Computing Regularity Descriptons of Cellular Textures

Larry S. Davis

Department of Computer Sciences

Austin, Texas 78712

February 1979 TR-89

This research was supported in part by the National Science Foundation under grant ENG 74-04986.

1.0 Introduction

This paper is concerned with computing the spatial structure of textures composed of texture elements arranged on a homogeneous background. By computing the "spatial structure" we mean constructing a qualitative description of the distribution of texture elements, such as might be made by a human viewing and describing the texture.

Our principle motive for pursuing this question is to increase our understanding of how to compute what are called strong texture measures. The notions of strong and weak texture measures were introduced by Haralick[1]. By a weak texture measure is meant one which is based only on the description of the properties of individual texture elements which comprise the texture. For example, the average size of the texture elements is a weak texture measure.

Since the same set of texture elements can be spatially arranged in different ways to produce perceptually different textures, it is clear that weak texture measures are not sufficient to account for all texture discriminations. Therefore, one must also consider the class of strong texture measures, which additionally take into acount the spatial relationships between the texture elements.

Strong texture measures have been investigated by Maleson et al[2] who compute spatial relations between segmented texture elements (connected components of constant grey level); by Davis

et al[3] who compute spatial relations between local image features (edges, lines, etc.); and by Ehrich and Foith[4], who forms a relational tree between local maxima and minima of grey level. Haralick[1] contains other examples.

In this paper, we will be concerned with textures composed of identical elements - i.e., we will ignore the added information that is potentially available from the properties of the individual texture elements such as their size, shape, color, etc. We do not mean to imply that such properties play a secondary role in texture analysis. There is psychological evidence (see, e.g., Pickett[5]) that size, shape, etc. are more important for human texture discrimination than variations in element density or element arrangement. We choose to ignore them in order to focus attention on element arrangement.

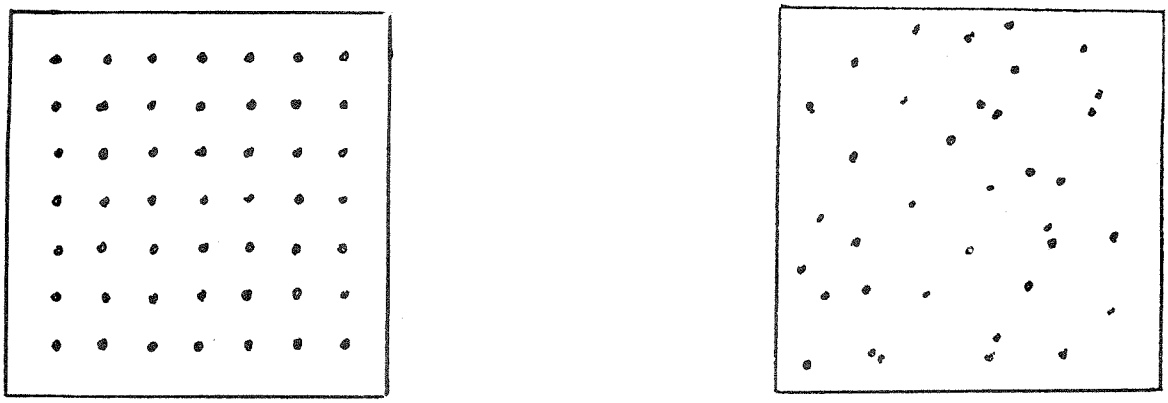
Therefore, a texture for out purposes is simply a distribution of points in the plane. If this seems unreasonably restrictive, then consider the different "textures" in Figure 1, which are composed of just points in the plane. There are clear differences between the patterns. It is these differences that we would like to be able to compute and describe. Note that a variety of processes can be defined to place dots in a field to synthesize a texture (see, e.g., Rosenfeld and Lipkin[6]). We will restrict our attention to just two- random and regular- which we will define more formally in Section 2. (We specifically exclude from consideration situations where the dots themselves group to form texture elements.)

We will also describe an application of these ideas to a real image analysis problem involving the discrimination of healthy and infested trees in aerial photographs of citrus orchards. Figure 2 contains an example of one such image. The technical aspects of this problem as a pattern recognition problem need not concern us here; they are detailed in Williams[7]. Instead, we will discuss the relevance of texture analysis to the computation of tree locations in such images. As can be seen in Figure 2, the trees roughly define a rectangular grid. There are, of course, minor deviations from perfect rectangularity (i.e., the tree "centers" do not actually lie at the grid points of what we perceive to be the underlying grid); furthermore, there are missing trees (trees have been cut down and certain infestations cause trees to appear deformed in these images). If we could somehow compute the grid structure from an analysis of the image then we could:

1) predict the location of the "missing" trees so that these areas could be subsequently investigated in order to determine if the trees are truly missing or if the trees are infested, and

2) predict the location of trees in sections of the image which we have not yet analyzed.

In the next section we will discuss methods for computing the spatial structure of dot textures, and then in Section 3 we will discuss the application of these procedures to the tree location problem.



a) regular

b) random

Figure 1. Dot patterns.

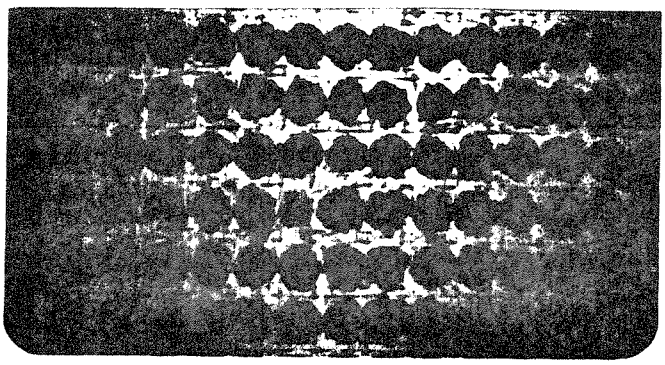


Figure 2. Original orchard image

2. Computing descriptions of spatial structure

In this section we will be concerned with computing the spatial structure of dot patterns. We are specifically concerned with describing square or rectangular grid patterns, although the procedures which we will present can be extended to other regular patterns. Rectangular grid patterns can be described by a triple, (b1,b2,dir), where b1 and b2 are the grid spacings, and dir is the orientation of the grid with respect to an arbitrary coordinate system.

In order to describe dot patterns we will first test the hypothesis that the pattern is regular (i.e., that the pattern is a square or rectangular grid) against the hypothesis that the pattern is random. The particular random process which we will consider for forming random patterns is the Poisson point process[8]. The Poisson point process produces patterns by randomly dropping points onto a bounded region of the plane. The process is completely characterized by its density, p1, which is the expected number of points dropped onto a region A of unit area. (See Kendall and Moran[9] for an accessible, introductory discussion of distributions of points in the plane).

If it is decided that the pattern is regular, then (b1,b2,dir) will be computed. In the discussion that follows we will consider only square grids. For square grids we need only compute a pair (b,dir) instead of a triple. The extension to rectangular grids is straightforward and is outlined in Section 3.

What is there that distinguishes a so-called regular pattern from a random pattern? Clearly, it has something to do with the repetitive placement according to some spatial constraint of the dots in the perceptual field. The heuristic procedures which we describe below attempt to measure that repetition. They are based on the observation that for a square grid pattern the distance of a point from its k<=4 nearest neighbors will have a very peaked distribution near b. The direction between a point and its k<=4 nearest neighbors will also have a peaked distribution, but there will be two peaks separated by pi/2 radians (assuming that we coalesce directions separated by pi).

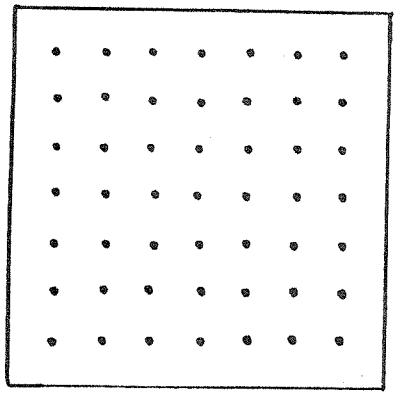
The heuristic procedure involves histogramming the distance between each point and its k nearest neighbors. This histogram is called the size histogram and is denoted by Hs. Hs is then examined for dominant peaks. The search for peaks in the size histogram represents an attempt to gauge the "granularity" of the pattern.

The dominant peak of smallest distances (i.e., the left-most peak in Hs) is then detected, and is used to create a filtered histogram of the directions between a point and a subset of its k nearest neighbors. We will refer to this subset as the ks-nearest neighbors of the point. Here, s indicates the mode of the peak. The filtered histogram is called the direction histogram and is denoted by Hd. (Histograms based on near neighbors have also been used recently by Stevens[10] to discover locally parallel structure in dot patterns - another aspect of regularity in dot

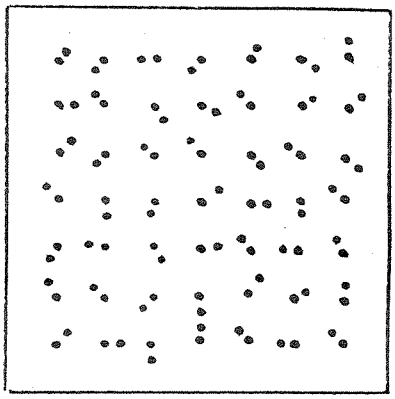
patterns.)

Note that computing Hs using only nearest neighbors and then searching Hs for the left-most peak represent two applications of the Gestalt law of proximity. The first application reflects our belief that the information attended to in perceiving such patterns is the distance between proximal points, while the second application says that this information is then subsequently interpreted based on proximity - i.e., the spatial relations between points in the peak of smallest distance determines our perception of the pattern.

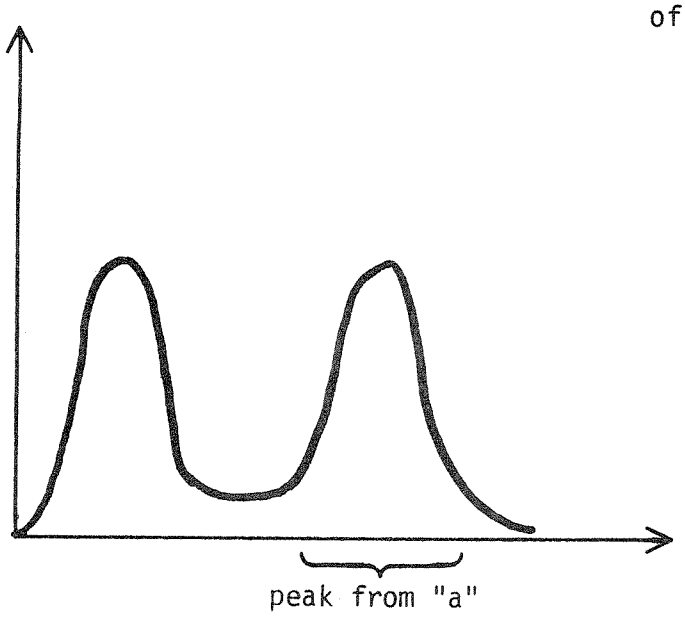
The motivation for this approach can be illustrated using Figure 3. Here, we have taken a square grid and "corrupted" it by randomly adding points to it near the vertices of the grid. The result is that the perception of the grid is interfered with by the addition of these points. Figure 3a contains the square grid pattern, and Figure 3b contains the same pattern with the addition of one extra point near each of the points of Figure 3a. The square grid pattern has been effectively masked by the addition of these points. For patterns such as the one in Figure 3b, the size histograms between points and their 3 or 4 nearest neighbors should have a strong and distinct peak produced by pairs of original grid points, in addition to the peak produced by the immediate nearest neighbors. Figure 3c displays such a histogram with the marked peak corresponding to the original grid point pairs, and the peak of small distances corresponding to the immediate neighbor pairs. If we considered the peak produced by



a) regular pattern



b) addition of "noise" points near vertices of "a".



c) H_s for (b)

Figure 3. Motivation for using left-most peak of H_s

point pairs in Figure 3a in describing the pattern of Figure 3b then we would (using the remainder of the procedure which is described below) decide that the pattern is regular, in contrast to our perception of that pattern. It is for this reason that we only consider the left-most peak in Hs.

