Technical Report CS79001-R

Segmentation of Images with Incompletely Specified Regions

by

Peichung F. Lai

and

Roger W. Ehrich

Department of Computer Science

lytechnic Institute and State University

Blacksburg, Virginia 24061

February 1979

Captions

1. Examples of images with incompletely specified regions.
2. Classification of a point to category of nearest training sample.
3. Completion of regions by nearness procedure.
4. (a) A 32 x 32 sample with 87 centers, 2 regions, and step-by-step iterations using 4-neighbor diffusion procedure are shown after (b) 1 iteration, (c) 3 iterations, (d) 7 iterations.
5. (a) Original texture sample (SKY-CLOUD), (b) Classified local maxima, (c) Partitioning result by nearness procedure using the absolute value distance, and (d) Partitioning result by diffusion procedure using 4-neighbors.
6. (a) Another texture sample (TREE-CLOUD), (b)-(d): same as in Figure 5.

Table 1 - Computation times for various samples. (in all cases, L=2)

ABstract

Sometimes image regions are incompletely specified in the sense that only a few representative points in each region are known. In order to determine a segmentation of such an image it is necessary to construct the region boundaries. A fast algorithm called a diffusion algorithm is given that determines region boundaries by diffusing region labels outward from the known points.

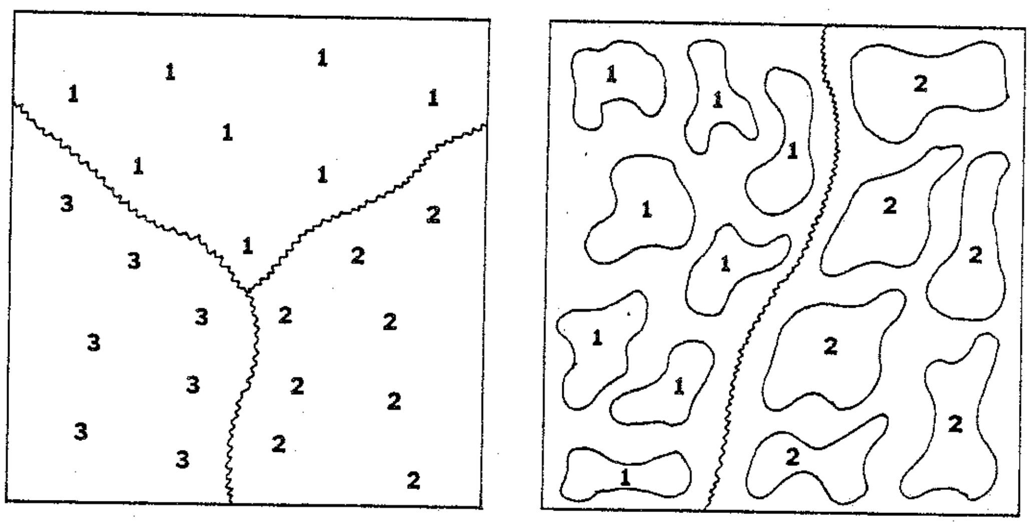
Key Words

Clustering, nearest neighbor, pattern analysis, region growing, segmentation, texture analysis. Computing reviews catagory: 8.2.

The work reported here was supported by the National Science Foundation under Grant ENG-7712105.

Introduction

A problem that sometimes arises in region growing procedures in image analysis concerns the generation of smooth, well defined connected regions given only a few isolated points from each region. Such regions are called incompletely specified regions. Two examples are shown in Figure 1. In the first example there are three adjacent clusters of labeled points, and in the second example there are two adjacent clusters of labeled blobs. In both cases the problem is to construct boundaries between the clusters of objects with different labels.



a b

Figure 1 - Examples of Images with Incompletely Specified Regions.

The problem above is equivalent to the pattern recognition problem of determining piecewise linear decision boundaries that separate the clusters of points with different labels [Ni65]. In this correspondence, the problem of determining the boundaries is solved by a "diffusion" algorithm in which the regions are enlarged until they meet and form a boundary which is piecewise linear.

In pattern analysis, a pattern is usually represented by a point in some d-dimensional Euclidean pattern space E^d. Each point in E^d is mapped by a classifier into the category numbers, 1,2,...,L. There are basically two ways to implement a pattern classifier. In the "decision region" approach a dictionary is stored that contains the classification of each point in E^d. Since this is usually computationally infeasible, a "decision boundary" approach is often used. In the decision boundary approach the regions of E^d are stored by encoding their boundaries in some way. Then the classifier makes use of these encodings to deduce the category number of an arbitrary pattern.

