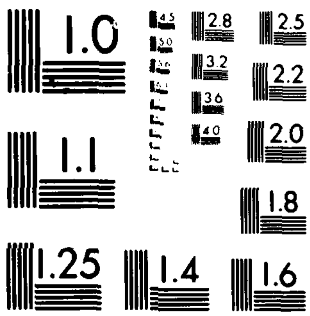
AD-A094 281 SOUTHEASTERN MASSACHUSETTS UNIV NORTH DARTMOUTH DEPT --ETC F/G 12/1
SOME EXPERIMENTAL RESULTS ON LINEAR ESTIMATION FOR IMAGE ANALYS--ETC(U)
JAN 81 C H CHEN, R WU, C YEN N00014-79-C-0494
UNCLASSIFIED SMU-EE-TR-81-4 NL

END
DATE
FILMED
2 81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

LEVEL II

Technical Report
Contract Number N00014-79-C-0494
SMU-EE-TR-81-4
January 28, 1981

SOME EXPERIMENTAL RESULTS ON LINEAR ESTIMATION FOR IMAGE ANALYSIS*

BY
C. H. Chen
Rong-Hwang Wu
Chihsung Yen
Department of Electrical Engineering
Southeastern Massachusetts University
North Dartmouth, Massachusetts 02747

*The support of the Statistics and Probability Program of the Office of Naval Research on this work is gratefully acknowledged.

SOME EXPERIMENTAL RESULTS ON LINEAR ESTIMATION FOR IMAGE ANALYSIS

C. H. Chen
Rong-Hwang Wu
Chihsung Yen

1. Introduction

In a previous report (ref. 1), a critical comparison is made on the various statistical image segmentation techniques. Linear estimation plays an important role in image segmentation which may be considered as an estimation problem. Included in the linear estimation methods considered in Ref. 1 are the parameter estimation for ARMA models, Fisher's linear discriminant, maximum likelihood and maximum a posteriori estimations for the regions, boundary estimation via split-and-merge algorithm, and decision-directed estimation using conditional population-mixture model. In this report, experimental results are presented for a textured subimage on the segmentation by maximum likelihood estimation, maximum a posteriori estimation, and Fisher's linear discrimant. Among the three methods, the maximum a posteriori estimation performs the best with 3.96% segmentation error, the Fisher's linear discriminant is a close second with 4.6% segmentation error and requires slightly more computation. The performance of the maximum likelihood estimation is much worse even though it requires less computation than the other two methods.

2. Construction of the Textured Subimage

The construction of the 64x64 textured subimage was kindly described by Dr. Charles W. Therrien of MIT Lincoln Laboratory who employed the textured subimage in his study of the linear filtering models for image segmentation and classification (Ref. 2). The desired subimage is constructed from subimages 5 and 9 (counting from upper left to lower right horizontally) of image B2568-38 of USC data base by using the following procedure. For each point (n,m), the following expressions are evaluated.

$$B1(n, m) = (n-36) + (m-40)^2/50$$

$$B2(n, m) = ((n-48)/8)^2 + ((m-20)/12)^2 - 1$$

Then if

$B1 < 0$ <u>or</u> $B2 < 0$, take point from subimage 5

if $B1 > 0$ <u>and</u> $B2 > 0$, take point from subimage 9

This results in a new subimage with the texture from subimage 5 above the parabolic boundary and within the elipse and the texture from subimage 9 everywhere else. The texture from subimage 5 is denoted as type 1 while the texture from subimage 9 is denoted as type 2. Thus the segmentation is to classify for each pixel whether it is from the type 1 or from the type 2 texture. For the results in Ref. 2, the filters were 4x4 causal (first quadrant) computed by solving two-dimensional normal equations with an estimated correlation function. Transition probabilities computed in the paper made use of 5x5 neighborhoods.

By using the lineprinter display with 16 levels, Fig. 1a, 1b and 1c show respectively the original subimages 5, 9 and the constructed subimage. Two-level displays of these subimages from the Tektronix 4010 terminal are shown in Fig. 2. Fig. 3 shows the ideal segmentation as defined by the parabola and elipse in the above quations.