After Hd is computed, we examine it for pairs of peaks which are separated by pi/2. All such pairs are then compared using some evaluation function, and the best is chosen as the description of the square grid. The details of the algorithm can be understood by following its application to a simple example.

Figure 4 contains a regular point texture. The figure was generated by placing a dot at all vertices of a 10x10 grid, with grid size b. Then p0=50% of the dots were randomly erased. Similar to the Poisson process, we call p1=1-p0 the density of the regular process. Finally, the positions of the remaining dots were perturbed by moving them, at random, to a position in the circle of radius .3b centered at that point. Figure 5 contains Hs for k=2 and 4.

Figure 6, then, contains Hd for the smallest distance peak in the histograms of Figure 5. The next step in the process is to detect all pairs of peaks in the direction histogram which are separated by approximately pi/2 and to assign some evaluation to each of these peak pairs. Then, the best peak pair is chosen, and its directions define the orientation of the grid.

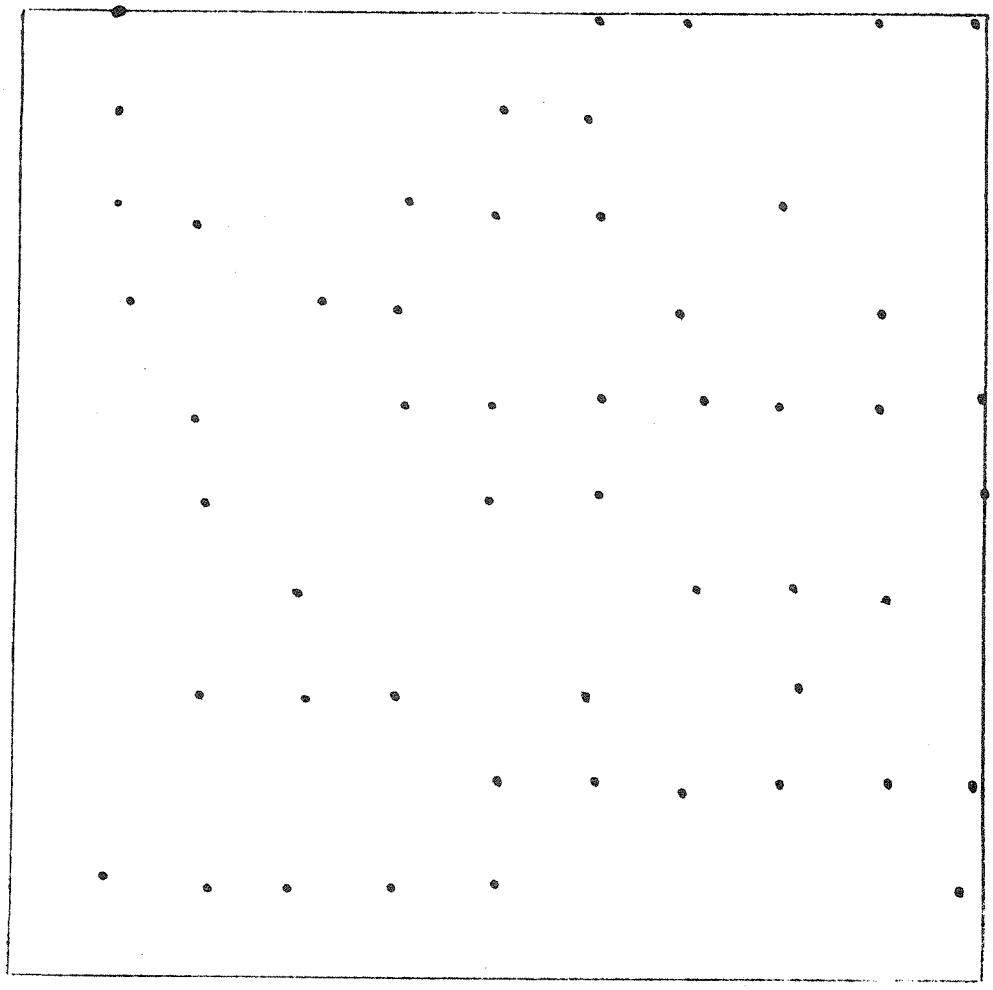
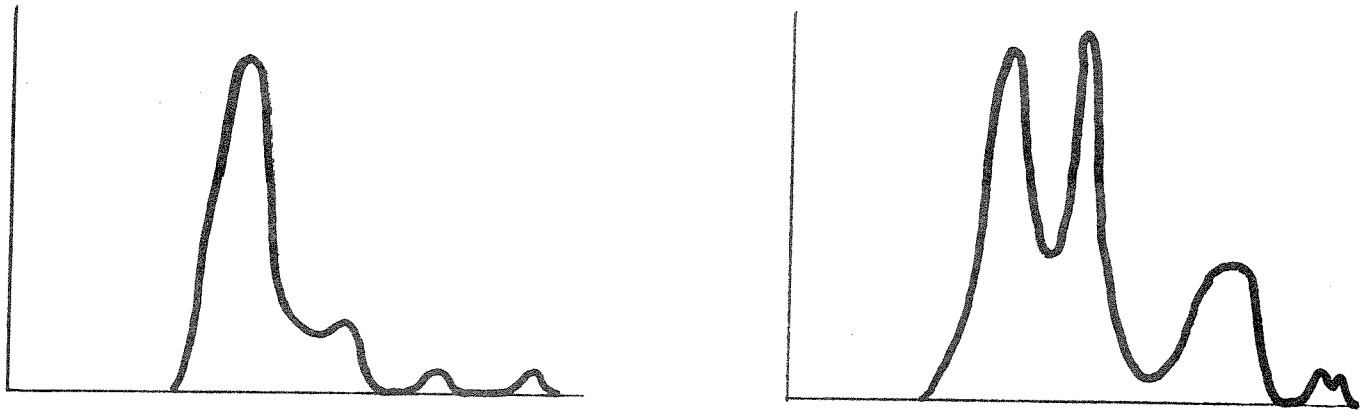
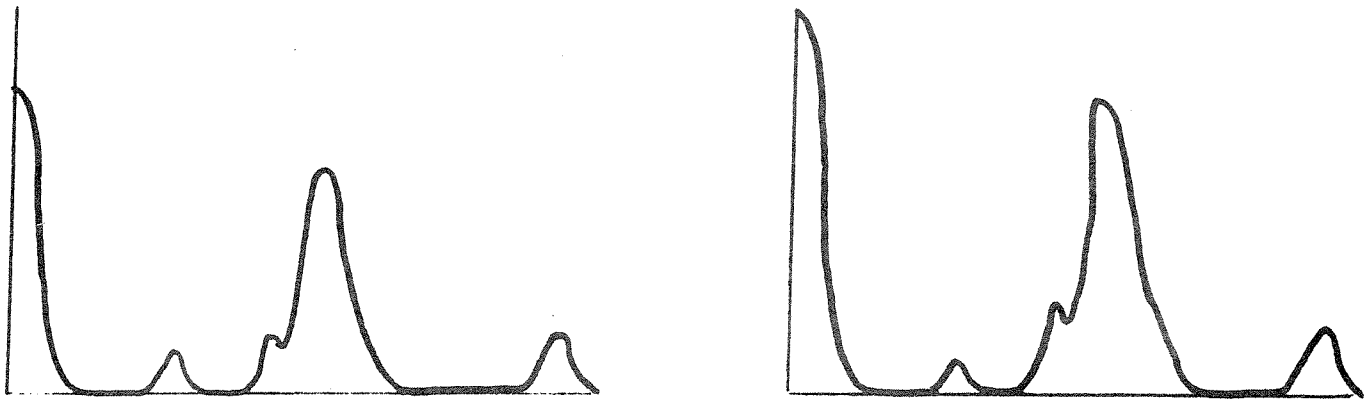


Figure 4. Dot pattern



a) K = 2 b) K = 4

Figure 5. H_s for Figure 4.



a) K = 2 b) K = 4

Figure 6. H_d for left-most peaks of Figure 5.

Once the best peak pair in the direction histogram has been determined, we can use the descriptions of those peaks to compute a measure which will reflect the degree of regularity of the original point pattern. If the degree of regularity is not high enough, then we will not describe the point pattern as being regular. This measure is defined as follows. Let (d1,d2) be the bin names of the modes of the best direction peak pair, and let N1, N2 be the respective relative frequencies of these two bins. We can now define the regularity of the pattern as

$$r = N1 + N2$$

For a perfect, regular texture we would obtain r=1. This is because each point will have k<=4 nearest neighbors at neighboring grid points along the principle directions of the pattern. Now, consider a random point pattern. If the direction histogram has been quantized to t bins, and if the positions of the points in the random pattern have been sufficiently, finely quantized, then since the distribution of directions to nearest neighbors for a random pattern is uniform[8], each bin in the direction histogram will have relative frequency 1/t. Therefore, for a random pattern, we have

$$r = N1 + N2 = 2/t$$

Finally, we take as our regularity measure for an arbitrary point pattern

$$R = r/(2/t) = tr/2$$

which represents the ratio of the regularity of the given pattern to the regularity of a random pattern.

In order to gauge the utility of this measure for discriminating between random and regular patterns, we performed the following simple experiment. Regular patterns were generated using values for density, p1, of .2, .3, .4, .5, .6, .7, and .8. Table 1 shows the value of R as a function of p1 for k=2 and 3. It should be noted that for computing the histogram of directions between a point and its k nearest neighbors we used a value of t=8. Therefore, the maximum value which R can take is 4. Random patterns were also produced using the same values for density (number of points dropped per unit area) as the value for p1. Table 2 contains the values of R as a function of that density, d, again using values for k of 2 and 3. Ideally, for these random patterns R should be 2/t=.25.

It can be seen from Tables 1 and 2 that there is a clear difference between the values of R for the regular and random pattern. We could reliably use a threshold value of 2.0 to discriminate between regular and random patterns. We will do this in the next section to discover those orchard images whose tree centers do not form a regular pattern. In this way we will avoid the problems that would ensue from computing a description of the regularity for such images and then using such a description to predict the location of undetected trees.

p_1	R	
	k=2	k=3
.2	2.8	2.4
.3	2.7	2.4
.4	2.7	2.0
.5	3.1	2.7
.6	3.7	3.2
.7	3.5	2.9
.8	3.8	3.4

Table 1. R as a function of p1

d	R	
	k=2	k=3
.2	1.6	1.3
.3	1.9	1.4
.4	1.3	1.3
.5	1.4	1.2
.6	1.3	1.3
.7	1.2	1.2
.8	1.3	1.2

Table 2. R as a function of d

3.0 Application to the description of natural images.

In this section we will discuss an application of the procedures described in the preceding section to the description of a set of real textures. As described in the introduction, the problem is to detect trees in aerial photographs of citrus groves, and then to classify the trees as being either healthy or infested. We are concerned here only with the process of tree detection. We will detect trees by first detecting a set of "obvious" trees, and then using the locations of those trees as input to the processes described in the preceding section. Those procedures will predict the locations of other trees in the image; these locations can be used by subsequent programs to detect and classify trees not found by the first stage. We describe in Section 3.1 our procedure for detecting "obvious" trees and then Section 3.2 describes the regularity measures computed based on the locations of these trees.