If one regards a digital image as though it is a bounded region of a discrete two-dimensional pattern space, it becomes possible to describe the pattern classes by enumerating the classifications of all the points in the bounded space. Therefore the approach discussed in this paper is a region approach, and it is particularly useful when the shapes of the decision boundaries are extremely complex.

Construction of Decision Regions

Suppose we are given a training set of m categorized patterns, and assume that there are L categories. Constructing the decision regions is equivalent to determining an extended mapping of all points in the pattern space into category numbers 1,2,...,L so that all points in E^d will be correctly classified. In particular, let R be the training set, d(x,y) a distance measure on E^d, and let

$$f_R: R \rightarrow \{1,2,...,L\} \quad , \qquad R \subset E^d$$

be the mapping from the training set into the category numbers. Then we wish to determine an extended mapping

$$f: \ E^d \rightarrow \{1,2,...,L\}$$

such that $f|_R = f_R$ and such that $\forall\, p \in \{E^d - R\}$,

$$f(p) = f_R(r) \text{ for some } r \in R$$

if and only if there is no $q \in R$, $q \neq r$, such that $d(q,p) < d(r,p)$. An example is shown in Figure 2.

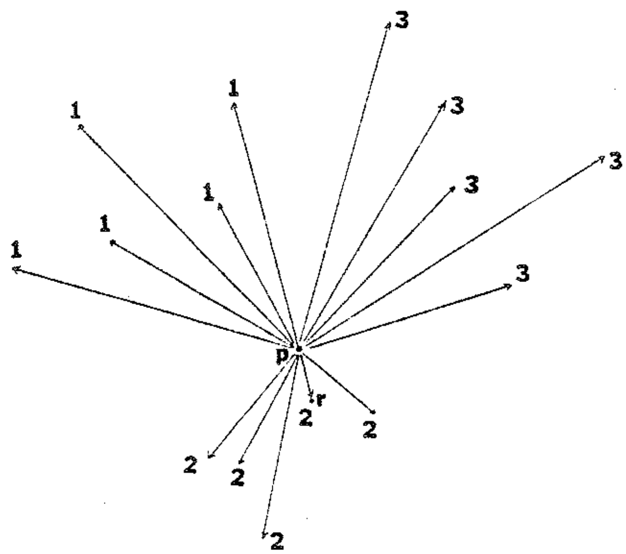


Figure 2 - Classification of a Point to Category of Nearest Training Sample.

Next, two procedures are given for computing f. The first is a straighforward "nearness" procedure, and the second is called a "diffusion" procedure that is particularly amenable to a parallel implementation. These procedures produce identical results except when a point, p, has two equidistant training patterns in R that map into different categories. However, the diffusion procedure will be shown to be substantially faster. In the following discussion, each element in R is referred to as a "center."

(l) Nearness Procedure:

First, define three common distance functions d(x,y) in the pattern space [Du73]. They are,

(i) Euclidean Distance:

$$d(x,y) = SQRT(SUM((x_1-y_1)^2, \ldots, (x_d-y_d)^2))$$

(ii) Absolute Value Distance:

$$d(x,y) = SUM(ABS(x_1-y_1), \ldots, ABS(x_d-y_d))$$

(iii) Maximum Value Distance:

$$d(x,y) = MAX(ABS(x_1-y_1), \ldots, ABS(x_d-y_d))$$

The procedure is described by the following steps.

For each individual point in $\{E^d-R\}$,

1) compute the "distance" between the point in the pattern space and each center.

2) search for the "nearest" center.

3) assign the category number of the selected center to the point.

Figure 3a is an example of 2-dimensional pattern with 87 centers, and 2 categories. Figure 3b shows the result after the nearness procedure is applied using distance measure. Three connected regions were obtained, two of which have the same label.

