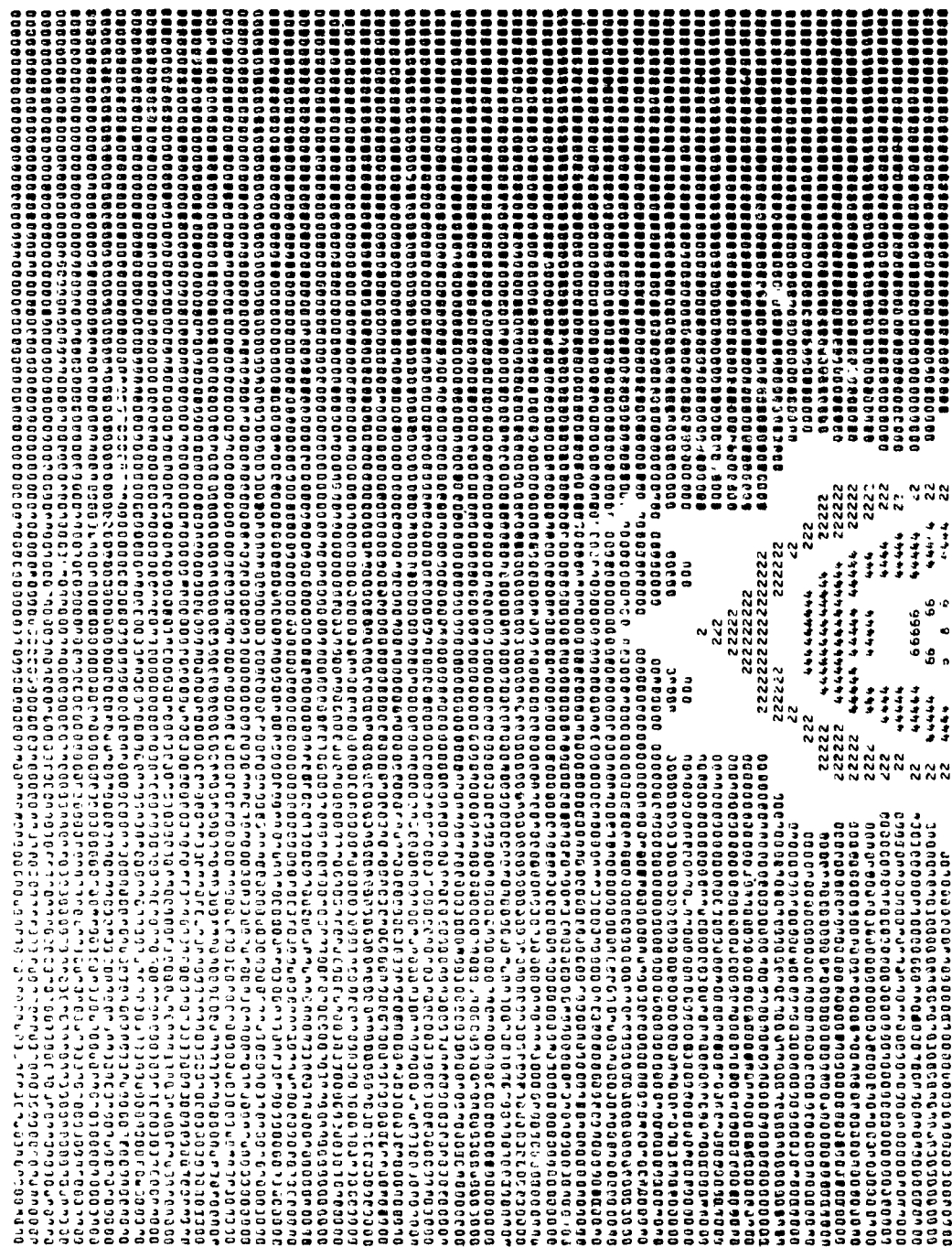


FIGURE 10. R-10, A-5, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-0, RNOUT2-0, W-N/A.



FIGURE 11. R-10, A-5, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-1, RNOUT2-1, W-N/A.