3.1 Detecting "obvious" trees.

The procedures for detecting trees consists of differentiating the image, and then using a Hough transform (Kimme et al[11], Shapiro[12]) to find tree centers in the differentiated image.

The image is differentiated by an edge detection procedure which first associates with each point in the image the magnitude of the maximal response from a set of directionally sensitive edge operators. These operators are very similar to the Kirsch operator[13] except that they operate on 5x5 neighborhoods. The procedures also associate the direction of the maximal response with each point. The next step is a local non-maxima suppression step. For each point, a small cone shaped neighborhood whose axis is orthogonal to the direction associated with that point is examined. If there is a point in that cone shaped neighborhood with similar direction and greater magnitude than the point at the apex, then the point at the apex is suppressed (i.e., its magnitude is marked as 0). The points which survive the non-maxima suppression process are input to the Hough transform programs. The encoding for the eight different types of edges which these procedures compute are shown in Table 3.

The next step is to compute the circular Hough transform for this edge image. This is done as follows. First, a range of circle radii is specified. The purpose of specifying a range of radii rather than a single radii is:

1) to allow for a certain amount of "eccentricity" in the circles which we will detect - i.e., the procedures will actually detect ellipses of fixed eccentricity as well as circles, and

2) to allow for a range of tree sizes.

Given this range of radii, the Hough transform is computed as follows. Each non-zero point in the edge image is examined and makes a contribution to certain cells in The Hough transform. The affected cells are those that lie on the set of semicircles whose radii lie in the specified range and whose points all lie in the positive half plane defined by the direction associated with the edge point (see Figure 7). The reason for only considering the positive half plane is that we have prior knowledge that the trees are brighter than their surrounds, so that there is no need to detect dark circles.

Given the complete Hough transform we search for circles in the image using the following procedure:

1) Find the point in the Hough space with the highest magnitude - this will correspond to the point in the image surrounded by the greatest number of edges within the set of sepcified radii.

2) Return to the image and examine the set of edge points which gave rise to the peak in the Hough transform. Create a histogram of eight bins, where the i'th bin contains the number of edge points in the i'th octant (with origin at the center point) which have the "correct" direction associated with them (see Figure 8). Call this histogram D.

3) Apply some evaluation function to D to determine if the spatial distribution of edges should be labeled as a circle. In the examples which follow, we adopted the criterion that a circle must have at least five non-empty bins in the histogram, and the total number of points in the histogram must be at least 10. If the point passes this test, then enter this point in the list of circle centers.

4) Delete from the Hough transform the contributions of all of the points which were used to form the circle detected in steps 1-3. This is a standard step in analysis using the Hough

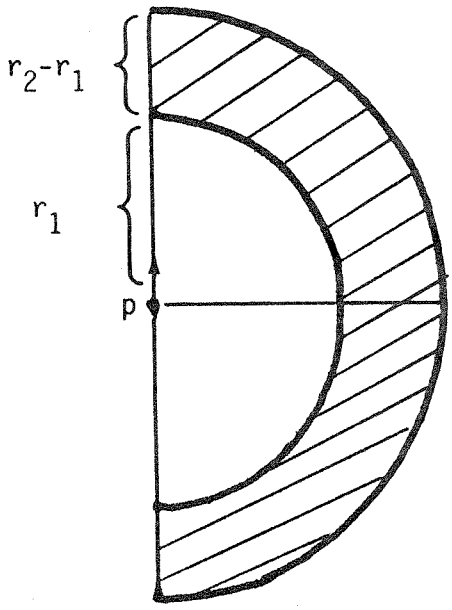
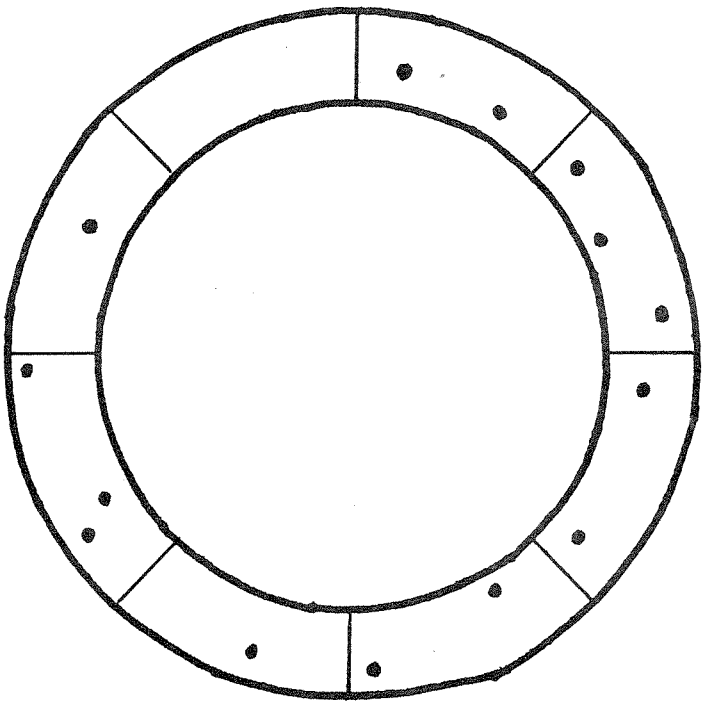


Figure 7. Hash marks show loci of possible circle centers for point p.



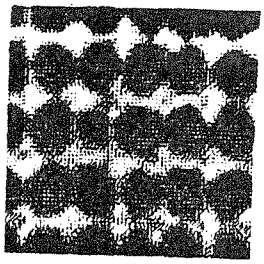
i	D(i)
1	2
2	3
3	2
4	2
5	1
6	3
7	1
8	0

Figure 8. Circle evaluation.

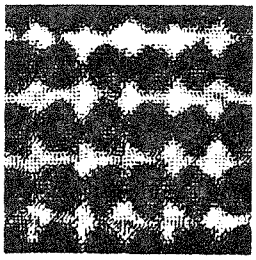
transform, and its purpose is to delete spurious peaks in the Hough transform.

5) Find the next highest point in the edited Hough space. If its magnitude is less than some threshold, then the procedure stops. Otherwise, we return to step 2.

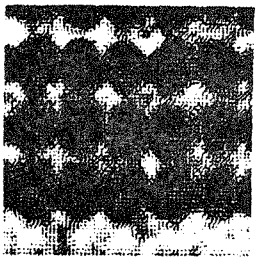
The results of applying this procedure to four 64x64 subimages of the large orchard image , are shown in Figures 9-10. Figure 9 contains the digitized orchard samples, and Figure 10 contains the edge images (using the encoding contained in Table 3), where the special symbol 'C' denotes the center of a circle detected by the Hough transform programs.



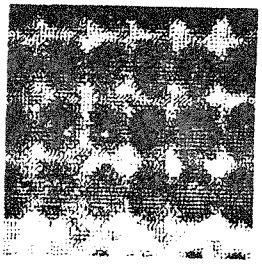
(a)



(b)



(c)



(d)

Figure 9. Four orchard images.

edge orientation	symbol
0	→
45	↗
90	↑
135	↖
180	←
225	↙
270	↓
315	↘

Table 3. Symbols for edge orientations.

Figure 10	(b_1, b_2, dir)	r
a	(15,11,.04)	3.2
b	(17,11,.05)	2.9
c	(16,11,.10)	3.2
d	(16,11,.05)	2.7

Table 4. Grid descriptions and regularity measures for orchard images.

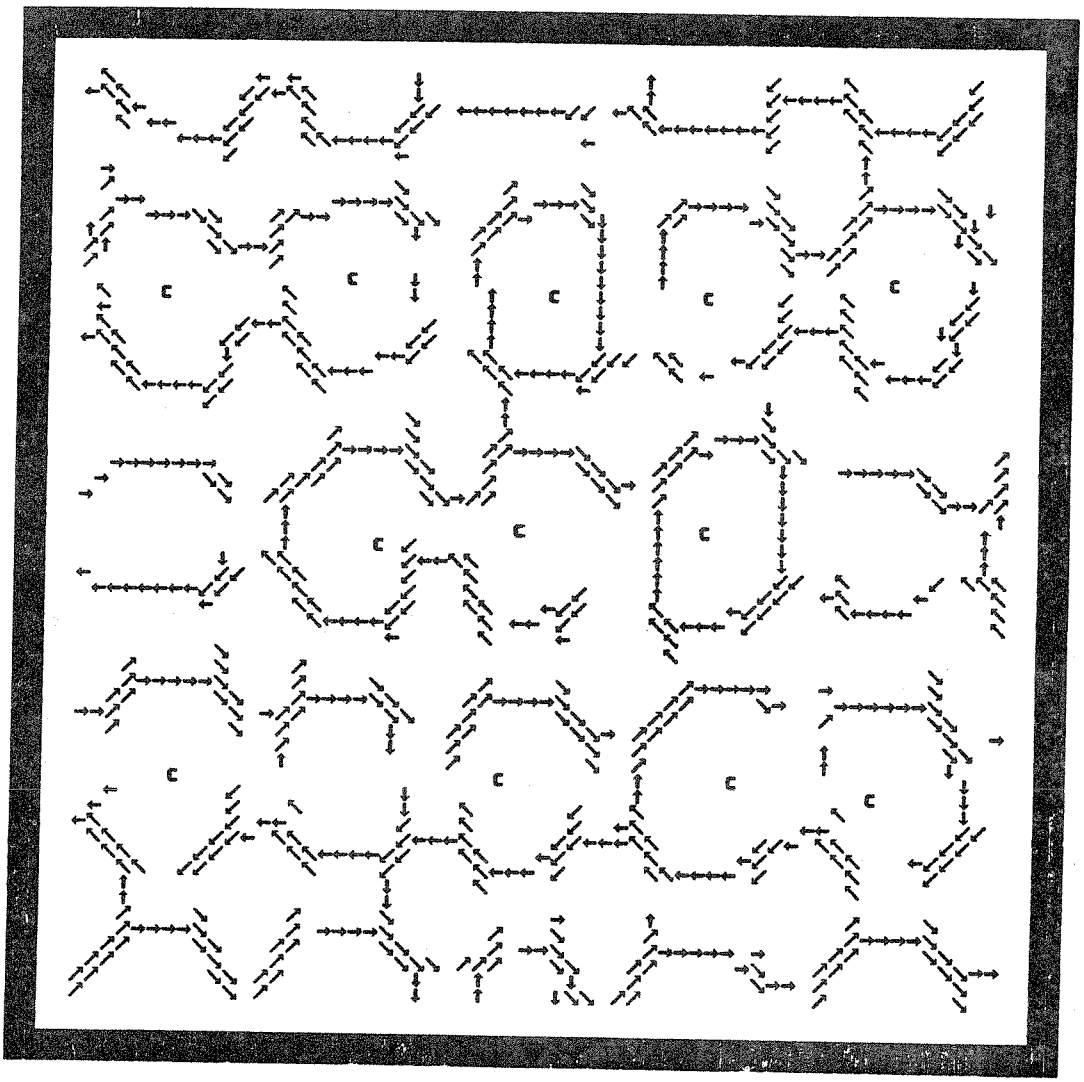


Fig. 10a. Circles centers for Fig. 9a.

