D47

N78-14577

BLOB: AN UNSUPERVISED CLUSTERING APPROACH TO SPATIAL PREPROCESSING OF MSS IMAGERY*

R. J. Kauth, A. P. Pentland, and G. S. Thomas**

Environmental Research Institute of Michigan
Ann Arbor, Michigan

ABSTRACT

A basic concept of MSS data processing has been developed for use in agricultural inventories; namely, to introduce spatial coordinates of each pixel into the vector description of the pixel and to use this information along with the spectral channel values in a conventional unsupervised clustering of the scene. The result is to isolate spectrally homogeneous field-like patches (called "blobs"). The spectral mean vector of a blob can be regarded as a defined feature and used in a conventional pattern recognition procedure. The benefits of use are: ease in locating training units in imagery; data compression of from 10 to 30 depending on the application; reduction of scanner noise and consequently potential improvements in classification/proportion estimation performances.

1. INTRODUCTION

For processing of MSS data, improved methods of extraction of training data, of data compression, and of classification/proportion estimation are needed. A basic technique which creates opportunity for improvements in all these aspects of MSS data processing has been developed. The basic technique is to incorporate the rudimentary concept of spatial nearness into the preprocessing steps in a simple and natural way, and to extract, as features, spatially homogeneous units from the MSS image. Incorporated into a complete processing system for MSS data the technique is helpful in all of the above mentioned ways.

2. DESCRIPTION OF BLOB

The basic technique is incorporated in an algorithm called BLOB [1]. It consists of augmenting each pixel vector with two additional components which describe the pixel's spatial coordinates (i.e., the line number and the point number). These augmented pixel vectors are used in a conventional clustering algorithm [2] to accomplish "spectral-spatial" clustering. Spatially and spectrally similar pixels are grouped together in each cluster (called "blobs"). In an agricultural scene the result is to build field-like structures, as shown in Figure 1. Figure 1 is a typical unsupervised blob map of an agricultural scene.

Ten grey levels were chosen for this display and these are assigned to the blobs in an arbitrary manner. The most notable characteristics of BLOB which can be observed in this figure are the use of spatial information to smooth over noise in individual pixels, and the treatment of boundary pixels. These will be discussed in more detail in following sections.

*This work was supported by the National Aeronautics and Space Administration through their Earth Observations Division, Johnson Space Center, Houston, Texas, under Contract NAS9-14988.

**The authors are members of the Information Systems & Analysis Department, Infrared & Optics Division, ERIM, Ann Arbor, Michigan.

[NOTE: Extensive preprocessing steps, in addition to blobbing, are routinely incorporated in ERIM's processing of Landsat MSS data and are in part responsible for the overall quality of the results presented [3]. Briefly, these steps include: screening, to identify clouds, cloud shadow, water, and bad (excessively noisy) pixels and to calculate diagnostic features for later use in atmospheric effects correction; satellite correction, to normalize Landsat 1 data to be commensurate with Landsat 2; solar zenith angle correction, to normalize data collected at various zenith angles to a single zenith angle; atmosphere (haze) effects correction, to normalize data collected under a variety of haze conditions to a single standard haze condition using the XSTAR algorithm [4]; Tasselled Cap transform [5], producing a set of linear combination features which emphasize physical characteristics of the data, primarily soil brightness and green vegetative development; and multitemporal augmentation, combining registered data from passes at two or more times.]

2.1 FEATURE EXTRACTION

The important information about each blob is recorded and comprises a compressed feature description of the scene. Two outputs are produced: a pixel output tape which defines for each pixel which blob it belongs to, and a blob feature output tape containing a redefined set of features for each blob. These features include the mean vector of all the pixels contained in the blob (or alternatively, only the interior pixels, as discussed in the next section) and the number of points in each blob. The blob mean vector includes the line mean and the point mean which serve as a description of the position of the blob in the scene.