FIGURE 11. Continued

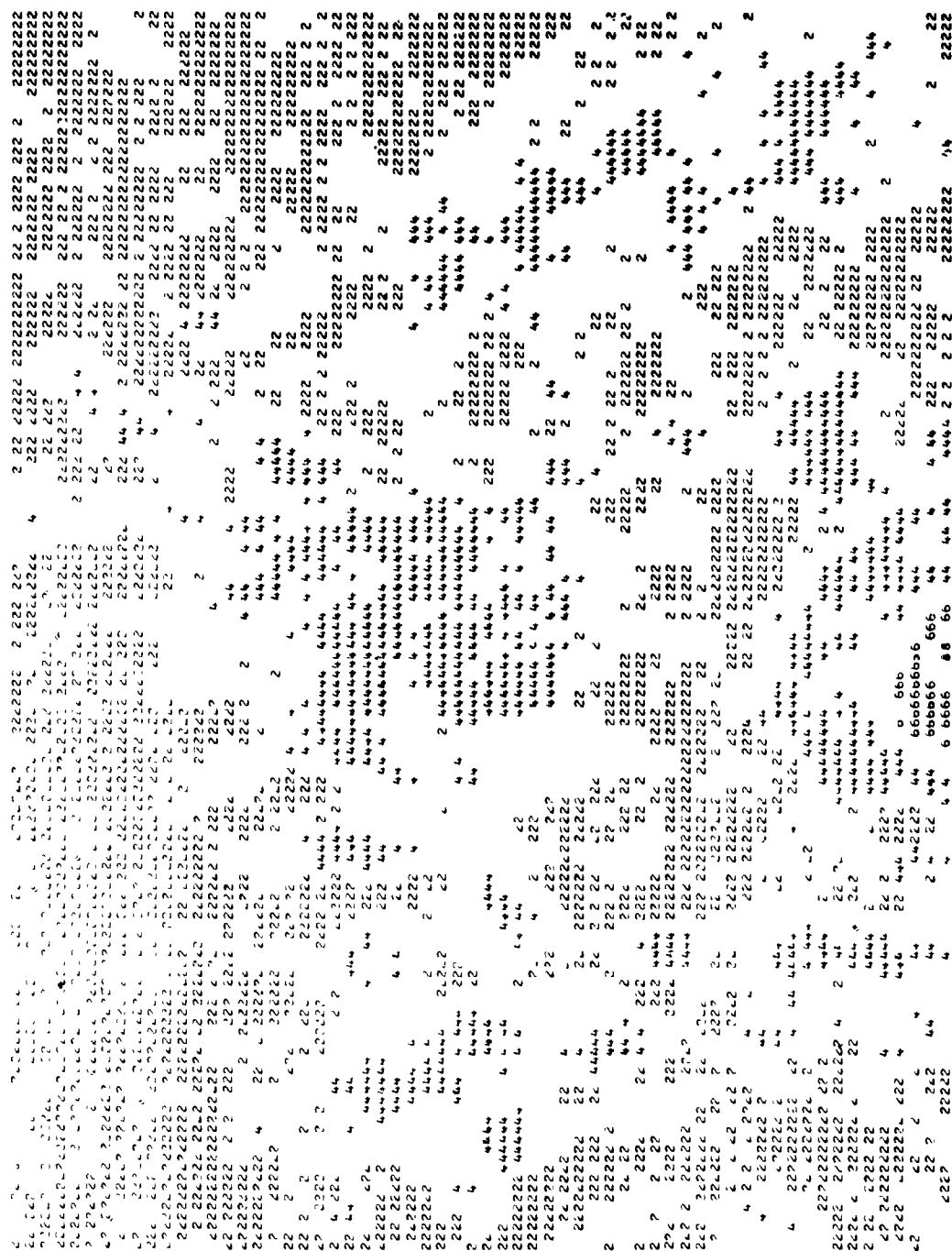


FIGURE 12. R-10, A-5, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-2, RNOUT2-2, W-N/A.

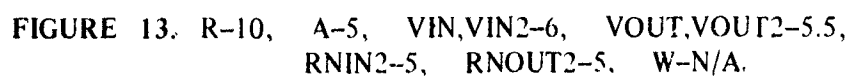


FIGURE 13. Continued

FIGURE 14. R-7, A-3.5, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-5, RNOOUT2-5, W-N/A.

37



FIGURE 15. R-7, A-3.5, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-5, RNOUT2-5, W-3.

39



FIGURE 16. R-7, A-3.5, VIN,VIN2-6, VOUT,VOUT2-5.5, RNIN2-5, RNOUT2-5, W-10.