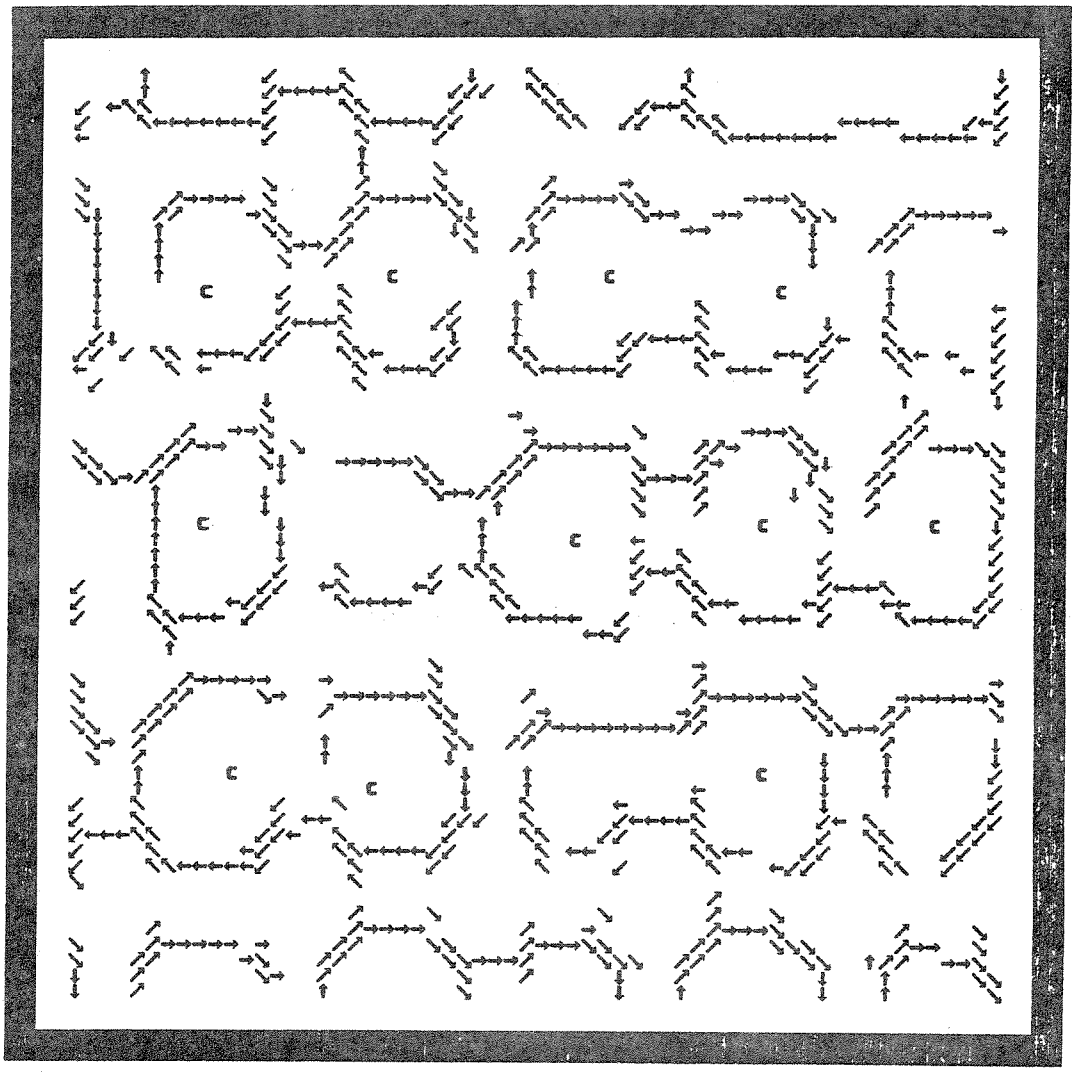


Fig. 10b. Circles centers for Fig. 9b.

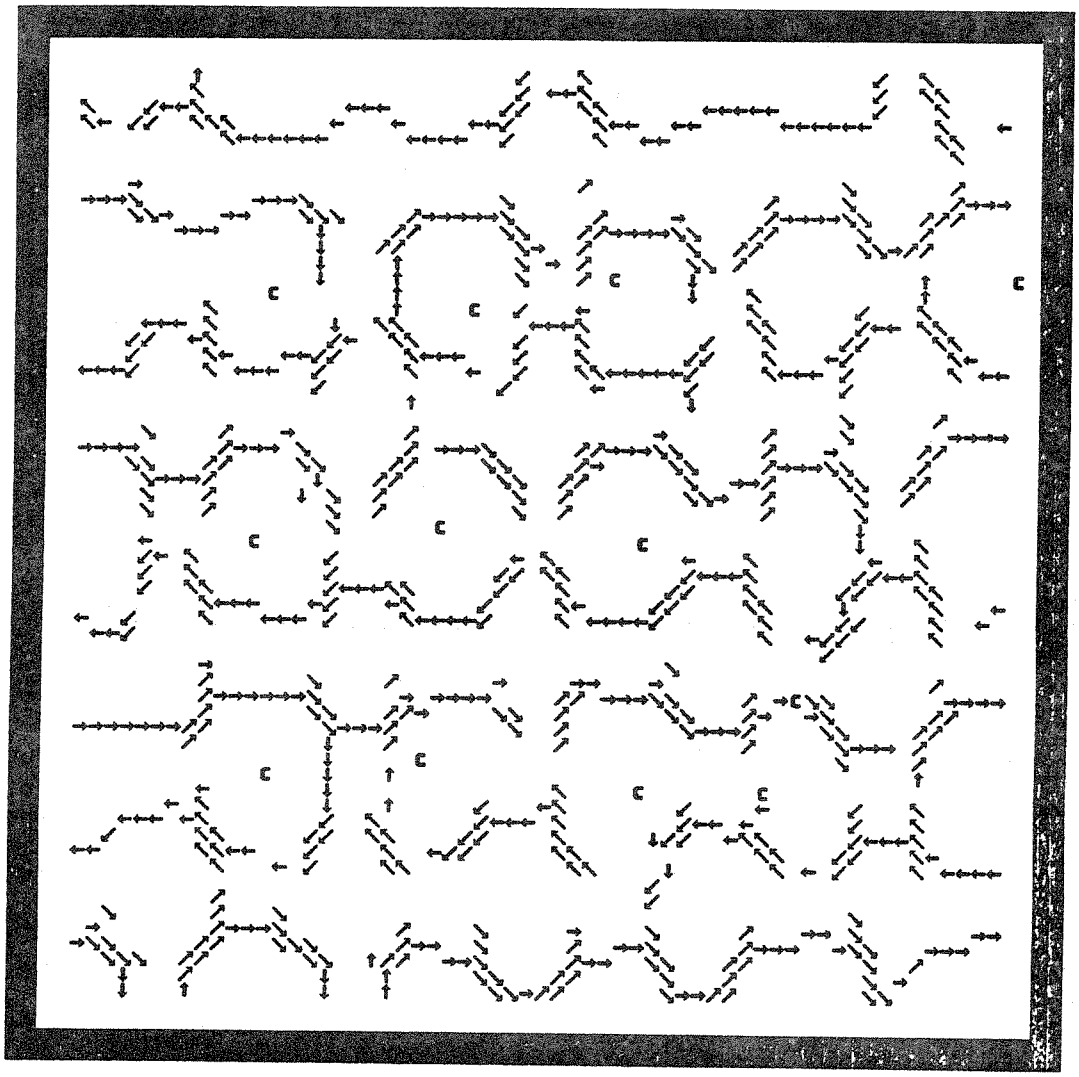


Fig. 10c. Circles centers for Fig. 9c.

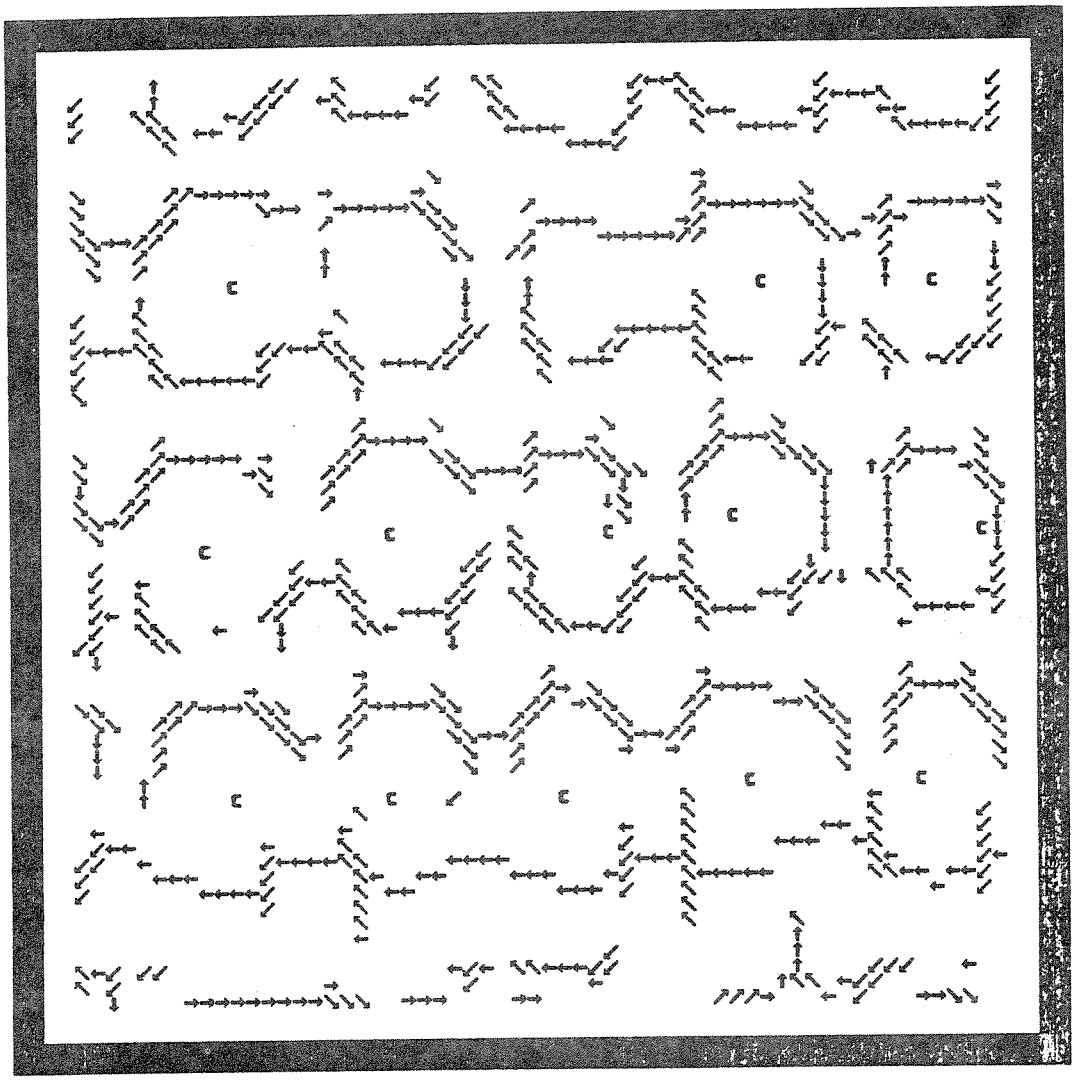


Fig. 10d. Circles centers for Fig. 9d.