2.2 EXTRACTION OF TRAINING UNITS

For purposes of training a classifier one would like to use only pixels which are "pure" examples of the classes of materials being trained. Thus, it is desirable for purposes of training to strip away boundary pixels, i.e., pixels of a given blob which are adjacent to another blob. The pixels remaining after stripping are nearly always pure pixels, and constitute "stripped blobs".

Figure 2 shows a segment of Landsat data which has been subjected to multitemporal spectral-spatial clustering, using four dates as follows:

23 October 1973 9 May 1974 27 May 1974 14 June 1974.

In Figure 2, the ground-truth field lines (and field numbers), and an encompassing rectangle are overlaid on the blob presentation. In this presentation the field center pixels are left blank while near-boundary pixels are shown as asterisks. This results in some fields being entirely missing, since they were formed into such small initial blobs. This is no particular drawback since such small or ragged fields aren't likely to make very good candidates for training fields anyway. Some large fields are converted into two blobs and so would be used as two separate training fields.

In several cases throughout the scene blobs run across field boundaries. We have identified 13 such cases. In 10 out of 13 cases the adjacent field ID's match. In two of the cases the infringement of the blob across field boundaries amounts to only a few pixels. One case is unexplained. This is fairly typical of our experience, namely that blobs seldom cross the boundary between two spectrally distinct classes.

2.3 TRAINING AND CLASSIFICATION

Having identified pixels which are suitable for training, various alternatives are available. Training can be carried out using the individual pixels themselves as training samples. In this case one would follow with pixel-by-pixel classification. A more practical procedure, and one of the objectives of developing BLOB in the first place, is to train using the blob feature vectors as training samples, and follow with blob classification. In blob classification we classify each blob according to its feature vector. Then we classify each pixel to the same class as the blob to which the pixel belongs, in order to produce a classification map. If we wish to make only a proportion estimate of each class we accumulate the total number of pixels in each blob belonging to each class, so that it is not necessary to process the individual pixels again.

Blob classification has the obvious advantage of data compression. It also carries whatever advantages or disadvantages may have accrued from the spatial preprocessing. To help understand what these may be we describe the algorithm in more detail in the next section.

3. DETAILS OF ALGORITHM

In the following paragraphs we describe the details of the BLOB algorithm and discuss its treatment of interior pixels, boundary pixels and small fields.

3.1 MATHEMATICAL DESCRIPTION

The clustering algorithm upon which BLOB is based was developed by A. P. Pentland [2]. The basic steps in clustering are as follows:

1. For each pixel decide which existing cluster it is closest to; the distance measure used is

$$d_i^2 = (x-\bar{x}_i)^T D_i^{-1}(x-\bar{x}_i) + \frac{n}{2} \ln |D_i|$$

where

x is a column vector of dimension n, the pixel vector

$\bar{x}_i$ is the sample mean of the pixels already included in an existing cluster, i.

D_i is a diagonal matrix of n sample variances of the pixels already included in an existing cluster, i.

2. If $d_i^2 > \tau$ for every existing cluster decide that the pixel belongs to none of the existing clusters, and start a new cluster with mean $\bar{x}_j = x$ and D_j = default.

3. Whenever a pixel is classified to cluster, i, update the statistics of this cluster by recursively computing the mean, $\bar{x}_i$, and the variances, D_i, including the current pixel.

4. A variety of procedures have been utilized to establish the initial clusters, including randomly selecting a number of pixels from the scene to form starting clusters, but usually the algorithm is run successfully with no seeding at all.

In the BLOB algorithm the distance measure, d^2, is modified by including additional components relating to the spatial position of the pixel. We have in many cases found it satisfactory to use a fixed covariance matrix common to all of the clusters (i.e., the "blobs"), so that it is not necessary to update the variances or to include the $\ln|\ |$ in the distance measure. In the simplest implementation the distance measure then becomes,

$$d_i^2 = w(x-\bar{x}_i)^T M^{-1}(x-\bar{x}_i) + \frac{(\ell-\ell_i)^2}{v_\ell} + \frac{(p-\bar{p}_i)^2}{v_p}$$

where

- x is the spectral pixel vector
- ℓ is the pixel line number
- p is the pixel point number within the line
- M is a fixed $n \times n$ spectral covariance matrix
- $\bar{x}_i$ is the spectral mean of blob, i.
- $\bar{\ell}_i$ is the line mean of blob, i.
- $\bar{p}_i$ is the point mean of blob, i.
- w is a relative weight between the spectral and spatial contribution to distance.

