FINDING EDGES IN NATURAL TEXTURES

Amar Mitiche
Larry S. Davis

TR 80 - 4 October 1980

This research was supported by funds derived from the Air Force Office of Scientific Research under contract F49620-79-C-0043 and the Joint Services Electronics Program under contract F49620-77-C-0101.

1. Introduction

Several recently proposed texture description models have been based on computing statistics which measure the spatial distribution of local features in textures. The most salient local features are the edges of the texture elements comprising the texture. Davis et al [1,2] have suggested generalized cooccurrence matrices (GCMs) as a source of such statistics. GCMs are an extension of gray level cooccurrence matrices (Haralick et al [3]) to sparse edge arrays. Other edge based texture models have been proposed by Nevatia and Price [4] and Marr [5].

The general edge detection problem has received considerable attention (see, e.g., Nahi [6], Modestino and Fries [7], Shanmugan et al [8], Cooper and Elliot [9], Marr and Hildreth [10]), but the problem of developing robust edge detectors specifically for textures - which can often be modelled in terms of texture element size, shape, color and placement (Schachter et al [11], Ahuja [12]) - has not been extensively investigated.

An optimal edge detection procedure for cellular textures was developed in Davis and Mitiche [13, 14]. This procedure is based on a class of one dimensional edge operators and one dimensional cellular texture models. It involves thresholding the values of the edge operator, e_k and performing local maxima selection of the above threshold values.

Given a one dimensional image, f(i), i = 1,...,n, e_k is defined as follow :

$$e_k(i) = (1/k) \left[\sum_{j=1}^{k} f(i+j) - \sum_{j=1}^{k} f(i-j) \right]$$

The analysis showed the dependence of the values of a best threshold, in the minimum error sense, and peak selection radius on k, the neighborhood size of the edge detector, P(w) the distribution of cell sizes in the texture and the distributions of gray levels in the texture cells. Those gray level distributions were assumed to be completely specified by their means, m_i and their variances, v_i . This paper describes the implementation of a system for the automatic detection of edges in cellular textures based on the results of the analysis in [13,14] and is organized as follow : Section 2 contains a decription of the main steps of the system. Section 3 and 4 are concerned with some of the most relevant mechanisms used in this system. Section 5 describes applications of the automatic edge detection procedure to synthetic and real textures. Finally Section 6 contains conclusions.

2. System Description

The main computational steps of the system which computes the 'best possible' cellular texture segmentation for a given value k of the neighborhood size are listed below. The meaning of the word 'best' is explained in the discussion of the motivation for this system.

a) Compute an initial segmentation of the texture

b) Compute a minimum error threshold based on that segmentation

c) Compute a new segmentation by applying thresholding and local maxima selection, using the minimum error threshold computed in (b)

d) Compare this new segmentation with the previous one according to some specific measure of closeness (to be discussed subsequently)

e) If the segmentations are similar enough then terminate the process. Else go to step b.

Figure 1 shows a schematic diagram of the system described above. The parameter of the system is the threshold level. This threshold could be computed by selecting a point in the valley of the histogram of e_k values at edges and interior points This would eliminate the loop back to step b from step e. But a threshold computed

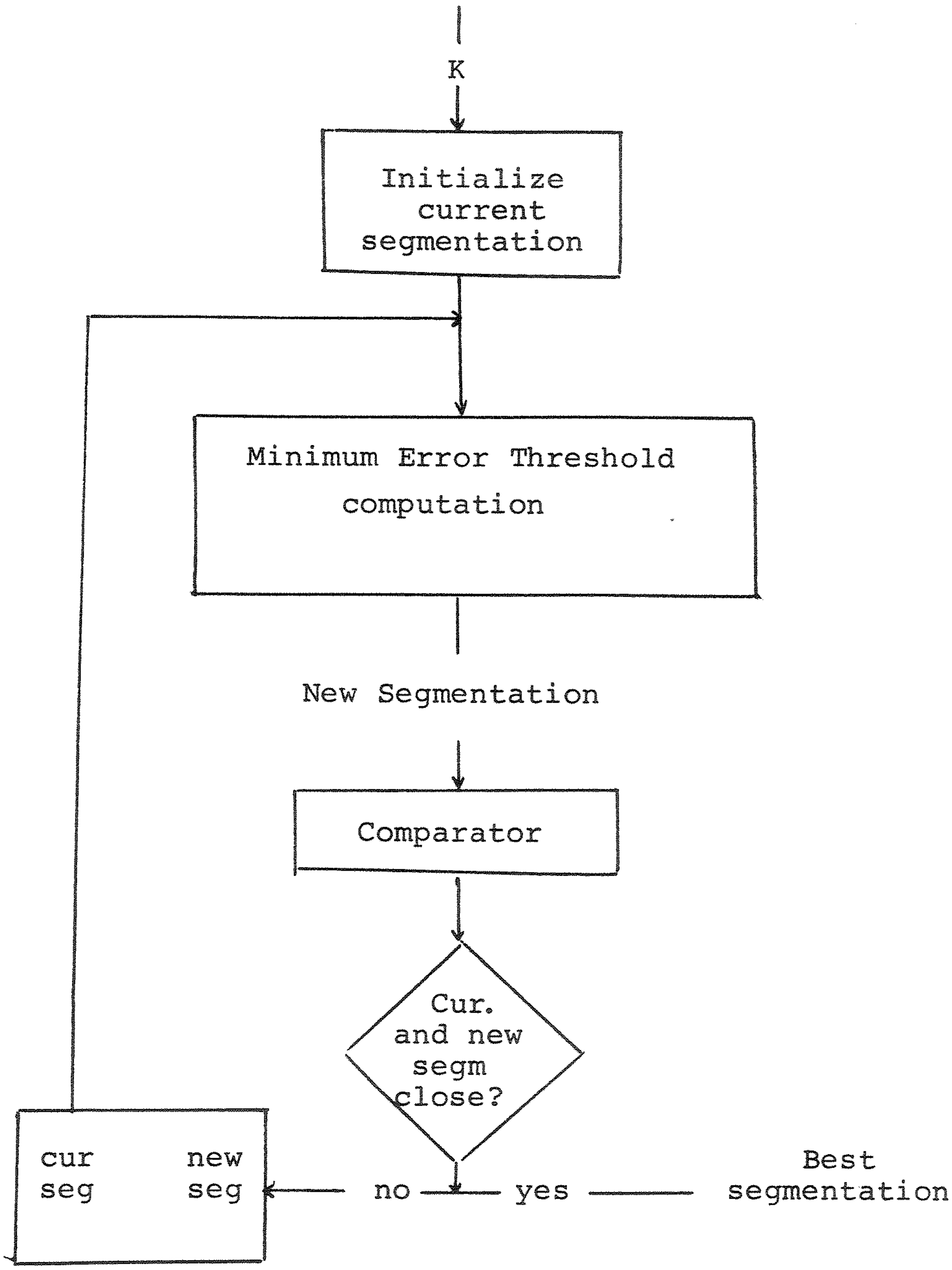


Figure 1

Diagram of a system to compute the segmentation for a given value of K.

in this manner will be entirely dependent on the method for selecting the texture edge/interior points and will therefore be unreliable. Instead, an adaptive threshold can be computed if a form for the distribution of the edge oprerator values at edges and interior points is assumed. At step b the threshold level is determined as follow :

Let z_1 and z_2 be the random variables that decribe the values of e_k at edges and interior points and let $p(z_1)$ and $q(z_2)$ be the respective probability density functions. Finally, let Θ be the fraction of points occupied by edge points in the set of edge/interior points. Then if t is the selected threshold, the overall classification error is

$$\mathrm{Error}(t) = \Theta \int_{-\infty}^{t} p(z_1)\,dz_1 + (1 - \Theta) \int_{t}^{\infty} q(z_2)\,dz_2$$

The value of t that minimises this error is computed by differentiating with respect to t and setting the result to zero, obtaining

$$\Theta\, p(t) - (1 - \Theta)\, q(t) = 0$$

A threshold determined this way will also be dependent on the selected sets of edge and interior points. But the next sections describe how the adaptive threshold method weakens this dependency. In the applications reported in this paper, z_1 and z_2 are assumed to be normally distributed with means and variances computed from the current segmentation. The validity of this assumption was discussed in [13]. The size of the interval in which local maxima selection is performed is not considered a parameter of the system since it was shown in [14] that it does not significantly influence the system performance when it is kept small enough to avoid interference between edges. The motivation for the design of this procedure is the fact that it produces a consistent segmentation. A segmentation, δ_1, is said to be consistent if it yields, following the steps listed above, a threshold value such that thresholding along with local maxima selection produce a new segmentation, δ_2, that is sufficiently similar to δ_1 according to a selected measure of similarity. The following sections describe the main components of the system

3. Current and new segmentations

The edge detection procedure requires an initial segmentation of the texture into edge and interior points. This initial segmentation is computed by applying e_k to the texture and selecting the peaks as edges, i.e., only applying local maxima selection. This procedure has a high false alarm rate but a negligeable false dismissal rate.

As noted in the preceding section, in order to compute a minimum error threshold, and thus a new segmentation from the current segmentation, it is not only necessary to decide which points in the texture are edge points, but also which are interior points. Edge points are the computed edges of the current segmentation. Selecting interior points is more difficult.

The first and simplest way to extract interior points is to consider any point at which the value of the edge operator is below threshold to be an interior point. This solution generally selects too many points and has a tendency to slow the convergence of the edge detection process.

A more efficient method would declare all points in the texture to be interior points except edge points (points at which edges have been computed) and points a fixed distance from these edge points. This solution yields better results than the preceding one. Additional improvement can be obtained by making this distance a decreasing function of the neighborhood size. In this case care should be taken to keep the fraction of interior points from becoming zero when the neighborhood size becomes large. More specifically, if d(k) denotes this distance, n the total number of points in the texture and m the number of edges in the current segmentation then the fraction I of interior points in the texture can be computed as

$$I = \max (I_{min} , 1 - (m + 2d(k))/n)$$

where Imin is a constant representing a lower bound on I.

As indicated earlier, once a new segmentation has been computed it is compared for similarity to the preceding one. The simplest way to compare two segmentations is to compare the threshold values derived from these segmentations. Then the segmentations are declared sufficiently similar if these thresholds differ by less than a given tolerence. The two segmentations are compared with more precision if they are checked directly against each other on a point by point basis with some lag allowed. They are then declared sufficiently similar if they differ by less than a given fraction of the points. This alternative is of course more constraining and requires more computational overhead.

4. The Effect Of Neighborhood Size

The system described above takes a given value of neighborhood size as input and produces a corresponding 'consistent' segmentation. In the following, a method is proposed for selecting the best overall neighborhood size from which to compute the final segmentation. The assumption is that there is indeed a measure of goodness of a segmentation that is a function of k, the neighborhood size.

First, the probability of error as a function of k, er(k), as described in [13] can be used as a criteria for selecting a best k. This would require assuming a parametric form (normal as in [13] for example) for the distribution of e_k at edges and interior points. Although the threshold determined by using the normality (or other distribution) assumption may be nearly correct because the means of edge operator values at edges and interior points are expected to be widely separated, the error associated with it may not be accurate for most real textures. Therefore the minimum error er(k) should be considered only of marginal practical value.