3.2 Hypothesizing circle centers

Once the set of circle center points has been detected, we are in a position to apply the procedures developed in the previous section for computing the description of the underlying grid pattern. However, some changes need to be made to those procedures since the grid patterns for the orchard images are rectangular rather than square. This means that instead of having to compute only a single distance,b, as we did for square grids, we need to compute a pair of distances b1 and b2. Note that these two distances can be computed from only Hs since the two left-most peaks will correspond to b1 and b2.

Figure 11 contains the size histograms between a point and its three nearest neighbors for the images of Figure 9. For these patterns, the two major peaks in Hs correspond to b1 and b2 in the underlying rectangular grid. We next compute two filtered direction histograms, one for b1 and one for b2. Each is searched for dominant peaks. The direction of the grid is determined by the pair of peaks, one from the direction histogram for b1 and the other from b2, whose sum of sizes is greatest and whose angular difference is tolerably close to pi/2. It should be noted the for the images which we investigated, Hd, the filtered histograms for b1 and b2 never contained more than a single major peak, so that computing the orientation of the grid was straightforward. Table 4 lists b1, b2, the grid orientations, and the regularity measure for the patterns corresponding to the loci of circle centers for the four orchard images. Notice that in all four images the

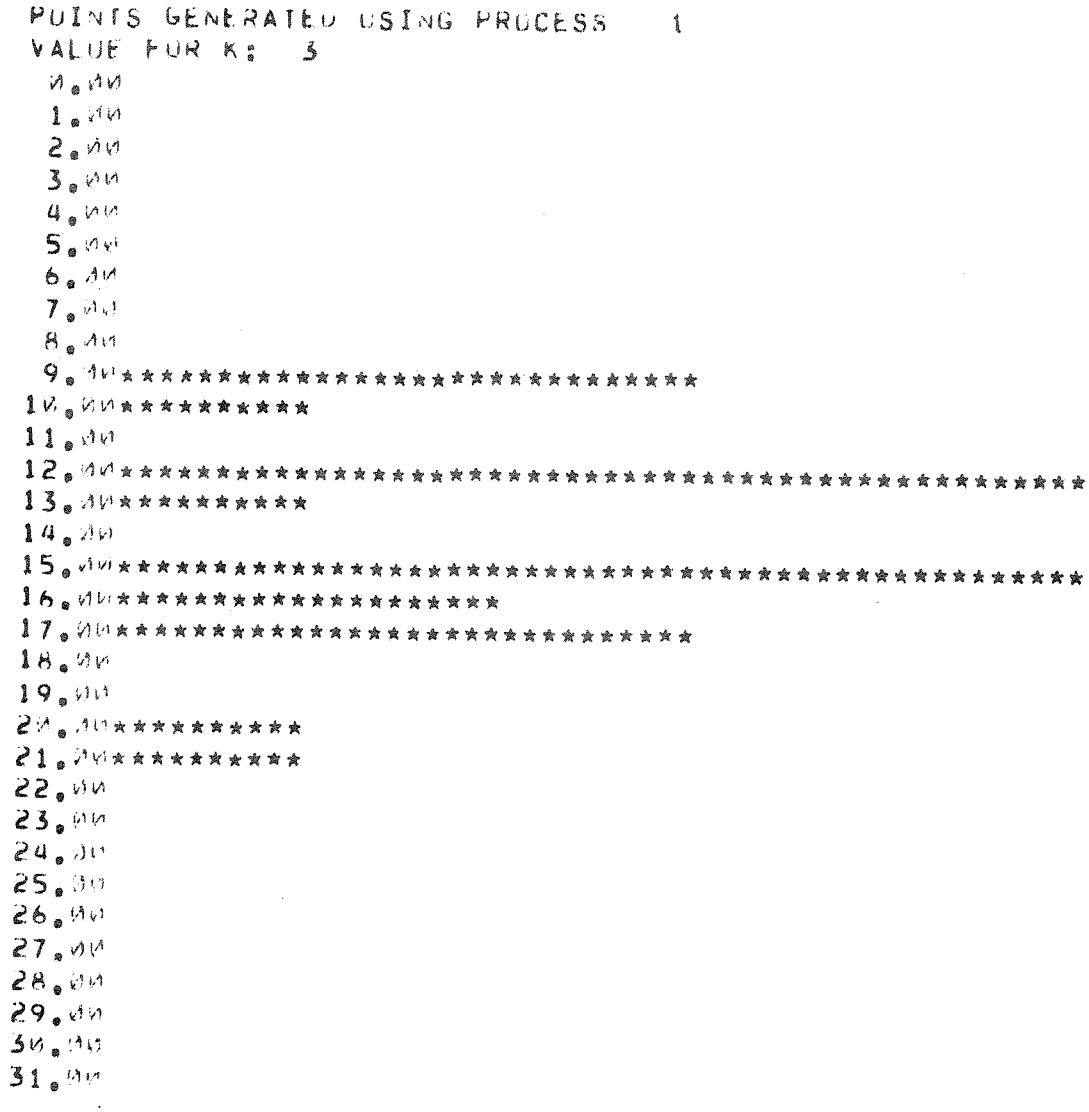


Fig. 11a. H_s (K=3) for Fig. 9a.

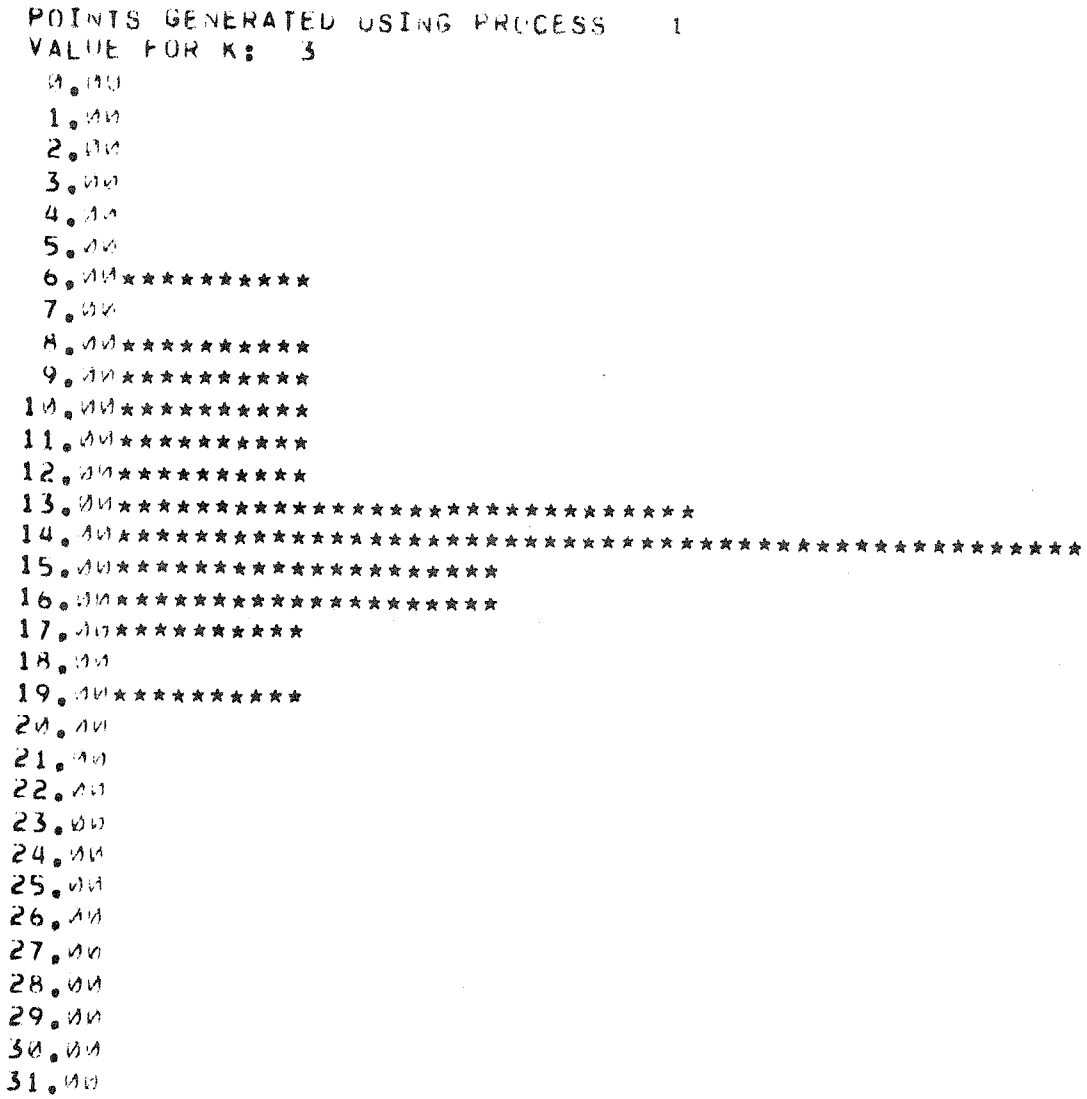


Fig. 11b. H_s (K=3) for Fig. 9b.