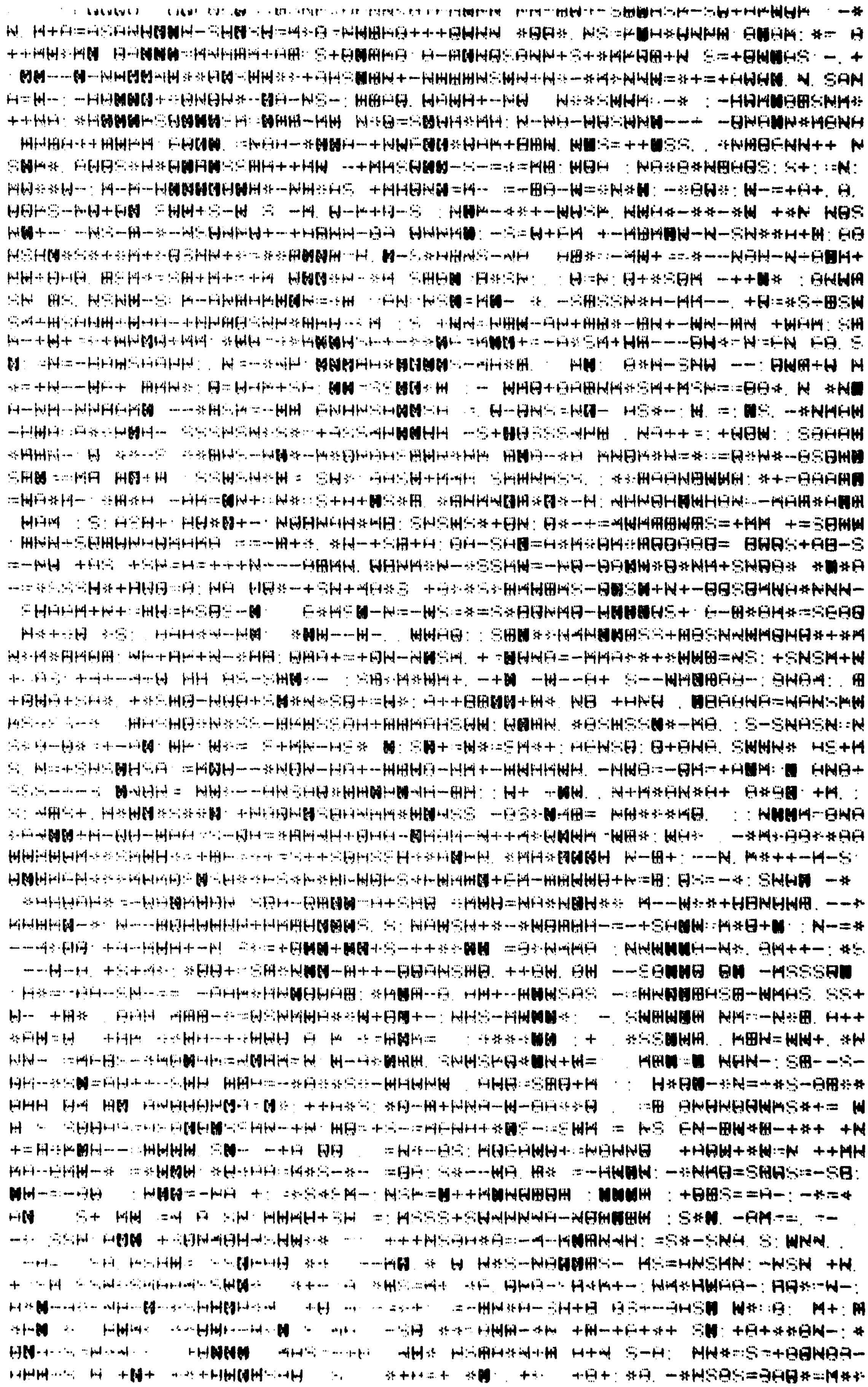


Fig. 1a
Subimage 5

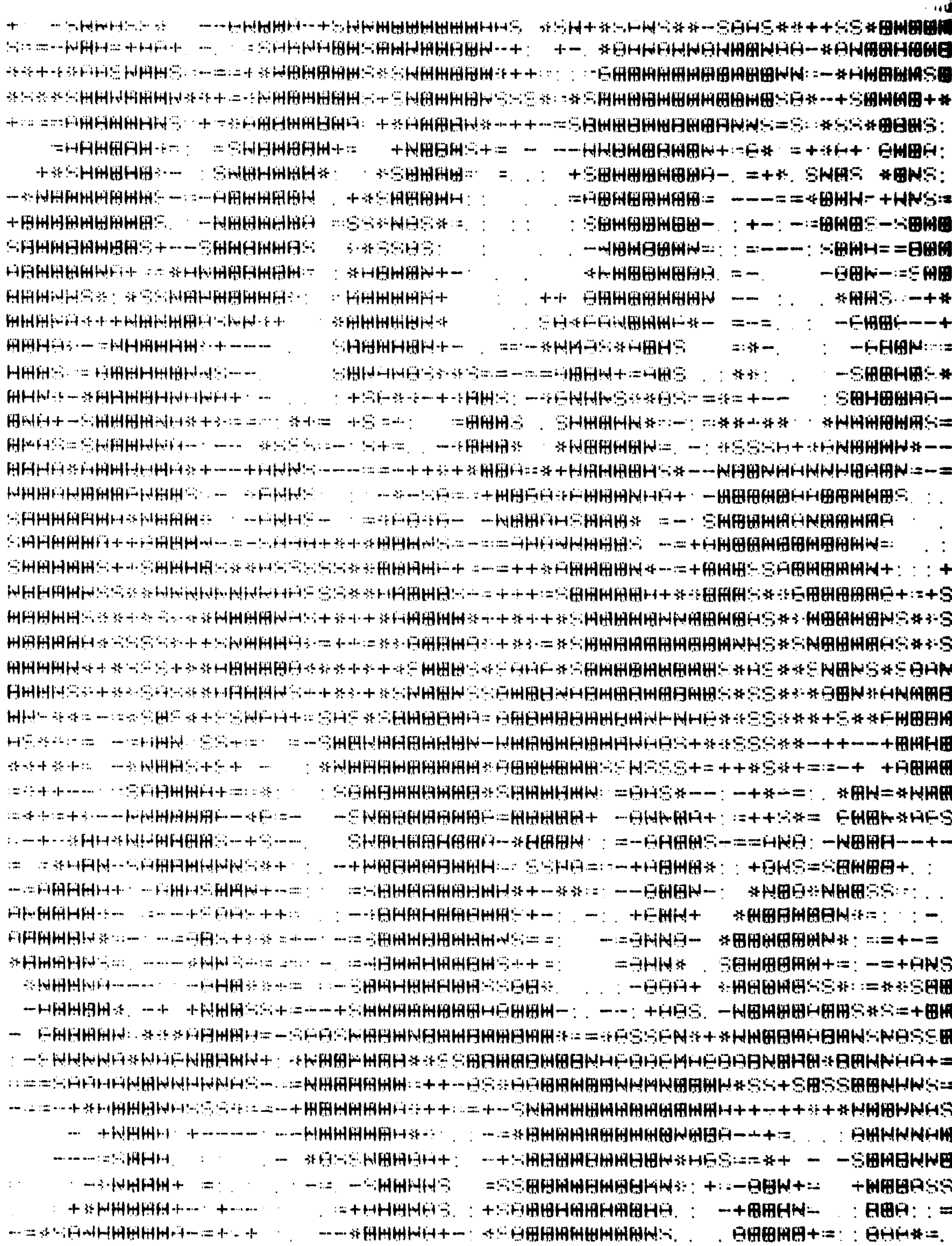


Fig. 1b
Subimage 9

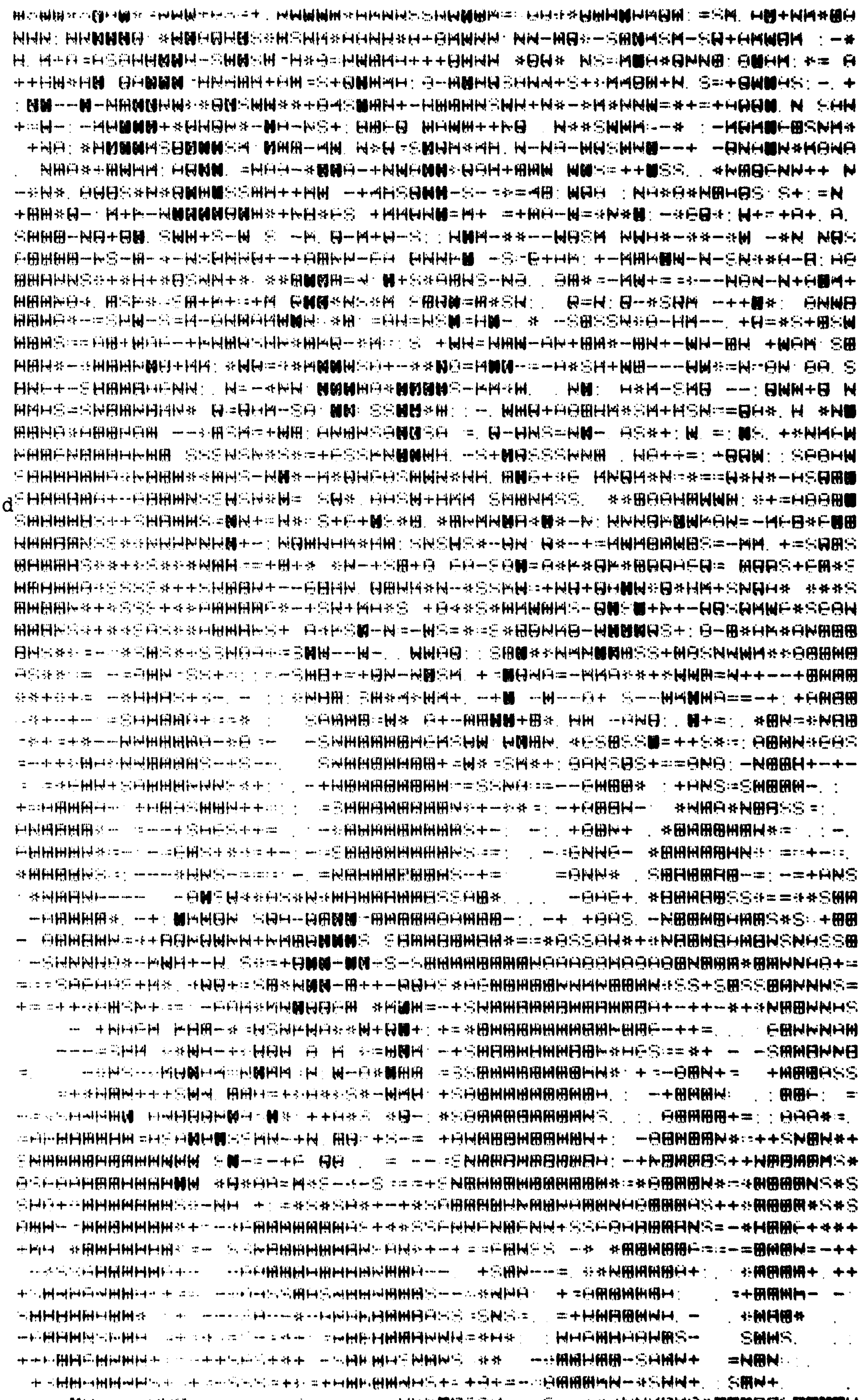
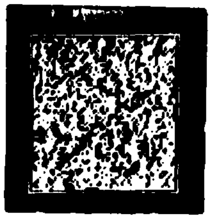


Fig. 1c Constructed Subimage