A measure that is better suited for natural textures is obtained from the segmentation itself. This measure is the ratio

$$r = E^2/V$$

where E is the mean absolute difference in the gray level average between adjacent intervals of the segmentation and V is the variance of such differences.

More specifically, let f(i) i = 1, n be a one dimensional digitized image function (the string of rows of the two dimensional texture) and let X_j , j = 1, m with $X_1 = 1$ and $X_m = n$ be the edge positions of the current segmentation (Edges are added at the beginning and end of each row of the 2-dimensional texture). The expression for the average gray level, a_i, for the i-th interval in the segmentation is given by

$$a_i = \sum_{X_i \le j < X_{i+1}} f(j)/(X_{i+1} - X_i) \qquad 1 < i < m-1$$

Also let $\quad b_i = |a_{i+1} - a_i| \qquad 1 < i < m-2$

Then E is defined as

$$E = \sum_{i=1}^{M-2} b_i/(m-2)$$

and the expression for V is

$$V = \sum_{i=1}^{M-2} (b_i - E)^2/(m-2)$$

Intuitively the ratio reflects how well a piecewise constant function fits the data and thus implicitely assuming that the texture is composed of pieces of fairly uniform gray level. The selected value of k is the one that yields the segmentation with the highest value for this ratio. The higher the number of edges computed at the right positions and the lower the number of edges computed at the wrong positions in the image, the higher the value of this ratio. This ratio thus is an indicator sensitive to the false alarm and false dismisal rates and is such that it takes on higher values for lower error rates. As a qualitative example consider varying the neighbohood size, k, and computing r(k) for a given texture. The relation between k and the error rate was described in [13]. This error rate will have a high value for a small enough k, decrease to an optimum value and start increasing again for higher values of k. Therefore r(k) is expected to have a low value for a sufficiently small k, increase to an optimum value and then start decreasing for higher values of k.

Although being definitely a better measure than the minimum error er(k), the ratio r described above is not sensitive enough to differentiate between segmentations which differ only locally around their edges. This is particularly true for dense segmentations in terms of the number of edges they contain. In other words, the ratio as a criteria will not be responsive to such differences since small displacements of the edges from their original position will not significantly affect the value of the expected value of the gray level average of the individual intervals of the segmentation.

To make the process of selecting the neighborhood size more sensitive to these differences, the average absolute gray level difference around the computed edges Σ can be measured and entered in the ratio r. Formally, let f be an image function and $\{X_i\}$ i =1, m be a set of edges as before. Then if Δ is a given constant,

$$\Sigma = (1/\Delta m) \sum_{Y=1}^{\Delta} \sum_{i=1}^{M} \left| f(X_i - Y) - f(X_i + Y) \right|$$

or simply,

$$\Sigma = (1/m) \sum_{i=1}^{M} \left| e_\Delta(X_i) \right|$$

This mean, Σ , contributes to the accuracy of the process by the fact that it takes a high value for segmentations whose computed edges are close to true edges. Thus the measure $r'(k) = r(k) . \Sigma$ or simply $r'(k) = E\Sigma/V$ should be used instead of r(k) as a measure of goodness of a segmentation.

Now a best overall neighborhood size can be determined by computing $r'(k)$ for a set of of values of k and select the one that gives the highest value for r'.

A more efficient method would compute r' for k_1, k_2 and a third point k_3 between k_1 and k_2 and fit a smooth curve (a cubic spline for example) through these points. The optimum value of k would correspond to the integer value that has the highest value on the fitted curve.

5. Application to synthetic and natural textures

Edges have been computed succesfully in synthetic as well as natural textures using the automatic edge detection process described in the above sections. The edge detection has been performed in the horizontal and vertical directions. The normality assumption was used to determine a minimum error threshold. The interior points were selected to be all the points of the image exept the edge points and points distance k from these edges, k being the size of neighborhood. The fraction of interior points in the image is set to a fixed value when k becomes too large (i.e. when the number of edges and points distance k away from these edges approaches the total number of points in the image). The measure of goodness of a segmentation in both direction is the ratio r'.

Figure 2 shows a synthetic texture which is a 64x64 random checkerboard generated by a constant cell width model with parameter b = 8 and two coloring processes with means m1 = 30 and m2 = 40 and variances v1 = v2 = 10 .

Figure 3 is a plot of the ratio r' as a function of k, the neighborhood size, for both the horizontal and vertical directions. This plot enables the choice of a best k at k = 6. Table 1 reports the corresponding thresholds. It should noted that these thresholds are very similar for k =6 , 7 and 8. It should also be pointed out that although k = 6 gives the best segmentation, the segmentations for k = 7 and k = 8 are comparably good. This is because the

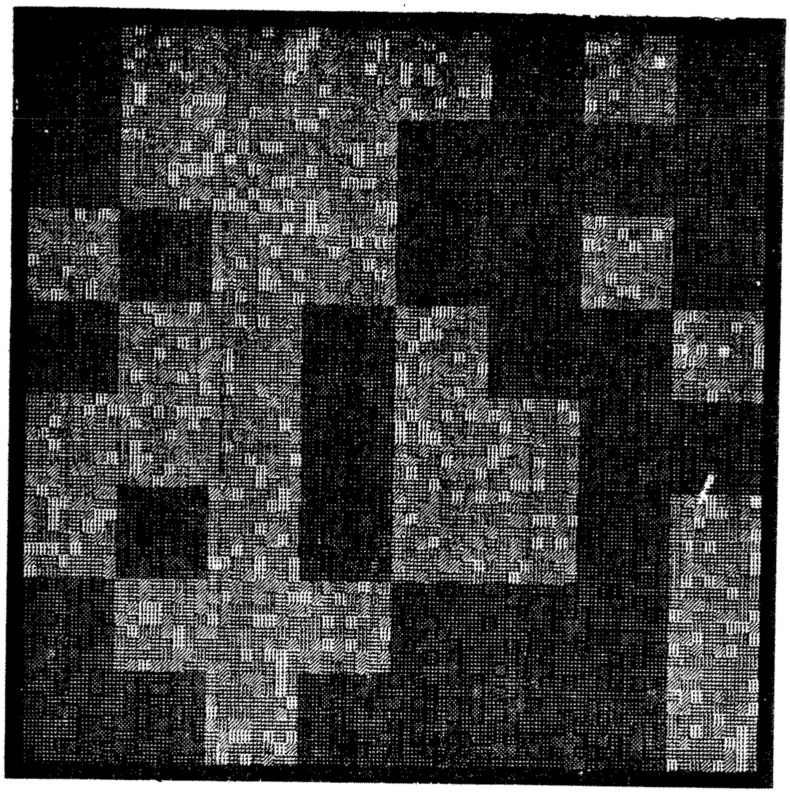


Figure 2

64 x 64 random checkerboard Texture

Darker region	: mean gray level 40 variance 10 mean cell width 8
Lighter region	: mean gray level 30 variance 10 mean cell width 8

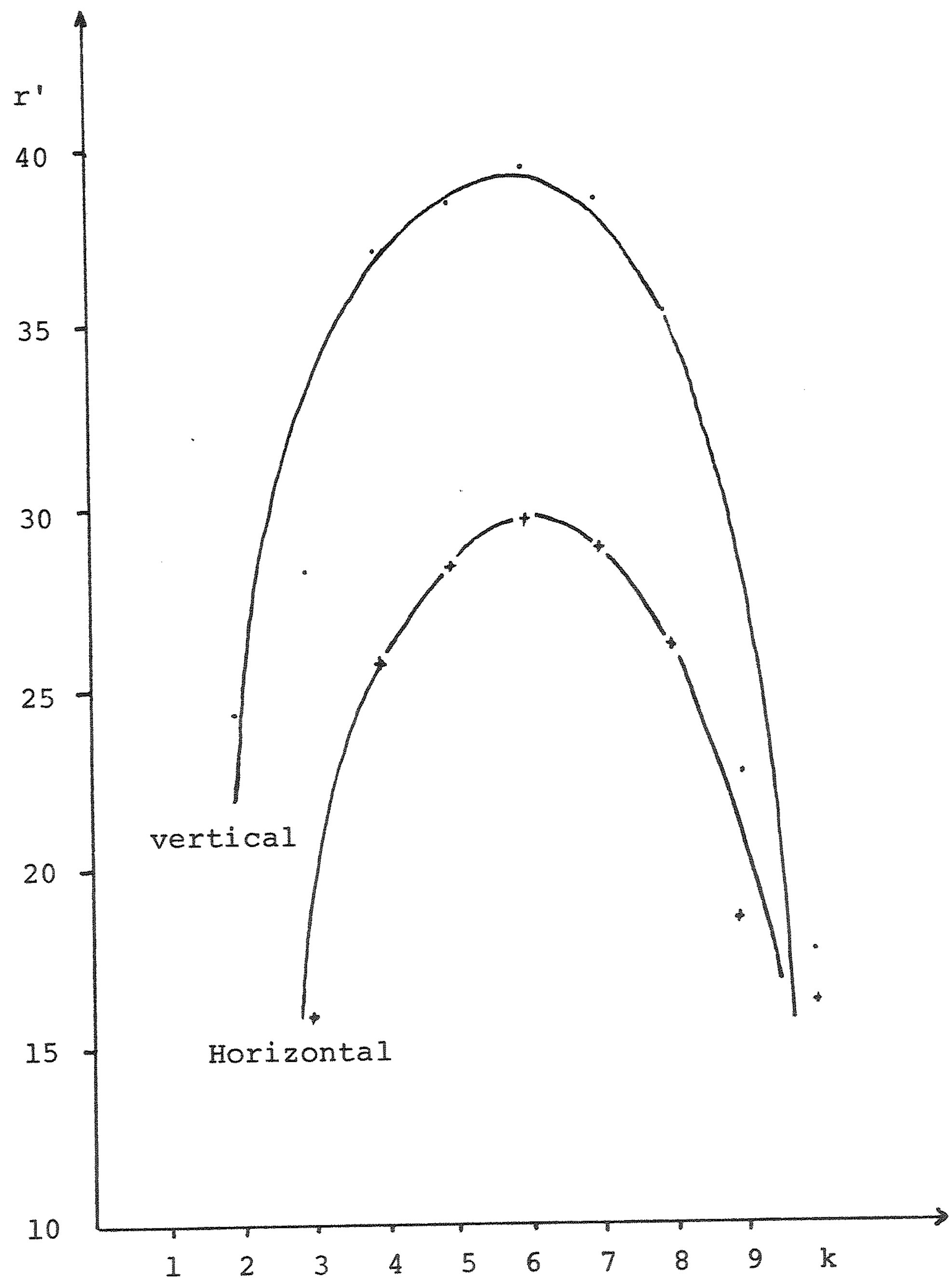


Figure 3

r' versus k in the horizontal and vertical directions for the checkerboard texture of figure 2.

k	Threshold horizontal	Threshold vertical
2	2.55	2.50
3	7.46	2.39
4	10.36	5.36
5	11.96	6.86
6	13.36	8.53
7	14.86	10.76
8	15.12	11.08
9	15.06	10.01
10	14.56	9.16

Table 1

Horizontal and vertical thresholds for the checkerboard texture of figure 2.

error rates corresponding to these values are very similar.