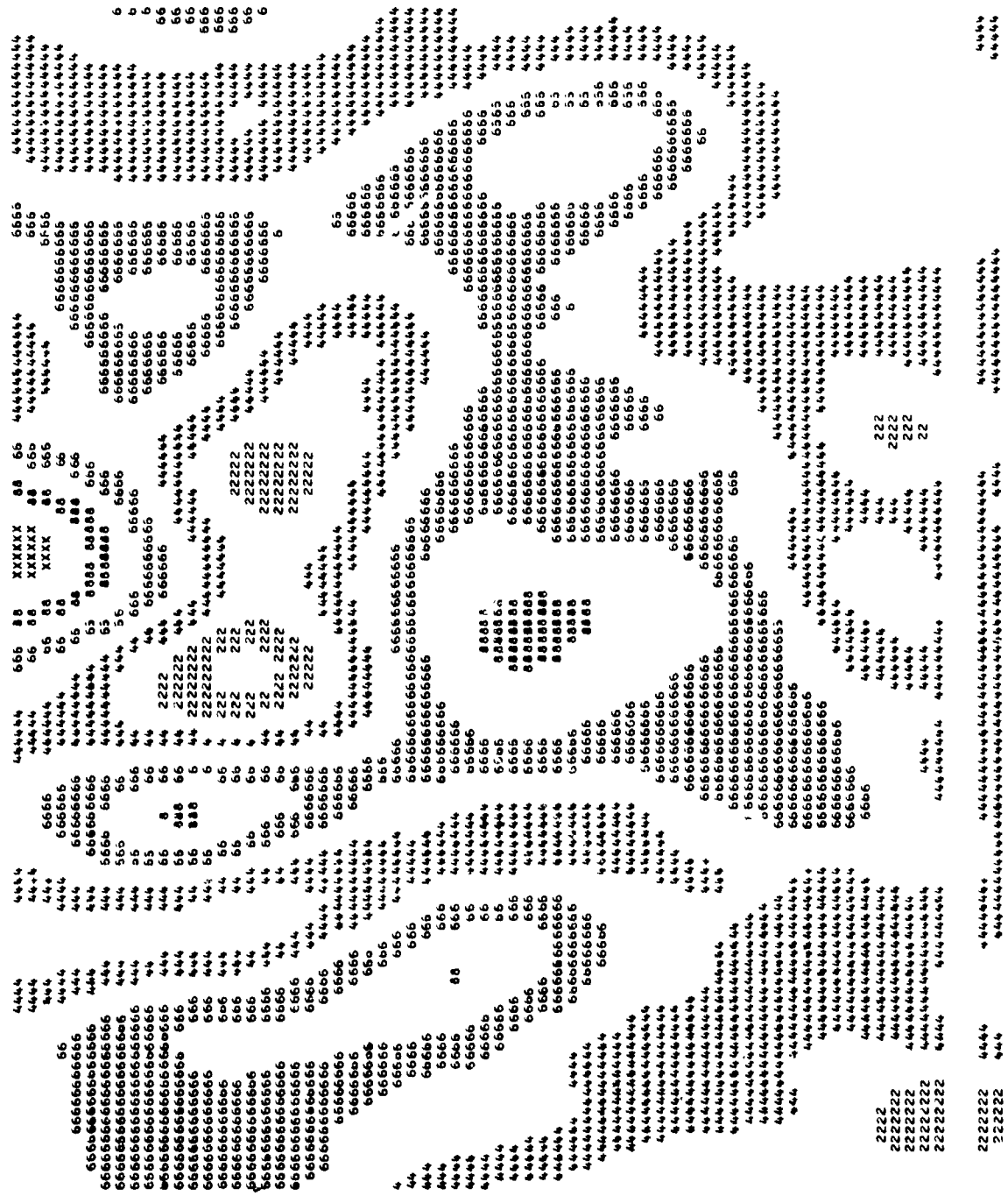


FIGURE 16. Continued

