Pattern Recognition Vol. 12, pp. 251–260.
Pergamon Press Ltd. 1980. Printed in Great Britain.

0031-3203/80/0801-0251 $02.00/0

NEIGHBOR GRAY LEVELS AS FEATURES IN PIXEL CLASSIFICATION

NARENDRA AHUJA and AZRIEL ROSENFELD
Computer Science Center, University of Maryland, College Park, MD 20742, U.S.A.

and

ROBERT M. HARALICK*
Dept. of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, U.S.A.

(*Received* 28 *October* 1977; *in revised form* 14 *November* 1979)

Abstract – In segmenting an image by pixel classification, the sequence of gray levels of the pixel's neighbors can be used as a feature vector. This yields classifications at least as good as those obtained using other local properties (e.g., averages or values of difference operators) as features.

Image Segmentation   Pixel classification   Neighborhood operators

1. INTRODUCTION

This paper describes the application of a supervised classification technique for image segmentation which explicitly uses a pixel's neighborhood information. There is, of course, an abundance of previous research in this area – e.g., for multispectral image classification,[1] FLIR image analysis,[2] and general image segmentation, [3,4] to list just a few. The experiments described here differ from those of previous investigations in that they indicate the superiority of using the pattern of gray levels in the neighbourhood of a point, rather than features computed over these neighbourhoods.

2. SUPERVISED IMAGE SEGMENTATION

In this section we will discuss the application of minimum mean squared error procedures to segmentation using neighborhood grey tone information. Let I be an $m \times m$ image and let $X_{i,j}$ be a feature vector of length a associated with pixel $I(i,j)$. $X_{i,j}$ may consist of, for example, the gray level of $I(i,j)$ and the average gray levels of the 3×3 and 5×5 neighborhoods surrounding $I(i,j)$. In the experiments described here, $X_{i,j}$ is a vector defined by the gray levels in a $k \times k$ neighborhood of $I(i,j)$. The organization of $X_{i,j}$ will be discussed later.

Now, suppose that I has been segmented and that each pixel is labeled with an integer representing the segment to which it belongs. Suppose there are r classes labeled with the integers $L = \{l_1 \ldots l_r\}$. Let

$$Y = \begin{pmatrix} X^t_{1,1} \\ X^t_{1,2} \\ \vdots \\ X^t_{m,n} \end{pmatrix}, \quad c = [c_1, \ldots, c_{mxn}]\ (c_i \in L),$$

and let $w = [w_1 \ldots w_s]^t$. The w that minimizes $Yw - c$ is then $(Y^tY)^{-1}\, Y^t c$[5]. In the two-class case ($r = 2$) if we take $l_1 = -1$ and $l_2 = 1$ then the mean squared error solution approaches a minimum mean squared error approximation to the Bayes discriminant function. If, from the $m \times n$ samples, n_1 are from class 1 and n_2 are from class 2, then choosing $c = [m \times n/n_1, -m \times n/n_2]^t$ results in the Fisher linear discriminant [5].

The weight vector w is used to classify pixels by computing $X^t_i\, w$, where X_i is the feature vector of the pixel. If $X^t_i w > 0$ then the pixel is assigned to class 2 (in the two-class case); otherwise it is assigned to class 1.

In the first set of experiments discussed below, two choices of X_i were investigated:

(1) the average gray levels of 5×5, 3×3 and 1×1 neighborhoods of a point, and

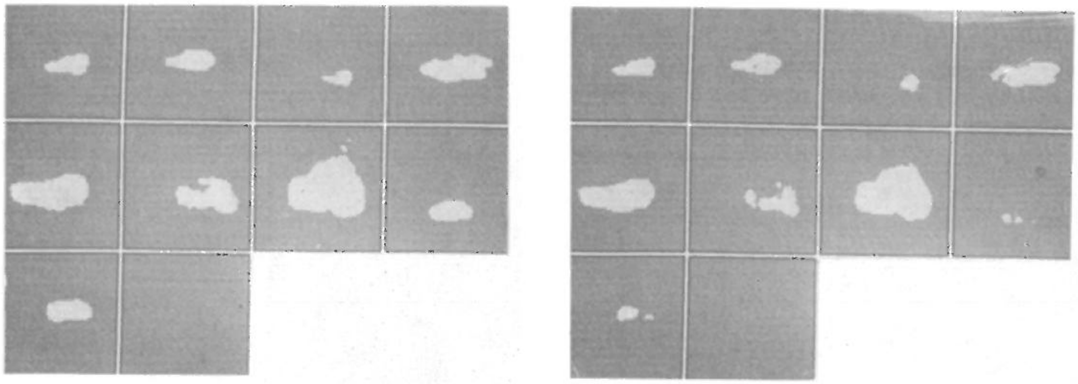
(2) the individual gray levels in a $k \times k$ neighborhood of a point.