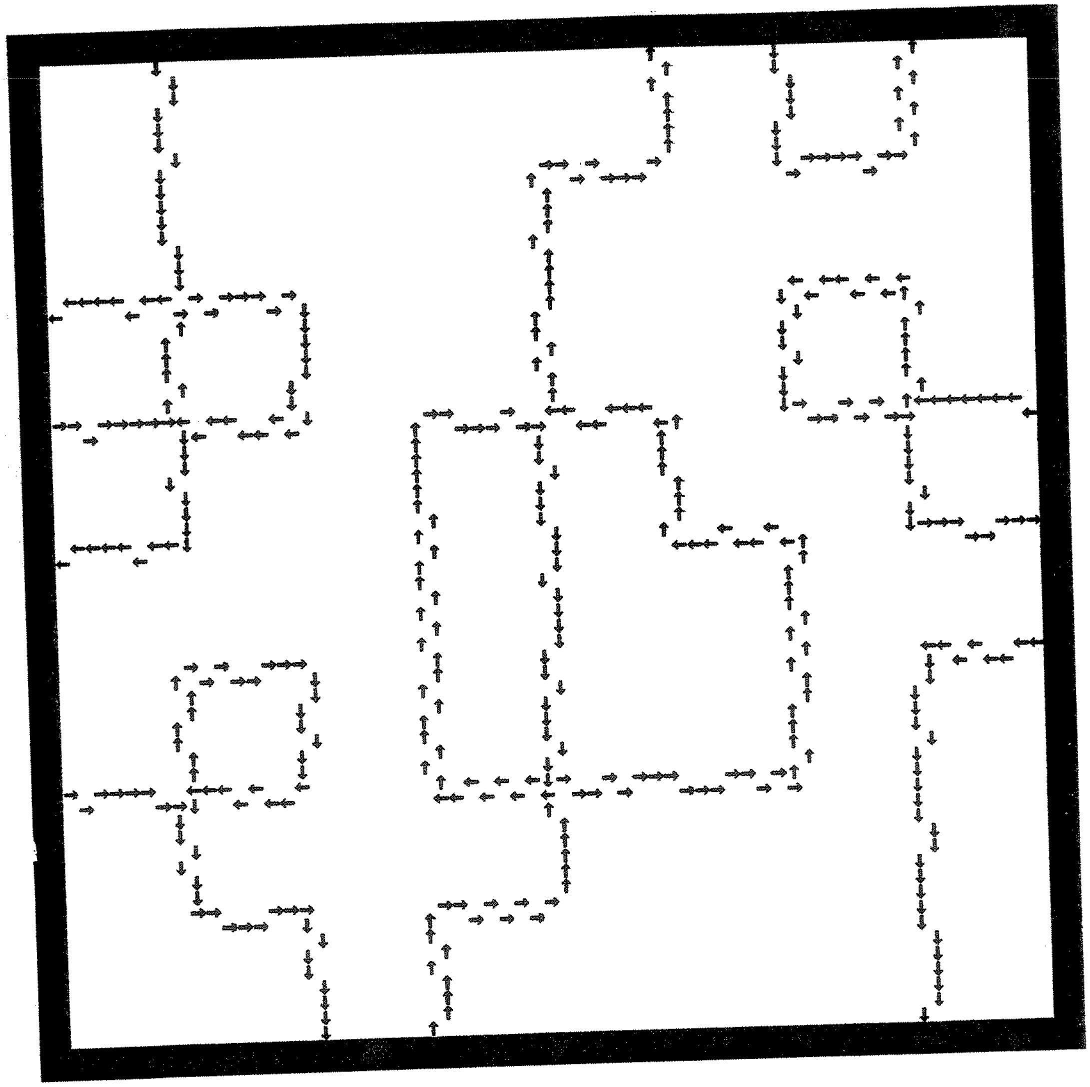
Figure 4a is the edge map that resulted from the application of the automatic edge detection procedure to the entire image. Figures 4b through 4e show the segmentations obtained when the procedure is applied to the four 32x32 quadrants of the checkerboard individually. It can be seen that the performance of the procedure is still very good on these small windows. Table 2 lists the thresholds and the optimal values of k corresponding to these windows.

The images in figures 5a through 5e are textures of orchards, gratings, concrete, pebbles and bricks respectively. It can be noticed that the texture elements are not well defined in some of the textures and that there are several levels at which discontinuities in image gray scale occur, making the process of detecting edges more difficult. The results of segmenting these texture in the horizontal and vertical directions is illustrated in figures 6a through 6e. Table 3 shows how, for k = 10, the number of the above threshold points and the number of edges in the gratings texture vary as a function of iteration.

Figure 7 is a plot of r' as a function of k for the gratings of figure 5b and figure 8 is a plot of er(k) for the same texture. It can be noticed that er(k) does not allow a confident choice of a best k.

To emphasise the importance of k in the performance of the edge detection process it has been applied to the same natural textures with the common value k = 6 and the same thresholds as before. The results are illustrated in figure

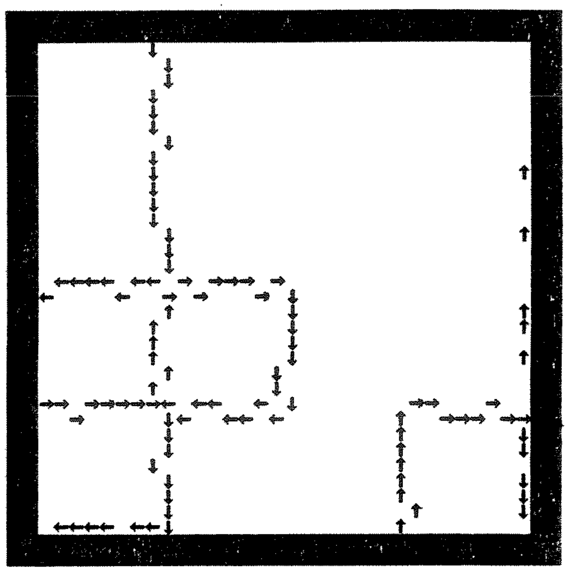
9a through 9e. Notice that the edge map for the pebble texture is acceptable while the edge maps for the other textures are totally inaccurate with a high rate of error.



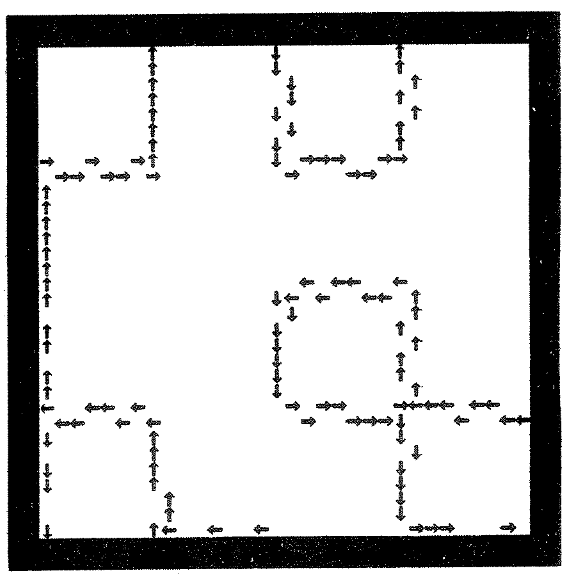
(a)

Figure 4

(a) edge map for the texture of figure 2.
(b)(c)(d)(e) edge maps for the 4 32 x 32 windows of the texture of figure 2.

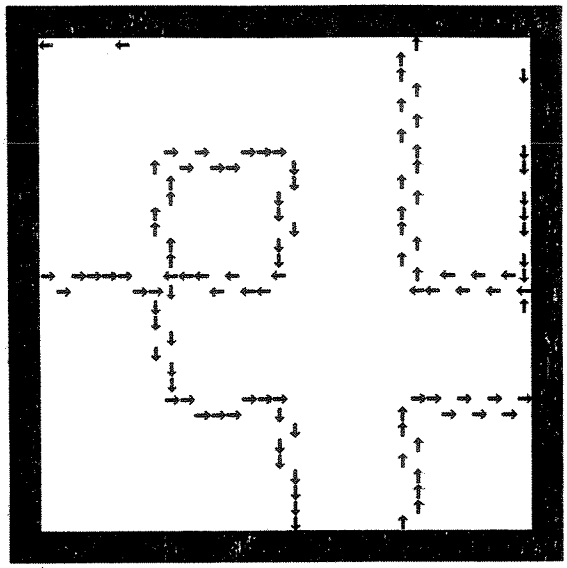


(b)

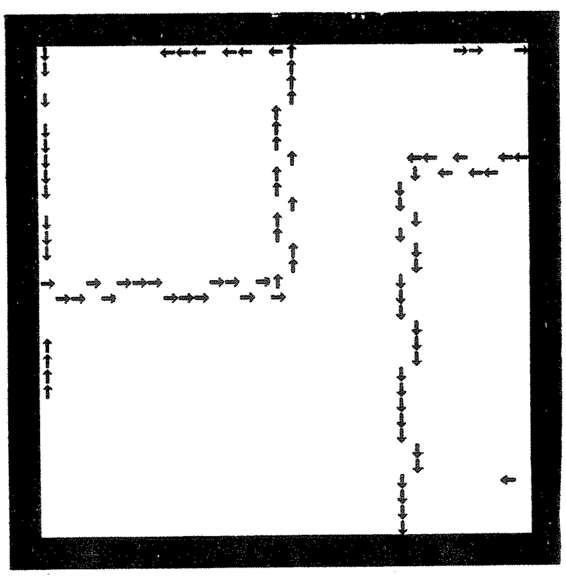


(c)

Figure 4 (cont'd)



(d)



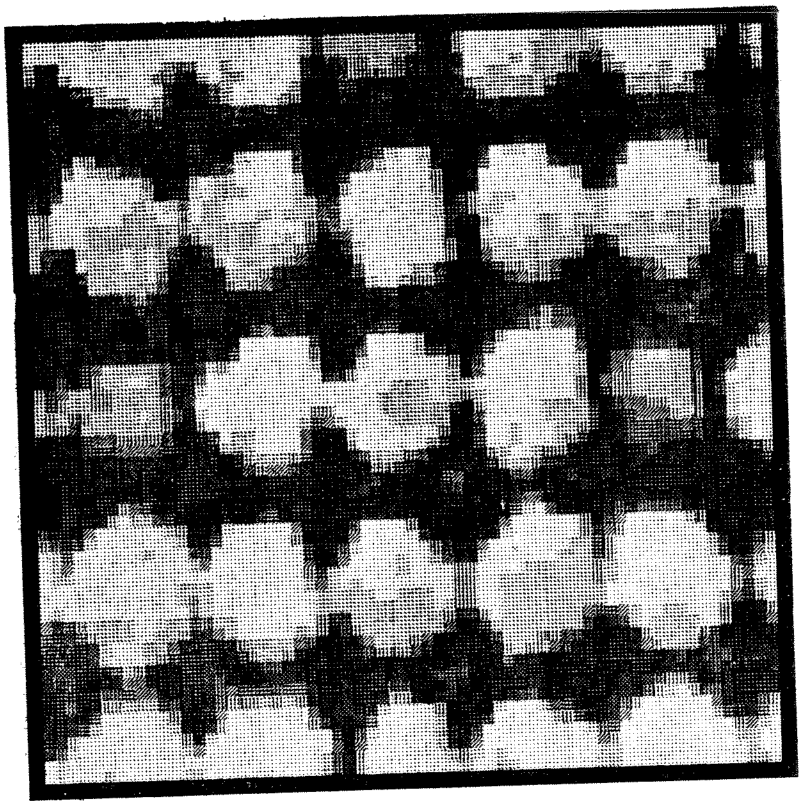
(e)