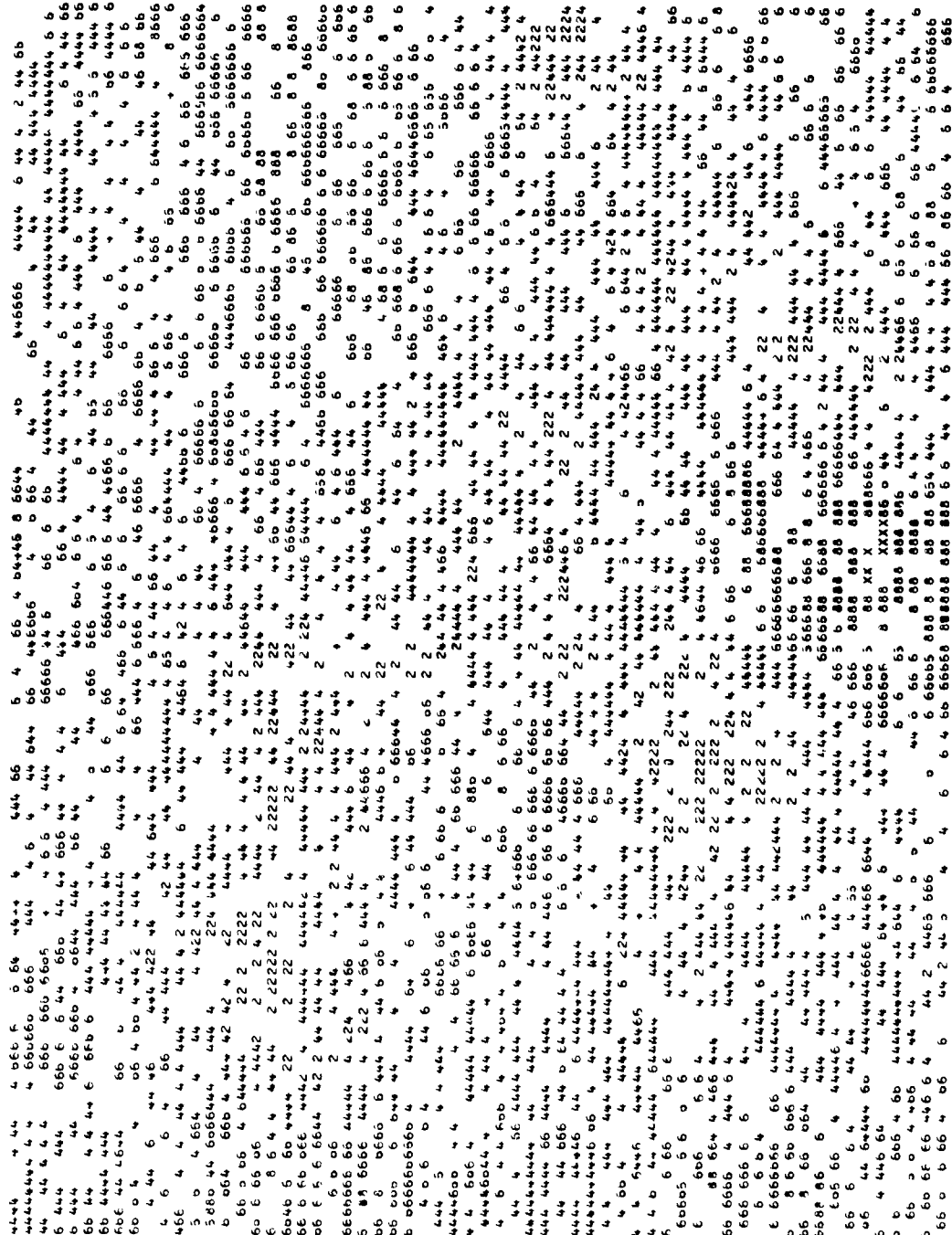


FIGURE 17. Continued

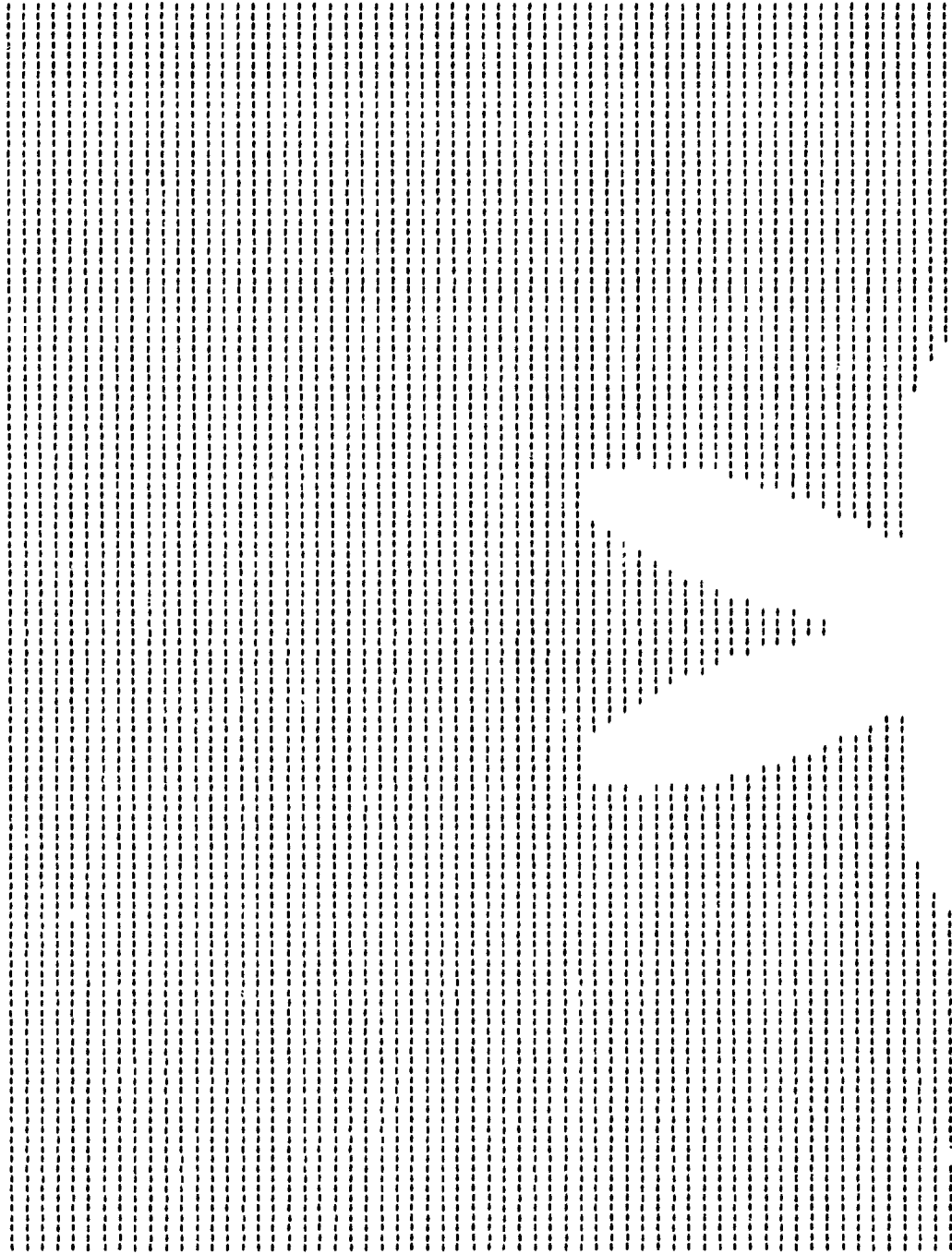