In the second case, one would suppose that the arrangement of the gray levels in the vector X_i should be important. For example, suppose that X_i were constructed by a raster scan of the $k \times k$ neighborhood of a point, and that one of the classes in our training set is composed of gray level patterns that have strong directional properties. Then if subsequent images contain objects in this class in different orientations from those of the training image, we would expect the classification results to be poor for those objects.

One possible solution to this problem is to construct

---

* Reprint requests to Dr. Haralick.



(c) (d)

Fig. 4. Analogous to Fig. 3, using as features the 9 gray levels in a 3 × 3 neighborhood of each point.

14T
THR WEIGHT = 2.76

−0.11	−0.00	−0.01	0.05	−0.14
−0.00	0.06	0.05	0.10	0.10
0.02	0.09	0.18	0.03	0.02
0.02	0.04	0.04	−0.00	0.04
−0.08	−0.01	0.01	0.02	−0.11

31T
THR WEIGHT = 12.68

−0.00	0.08	0.01	0.00	−0.10
0.08	0.13	0.02	0.09	−0.04
0.03	0.05	0.25	0.07	0.00
0.02	0.16	0.08	0.15	0.06
−0.11	0.03	0.01	0.02	−0.13

28T
THR WEIGHT = MISSED

−0.03	0.01	0.00	−0.02	−0.05
0.04	0.09	0.03	0.09	0.01
0.06	0.13	0.32	0.05	0.07
0.01	0.14	0.07	0.16	0.07
−0.14	−0.05	−0.05	−0.00	−0.07

33T
THR WEIGHT = −3.19

0.05	0.01	0.04	0.04	−0.00
0.01	0.06	0.01	0.03	0.00
0.00	0.02	0.09	0.05	−0.00
0.55	0.06	0.06	0.08	−0.05
0.00	0.03	−0.00	0.04	−0.05

Fig. 5. Weight matrices for the 25 features used in Fig. 3.

14T
THR WEIGHT = 2.52

−0.03	0.04	0.05
0.10	0.21	0.10
−0.02	0.02	−0.04

31T
THR WEIGHT = −12.30

0.16	0.04	0.05
0.10	0.24	0.10
0.10	0.07	0.10

28T
THR WEIGHT = 11.49

0.10	0.05	0.11
0.11	0.32	0.09
0.01	0.02	0.10

33T
THR WEIGHT = −2.04

0.10	0.06	0.07
0.03	0.09	0.02
0.09	0.08	0.04

Fig. 6. Weight matrices for the 9 features used in Fig. 4.