Motivated by a desire to make the structure of the blobs match our notion of the East-West/North-South rectangular structure of the fields of an agricultural scene we have incorporated modifications and combinations of modifications of the spatial distance measure, as follows:

1. Line and point numbers are replaced by East and North coordinate values. (The mean positions of the blobs are then transformed back to line and point coordinates, for presentation to the user, for locating the position of the blob in imagery.)

2. Two additional varieties of spatial distance measure have been tried using these revised line and point coordinates, as follows:

   a. $$\text{spatial distance} = \sqrt{\frac{(\ell-\bar{\ell})^4}{V_\ell^2} + \frac{(p-\bar{p})^4}{V^2}}$$

   b. $$\text{spatial distance} = \max\left\{\frac{(\ell-\bar{\ell})^2}{V_\ell}, \frac{(p-\bar{p})^2}{V_p}\right\}$$

Case a. produces spatial iso-distance contours which form "super-ellipses". Case b. corresponds to spatial iso-distance contours which form rectangles.

Our experience is that these variations make no practical difference since the blobs tend to shoulder together to fill the space available along natural (spectral) boundaries in any case (see again Figure 1). Furthermore, in cases where a large naturally homogeneous area is broken into two or more blobs the shape of the boundary is determined by the process of sequentially adding pixels to blobs and updating the mean, rather than by the detailed form of the distance function (see the top central area of Figure 2 for an example case). In current practice* we use the transformation 1. above, and the maximum squared distance measure, 2.b. above. Also, in current practice, the spectral features which are used are the Tasselled Cap "brightness" and "green" features from one or several times; since these are nearly uncorrelated to one another only the diagonal terms of the covariance matrix are used.

*[NOTE: The current working version of BLOB has been completely revised and reprogrammed by W. Richardson, including in particular the introduction of the maximum squared distance measure, and including numerous techniques to speed up the calculation by a factor of 5 or 6. At present BLOB requires about 33 msec per pixel for 9 channel data running on ERIM's 7094 computer, or .5 msec per pixel on the University of Michigan's Amdahl 470/V computer, both times being comparable to conventional quadratic classification rules. Running time is not a strong function of the number of channels being used.]

3.2 BLOB TREATMENT OF INTERIOR PIXELS

The net effect of BLOB is to smooth out the identifications of within field pixels. Thus, if an isolated pixel in the middle of a field is spectrally somewhat different than its neighbors it will nevertheless be included with them in the same blob. If the pixel is much different, spectrally, than its neighbors it will be identified as a separate (one pixel) blob, or will be included with some nearby blob of similar spectral character. The net effect is to reduce the "salt and pepper" appearance of conventional classification maps, or of conventional spectral clustering maps without consideration of spatial information.

3.3 BLOB TREATMENT OF BOUNDARY PIXELS

It is informative to consider how BLOB processes mixed pixels at the boundary of two fields. In conventional classification such pixels may masquerade as a third class of material which may be located geographically some distance away in the scene. Figure 4 illustrates this idea. In Figure 4(a) a mixed pixel straddles a boundary between two classes, A and B. The signal "x" from this pixel is typical of the signals which would come from a Class C, and if no further information is given the pixel will be identified as Class C. In Figure 4(c) the joint density of signal and position are shown and it is clear that the pixel (x',d') will be classified as Class A or as Class B, but certainly not as Class C. (Pixels along boundaries will tend thus to be equally divided between the adjacent classes.)

Of course, if the Class C field is geographically nearby, then the pixel will still be misclassified, but such instances will be few in number.