FIGURE 18.

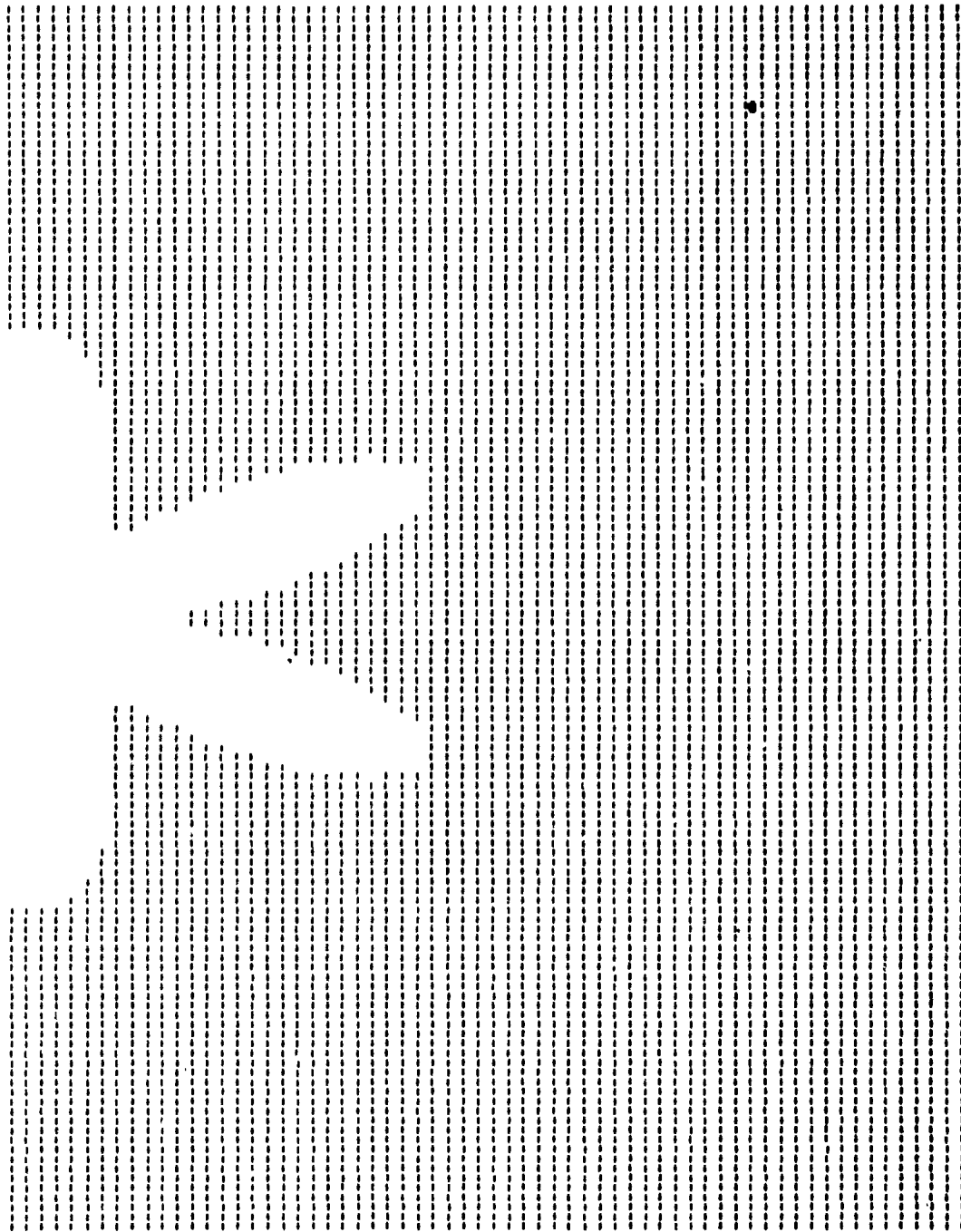


FIGURE 18. Continued.