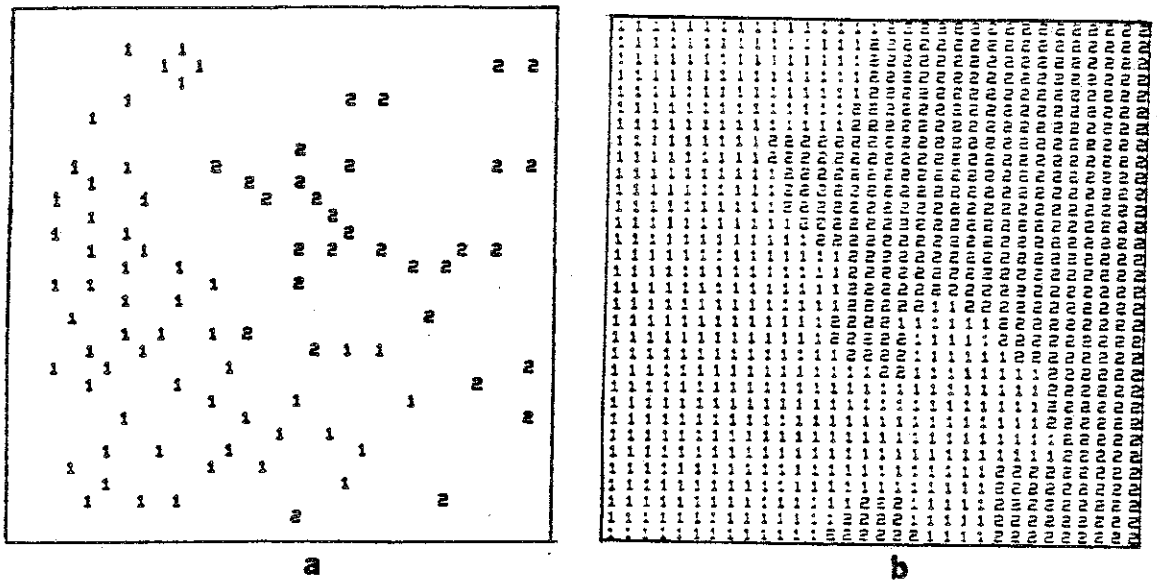


Figure 3 - Completion of Regions by Nearness Procedure.

(2) Diffusion Procedure:

In the nearness procedure, the computation time required is proportional to m, the number of centers, and usually is very large. The diffusion procedure described below has the advantage that the computation time required is proportional to L, the number of categories, which is usually very small.

The idea of the diffusion procedure is try to expand or "diffuse" each center outward in all directions into the entire pattern space. For the purpose of describing the procedure, we shall call all points in $\{E^d-R\}$ as "grounds," and an individual point in $\{E^d-R\}$ as a "ground point". The local neighborhood around a point is the set of points adjacent to it. For example, the "maximal neighbor set" contains all points adjacent to a point and has 3^d-1 members. Hence, in 2-dimensions, the local neighborhood would consist of those points that are 8-adjacent to the given point.

The diffusion procedure is an iterative procedure. In each iteration, each categorized point is expanded by assigning its category number to its neighbors, and the procedure is repeated until all ground points are categorized. For some ground points, however, there will be ties when they are simultaneously in the neighborhood of several classified points that have different category numbers. A reasonable strategy for resolving such ties is to use a majority voting scheme.

Whenever a tie occurs, the ground point is assigned the category number that occurs most frequently among its classified neighbors. Only when there is no majority is a random classification made.

An efficient way to implement the voting scheme is by using the concept of the indexed counter. It is described as follows. Define

$$g(i_1,\ldots,i_d) = f(p)$$

Where $(i_1,\ldots,i_d)$ is the location of the point $p \in E^d$. The first step of the procedure is to associate every center with its category number and every ground point with the null category, denoted Ø. Thus set

$$f(p) = f_R(p) \quad , \quad \forall\, p \in R$$

and set $f(q) = \emptyset$, $\forall\, q \in \{E^d - R\}$.

For each point in category Ø, let VOTES(Ø),...,VOTES(L) be counters for tallying the votes for each of the L+1 categories. If one defines the local neighborhood set as the maximal-neighbor set, the diffusion operation on an individual ground point at location ($i_1, \ldots, i_d$) is the following:

1) Initialize VOTES(Ø),...,VOTES(L) to Ø.

2) Increment VOTES(g($i_1-1, i_2, \ldots, i_d$)) by 1

   Increment VOTES(g($i_1+1, i_2, \ldots, i_d$)) by 1

   .

   . ((3^d-1) combinations)

   .

   Increment VOTES(g($i_1, i_2, \ldots, i_d+1$)) by 1.

3) Find the category number, k, for $Ø \leq k \leq L$, for which VOTES(k) is maximum.

4) Assign the category number k to g($i_1, \ldots, i_d$).