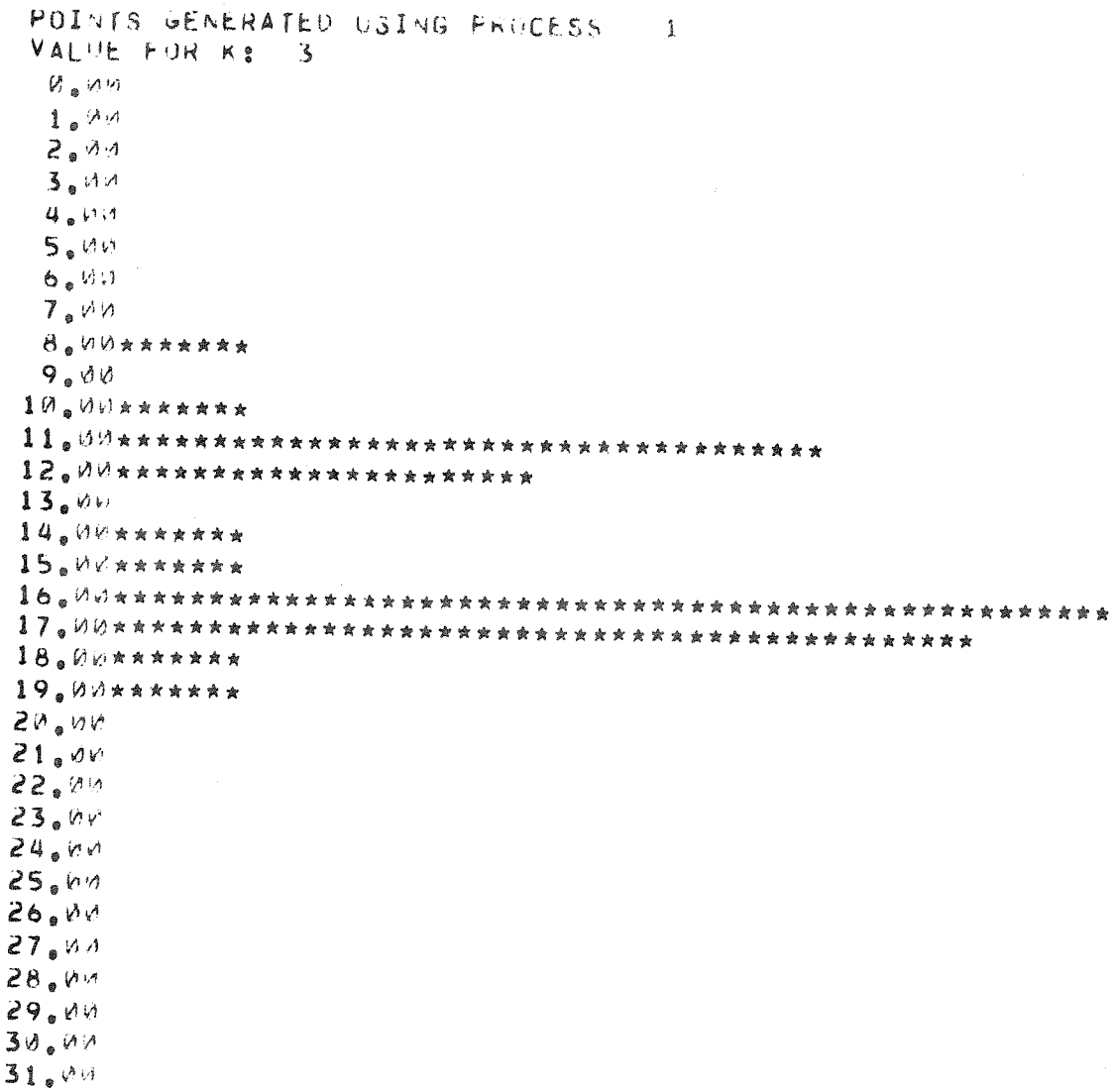


Fig. 11c. H_S (K=3) for Fig. 9c.

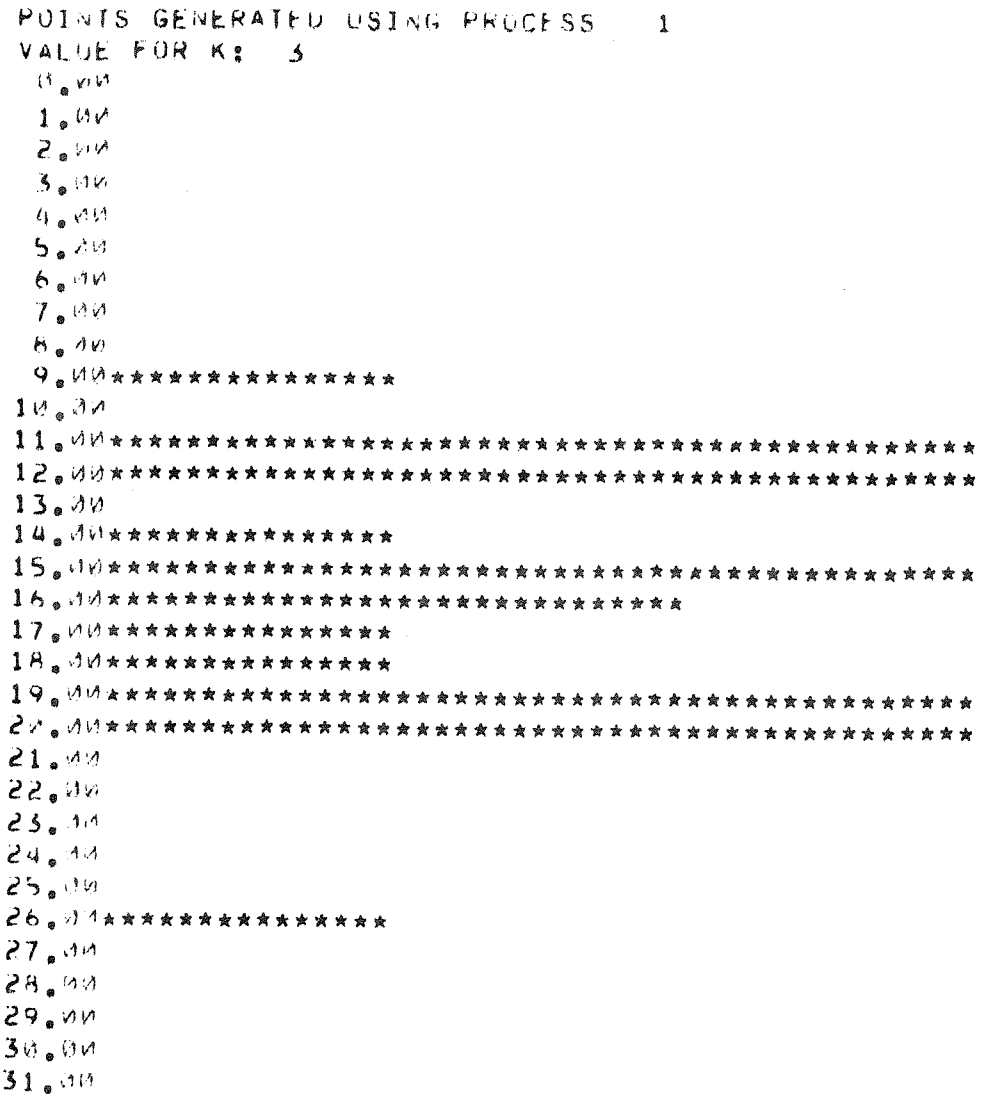


Fig. 11d. H_s (K=3) for Fig. 9d.

regularity measure is significantly higher than the threshold of 2.0 which we determined in Section 2.

Once the grid pattern is specified, it remains to hypothesize the locations of the undetected circles. The procedures to do this are defined as follows: Let C = { ci= (xi,yi)} be the set of detected circle centers, and let (b1,b2,dir) define the rectangular grid pattern. In the discussion which follows we will refer to the "grid for circle center ck." This is the grid having sizes b1 and b2 and orientation dir which is centered at ck - i.e., for each ck, we translate the grid so that ck is at the origin. Let Dk(p) be the distance of a point p from the nearest grid point in the grid for circle center ck. If p lies exactly on a grid point in the grid for center ck, then Dk(p)=0.

Now, let F(p,c) be a predicate defined over pairs of point positions, where p is an arbitrary image point and c is an element of C. Two examples of such predicates are:

1)F1(p,c) is true if the distance between p and c is less than some threshold t.

2)F2(p,c) is true if c is one of the k nearest neighbors of circle centers to the point p.

Now, we define:

$$D(p) = \sum_{ci} Di(p)*F(p,ci)$$

Here, we intrepret TRUE as 1 and FALSE as 0 for computing this sum. D can be regarded as an image. The value of D(p) is an

indicator of how close the point p fits the rectangular grid pattern defined by the ci. We then compute as our hypothesized centers the local minima of the D image. That is, the point p is a hypothesized center if D(p) <= D(q), where dist(p,q) < d. Here, d determines the radius of the local minima operator.

We found, empirically, that the location of the circle centers was not substantially effected by the choice of spatial predicate. Figure 12, for example, shows the hypothesized circle centers, labeled as H's, for the orchard in Figure 9a, using the spatial predicate F1 with t=40 and 80, and the spatial predicate F2 based on k=3 nearest neighbors. The results are quite similar. Figure 13, then, shows the hypothesized circle centers for the remaining three orchard images using F2 with k=3. (Note that the H's near the boundaries of the image are due to border effects of the non-maxima surpression and should probably be discounted.)

We should mention that the reason for introducing the spatial predicate in computing the D image instead of simply basing the computation of D on all circle centers is to make the computation of the D image more sensitive to local changes in the structure of the underlying grid. This would be more significant if we were directly processing larger image areas. In such a situation, we might want to associate a grid description with each detected circle center rather than with an entire image; the computation of D using a spatial predicate would then reflect any gradual changes in the grid description as we moved around in the image.

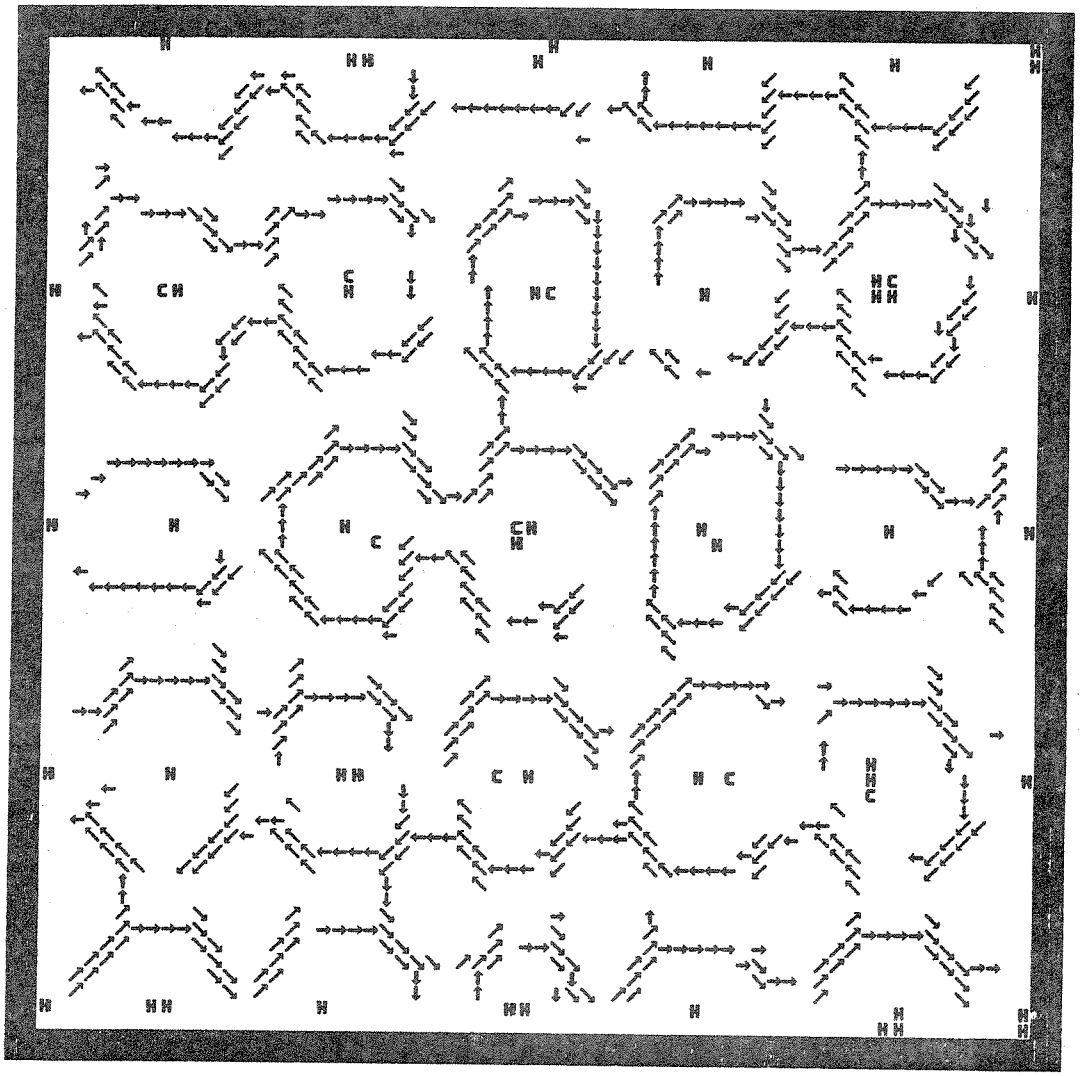


Fig. 12a. F1, t=40 for Fig. 9a.