APPENDIX A - LIST OF VARIABLES

- EDGE: The length of an edge - in most cases, 128.
- R: Radius of the figure; the length from the center of the figure to a point halfway up one of the arms.
- 2A: The length of an arm.
- NARMS: The number of arms.
(The border of the star is measured from the center of the figure and can be expressed as $R + A \cos(\text{NARMS} \cdot x)$, where x varies between 0 and 360 degrees, and the distance between the center and the edge of the star thus varies between $R - A$ and $R + A$).
- VIN: Numerical value assigned to points inside the star.
- VOUT: Value assigned to points outside the star.
- RNIN: Amplitude of random noise assigned inside the star; thus the total value inside the star will vary between $VIN - RNIN$ and $VIN + RNIN$.
- RNOUT: Amplitude of random noise assigned outside the star; thus the total value outside the star will vary between $VOUT - RNOUT$ and $VOUT + RNOUT$.
- RRAN: The initial value for the random number generator.
When two figures were being correlated, separate values (VIN2, VOUT2, RNIN2, RNOUT2) were read in. In all our examples, $VIN = VIN2$ and $VOUT = VOUT2$, $RNIN = RNOUT = 0$ (the noiseless image), and RNIN2 and RNOUT2 were varied as desired.
- W: Value used in filtering function; a small value of W produced a sharp filter (noise effects on correlation substantially reduced), while a large value of W produced a wider, fuzzier filter (noise effects on correlation only somewhat reduced).

APPENDIX B - LISTING OF COMPUTER PROGRAMS

```

PROGRAM AUTOCOR(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT)
COMPLEX C(128)
COMPLEX STAR1, STAR2
COMMON STAR1(128,128), STAR2(128,128)
CALL COOL(C)
CALL OUTPUT
READ 13, N1
13 FORMAT(1I3)
DO 7 I=1, N1
  READ 100, EDGE, R, A, NARMS, VIN, VOUT, RNIN, RNOUT, RRAN
  C 100 FORMAT(3(F10.3,10X),1I10,/,5(F10.3,5X))
  ALSO PRINT OUT STARTING VALUES
  PRINT 200, EDGE, R, A, NARMS, VIN, VOUT, RNIN, RNOUT, RRAN
  200 FORMAT(1/10H EDGE = ,1F10.3,10X,5H R = ,1F10.3,/,
  20015H A = ,1F10.3,10X,5H NARMS = ,1I5,10X,7H VIN = ,1F10.3,/,
  20028H VOUT = ,1F10.3,10X,8H RNIN = ,1F10.3,/,
  20039H RNOUT = ,1F10.3,10X,8H RRAN = ,1F10.3)
  READ 3, VIN2, VOUT2, RNIN2, RNOUT2
  3 FORMAT(4(F10.3,5X))
  PRINT 4, N1, VIN2, VOUT2, RNIN2, RNOUT2
  4 FORMAT(6H TEST ,1I2,/,21H VALUE IN STARFISH = ,1F10.3,/,
  4123H VALUE IN BACKGROUND = ,1F10.3,/,21H NOISE IN STARFISH = ,
  421F10.3,/,23H NOISE IN BACKGROUND = ,1F10.3)
  CALL STARFISH(EDGE, R, A, NARMS, VIN2, VOUT2, RNIN2, RNOUT2, RRAN)
  CALL SWITCH
  C CALL STARFISH(EDGE, R, A, NARMS, VIN, VOUT, RNIN, RNOUT, RRAN)
  NOW TRANSFORM THE ARRAYS
  N=1
  CALL TWOEE(N)
  N=2
  CALL TWOEE(N)
  CC NOW MULTIPLY TOGETHER AND TAKE CONJUGATE
  CALL MULT
  N=1
  CALL TWOEE(N)
  CALL PUTOUT
  CALL POTOUT
  7 CONTINUE
  STOP
  END

SUBROUTINE STARFISH(EDGE, R, A, NARMS, VIN, VOUT, RNIN, RNOUT, RRAN)
COMPLEX STAR
COMMON STAR(128,128)
HEDGE=EDGE/128.0
PI=3.14159
TPI=2.0*PI
DO 1000 II=1,128
  DO 1000 JJ=1,128
    XX=(FLOAT(II)-64.5)*HEDGE
    YY=(64.5-FLOAT(JJ))*HEDGE
    RR=SQRT(XX*XX+YY*YY)
    ALFA=ACOS(XX/RR)
    IF(YY.LT.0.0) ALFA=TPI-ALFA
    RFISH=R*A*COS(NARMS*ALFA)
    IF(RR-RFISH)1,1,2
  1 STAR(II,JJ)=CMPLX(VIN,0.0)
    IF(RNIN.NE.0.0) STAR(II,JJ)=CMPLX(VIN+RNIN*(2.0*РАНF(RRAN)-1.0),
  10.0)
    GO TO 1000
  2 STAR(II,JJ)=CMPLX(VOUT,0.0)
    IF(RNOUT.NE.0.0) STAR(II,JJ)=CMPLX(VOUT+RNOUT*(2.0*РАНF(RRAN)-1.0),
  10.0)
  1000 CONTINUE
  RETURN
  END