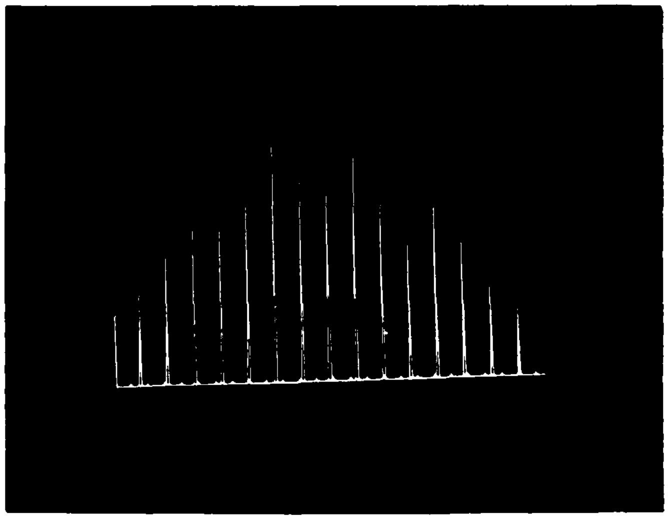
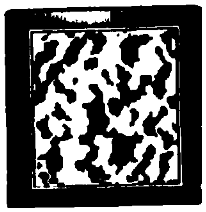


Fig. 2a Subimage 5 (above) in two-level display and histogram of the subimage (right). There are 16 levels in the original image.