Figure 4 (cont'd)

window	Optimal k horizontal	Optimal k vertical	Threshold horizontal	Threshold vertical
(1,1)	4	6	8.1	8.8
(1,2)	5	7	8.2	13.0
(2,1)	6	6	9.9	11.0
(2,2)	7	6	14.5	5.8

Table 2

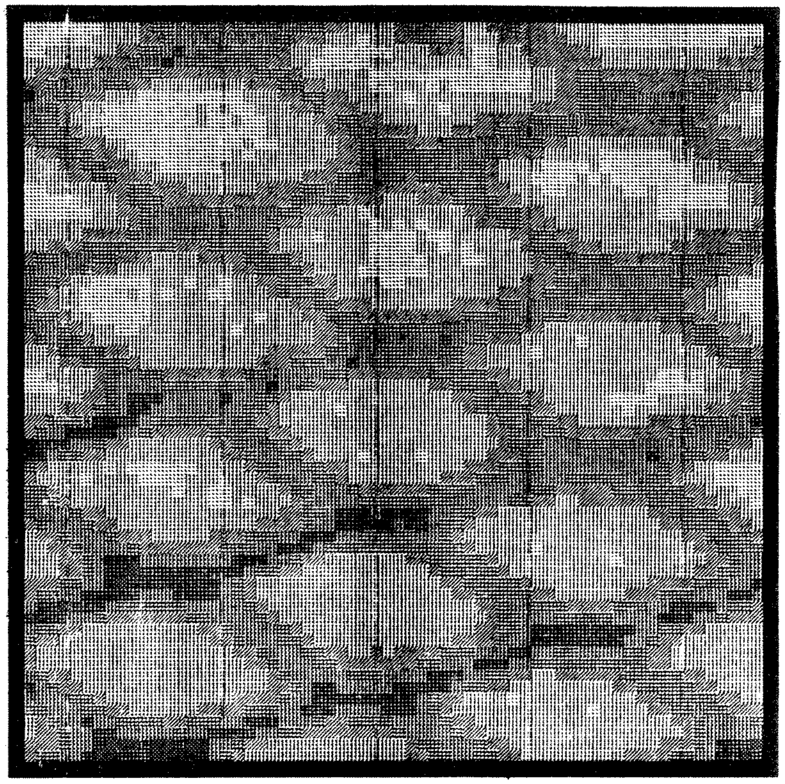
Optimum values of k and corresponding thresholds for the 4 windows of figures 4b - 4e.



(a)

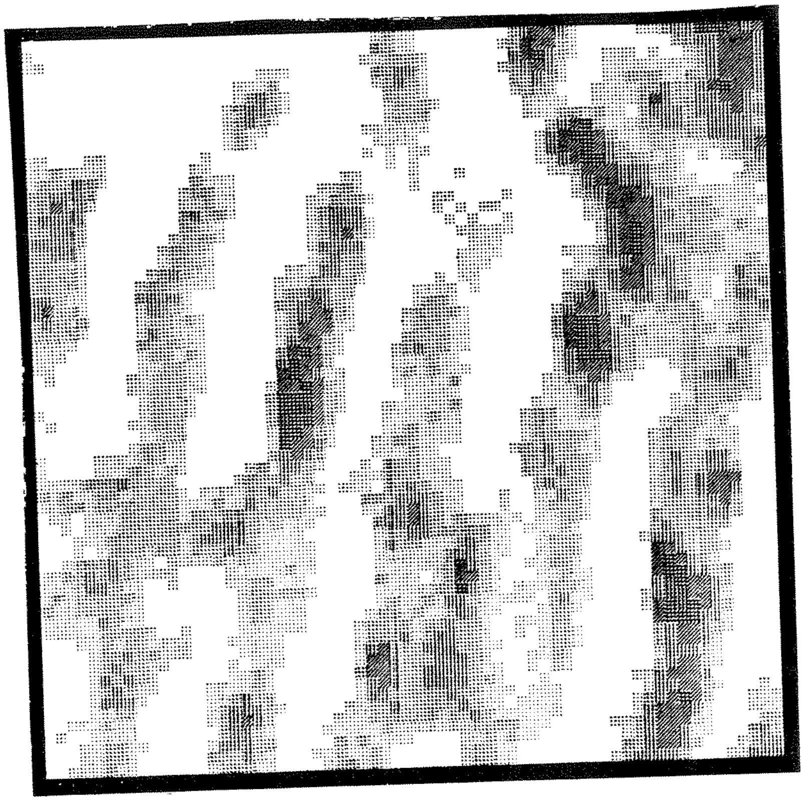
Figure 5

(a) Orchards
(b) Gratings
(c) Concrete
(d) Pebbles
(e) Bricks



(b)

Figure 5 (cont'd)

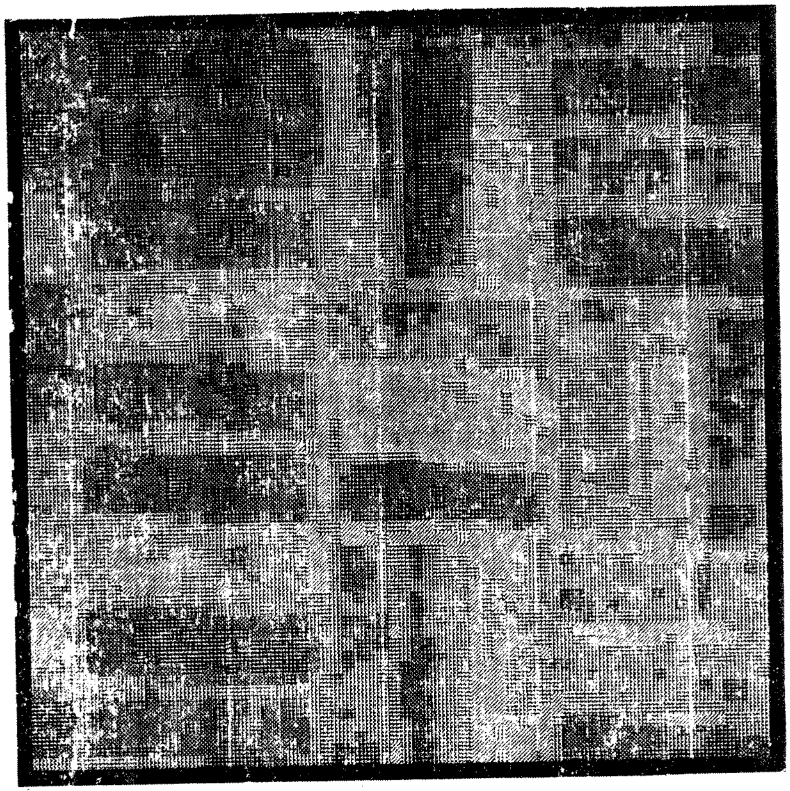


(c)

Figure 5 (cont'd)

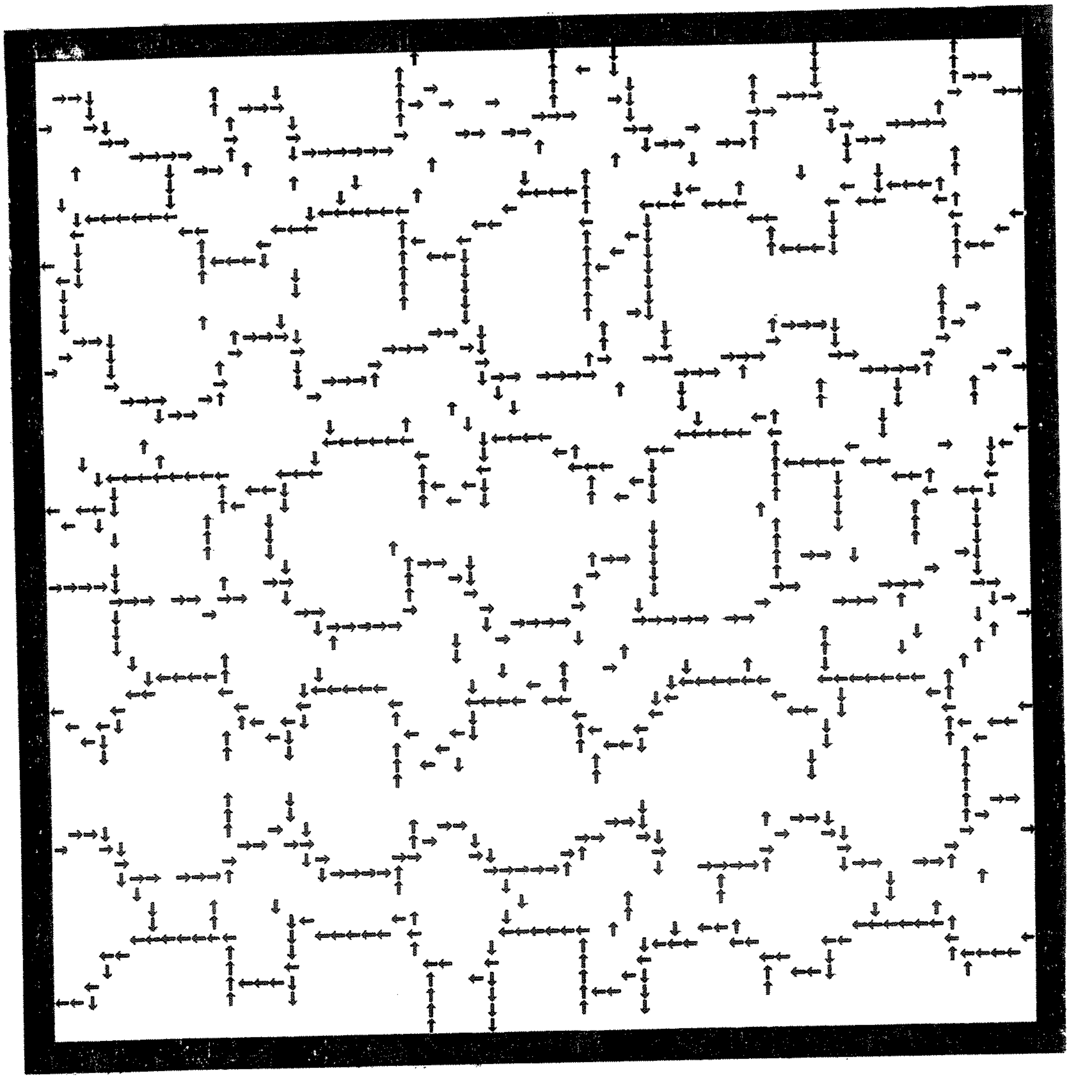
(d)