```

CS
C4
C5

APPENDIX B - Continued

```

SUBROUTINE COOL(C)
COMPLEX C(128),M(6,63),CC
DIMENSION NUM(6),JJ(128),N(7)
DO 10 I=1,128
  II=I-1
  DO 3 K=1,7
    IF (II-2** (7-K)) 1,2,2
1  N(8-K)=0
    GO TO 3
2  N(8-K)=1
    II=II-2** (7-K)
3  CONTINUE
    III=0
    DO 4 M=1,7
      IF(N(M).NE.1) GO TO 4
      III=III+2** (7-M)
4  CONTINUE
    JJ(II)=III+1
10  CONTINUE
    PI=3.14159265359
    ARG=2.0*PI/128.0
    WR=COS(ARG)
    WI=-SIN(ARG)
    W(1,1)=CMPLX(WR,WI)
    DO 100 NI=2,63
      W(1,NI)=W(1,1)*W(1,NI-1)
100 CONTINUE
    DO 150 NJ=2,6
      DO 150 NI=1,63
        W(NJ,NI)=W(NJ-1,NI)*W(1,NI)
150 CONTINUE
    DO 200 NK=1,5
      NUM(NK)=2** (7-NK)
200 CONTINUE
    RETURN
  ENTRY TRANS
  DO 600 LEVEL=1,6
    LDEL=NUM(LEVEL)
    ITOP=1
    IBOT=LDEL+1
    LD=LDEL-1
400  CC=C(ITOP)
    C(ITOP)=CC+C(IBOT)
    C(IBOT)=CC-C(IBOT)
    DO 500 IP=1,LD
      ITOP=ITOP+1
      IBOT=IBOT+1
      CC=C(ITOP)
      C(ITOP)=CC+C(IBOT)
      C(IBOT)=N(LEVEL,IP)*(CC-C(IBOT))
500  CONTINUE
    IF (IBOT.EQ.128) GO TO 600
    ITOP=IBOT+1
    IBOT=ITOP+LDEL
    GO TO 400
600  CONTINUE
    DO 700 M=1,127,2
      MM=M+1
      CC=C(M)
      C(M)=CC+C(MM)
      C(MM)=CC-C(MM)
700  CONTINUE
    DO 800 L=1,128
      K=JJ(L)
      IF (L.GE.K) GO TO 800
      CC=C(L)
      C(L)=C(K)
      C(K)=CC
800  CONTINUE
    RETURN
  END

```

APPENDIX B - Continued

```

SUBROUTINE MULT
COMPLEX STAR1,STAR2,CC
COMMON STAR1(128,128),STAR2(128,128)
DO 1 I=1,128
DO 1 J=1,128
CC=STAR1(I,J)*STAR2(I,J)*((-1)**(I+J))
STAR1(I,J)=CONJG(CC)
1 CONTINUE
RETURN
END

```

```

PROGRAM AUTOCOR(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
COMPLEX C(128)
COMPLEX STAR1,STAR2
COMMON STAR1(128,128),STAR2(128,128)
N=1
CALL OUTPUT(N)
CALL CUOL(C)
READ 7,N6
7 FORMAT(1,3)
DO 30 I=1,N6
READ 100,EDGE,R,A,NARMS,VIN,VOUT,RNIN,RNOUT,RRAN
100 FORMAT(3(F10.3,10X),1I10,/,5(F10.3,5X))
C ALSO PRINT OUT STARTING VALUES
PRINT 200,EDGE,R,A,NARMS,VIN,VOUT,RNIN,RNOUT,RRAN
200 FORMAT(/,8H EDGE = ,1F10.3,10X,5H R = ,1F10.3,/,
20015H A = ,1F10.3,10X,9H NARMS = ,1I5,10X,7I VIN = ,1F10.3,/,
20028H VOUT = ,1F10.3,10X,8H RNIN = ,1F10.3,/,
20039H RNOUT = ,1F10.3,10X,8H RRAN = ,1F10.3)
C NOW GENERATE THE ARRAY
CALL STARFSH(EDGE,R,A,NARMS,VIN,VOUT,RNIN,RNOUT,RRAN)
CALL SWITCH
READ 101,VIN2,VOUT2,RNIN2,RNOUT2
101 FORMAT(4(F10.3,5X))
PRINT 4,VIN2,VOUT2,RNIN2,RNOUT2
4 FORMAT(21H VALUE IN STARFISH = ,1F10.3,/,
4123H VALUE IN BACKGROUND = ,1F10.3,/,21H NOISE IN STARFISH = ,
+21F10.3,/,23H NOISE IN BACKGROUND = ,1F10.3)
CALL STARFSH(EDGE,R,A,NARMS,VIN2,VOUT2,RNIN2,RNOUT2,RRAN)
C NOW TRANSFORM THE ARRAYS
N=1
CALL TWODEE(N)
N=2
CALL TWODEE(N)
CC NOW MULTIPLY TOGETHER AND TAKE CONJUGATE
CALL MULT
C NOW FILTER AS OFTEN AS DESIRED
READ 7,N7
DO 30 J=1,N7
CALL STORE
N=2
CALL FILTER(N,EDGE)
CALL OPER(N,EDGE)
CALL TWODEE(N)
CALL PUTOUT(N)
CALL POTOUT(N)
300 CONTINUE
STOP
END