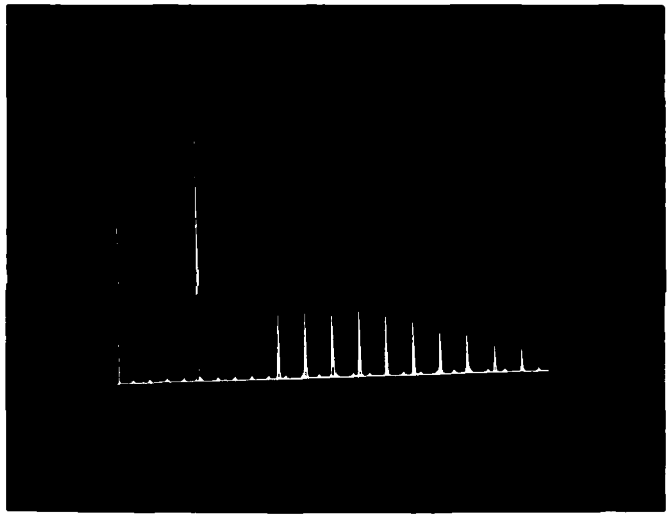
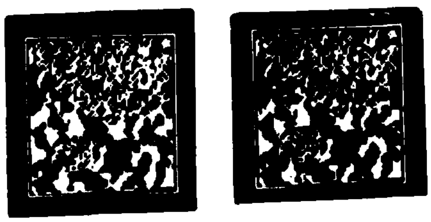


Fig. 2b Subimage 9 (above) in two -level display and histogram of the subimage (right).



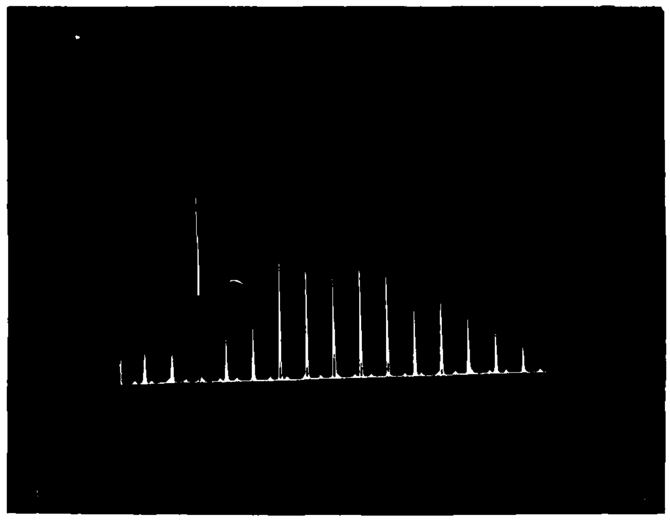


Fig. 2c Subimage constructed (two shown above) and the histogram of the constructed image (right).

3. The Maximum Likelihood Estimation

By assuming a first-order autoregressive model, we followed closely the mathematical development in Ref. 2. The maximum likelihood estimate for the regions is obtained by the minimization of the espression (10) in Ref. 2. Fig. 4 shows the maximum likelihood (ML) segmentation obtained by assigning pixels to white and black texture type 1 and 2 respectively. This result is very close to Fig. 2 (b) of Ref. 2 which employed a higher order autoregressive model as described in the previous section. Our first-order model made use of our earlier work reported in Ref. 3.

4. The Maximum a Posteriori Estimation

The maximum a posteriori estimation which uses the combination of the Markov chain with a Gaussian autoregressive process is to minimize the expression (11) in Ref. 2. To begin with, the image segmentation is considered as an assignment of the pixels to 0 and 1. Such an assignment for a point (n, m) is referred to as its "state". The state interdependence complicates the computation of the estimate and an iterative procedure must be employed. The transition probability that plays a major role in the maximum a posteriori (MAP) segmentation is counted from the numbers of state assignment of 0 and 1 for a 5x5 neighborhood. It is noted that MAP segmentation starts with the ML segmentation result. For clarity the sequence of lineprinter outputs is shown in Fig. 5. They include the ML segmentation (Fig. 5a), Iteration 1 (Fig. 5b), Iteration 4 (Fig. 5c), Iteration 7 (Fig. 5d) and Iteration 11 (Fig. 5e). At Iteration 11, the small false region in the upper left corner starts to disappear. It is noted that such false region was not

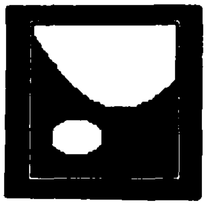


Fig. 3 Ideal segmentation

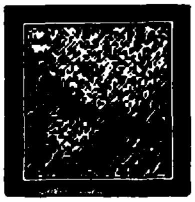


Fig. 4 ML segmentation

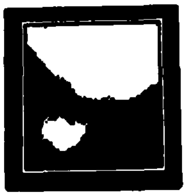


Fig. 6a MAP segmentation with 14 iterations

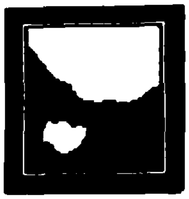


Fig. 6b Smoothing of Fig. 6a

Note: Learning samples for ML and MAP segmentations are based on 480 pixels selected from each type of texture. The following are parameters computed for each texture. Type 1 has mean of 108 and Type 2 has mean of 81.