3.4 BLOB BEHAVIOR FOR SMALL FIELDS

Consider the behavior of BLOB in case there are small fields, for example strip-fallow farming common in the northern Great Plains of the U.S. In this case blob 'A' may cover several wheat strips, and blob 'B' may cover several overlapping fallow strips. (This is possible because BLOB as presently implemented does not take any account of contiguity of classes -- only of nearness.) The positions of these blobs might be quite close together, and in particular the mean of A may fall in a fallow strip while the mean of B may fall in a wheat strip. In such cases it is almost certain that no pure pixels will be available for training, but it is still desirable to be able to visually inspect the coverage of each blob, to notice that it occurs in a strip fallow situation, etc. Improved methods of display are being developed but much remains to be done in this area.

As the size of fields gets smaller the number of blobs may increase until each blob is composed of a few isolated pixels which are spectrally alike; in other words, BLOB degenerates gradually to a pixel by pixel processor.

4. EVALUATION OF BLOB PERFORMANCE

Evaluation ought to imply a careful definition of measures of performance, of the algorithms being tested, and of the range of conditions of measurement. In this sense there has been no adequate evaluation of BLOB. BLOB is a heuristically evolving procedure, and testing has mainly been for the purpose of guiding our insight.

BLOB should be evaluated for performance in several areas: ease in extraction of training statistics; classification accuracy and/or proportion estimation accuracy; and data compression. We have adopted BLOB in large part because of the ease of training and the compression of data.

4.1 EXTRACTION OF GROUND TRUTH

In current practice blobs are labelled for training purposes by visual comparison between a ground truth map or image and a line printer blob map. Usually the line printer map is a map of stripped blobs, as in Figure 3. The blobs are numbered with a special modulo 50 symbol set. A second map keeps track of which set of 50 a given blob is in. Sometimes an auxiliary printout of unstripped blobs such as Figure 1 is used to make the visual association between image and line printer map easier. Even with such primitive arrangements the time to obtain wall to wall ground truth is significantly less than for previous competing methods.

4.2 DATA COMPRESSION

Data compression is particularly important because it allows us to concentrate on analyzing data dependencies over a wide range of ancillary conditions. Depending on the application, BLOB provides a data compression factor of 5-30. For example, suppose that 7-channel data are collected from a spacecraft multispectral scanner and that for certain applications, it appears suitable to blob with an on-board processor and transmit the blob mean spectrum for each blob and the blob identification number for each pixel. Suppose on average there are 30 pixels per blob. Then the average number of channels per pixel is 1 + 7/30 for a net data compression of approximately 6.

Much of our current work is concerned with proportion estimation rather than the production of a class map, and with studies of the signatures of classes under various conditions. For these purposes the net compression factor is closer to 30, since only the blob spectral mean and the number of pixels per blob need be retained.

4.3 CLASSIFICATION/PROPORTION ESTIMATION

Qualitatively we do not expect the classification/proportion estimation performance of BLOB to be any worse than pixel-by-pixel classification; in fact because of its rational treatment of boundary pixels BLOB should perform better in these categories. However, to date, no valid comparative tests have been made of blob training, classification and proportion estimation vs. a conventional pixel-by-pixel classification approach. Such tests are planned for the immediate future.

5. SUMMARY

The BLOB algorithm as presently implemented is simple in concept and in execution. We believe that it exploits most of the spatial information available in agricultural scenes of Landsat MSS data. For other classes of scenes or for higher resolution data additional sophistication may be warranted, such as the inclusion of textural features in the pixel vector, or the addition of the capability to annex (join together) adjacent blobs of similar spectral character.

The algorithm has been discussed with respect to three main functions; training extraction, classification/proportion estimation and data compression. We believe it has significant value in the first and the last of those categories. For classification performance it appears no worse than pixel by pixel, but this is unproved.