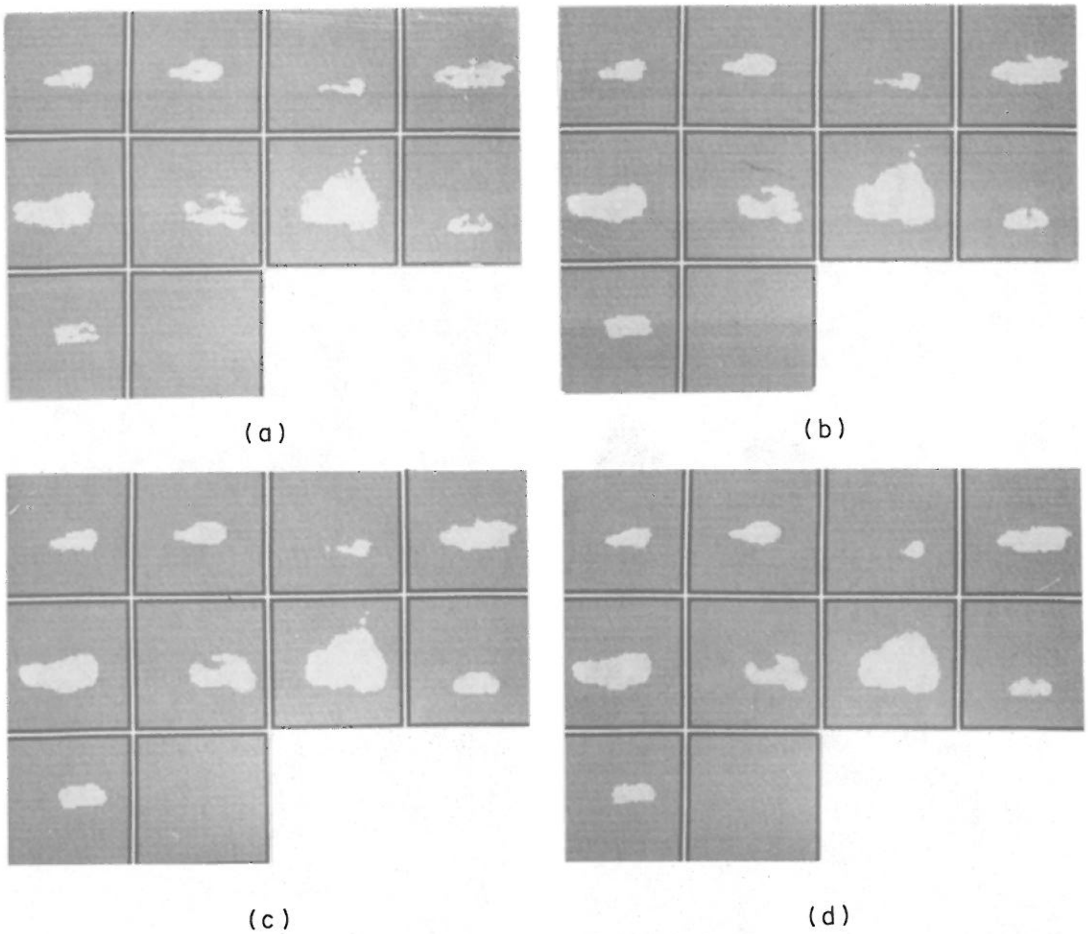


Fig. 7. Analogous to Fig. 3, using as features the average gray levels in 1×1, 3×3, and 5×5 neighborhoods of each point.

for best spatial edge coherency on each connected component.

To determine the advantage while still generating the ground truth automatically, we chose another set of Forward Looking InfraRed images, let the Superslice algorithm determine a segmentation which we took to be ground truth, and then added random noise to the imagery. Any pixel by pixel thresholding procedure, even one like Superslice, operating on noisy imagery should have some trouble segmenting the image well because the pixel to pixel grey tone may have wide variations and the local internal spatial coherency occurring in neighborhoods is not used.

In our second experiment, we selected a new set of 25 Forward Looking InfraRed (FLIR) images showing dark objects on a light background. The ten images used for training are shown in Fig. 9. To obtain training samples, the images were thresholded as shown in Fig. 10; the thresholds were chosen, as before, using the Superslice algorithm [6]. From each of the ten images, 12 points were chosen from the interior of the object and 12 from the interior of the background, yielding a total of 120 object samples and 120 background samples.

Three sets of features extracted from these samples were used to train classifiers. Let E be the gray level of a

14T
THR WEIGHT = 2.53
−0.31 0.48 0.26

31T
THR WEIGHT = −12.45
0.29 0.24 0.44

28T
THR WEIGHT = −11.99
0.14 0.26 0.54

33T
THR WEIGHT = −2.52
0.31 0.10 0.19

Fig. 8. Weight vectors for the 3 features used in Fig. 7.

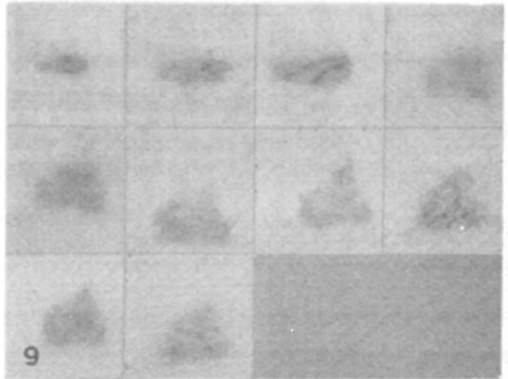
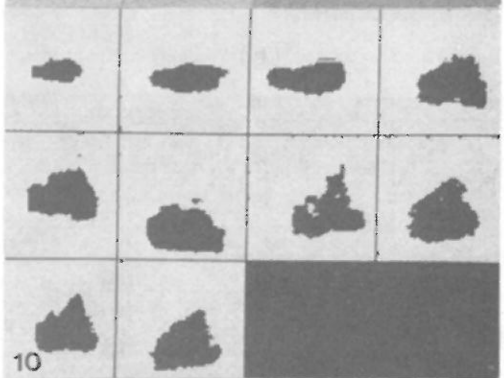
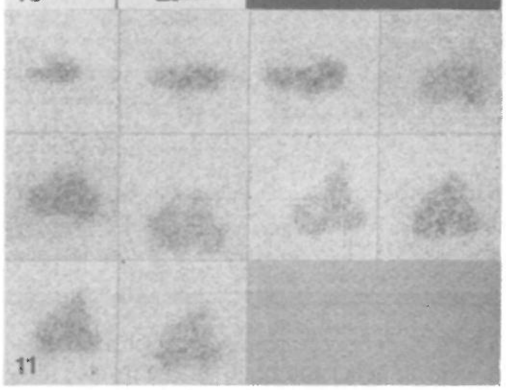