[illegible]

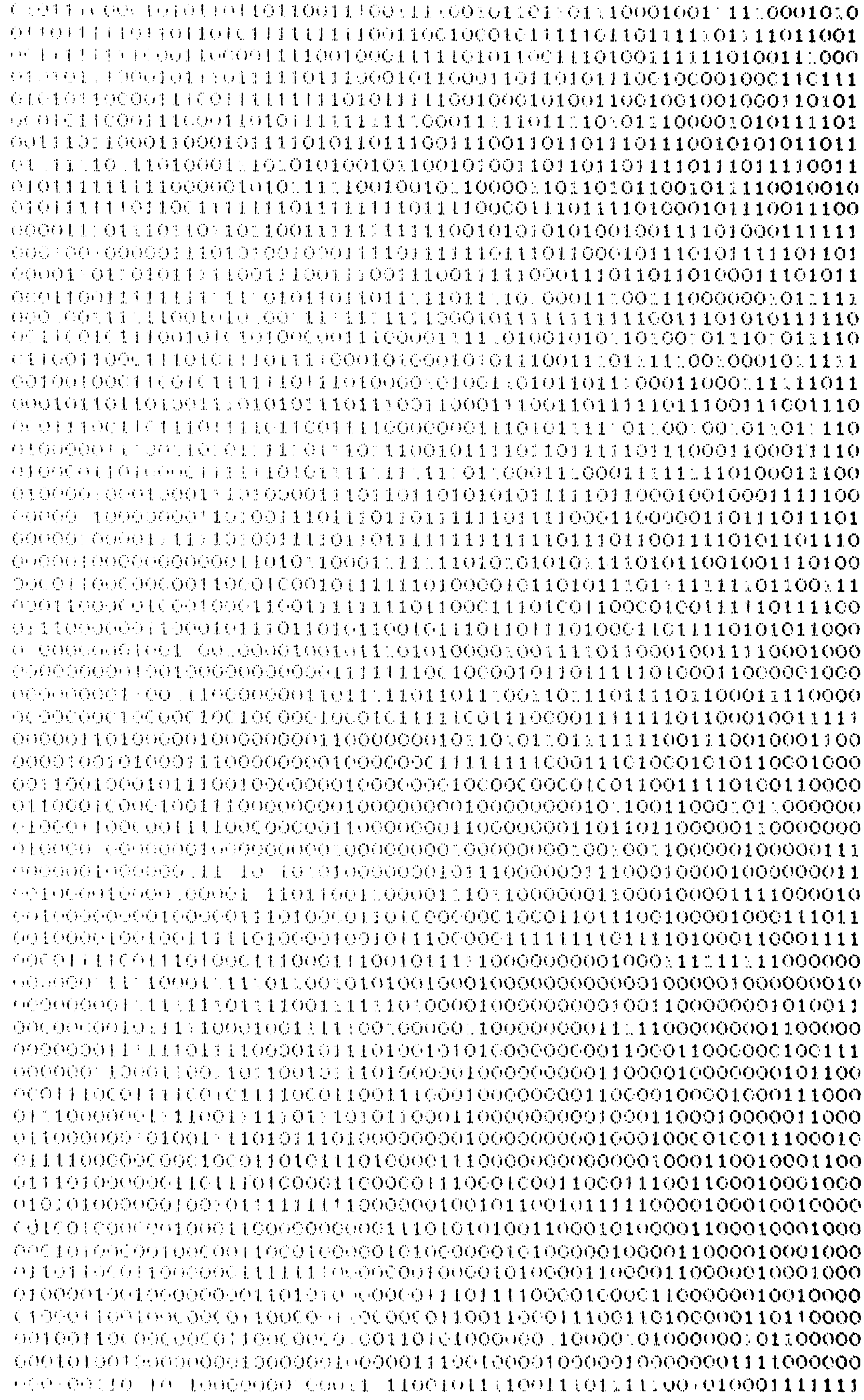


Fig. 5a
ML
segmentation

```
0001111110001111011111111100111110001111111110001111111101111000
0001111100010111011111111110011111000111111111111011000000111100
0001111100011100011111111111111111011111111111101101000000111100
0001111000011000111111111111111001111101111111111100001010111100
0001111101111000011111111111110011000111101111111000111111111100
0001111111111000111111111111111001111001111111111110111111110100
0001111111110000111111111111111110110000111111111110111111101111100
0001111111110011111101111111111111000011101111011111111111111100
0001111111111111111111011111111111110101011111101011111111111111100
0000111111111111111111110111111111111111011110101111101011111100
0000111111111111111111001111111111111111100111111111111000111111100
0000000111111111111101011111111111111111001111111111100000111111100
0000001111111111111111111111111111011111111111111111100101011111100
0000000111111111111101111001111111111111111111010010001011111100
0000000111111010111111111001111100101111100111111111111010101111100
0000001011111011111111111010001010111011111111111100001111111100
0000000011100111111111111110000000111111111111111111111100111111100
0000000011011111111111111111100000011111111111111111110010111011100
0000000110011111111111111111111000011110111111111111110001010111100
0000000100000111111111101111111111111111111111111101111110111101011111100
0000000000000111111111111111111111111111111111111101011100111111100
0000000000000000110011111111111111111111111111111101000110111111100
0000000000000010100111111111111111111111111111111110011110111111100
0000000000000000000101111111111111111111111111111010111111111111100
0000000000000000000101111111111111111111111111111011111111111111100
0000000000000000000011111111111100001111111110000111111111111000
0000000000000000000001011111010001001011111000110111111111000000
0000000000000000000000111111101000011111111111100111111100000000
0000000000000000000000011111111100001011111111111001100000000000
0000000000000000000000011111111011101111111111111110000000000000
0000000000000000000000000000100111110111000111111110100000000000
0000000000000000000000000000011110111001111111111100000001000
0000000000000000001000000000000000111000000011111001110000000000
0000000000000011100000000000000000000000000011111100000011001000000
0000000000000011100000000000000000000000000111001100000001000000
0000000000001111000000000000000000000000000010011000000000000000
0000000000000110000000000000000000000000000011000000000100000000
0000000000000000100000100000000000010000000110000000000000000000
0000000000001000111000000010000000000000001110000000000000000100
0000000000001000011111000000100000000000000011100000000000000000000
0000000000100010111110001000010000000000000000000000000110000000
0000000001011111111110001101100000000000000000000000000000000000
0000000111111111111100011111100100000000000000000000000000000000
0000000111111111111101111111001000000000000000000000000000000000
0000000111111111111001111111100000000000000000000000000000000000
0000000011111111110000111111110000000000000001000000000000000000
0000000001111111111101111110000000000000000000100000000000001101100
0000000011111111111100111100010000000000000000000000000000001000
0000100000000101111111111111000000000000000000000000000000000000
0001000000000011111111110100000000000000000000000000000000000000
0001000000000011111111111100000000000000000000000000000000000000
0001000000000101111110101000000000000000000000000000000000000000
0001000000000000111100000000000000000000000000000000000000000000
0001000000000000111100000000000000000000000010000000000000000000
0000000000000000011100100000000000010100000100000000000000000000
0000000000000000001000000000000000000000000100000010000000000000
0000000000000000000100000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000100000000000000100000
0000000000000000000000000000000001011000000000000010000000001011100000
```