Figure 5 (cont'd)



(e)

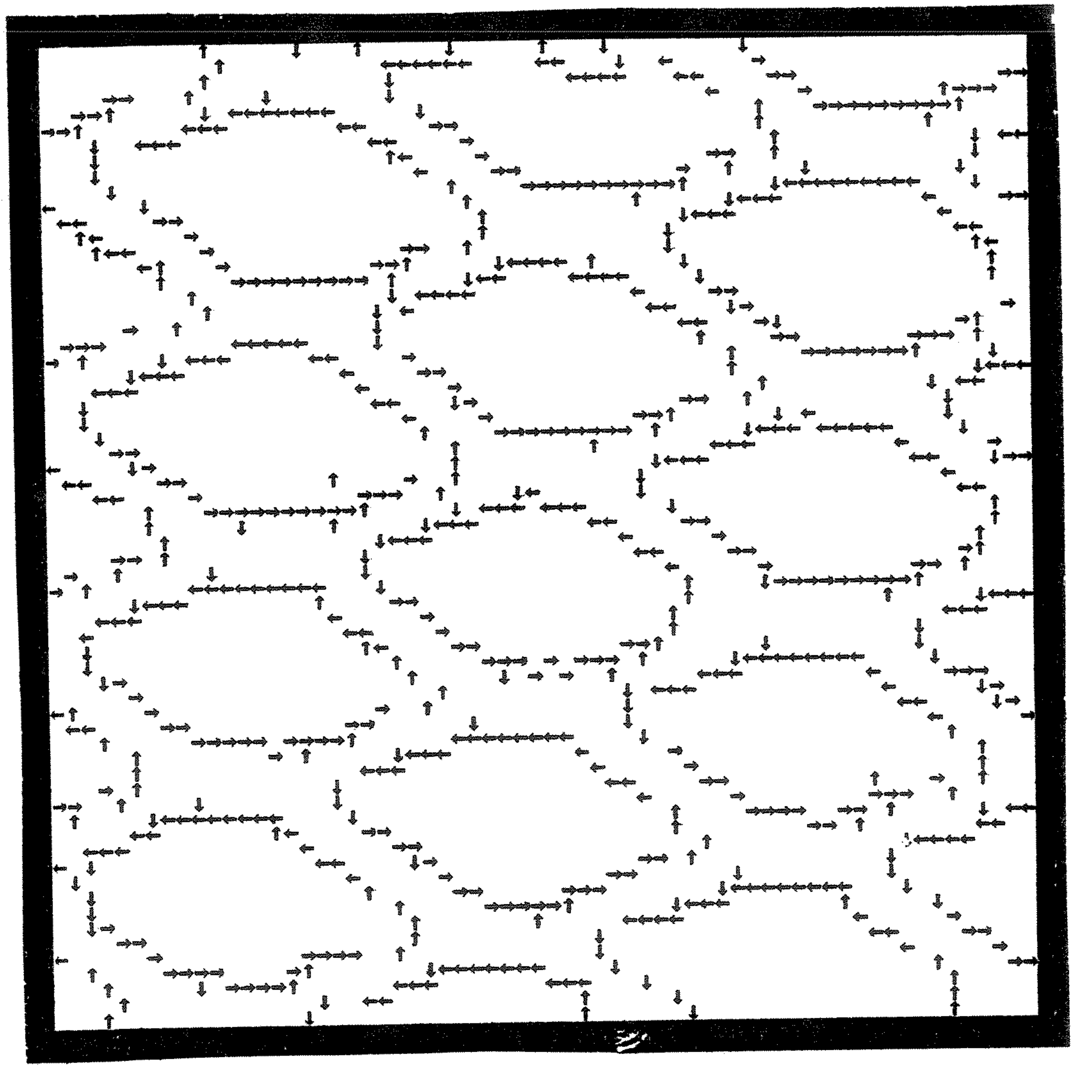
Figure 5 (cont'd)



(a)

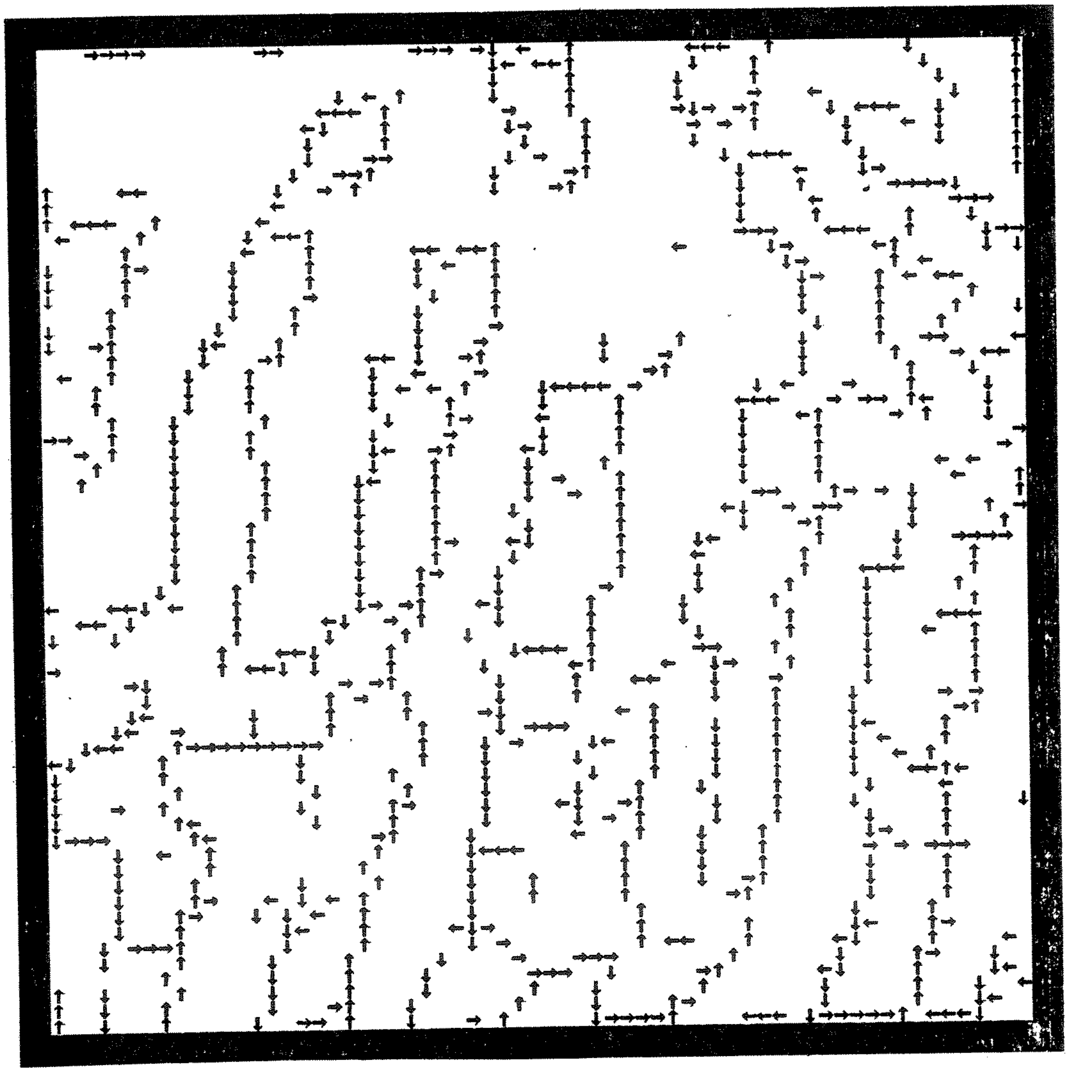
Figure 6

Edge map for (a) Orchards (b) Gratings
(c) Concrete (d) Pebbles (e) Bricks



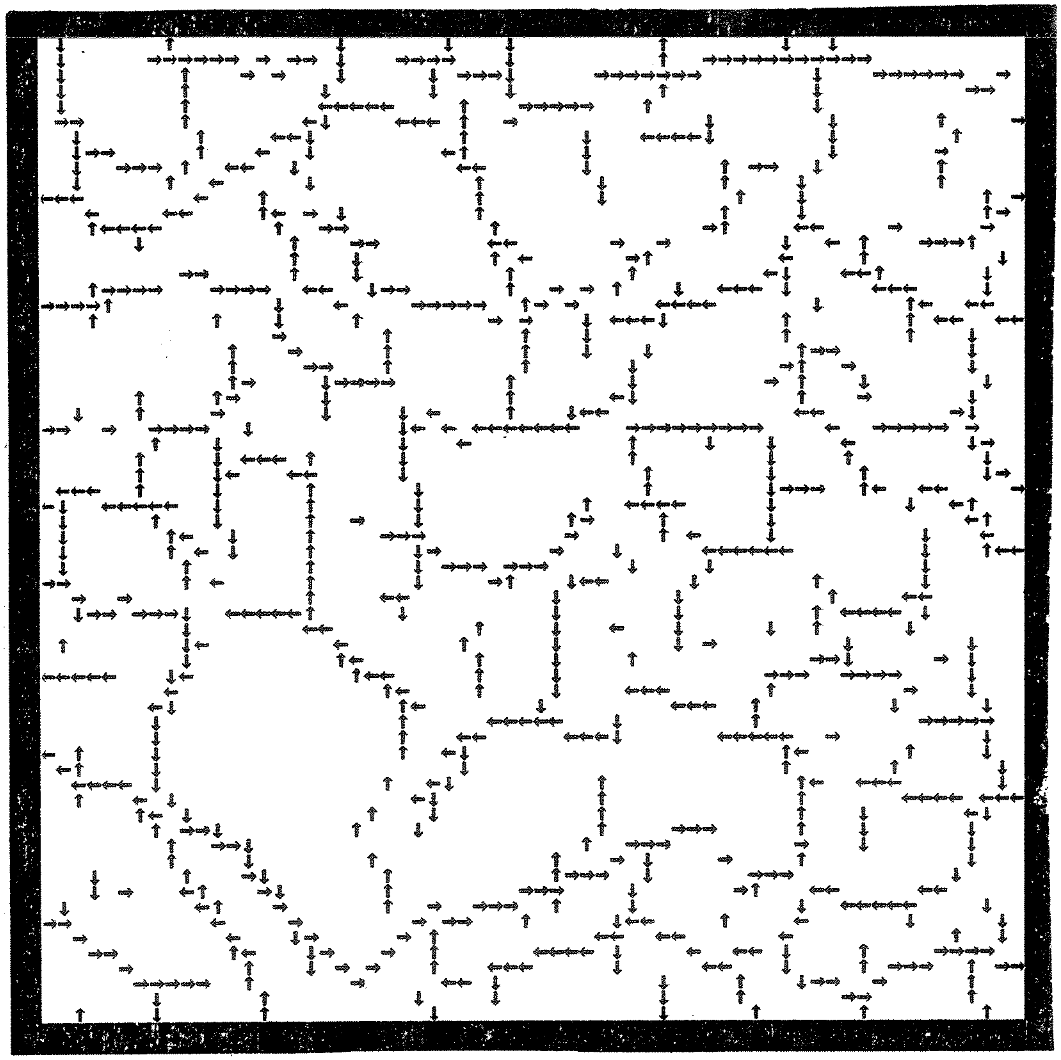
(b)