The greatest potential for the technique remains to be exploited, namely its use in an interactive mini-computer environment. We can envision training pixels being extracted from complex scenes, for example marshland or forest, merely by the analyst indicating a single pixel. The display then would present for inspection the entire blob to which that pixel belongs. The computer could further cluster blobs spectrally and the analyst could indicate which blobs to merge into the same class.

REFERENCES

1. Kauth, R. J. and G. S. Thomas, System for Analysis of Landsat Agricultural Data: Automatic Computer-Assisted Proportion Estimation of Local Areas, Report No. 109600-67-F, Environmental Research Institute of Michigan, Ann Arbor, Michigan, May 1976.

2. Pentland, Alex P., The Use of Unsupervised Clustering as a Classifier for LACIE MSS Data, Technical Memorandum 109600-39-R, Environmental Research Institute of Michigan, Ann Arbor, Michigan, October 1977.

3. Information Systems & Analysis Department, Partitioning Requirements for Signature Extension, Technical Memorandum 122700-13-R, Environmental Research Institute of Michigan, Ann Arbor, Michigan, January 1977.

4. Lambeck, P., W. Malila, and J. Gleason, Signature Extension Preprocessing Algorithm Development, Task 2, First Quarterly Report, No. 122700-5-L, Environmental Research Institute of Michigan, Ann Arbor, Michigan, September 1976.

5. Kauth, R. J. and G. S. Thomas, "The Tasselled Cap -- A Graphic Description of the Spectral-Temporal Development of Agricultural Crops as Seen by Landsat", Symposium Proceedings on Machine Processing of Remotely Sensed Data, Purdue University, W. Lafayette, Indiana, June 29 - July 1, 1976.