Fig. 9. Training images (no noise added).

Fig. 10. Superslice thresholds for training images.

Fig. 11. Training images with Gaussian noise added ($\sigma = 4$).

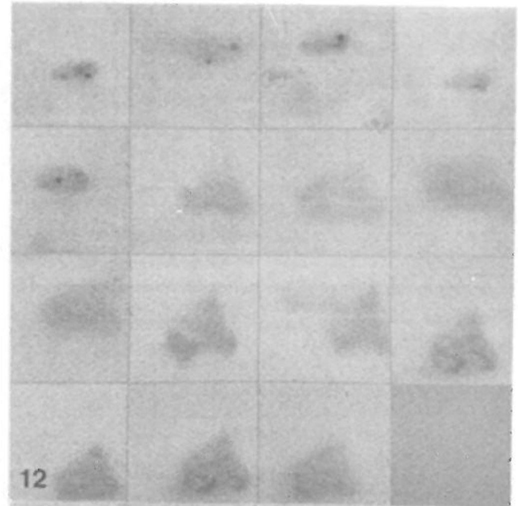
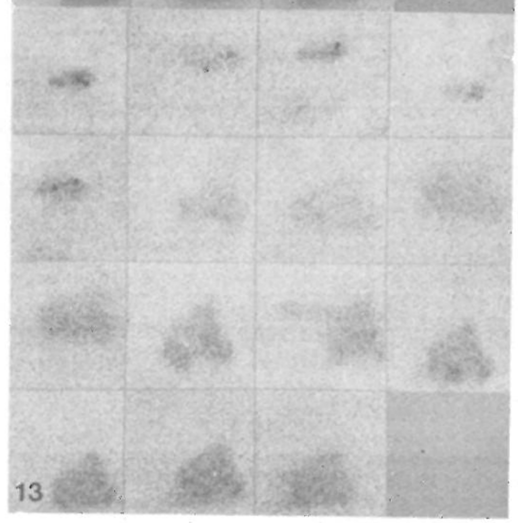



Fig. 12. Test images (no noise added)

Fig. 13. Test images with Gaussian noise added ($\sigma = 4$)

sample point, and let the gray levels of E's neighbors be

ABC
DEF
GHI

Then the three sets of features were as follows:

(a) E (pixel grey level)
(b) A,B,C,D,E,F,G,H,I (8-neighborhood)
(c) E; max (|D–E|,|E–F|); max (|B–E|,|E–H|); B + D + F + H − 4E (local features).

Each set of features was computed for the 240 sample points extracted from the images in Fig. 9. They were also computed for the same points in a set of noisy images obtained by adding independent Gaussian noise with $\mu = 0, \sigma = 4$ to the images of Fig. 9 (see Fig. 11).

In training the classifiers, object samples were labelled + 1 and background samples − 1. The resulting feature weights and thresholds for the four feature sets, using the images with and without noise, are given in Table 1.

The remaining set of 15 images, shown in Fig. 12, constituted the test data. Discriminant values were computed for every pixel in these images using each of the four classifiers obtained from the non-noisy traning data. The resulting numbers of errors are shown in Table 2. In addition, Gaussian noise with $\mu = 0, \sigma = 4$ was added to the test images (Fig. 13); discriminant values were computed using the classifiers obtained from the noisy training data. These errors are also shown in Table 2. The locations of the errors are shown in Figs 14 and 15 for the non-noisy training sets, respectively.