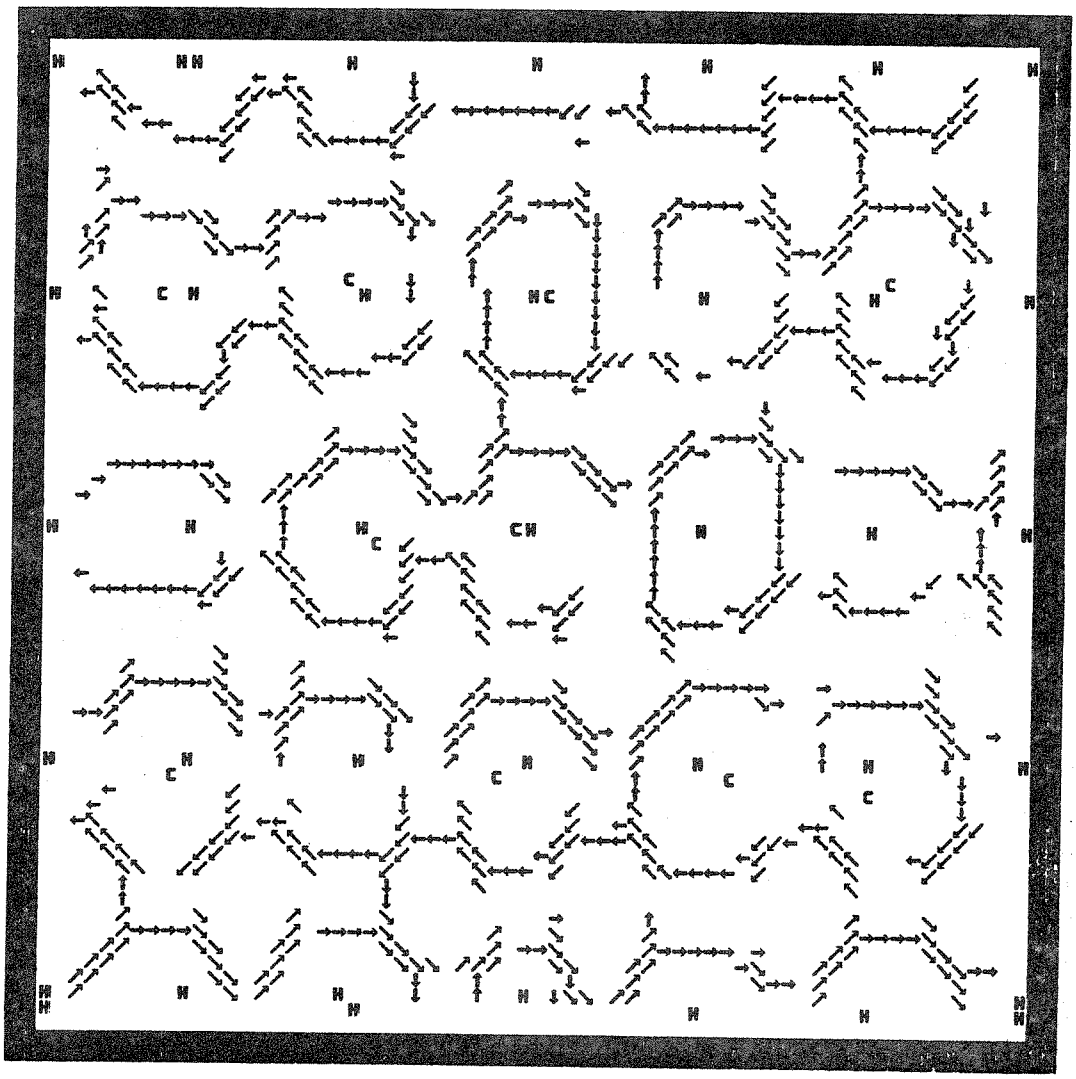


Fig. 12b. F1, t=80 for Fig. 9a.

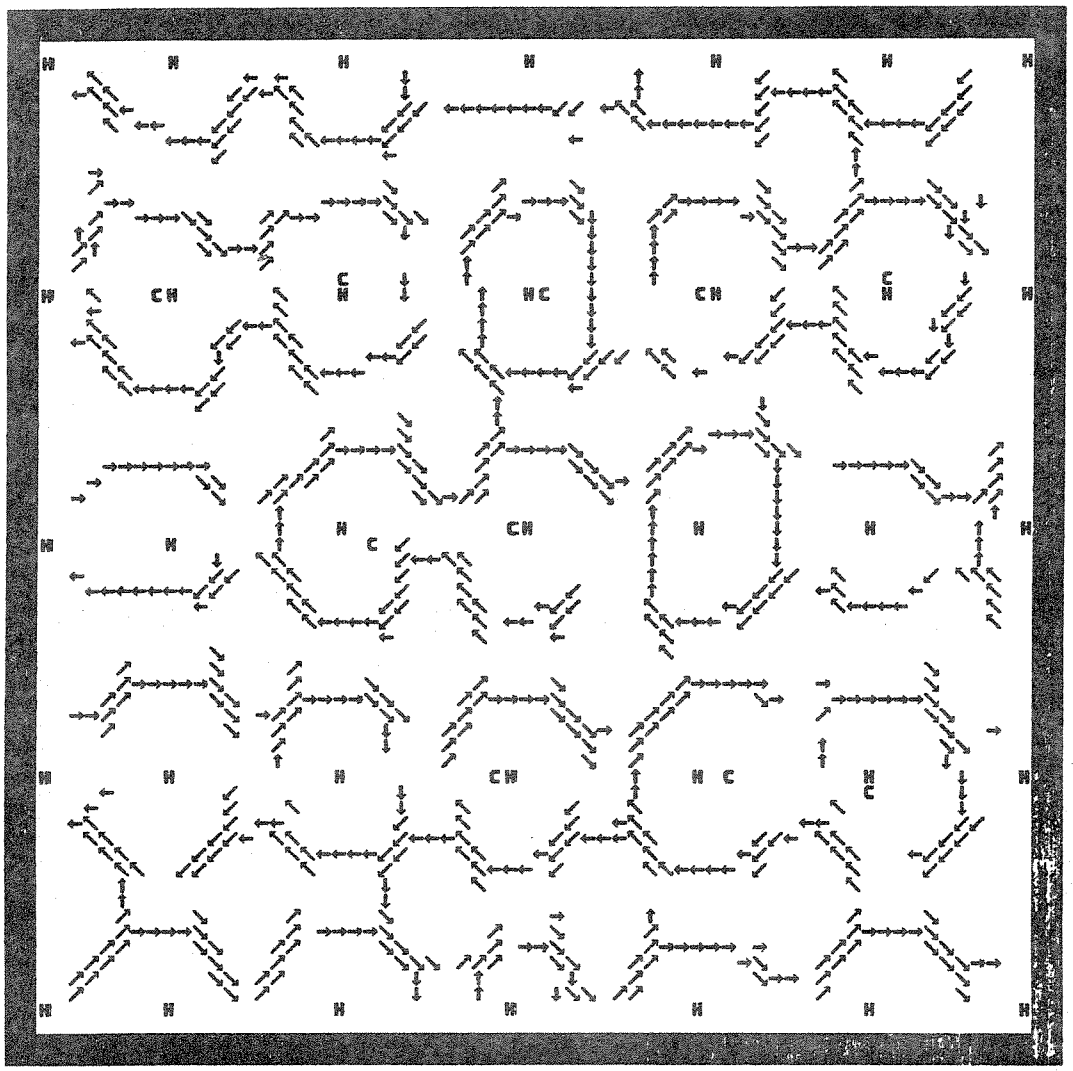


Fig. 12c. F2, k=3 for Fig. 9a.

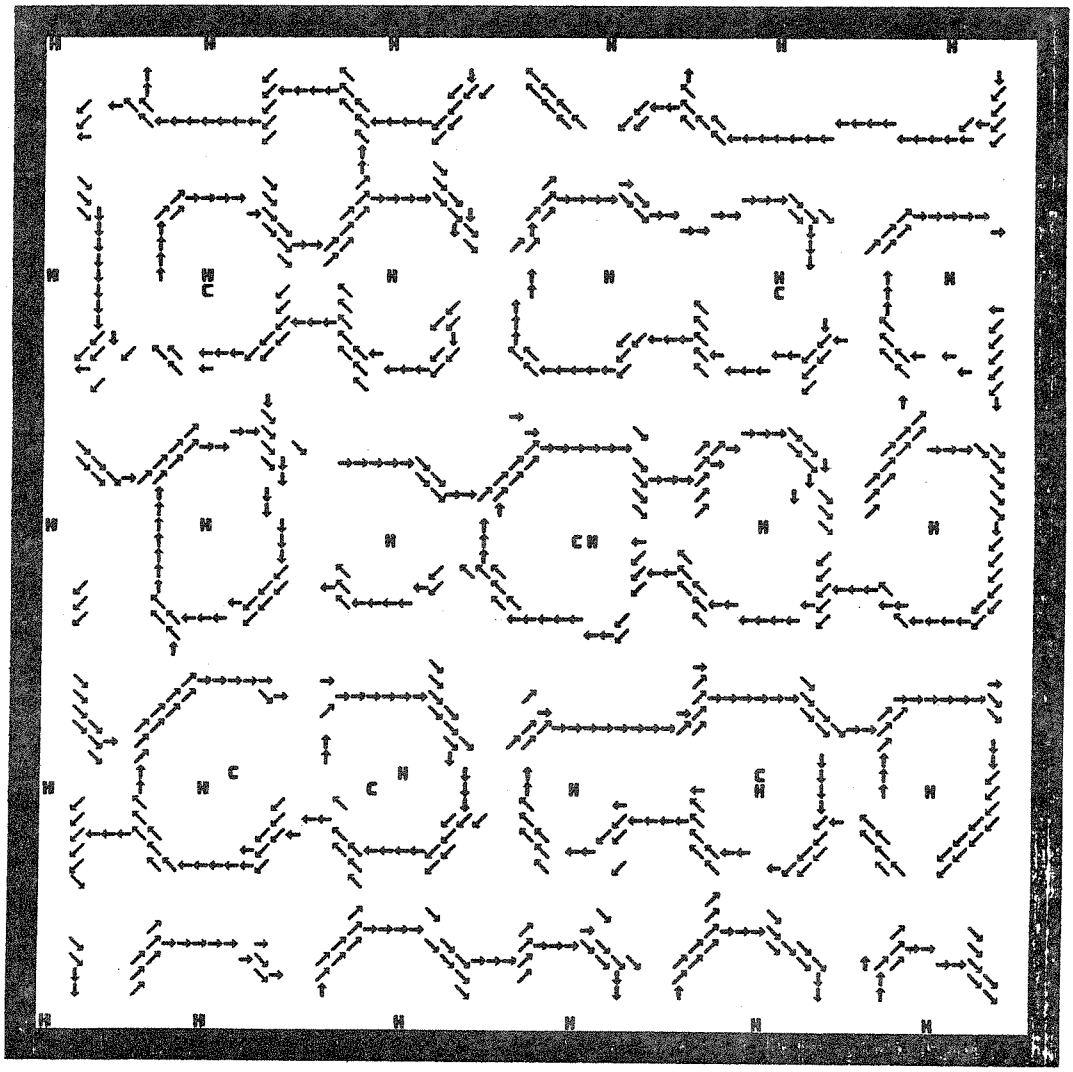


Fig. 13a. F2, K=3 for Fig. 9b.