Fig. 5b
Iteration 1
MAP
segmentation

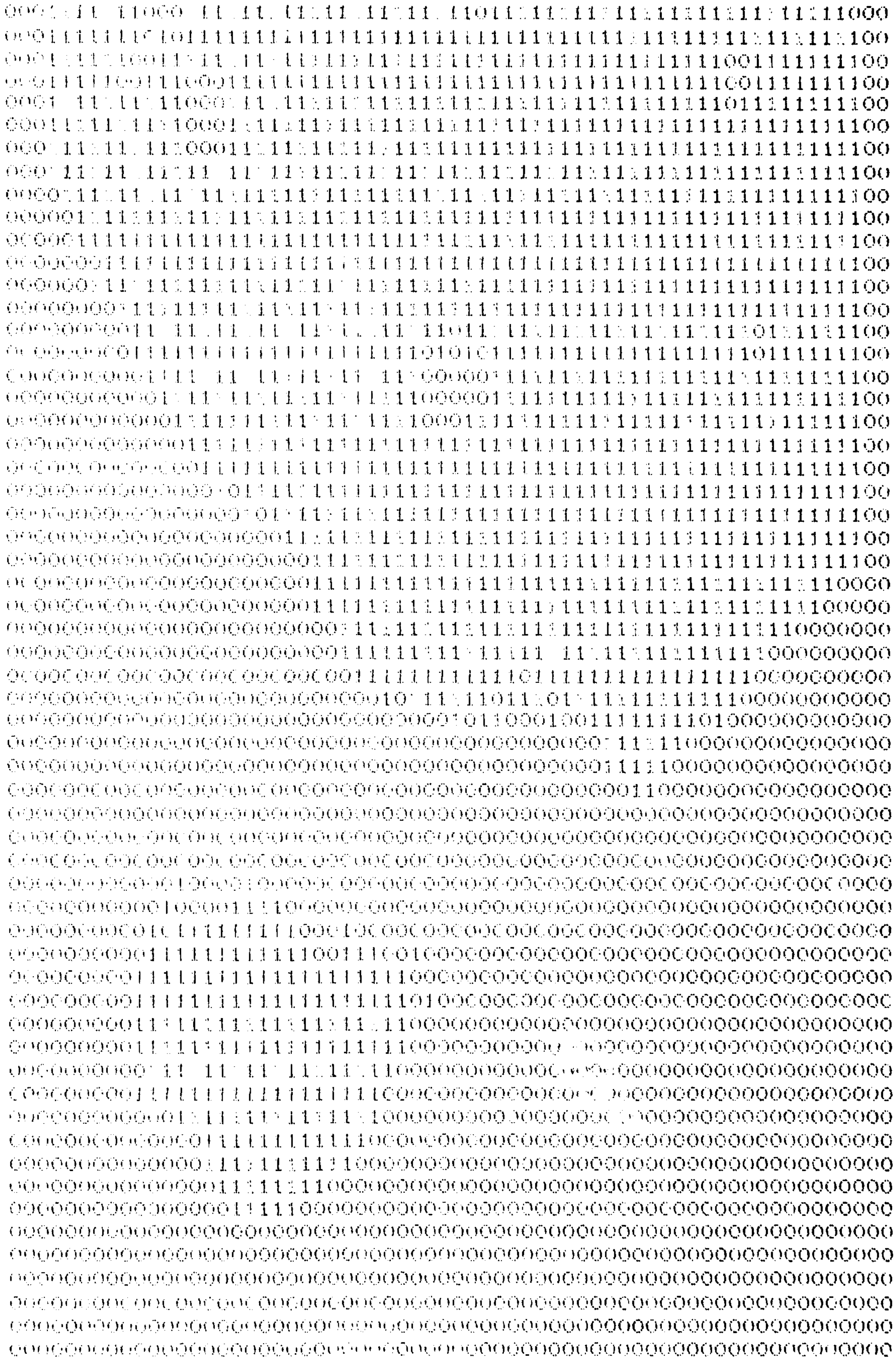


Fig. 5c
Iteration 4
MAP
segmentation

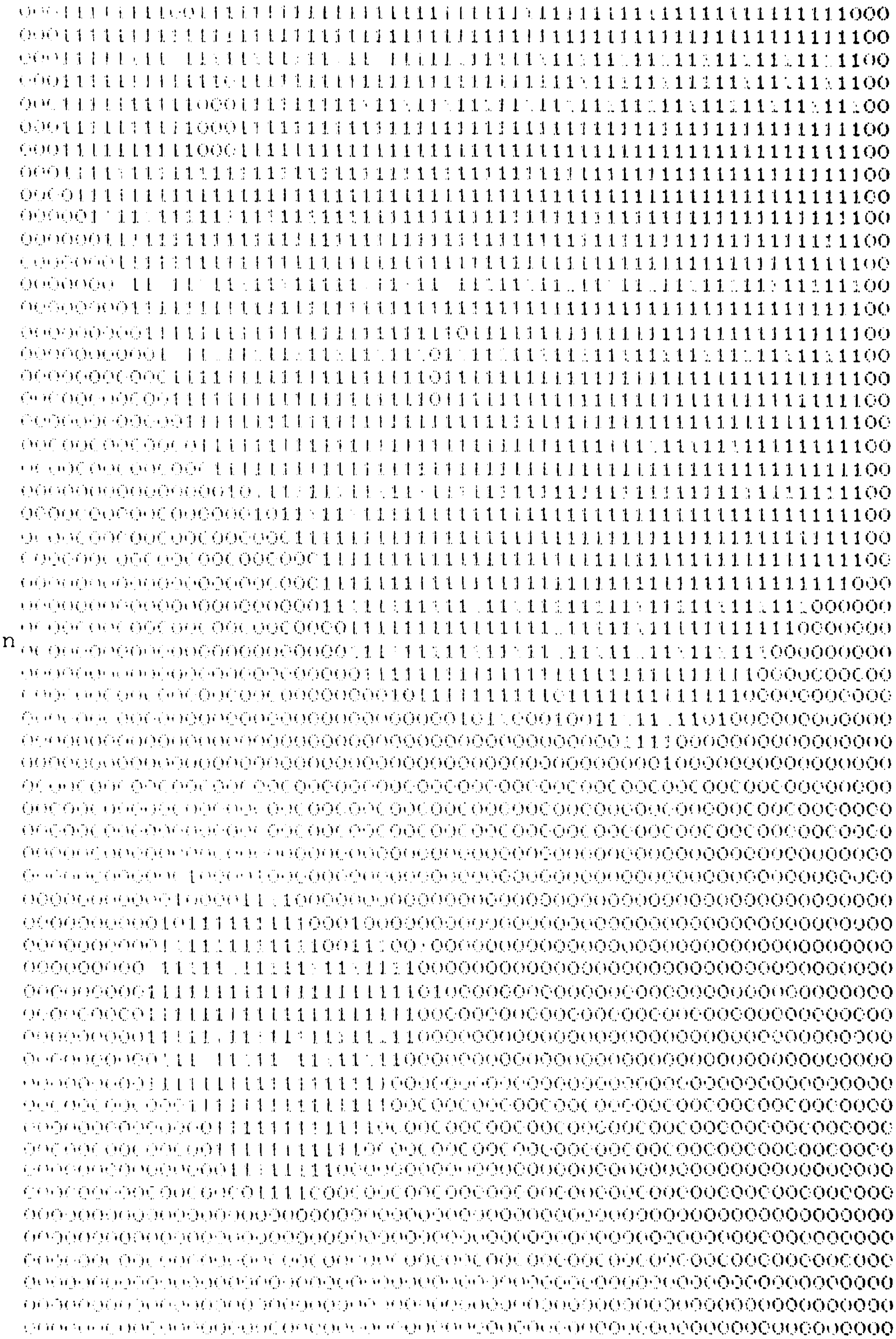


Fig. 5d
Iteration 7
MAP
segmentation

```
0001111111101111111111111111111111111111111111111111111111111000
0001111111111111111111111111111111111111111111111111111111111100
0001111111111111111111111111111111111111111111111111111111111100
0001111111111111111111111111111111111111111111111111111111111100
0001111111111111111111111111111111111111111111111111111111111100
0001111111111111111111111111111111111111111111111111111111111100
0001111111111111111111111111111111111111111111111111111111111100
0001111111111111111111111111111111111111111111111111111111111100
0000111111111111111111111111111111111111111111111111111111111100
0000011111111111111111111111111111111111111111111111111111111100
0000001111111111111111111111111111111111111111111111111111111100
0000000111111111111111111111111111111111111111111111111111111100
0000000111111111111111111111111111111111111111111111111111111100
0000000011111111111111111111111111111111111111111111111111111100
0000000001111111111111111111111111111111111111111111111111111100
0000000000111111111111111111111111111111111111111111111111111100
0000000000011111111111111111111111111111111111111111111111111100
0000000000011111111111111111111111111111111111111111111111111100
0000000000001111111111111111111111111111111111111111111111111100
0000000000000111111111111111111111111111111111111111111111111100
0000000000000011111111111111111111111111111111111111111111111100
0000000000000000011111111111111111111111111111111111111111111100