Table 1. Classifier weights and thresholds for training data

Feature set	No. of features	No noise Weight(s)	Threshold	Noise Weight(s)	Threshold
(a) pixel gray level	1	0.136	– 3.600	0.097	– 2.571
(b) 8-neighborhood	9	0.037 – 0.015 0.043 0.005 – 0.007 0.017 0.009 0.028 0.025	– 3.741	0.017 0.022 0.009 0.019 0.008 0.014 0.010 0.020 0.016	– 3.569
(c) local features	4	0.141; 0.004; – 0.021; 0.042	– 3.716	0.129; 0.003; – 0.003; 0.028	– 3.426

For the non-noisy data, using the pixel's gray level as the sole feature gives by far the best results. This is undoubtedly because the errors are defined in terms of a segmentation based on gray level thresholding (Superslice). Moreover, the training samples were all taken from the object and background interiors, where the neighbor gray levels are very close to that of the center pixel, and where the difference values are very small; thus when feature sets *b* and *c* are used, errors are common near the object/background borders, where the feature vectors are unlike those of the training data. Even if training samples had been taken at the borders, unambiguous classification of border pixels would still not be possible, since the borders lie in all orientations, so that given arrangements of neighbor gray levels can occur at both object and background points. Figure 14 confirms that the errors are indeed concentrated on the borders. Incidentally, the few errors obtained using set (a) are all on the image borders.

For the noisy data, on the other hand, the pixel gray level feature gives by far the worst results. This is because the noise makes single gray levels unreliable as features, whereas sets of neighbor gray levels provide redundancy. The error rate seems to depend primarily on the number of features used; it is lowest for the 8-neighborhood features (nine-dimensional), and about the same for the 4-neighborhood (not shown here) and local features, with five and four features, respectively. However, the 4-neighborhood feature set has two advantages over the local feature set.

(1) The features require no computation, since they are simply neighbor gray levels, not arithmetic combinations of such gray levels.

(2) They give the classifier maximum flexibility by allowing it to work with the basic gray level data, rather than forcing it to work with specified combinations of gray levels from which the original gray levels can no longer be retrieved (e.g., since absolute value and max operations are involved).

Table 2. Numbers of errors for non-noisy and noisy test data

Test Image	non-noisy test data Set (a)	Set (c)	Set (d)	noisy test data Set (a)	Set (c)	Set (d)
6T	15	16	14	196	21	29
9T	107	122	115	454	114	137
11T	195	182	170	547	217	244
14T	45	50	53	211	63	74
15T	297	304	288	897	398	448
22T	74	75	80	499	115	164
42T	334	315	338	721	396	438
46T	253	252	253	856	330	368
48T	182	192	178	605	186	208
76T	206	212	207	337	213	209
80T	192	199	209	397	206	238
95T	1	50	42	459	103	118
99T	45	57	45	224	70	71
105T	68	69	73	450	104	118
114T	201	185	196	585	209	229