Steps 1) through 4) define a local neighborhood operator called the diffusion operator which will be denoted δ. In one iteration, δ is applied in parallel to each point in category Ø so that the diffusion process is not dependent upon the order in which the points are considered. The procedure is iterated until no points have category number Ø.

For a 2-dimensional example, if one uses 4-neighbors as the local neighborhood set, Step 2 becomes:

2) Increment VOTES(g(i_1-1, i_2)) by 1
   Increment VOTES(g(i_1+1, i_2)) by 1
   Increment VOTES(g(i_1, i_2-1)) by 1
   Increment VOTES(g(i_1, i_2+1)) by 1

Figures 4a to Figure 4f show the results after various iterations for the example in Figure 3. Notice that the resulting regions are similar to those in Figure 3b, which were computed using the nearness procedure.

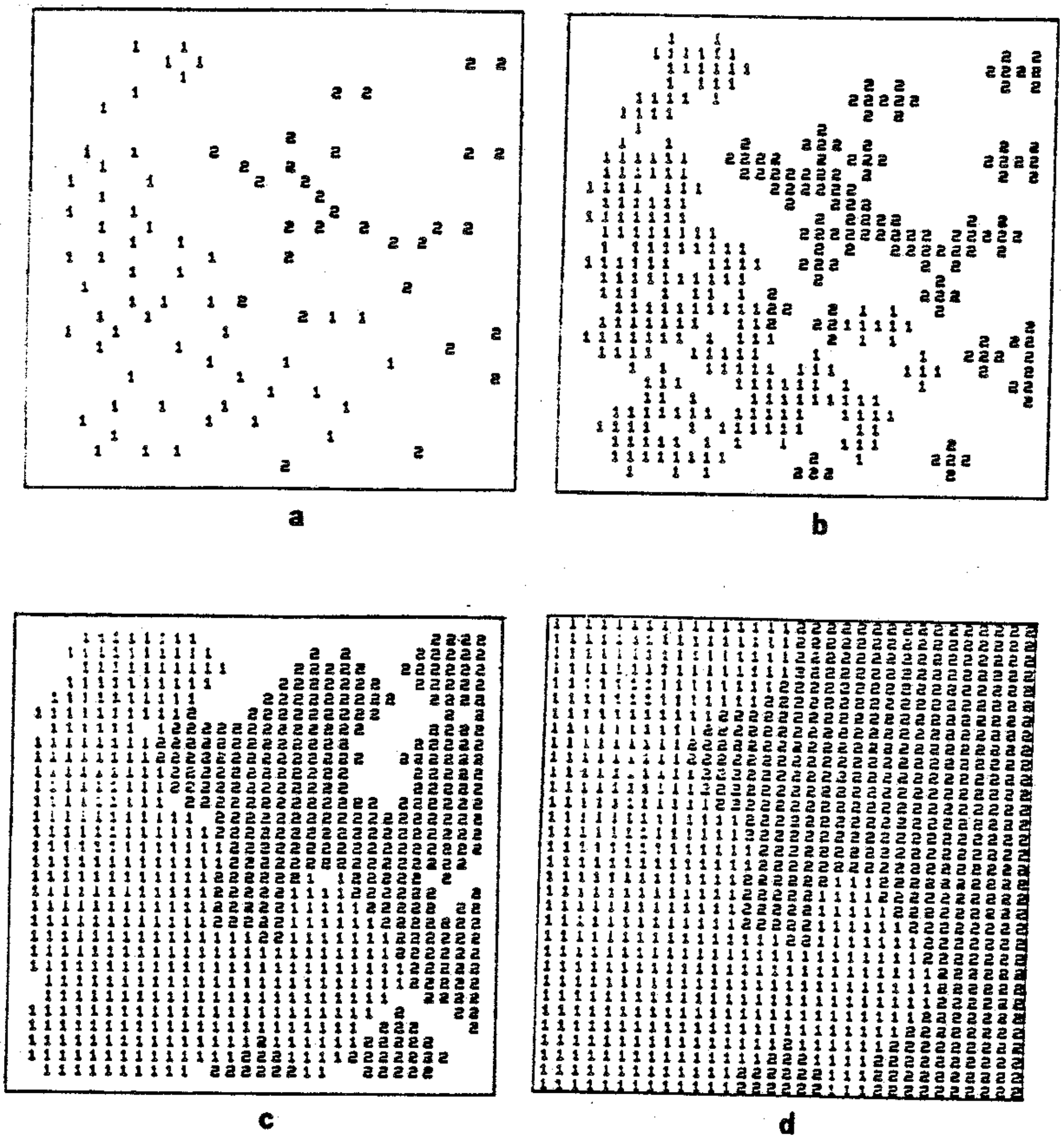


Figure 4 - (a) A 32x32 sample with 87 centers, 2 regions, and step-by-step iterations using 4-neighbor diffusion procedure are shown after (b) 1 iteration, (c) 3 iterations, (d) 7 iterations.