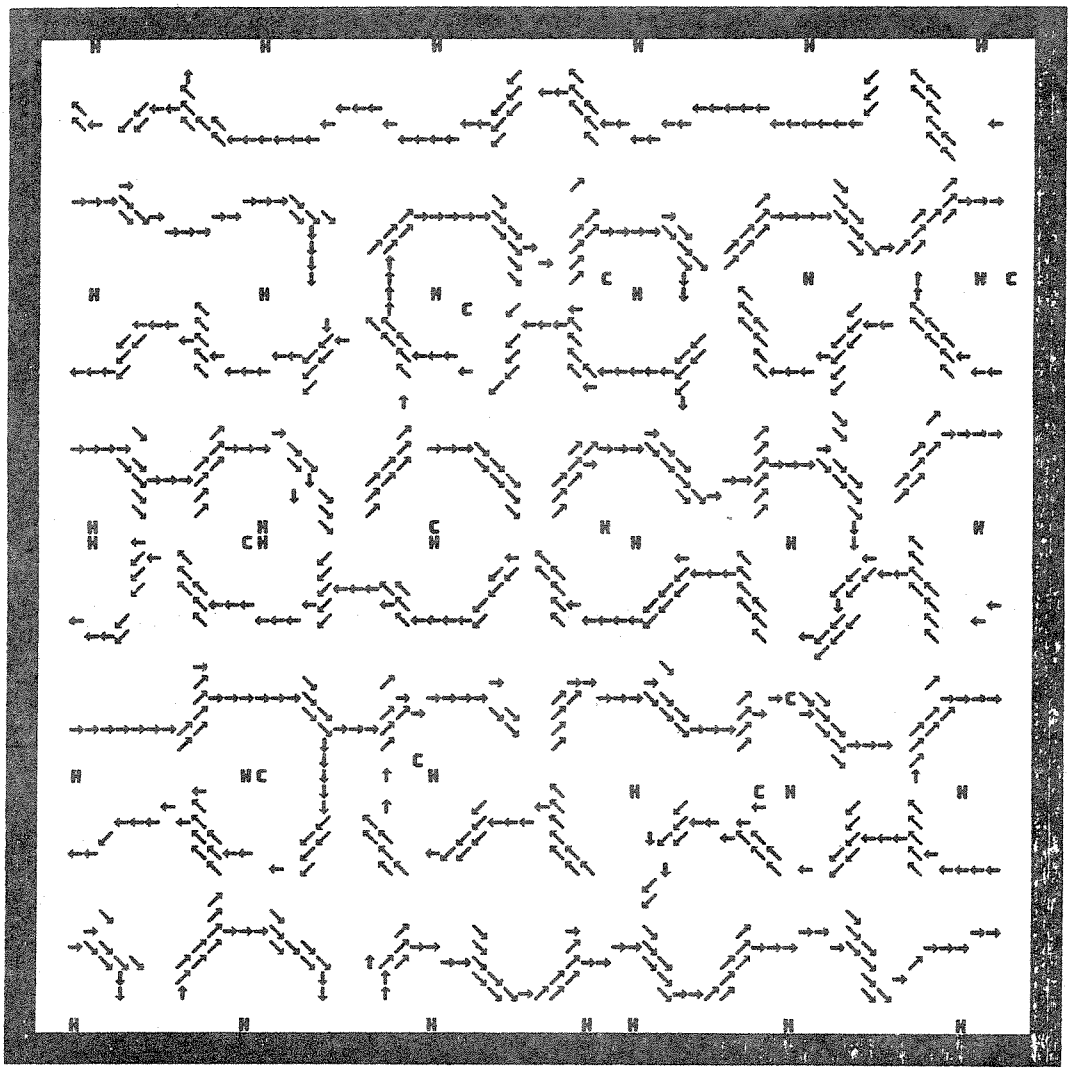


Fig. 13b. F2, K=3 for Fig. 9c.

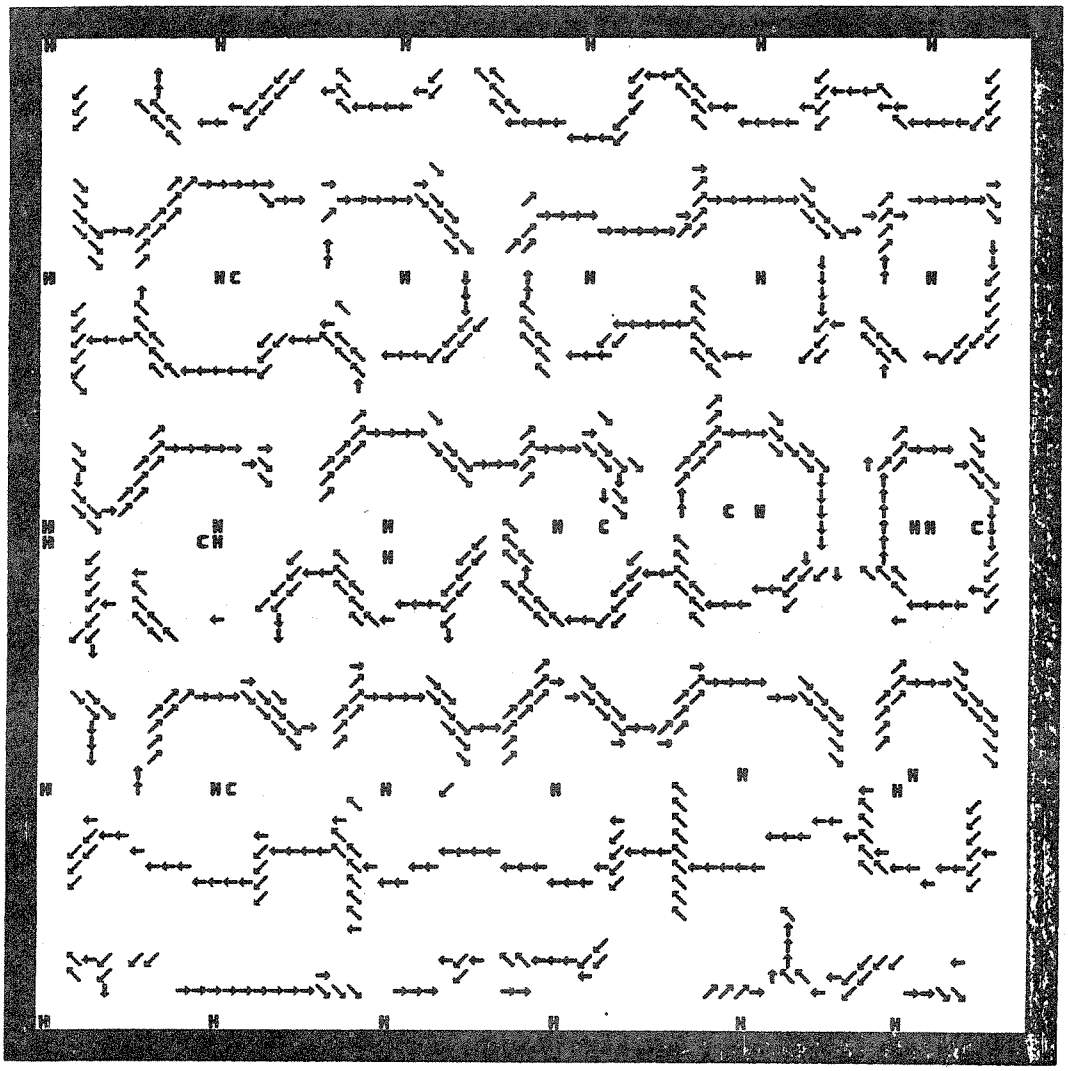


Fig. 13c. F2, K=3 for Fig. 9d.

4.0 Conclusions

In this paper we have concentrated on describing the spatial structure of dot patterns. These correspond to an idealization of textures, where all of the texture elements have identical properties. For such textures, the only available information for distinguishing between one and another is the spatial placement of the elements in the image. We based our method of discrimination on the fact that for random point patterns the distribution of directions between a point and its nearest neighbor is uniform, while for a regular texture, sharp peaks should be detectable in the direction histogram. We therefore proposed, as a method for computing the spatial structure of dot patterns, to construct the histogram of directions between a point and its ks-nearest neighbors, where s is determined by computing the peaks of the size histogram. Although we were specifically concerned with rectangular and square grids, the approach could be applied to other regular patterns as well - e.g., for a hexagonal point pattern we would expect three peaks in the filtered direction histogram (here $k>4$).

Finally, we described an application of these procedures to the analysis of images of orchards containing infestations. This involved first finding a set of "obvious" tree centers using Hough transform techniques, then describing the set of points corresponding to the tree centers as a rectangular grid, and finally using the description of that grid to predict the locations of other trees.

Acknowledgement

The author would like to thank Worthy Martin for his many careful readings of this paper and his help in preparing several of the figures.

References

1. Haralick, R. "A survey of texture analysis techniques," to appear in Proc. IEEE.

2. Maleson,J., C. Brown and J. Feldman, "Understanding natural textures," Proc. DARPA Image Understanding Workshop, Palo Alto, CA, 1977, 19-27.

3. Davis, L., S. Johns, and J. K. Aggarwal, "Texture analysis using generalized cooccurrence matrices," to appear in IEEE-Trans. on Pattern Analysis and Machine Intelligence, July 1979.

4. Ehrich, R., and J. P. Foith, "Topology and semantics of intensity arrays," Computer Visioo, Hanson and Riseman, eds., Academic Press, New York, 1978.

5. Pickett, R., "Visual analysis of textures in the detection and recognition of objects," in Picture Processing and Psychopictorics, ed. A. Rosenfeld and B. Lipkin, Academic Press, NY, 1970, pp. 289-308.

6. Rosenfeld, A. and B. Lipkin, "Texture synthesis," ibid., 309-345.

7. Williams, D. H., "Computer detection of citrus infestations using aerial color infrares transparancies," Ph.D. Thesis, Department of Electrical Engineering, University of Texas, Austin TX, 1977.

8. Schachter, B., and N. Ahuja, "A survey of random pattern generation processes," University of Maryland Computer Science Center TR-549, July, 1977.

9. Kendall, M.G. and P. Moran, Geometric Probability, Hafner Publishers, New York, 1963.

10. Stevens, K., "Computation of locally parallel structure," in Proc. DARPA Image Understanding Workshop, Nov., 1978, 92-102.

11. Kimme, C., D. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," Comm. ACM,19, 1975.

12. Shapiro, S., "Transforms of curves in noisey pictures," CGIP,12, 1978.

13. Kirsch,R., "Computer determination of the constituent structure of biological images," Computers and Biomedical Research,4, 1973.