0000000000000000010111111111111111111111111111111111111111111100
0000000000000000000011111111111111111111111111111111111111111100
0000000000000000000000111111111111111111111111111111111111111100
0000000000000000000000111111111111111111111111111111111111111000
0000000000000000000000111111111111111111111111111111111111000000
0000000000000000000000001111111111111111111111111111111110000000
0000000000000000000000001111111111111111111111111111111000000000
0000000000000000000000001111111111111111111111111111110000000000
0000000000000000000000000001011111111110111111111111000000000000
0000000000000000000000000000000010110001001111111000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000001000010000000000000000000000000000000000000000000000
0000000000010000111100000000000000000000000000000000000000000000
0000000000101111111100010000000000000000000000000000000000000000
0000000000111111111111001110010000000000000000000000000000000000
0000000001111111111111111111110000000000000000000000000000000000
0000000001111111111111111111101000000000000000000000000000000000
0000000001111111111111111111100000000000000000000000000000000000
0000000001111111111111111111100000000000000000000000000000000000
0000000001111111111111111111000000000000000000000000000000000000
0000000001111111111111111110000000000000000000000000000000000000
0000000000001111111111111110000000000000000000000000000000000000
0000000000000111111111111000000000000000000000000000000000000000
0000000000000011111111111000000000000000000000000000000000000000
0000000000000001111111100000000000000000000000000000000000000000
0000000000000000011110000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
```