Application to Image Segmentation

There are a number of image processing applications in which the goal is to segment an image into disjoint regions [Ro76]. Because of the large number of picture elements involved, frequently each region of the segmented image is represented either by a primitive or by a set of representative points. For example, a region might be represented by the points in its boundary [St73], or it might be represented by specifying local extrema [Eh78,Ca77]. To reconstruct the regions of an image, all image points must be assigned region labels. In this section two examples are given and the computation times for the nearness and diffusion procedures are compared.

Figure 5a shows a texture sample which is composed of two different regions (SKY-CLOUD). Figure 5b shows the spatial distribution of the primitives which are simply local maxima of the picture function. The number labels of the primitives indicate the categories to which they belong [Eh78]. In this example, $d = 2$, $L = 2$, $m = 424$, and the pattern space is (64 x 64). For the nearness procedure, the absolute value distance was arbitrarily used, and Figure 5c shows the result. Comparing with the original image in Figure 5a, the resulting boundary is well formed.

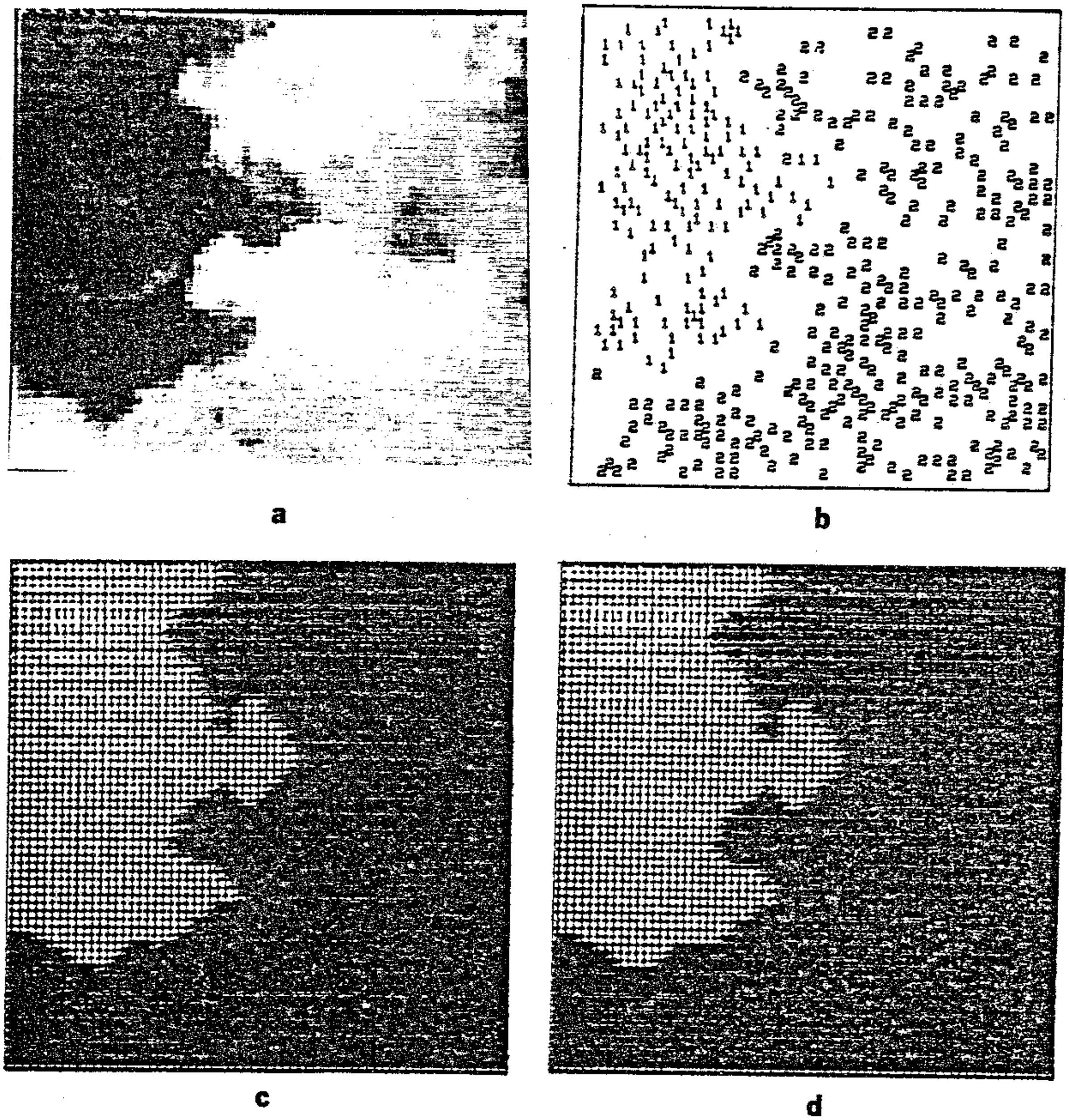


Figure 5 - (a) Original texture sample (SKY-CLOUD), (b) Classified local maxima, (c) Partitioning result by nearness procedure using the absolute value distance, and (d) Partitioning result by diffusion procedure using 4-neighbors.

For the diffusion procedure, the 4-neighbors formed the local neighborhood. Figure 5d shows the result which is very close to that in Figure 5c. The similarity depends upon a correspondence between the distance measure used in the nearness procedure and the definition of local neighborhood that is used in the diffusion procedure. That correspondence [Ro68] is

Euclidean - Alternating 4 and 8 neighborhoods
Absolute Value - 4 neighborhoods
Maximum Value - 8 neighborhoods.

Under these correspondences, the only possibility for differing results is due to the way in which ties are arbitrated. Figures 6a through 6d show the result for another example (TREE-CLOUD). Notice that the texture boundary between tree and cloud is again very acceptable.