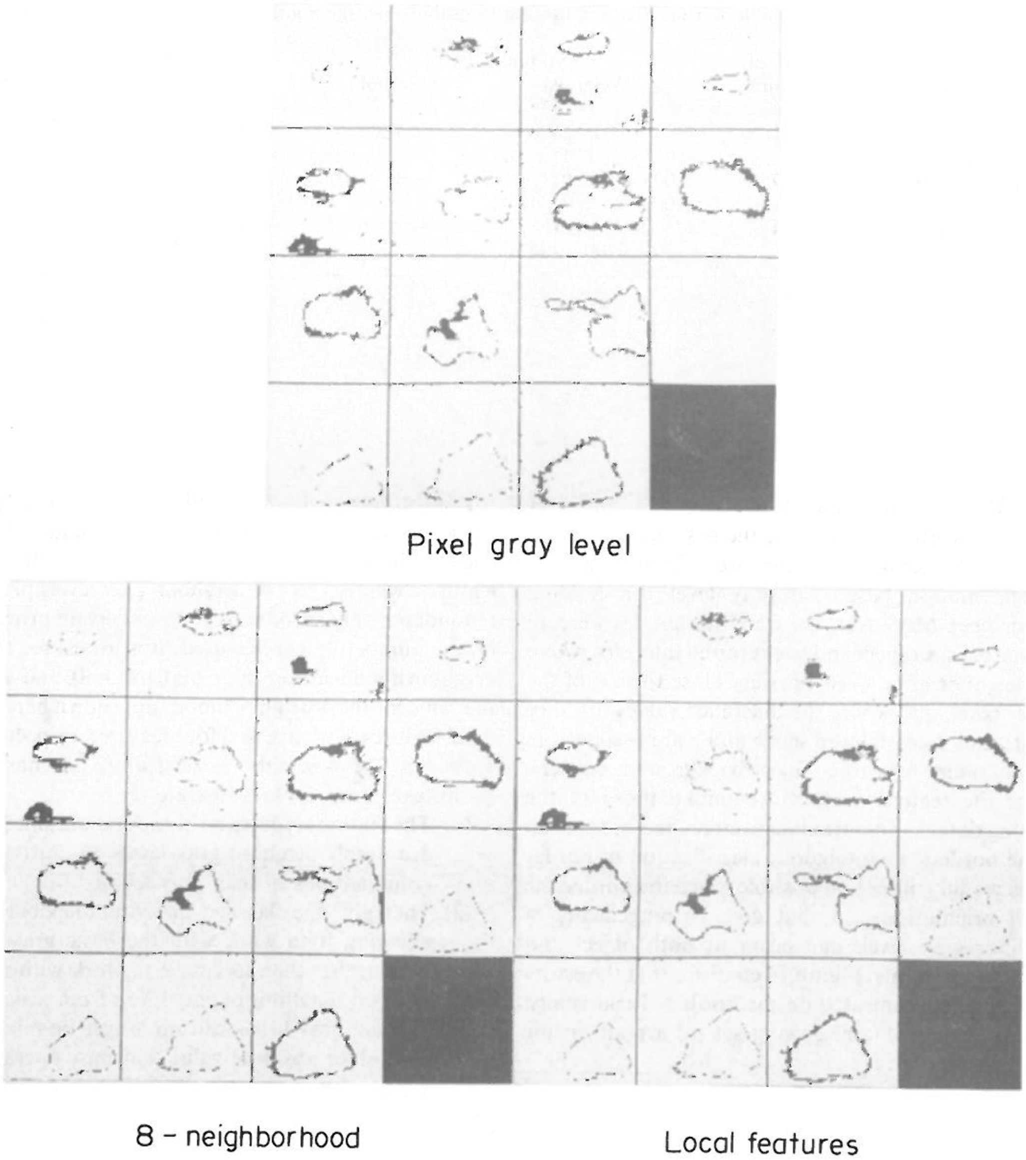


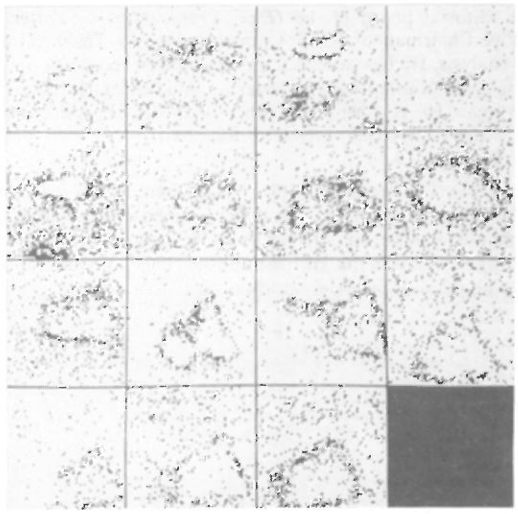
Fig. 14. Illustrates in black those pixels that were misclassified in the no-noise data.