Figure 6 (cont'd)



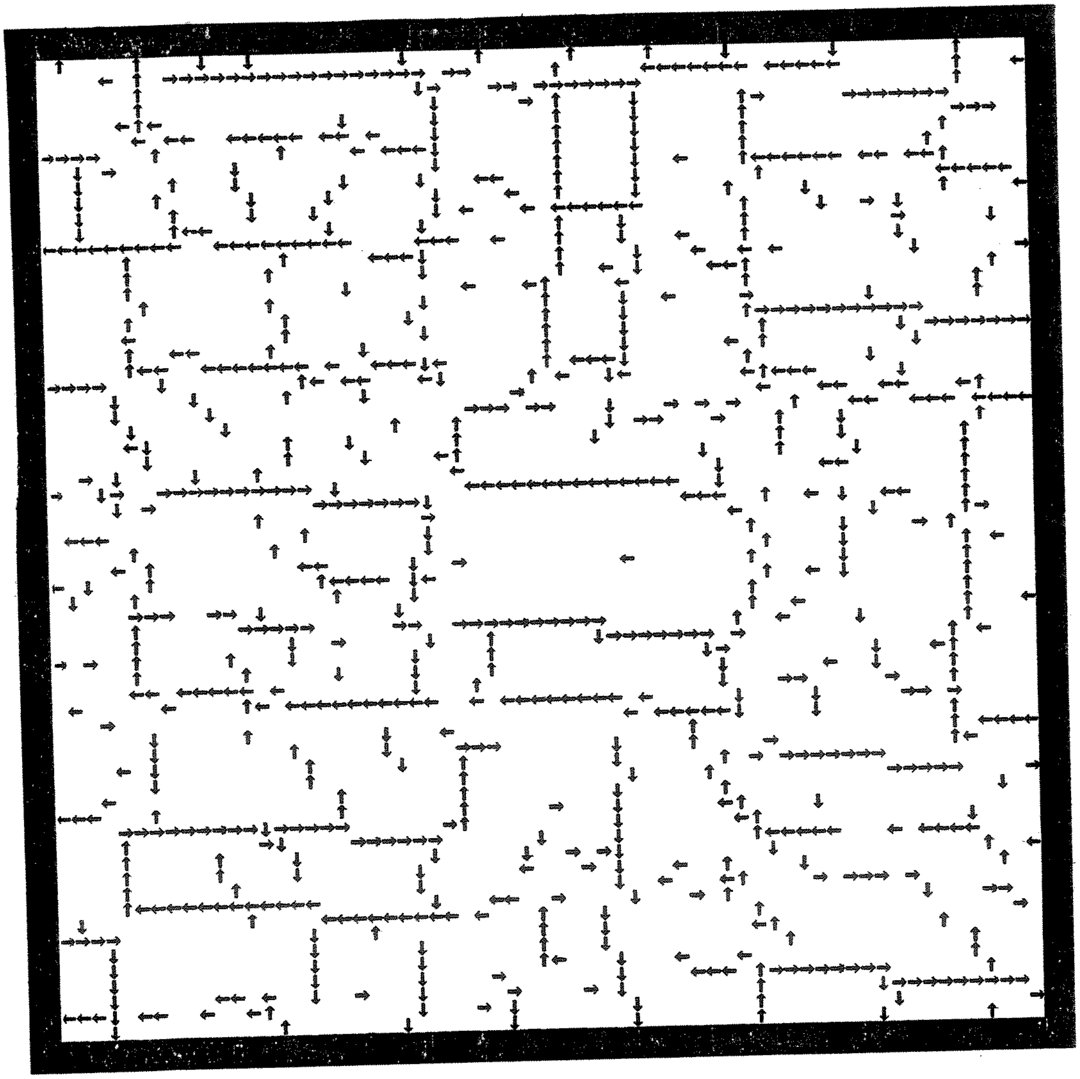
(c)

Figure 6 (cont'd)



(d)

Figure 6 (cont'd)



(e)

Figure 6 (cont'd)

Iteration	Above Threshold	Edges
1	399	77
2	1201	173
3	956	142

(a)

Iteration	Above Threshold	Edges
1	243	86
2	1226	322
3	1678	377
4	2310	430

(b)

Table 3

Number of points avove threshold and number of edges in the texture of figure 5b after each iteration of the system of figure 1 and for k=10. (a) Horizontal direction (b) vertical direction

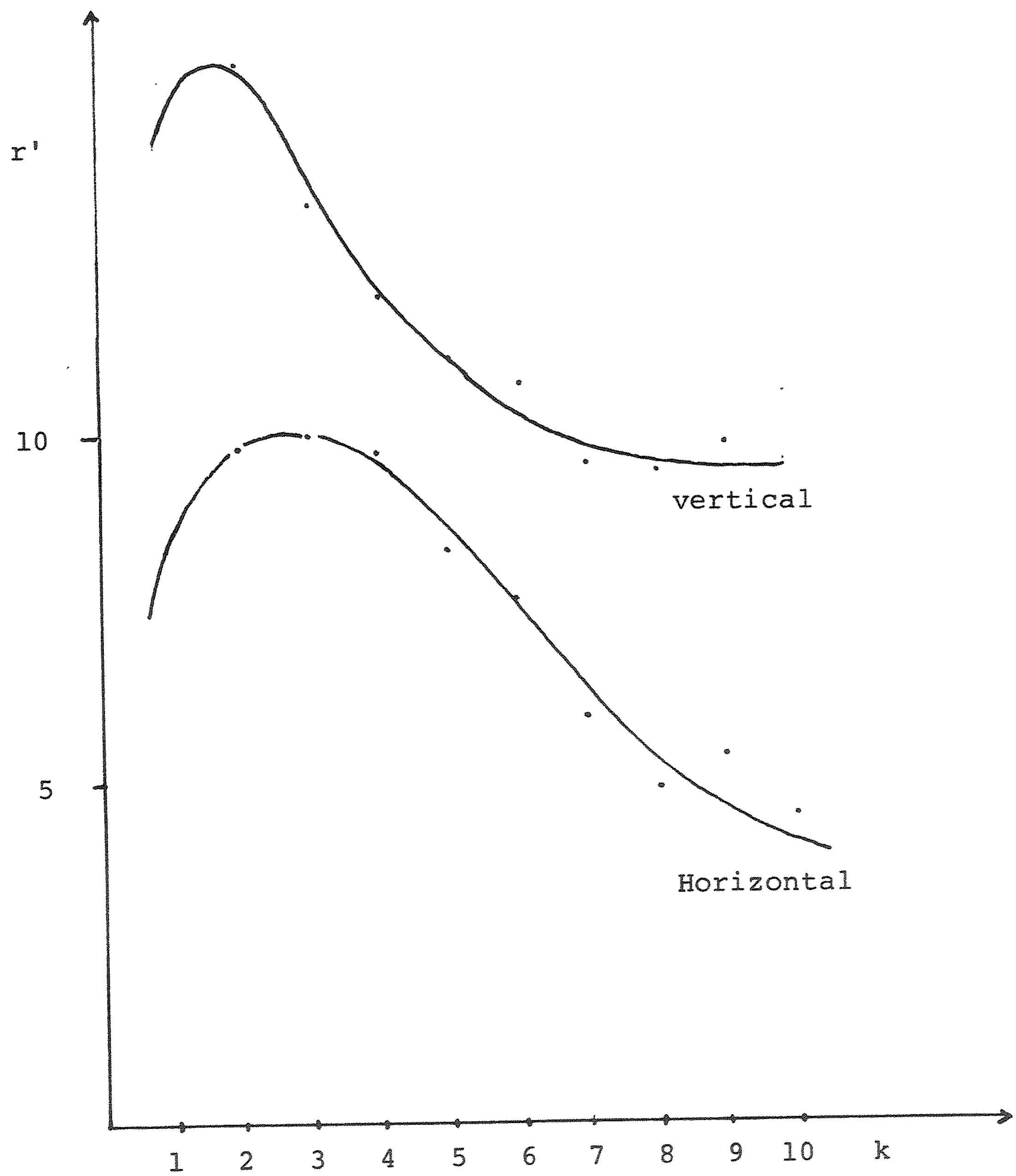


Figure 7

r' versus k in the horizontal and vertical directions for the gratings of figure 5b.