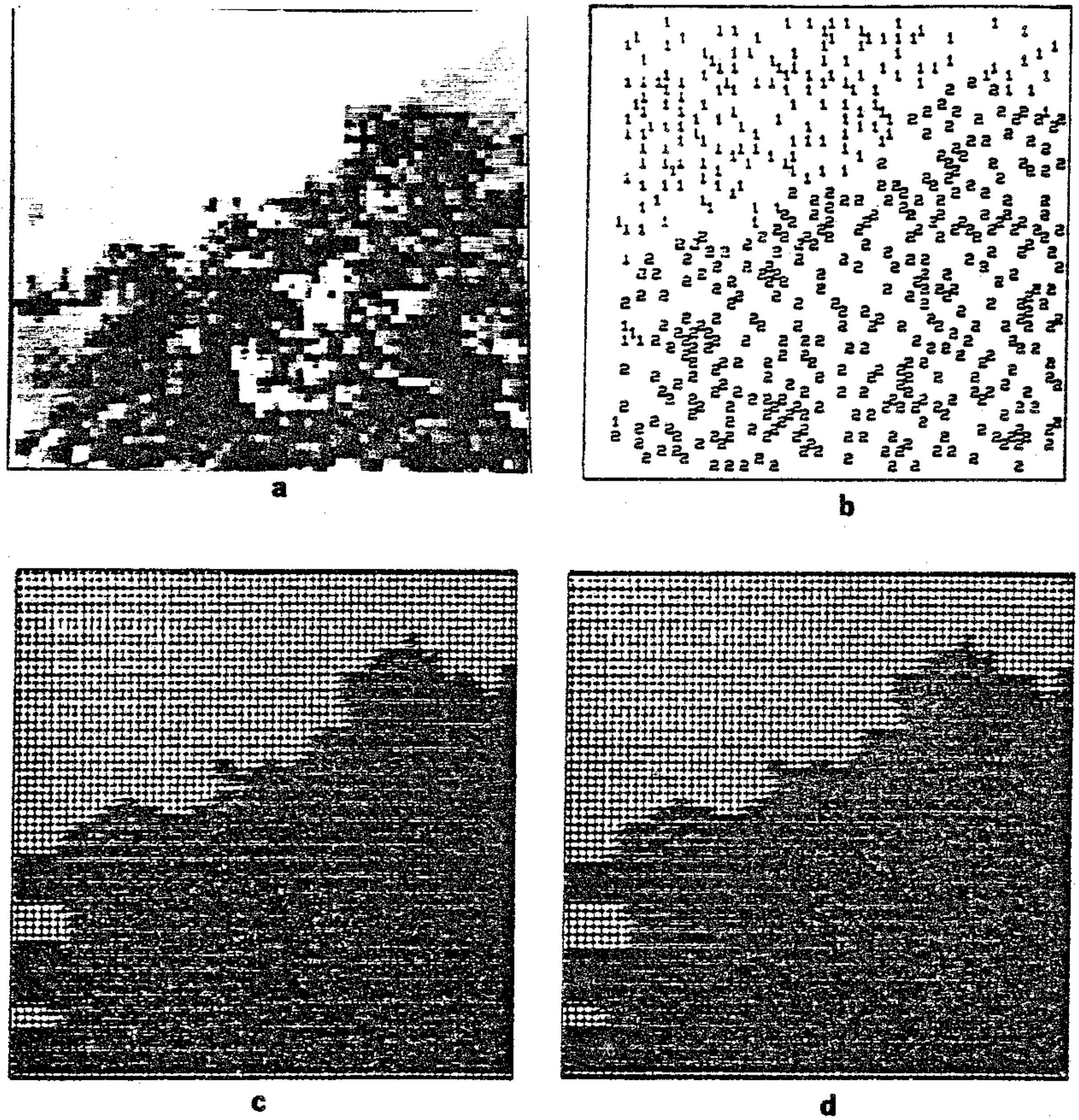


Figure 6 - (a) Another texture sample (TREE-CLOUD), (b)-(d): same as in Figure 5.

Insofar as computation time is concerned, we compared these two procedures in the 2-dimensional case. One would expect the diffusion procedure to be faster than the nearness procedure since $L \leq m$. For d=2 the number of computations required are proportional to $d_1 * d_2 * m$ for the nearness procedure, and for the diffusion procedure, the computations depend upon $d_1 * d_2 * L$, where m is the number of centers and L is the number of categorized regions. Table 1 gives the experimental computation times for two 32 x 32 samples and two 64 x 64 samples with m=87,103,424, and 492, respectively. In the nearness procedure, the average time required in the 64 x 64 case is about 16 times greater than that in the 32 x 32 case, while in the diffusion procedure the ratio is only 4. This is because in this application, the number of centers, m, grows with $d_1 * d_2$ while L remains constant.

Experiment has shown that both procedures produced satisfactory regions, and the diffusion procedure is computationally much less expensive than the nearness procedure. One reason that the diffusion algorithm behaves so well is that the centers used are local maxima, and regions are distributed evenly around the centers. As the computation progresses, the disputes on the boundary points usually occur in the final iterations, and the regions are nearly complete by that time.