3. CONCLUSION

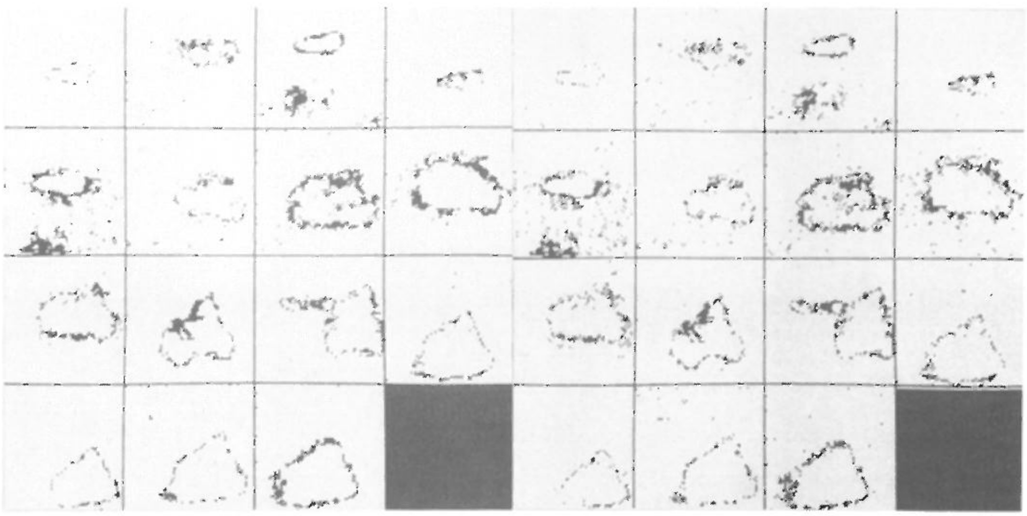
These results suggest that the gray levels of the pixel and its neighbors are a good set of features to use in pixel classification. There seems to be no advantage in using functions of the gray levels as features, even if these functions have simple intuitive interpretations in terms of the local image structure (i.e., they respond to edges, spots, etc.).

REFERENCES

1. R. M. Haralick and I. Dinstein, A spatial clustering procedure for multi-image data, *IEEE Trans. Circuits Systems* **CAS-22,** 440–450 (1975).
2. D. P. Panda and A. Rosenfeld, Image segmentation by pixel classification in (gray level, edge value) space, *IEEE Trans. Comput.* **C-27,** 875–879, (1978).
3. R. Ohlander, K. Price, and D. R. Reddy, Picture segmentation using a recursive region splitting method, *Comput. Graphics Image Processing,* **8,** 313–333, (1978).
4. B. J. Schachter, L. S. Davis and A. Rosenfeld, Some experiments in image segmentation by clustering of local feature values, *Pattern Recognition,* **11,** 19–28 (1978).
5. R. Duda and P. Hart, *Pattern Classification and Scene Analysis,* John Wiley, New York (1973).
6. D. L. Milgram, Region extraction using convergent evidence, *Comput. Graphics Image Processing,* **11,** 1–12, (1979).



Pixel gray level



8 – neighborhood Local features

Fig. 15. Illustrates in black those pixels that were misclassified in the noisy data.

About the Author – NARENDRA AHUJA was born in Achera, India, on December 8, 1950. He received his B.E. degree with Honors in Electronics Engineering from the Birla Institute of Technology and Science, Pilani, India (1972), the M.E. degree with distinction in Electrical Communication Engineering from the Indian Institute of Science, Bangalore, India (1974), and the Ph.D. degree in Computer Science from the University of Maryland, College Park, Maryland (1979).

From 1974 to 1975 he was Scientific Officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979 he was at the Computer Vision Laboratory, University of Maryland, College Park as a graduate research assistant (1975–1978), as a faculty research assistant (1978–1979) and as a research associate (1979). In 1979 he joined the University of Illinois at Urbana-Champaign where he is currently a Research Assistant Professor in the Coordinated Science Laboratory and an Assistant Professor in the Coordinated Science Laboratory and an Assistant Professor in the Department of Electrical Engineering. His research interests are in image processing and pattern recognition. He is a member of the ACM, the IEEE Computer Society and Eta Kappa Nu.

About the Author – ROBERT M. HARALICK received a Ph.D. from the University of Kansas in 1969 and is presently a professor in the Department of Electrical Engineering and in the Department of Computer Science at Virginia Polytechnic Institute and State University. He is an associate editor of the *IEEE Transactions on Systems, Man, and Cybernetics, Pattern Recognition,* and *Computer Graphics and Image*

Processing. He is on the editorial board of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and is the Vice-Chairman of IEEE Computer Society Technical Committee on Machine Intelligence and Pattern Analysis. He has done research in pattern recognition, image processing, remote sensing, texture analysis, image data compression, clustering, and artificial intelligence, publishing numerous research papers in these areas. He is a senior member of the IEEE and a member of ACM, Pattern Recognition Society, and Society for General Systems Research.

About the Author – AZRIEL ROSENFELD received the Ph.D. in mathematics from Columbia University in 1957. After ten years in the defense electronics industry, in 1964 he joined the University of Maryland, where he is Research Professor of Computer Science. He edits the journal *Computer Graphics and Image Processing*, and is president of the consulting firm ImTech, Inc. He has published ten books and over 200 papers, most of them dealing with the computer analysis of pictorial information.