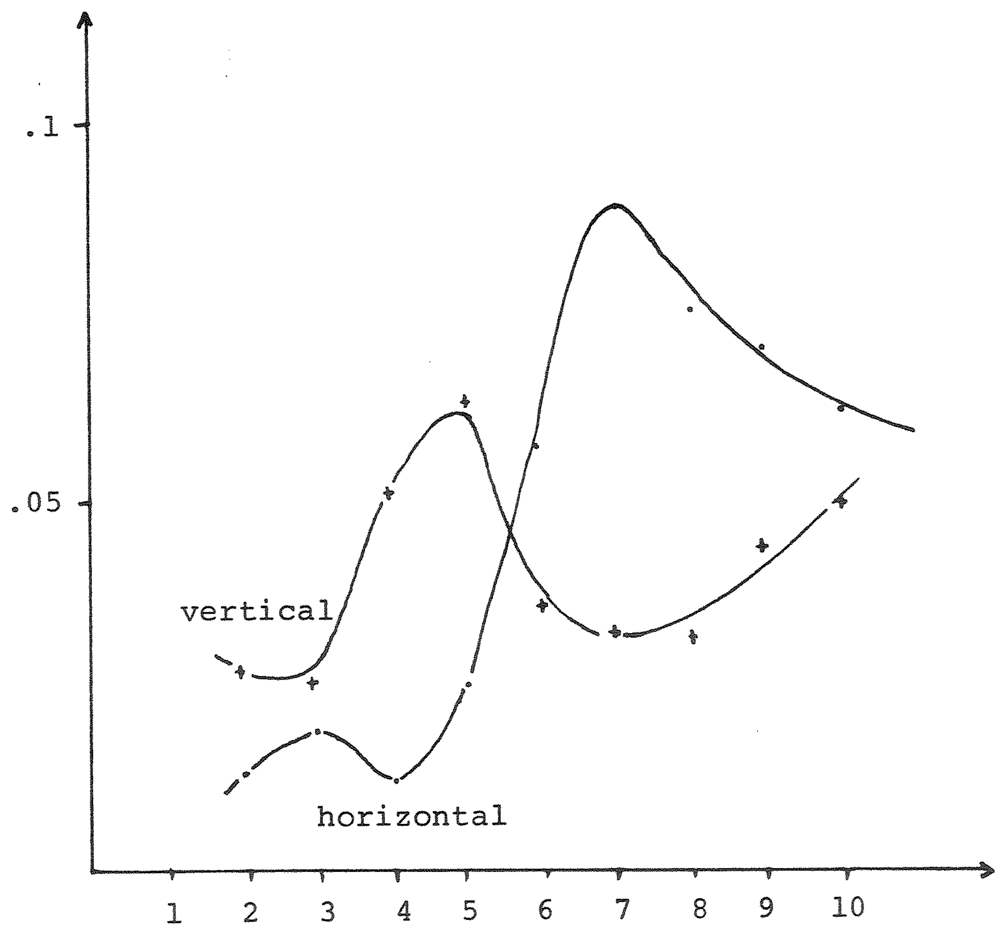
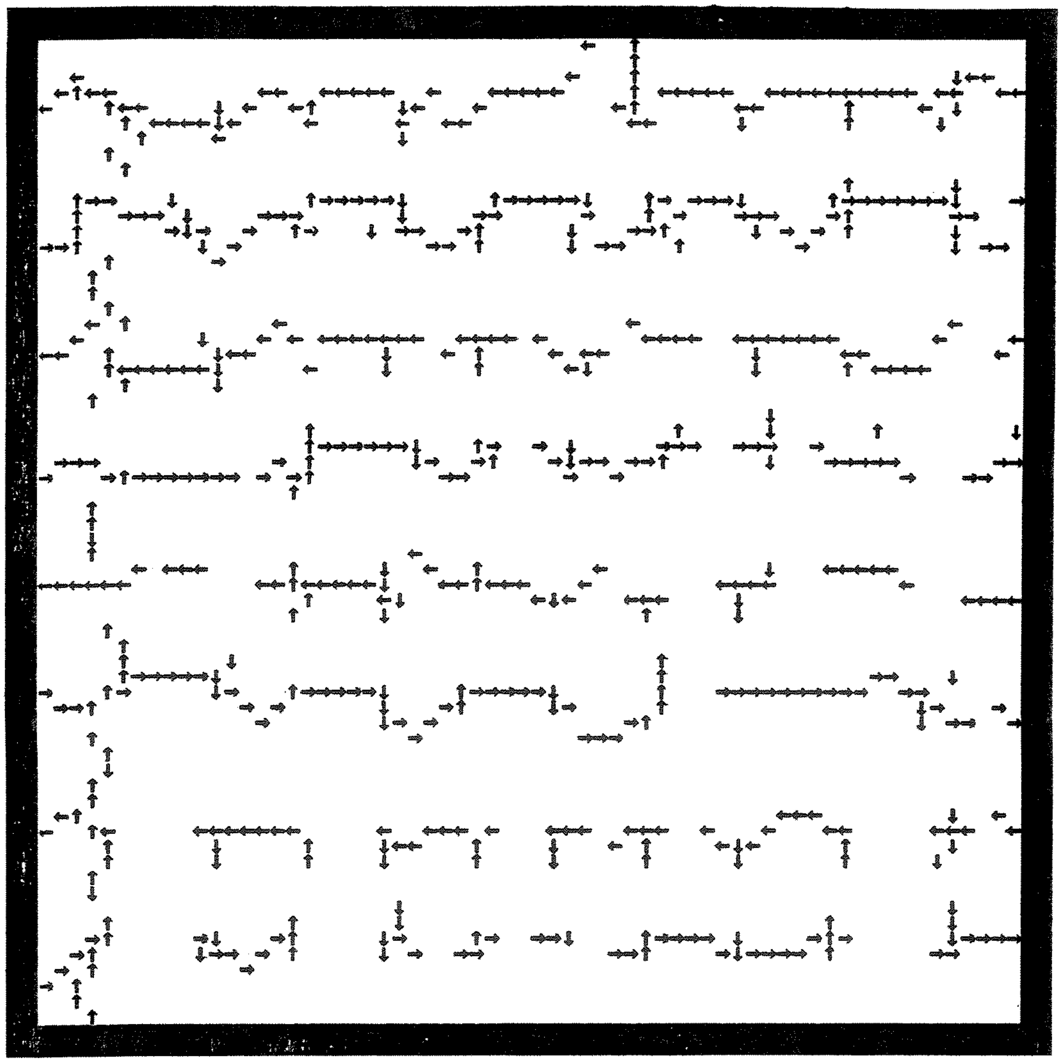


Figure 8

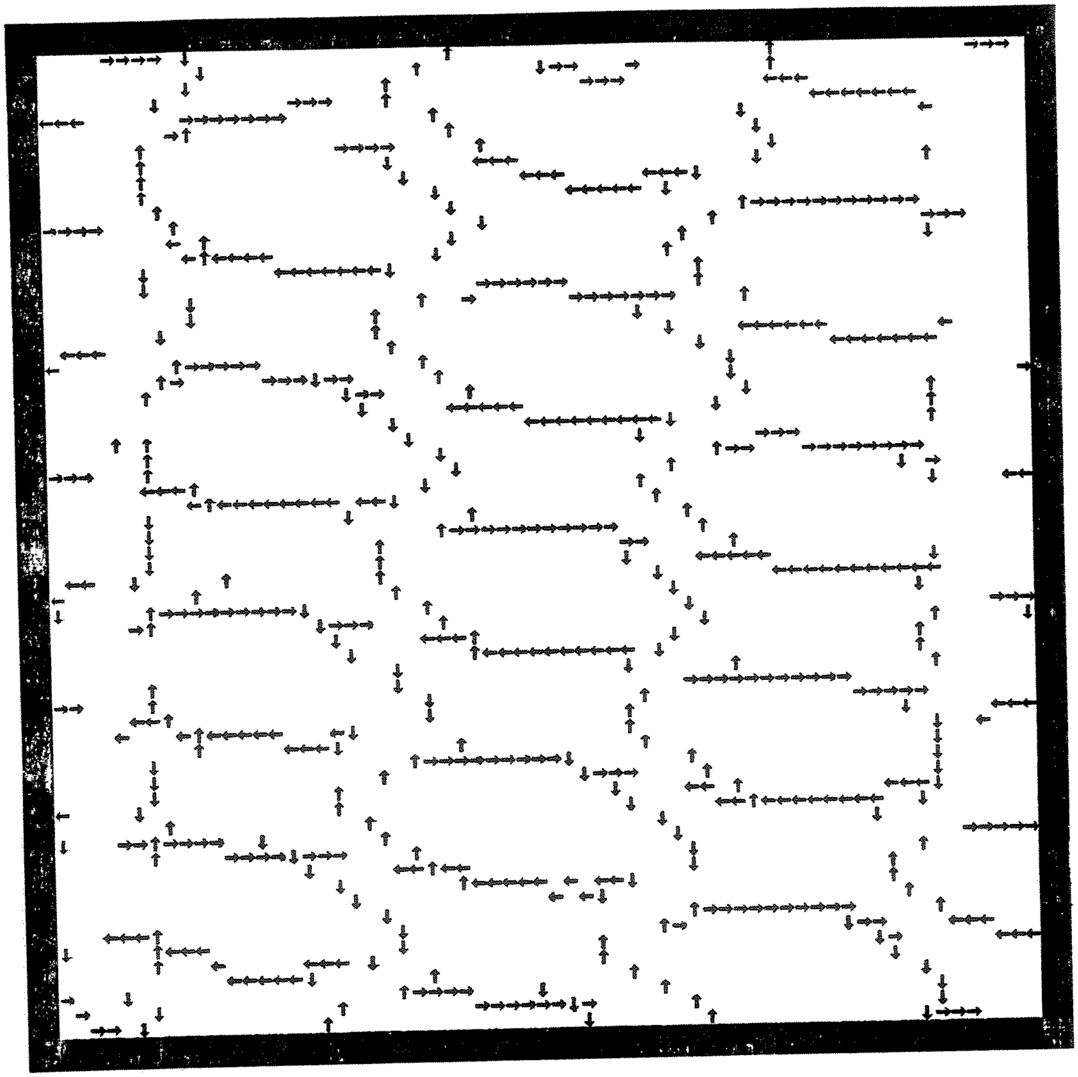
er(k) versus k in the horizontal and vertical directions for the gratings of figure 5b.



(a)

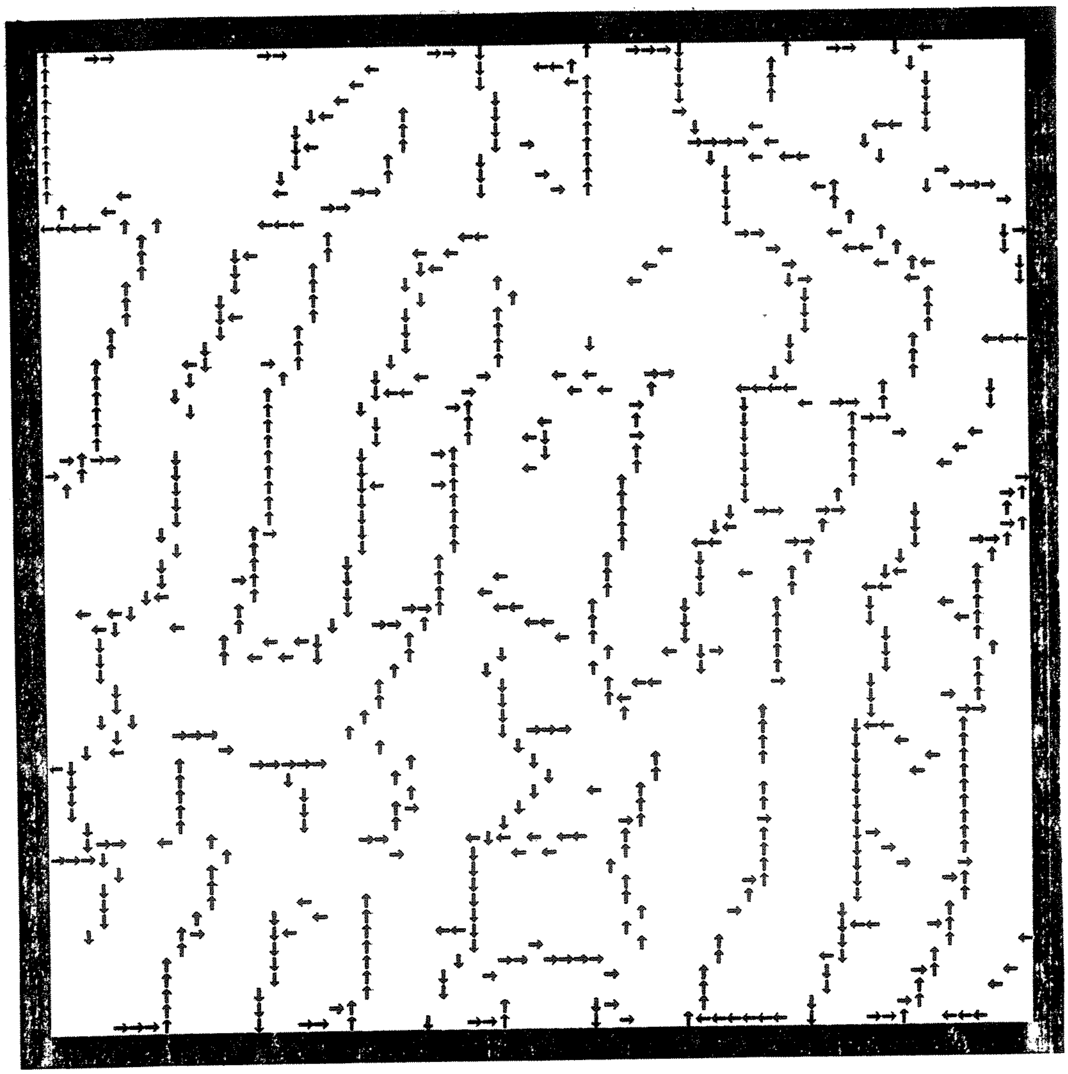
Figure 9

Edge maps with k = 6 for (a) Orchards (b) Gratings (c) Concrete (d) Pebbles (e) Bricks