Table 1 - Computation times for various samples.
(in all cases, L=2)

	nearness procedure			diffusion procedure		
	absolute value	maximum value	Eucli- dean	4-n	8-n	4-8
32x32 m=87	17 s.	15 s.	52 s.	5 s.	4 s.	4 s.
32x32 m=1Ø3	22 s.	43 s.	64 s.	4.5 s.	4.5 s.	4.5 s.
64x64 m=424	6 m.	11 m.	15 m.	2Ø s.	19 s.	19 s.
64x64 m=492	8 m.	14 m.	19 m.	18 s.	17 s.	17 s.

REFERENCES

Carlton, S.G. and O.R. Mitchell, "Image Segmentation Using Texture and Gray Level," Proc. IEEE Conference on Pattern Recognition and Image Processing, june 1977, pp.387-391.

Duda and Hart, Pattern Classification and Scene Analysis, 1973, by John Wiley and Sons, Inc.

Ehrich, R.W. and P.F. Lai, "Elements of a Structural Model of Textures," Proc. IEEE Conference on Pattern Recognition and Image Processing, June 1978, pp.319-326.

Nilsson, N.J., Learning Machines, 1965, by McGraw-Hill, Inc.

Rosenfeld, A. and J.L. Pfaltz, "Distance Functions on Digital Pictures," Pattern Recognition, Pergamon Press 1968, Vol.1, pp.31-61.

Rosenfeld and Kak, Digital Picture Processing, 1976, by Academic Press, Inc.

Strong III, J.P. and A. Rosenfeld, "A Region Coloring Technique for Scene Analysis," Communications of the ACM, Vol.16, No.4, April 1973, pp.237-246.