Fig. 5e
Iteration 11
MAP
segmentation

removed in Ref. 2 even after 16 iterations. At Iteration 14, the Tektronix display is shown in Fig. 6a which is then smoothed by using a procedure due to Sklansky with the result shown in Fig. 6b. The percentage segmentation error of Fig. 6b is computed as 3.96%. It is important to note that for all linear estimation methods considered in this report, the supervised learning samples are selected from regions of type 1 and type 2.

5. Fisher's Linear Discriminant

Our initial effort of using Fisher's linear discriminant follows the procedure employed in Ref. 4. The features are the gray levels of the pixels. Fig. 7a is a tabulation of the scattered matrices for Type 1(target) and Type 2 (background) textures. The two scatter matrices are nearly proportional by a factor of 3. The Fisher's linear discriminant simply cannot classify the two types of textures. A new feature selection procedure is used that extracts three features x(i), i=1,2,3 for a 3x3 neighborhood

$$\begin{matrix} A & B & C \\ D & E & F \\ G & H & I \end{matrix}$$

by computing

$$x(1) = E$$

$$x(2) = (A+B+C+D+E+F+G+H+I)/9$$

$$x(3) = (B+D+F+H) - 4E$$

The learning samples are taken from a 24x28 area with each type of texture region. With each 3x3 neighborhood represented by the feature vector (x(1), x(2), x(3)), the two scatter matrices as tabulated in Fig. 7b are quite different. The Fisher's linear discriminant can then successfully segment the subimage as shown in

Fig. 7a

[illegible]

*** SCATTER MATRIX II ***

[illegible]

Fig. 7b

[illegible]

Fig. 8a Segmentation by Fisher's linear discriminant

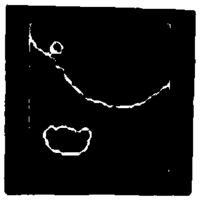


Fig. 8b Sobel gradient operation on Fig. 8a. (threshold is 3.162)

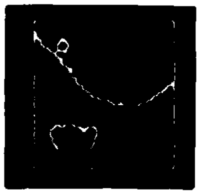


Fig. 8c Robert gradient operation on Fig. 8a. (threshold is 1)

Fig. 8a. The small false region, however, cannot be removed. The results of Sobel and Robert gradient operations on Fig. 8a are shown in Fig. 8b and Fig. 8c respectively. Due to the false region, the percentage segmentation error of Fig. 8a is 4.64%. The Fisher's linear discriminant requires slightly more computation time because of the scatter matrix and feature projection computations.

As a concluding remark, it is interesting to note that while both the maximum a posteriori estimation and the Fisher's linear discriminant are feasible linear estimation methods for image analysis, they employ quite different procedures for utilizing the statistical contextual information.

Acknowledgment We thank Dr. Therrien for helpful communications. The results reported in Sections 3 and 4 are due to R. H. Wu and the results reported in Section 5 are due to C. Yen.

References

1. C. H. Chen, "A comparative evaluation of statistical image segmentation techniques," Technical Report SMU-EE-TR-81-3, January 1981.

2. C. W. Therrien, "Linear filtering models for texture classification and segmentation," Proc. of the 5th International Conference on Pattern Recognition, pp. 1132-1135, Dec. 1980.

3. C. H. Chen and J. Chen, "A comparison of image enhancement techniques," Technical Report SMU-EE-TR-8, Dec. 1979. Also appeared in the Proc. of 1980 IEEE International Conference on Cybernetics and Society.

4. C. H. Chen and C. Yen, "Statistical image segmentation using Fisher's linear discriminant," submitted for publication, 1980.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO. AD-A094281

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

6 Some Experimental Results on Linear Estimation for Image Analysis.

5. TYPE OF REPORT & PERIOD COVERED

9 Technical Rep[redacted]rt,

6. PERFORMING ORG. REPORT NUMBER

14 SMU-EE-TR-81-4

7. AUTHOR(s)

10 C. H./Chen, Ron-Hwang/Wu [redacted] Chihsung/Yen

8. CONTRACT OR GRANT NUMBER(s)

15 N00014-79-C-0494

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Electrical Engineering Department
Southeastern Massachusetts University
North Dartmouth, MA 02747

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

NR 042-422

11. CONTROLLING OFFICE NAME AND ADDRESS

Statistics and Probability Program
Office of Naval Research, Code 436
Arlington, VA 22217

12. REPORT DATE

11 28 Jan[redacted]981

13. NUMBER OF PAGES

19

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

12 19

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)

Maximum likelihood estimation
Maximum a posteriori estimation
Transition probability
Fisher's linear discriminant

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Based on a constructed textured subimage, an experimental comparison is made on the three linear estimation methods for image segmentation, namely the maximum likelihood estimation, the maximum a posteriori estimation and the Fisher's linear discriminant. The last two methods are shown to be feasible for effective image segmentation.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

407932 gu

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DATE

FILMED

8