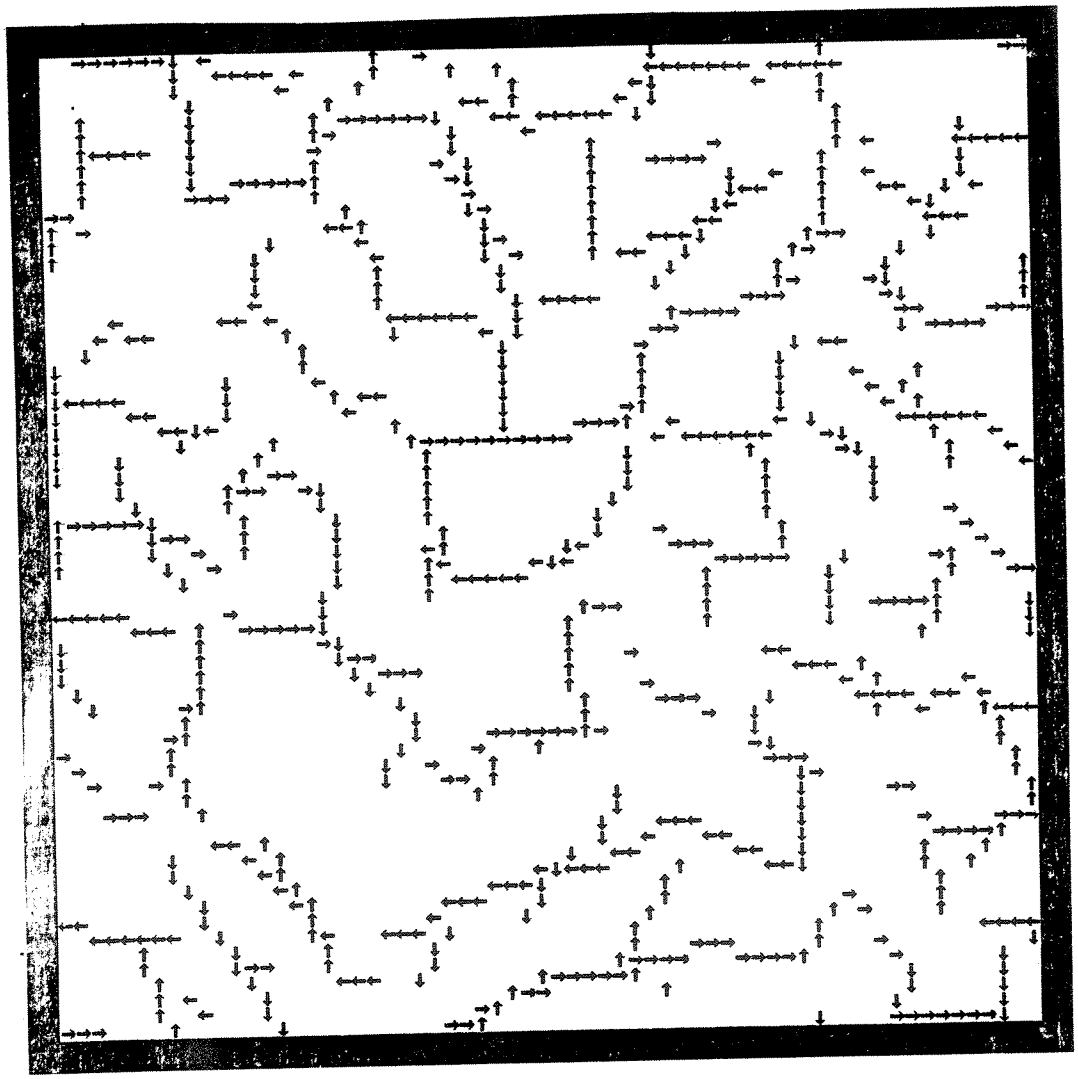
(b)

Figure 9 (cont'd)



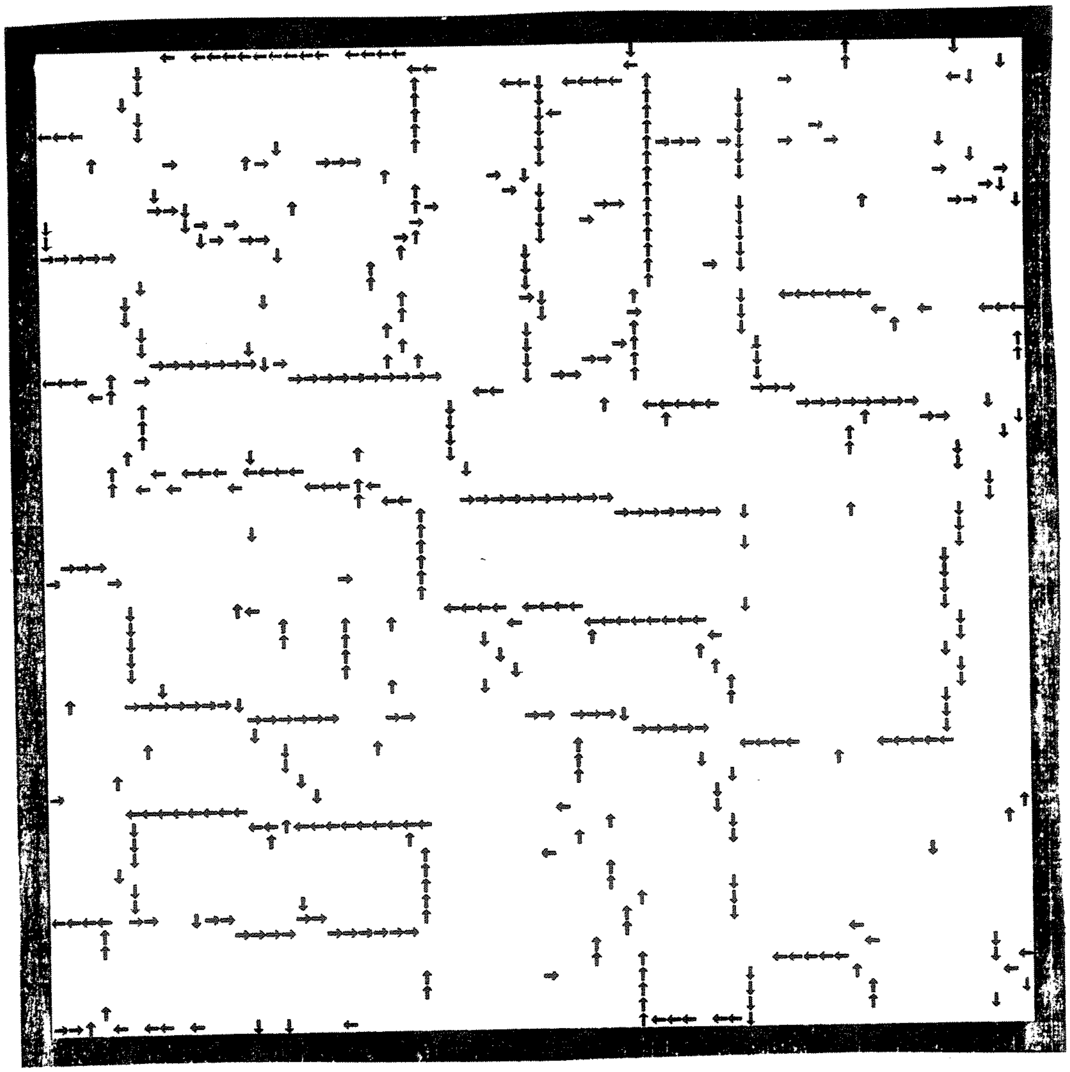
(c)

Figure 9 (cont'd)



(d)

Figure 9 (cont'd)



(e)

Figure 9 (cont'd)

6. Conclusion

This paper was concerned with the problem of applying the results of the analysis in the [13,14] to detecting edges in natural textures. An automatic edge detection procedure was used that was based on applying an edge sensitive operator to the texture and then thresholding the results of the edge operator and finally computing peaks from the above threshold points. The notion of segmentation consistency allowed automatic computation of the best possible segmentation for a given value of the neighborhood size. A measure of goodness of a segmentation was proposed to select the best overall neighborhood size from which to compute the final texture segmentation. Examples of choosing optimal edge detectors for natural textures given in this paper stress the importance of obtaining a good edge map for image analysis systems that use edges as a basic image description component.

References

1. L. Davis, S. Johns and J.K. Aggarwal, "Texture analysis using generalized cooccurrence matrices," IEEE Trans. Pattern Anal. Machine Intelligence PAMI 1, 1979, pp 251-258

2. L. Davis, M. Clearman and J.K. Aggarwal, "A comparative texture classification study based on generalized cooccurrence matrices," Proc. IEEE cof. on Decision and Control, Miami Dec 12-14, 1979.

3. R. Haralick, B. Shanmugan and I. Dinstein, "Texture features classification," IEEE trans. Systems, Man and Cybernetics, 3, pp 610-622.

4. R. Nevatia, K. Price and F. Vilnrotter,"Describing naturan textures," Proceedings of the ARPA Image Understanding Workshop, Palo Alto, Ca. April 1979, pp 55-60

5. D. Marr, "Early processing of visual information," Phil. Trans. Royal Society, B, 275, pp 483-524, 1976.

6. N. Nahi and S. Lopez-Mora, "Estimation detection of object boundaries in noisy images," IEEE trans. Automatic Contr. AC-23, 1978, pp 834-846.

7. J. Modestino and R. Fries, "Edge detection in noisy images using recursive digital filtering," Computer Graphics and Image Processing 6, 1977, pp 409-433.

8. K. Shanmugan, F. Dickey and R. Dubes, "An optimal frequency domain filter for edge detection in digital pictures," IEEE Trans. Pattern Anal. Machine Intelligence, PAMI-1, 1979, pp 37-49.

9. D.Cooper and H. Elliot, "A maximum likelihood framewor for boundary estimation in noisy images," Proc. IEEE Computer Society Conf. on Pattern Recognition and Image Processing, Chicago, may 31-june 2, 1978 pp 25-31.

10. D. Marr and E. Hildreth, "Theory of edge detection," Proc Royal Society Lond. B, 1979.

11. B. Schachter, A. Rosenfeld and L. Davis, "Random mosaic models for texture," IEEE Trans. Systems, Man and Cybernetics, 9, pp 694-702.

12. N. Ahuja, "Mosaic models for image synthesis and analysis," Ph.D Diss. University of Maryland, Computer Science Dept, 1979.

13. L. Davis and A. Mitiche, "Edge detection in textures," Computer Graphics and Image Processing 12, 1980, pp 25-39.

14. L. Davis and A. Mitiche, "Edge detection in textures - Maxima selection," TR 133, Computer Science Dept., University of Texas at Austin.