```

APPENDIX B - Continued

```

SUBROUTINE FILTER(N,EDGE)
COMPLEX STAR
COMMON STAR(128,128,2)
DIMENSION S(128)
READ 2,M
2 FORMAT(1F10.3)
PRINT 5,M
5 FORMAT(5H M = ,1F10.3)
SIGMA=M*128.0/EDGE
A=128.0/SIGMA
XYP=0.0
IF(A.LE.25.0)XYP=EXP(-A*A)
CONST=1+XYP
CONST=1/CONST
DO 3 I=1,128
A1=(FLOAT(I-1))/SIGMA
YXP=0.0
IF(A1.LE.25.0)YXP=EXP(-A1*A1)
A2=(FLOAT(129-I))/SIGMA
ZXP=0.0
IF(A2.LE.25.0)ZXP=EXP(-A2*A2)
S(I)=CONST*(YXP+ZXP)
3 CONTINUE
RETURN
ENTRY OPER
DO 4 J=1,128
DO 4 K=1,128
STAR(J,K,N)=STAR(J,K,N)*S(J)*S(K)
4 CONTINUE
RETURN
END

```

```

SUBROUTINE OUTPUT
COMPLEX STAR
COMMON STAR(128,128)
DIMENSION ALINE(128),ACHAR(11)
READ 10,(ACHAR(I),I=1,11)
10 FORMAT(11A1)
RETURN
ENTRY PUTOUT
BIG=REAL(STAR(1,1))
SMALL=REAL(STAR(1,1))
DO 100 II=1,128
DO 100 JJ=1,128
Z=REAL(STAR(II,JJ))
IF(Z.GT.BIG)BIG=Z
IF(Z.LT.SMALL)SMALL=Z
100 CONTINUE
RANGE=BIG-SMALL
PRINT 110
110 FORMAT(1H1)
DO 200 JJ=1,128
DO 150 II=1,128
MM=10.0*(REAL(STAR(II,JJ))-SMALL)/RANGE+1.5
ALINE(II)=ACHAR(MM)
150 CONTINUE
PRINT 160,(ALINE(I),I=1,128)
160 FORMAT(5X,128A1)
200 CONTINUE
PRINT 250,SMALL,BIG,RANGE
250 FORMAT(10H SMALLEST VALUE = ,1P1E10.3,/,17H LARGEST VALUE
2501 = ,1P1E10.3,/,8H RANGE = ,1P1E10.3)
RETURN
ENTRY POTUUT
PRINT 270
270 FORMAT(31H DETAILED PRINTOUT OF 65TH ROW )
READ 300,XCHAR
300 FORMAT(111)
DO 290 I2=1,128
N1=100.0*(REAL(STAR(I2,65))-SMALL)/RANGE+1.5
PRINT 310,I2,(XCHAR,I=1,N1)
310 FORMAT(1X,114,3X,101A1)
290 CONTINUE
RETURN
END

```

APPENDIX B - Continued

```

SUBROUTINE SWITCH
COMPLEX STAR1,STAR2
COMMON STAR1(128,128),STAR2(128,128)
DO 700 II=1,128
DO 700 JJ=1,128
STAR2(II,JJ)=STAR1(129-II,129-JJ)
700 CONTINUE
RETURN
END

```

```

SUBROUTINE TMODEE(N)
COMPLEX C(128)
COMPLEX STAR
COMMON STAR(128,128,2)
DO 200 I=1,128
DO 100 J=1,128
C(J)=STAR(J,I,N)
C 100 CONTINUE
NOW THE COLUMN WILL BE TRANSFORMED
C CALL TRANS(C)
NOW THE TRANSFORMED COLUMN WILL BE PUT BACK IN THE ARRAY
DO 200 JJ=1,128
STAR(JJ,I,N)=C(JJ)
C 200 CONTINUE
NOW EACH ROW OF THE NEW ARRAY WILL BE TRANSFORMED
DO 400 L=1,128
DO 300 K=1,128
C(K)=STAR(L,K,N)
C 300 CONTINUE
CALL TRANS(C)
DO 400 M=1,128
STAR(L,M,N)=C(M)
C 400 CONTINUE
RETURN
END

```