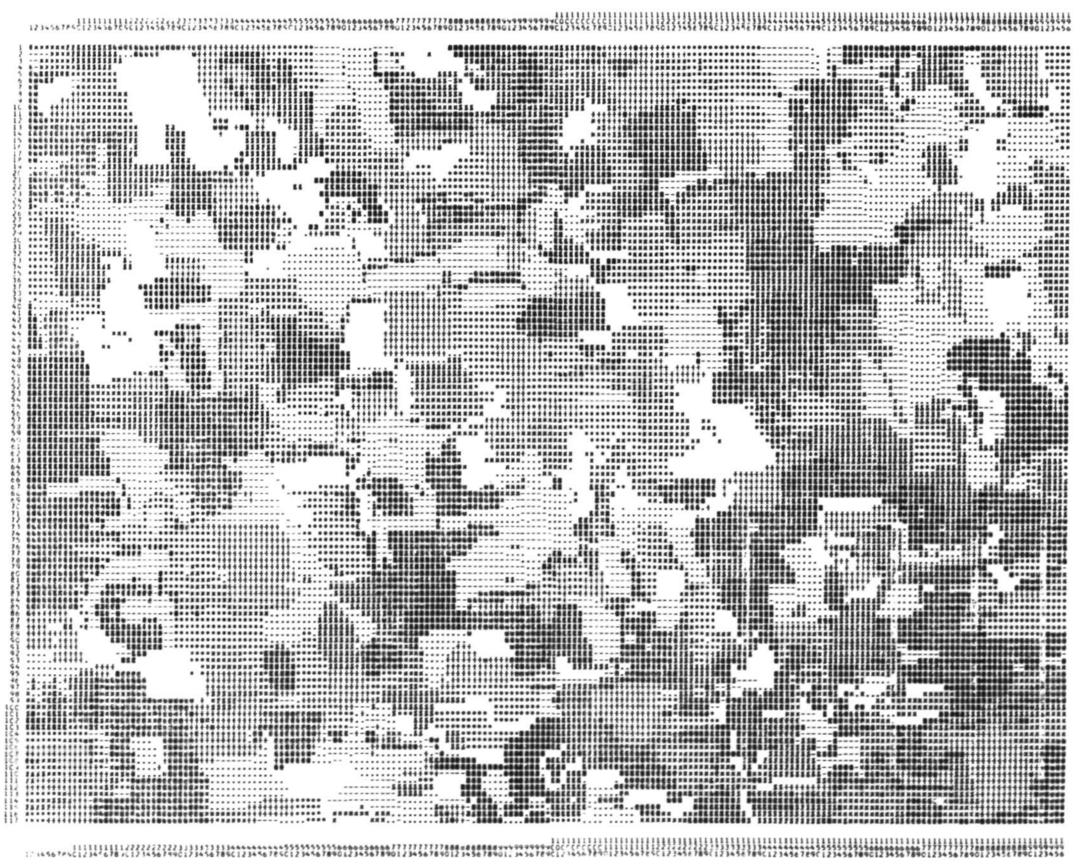

FIGURE 1. MULTITEMPORAL-SPECTRAL-SPATIAL BLOB MAP OF A 5 x 6 SEGMENT IN KANSAS. The dates used for blobing are 7 Nov 75, 6 May 76, 1 Jun 76.

ORIGINAL PAGE IS
OF POOR QUALITY

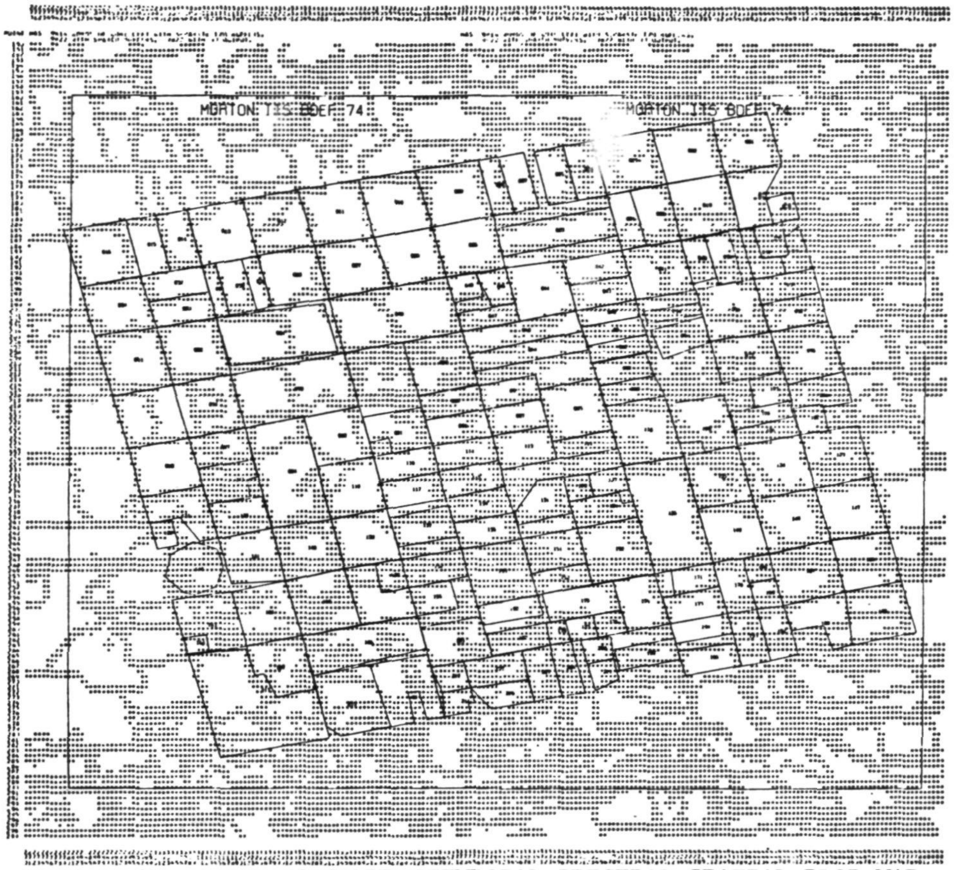

FIGURE 2. MORTON ITS MULTITEMPORAL SPECTRAL SPATIAL BLOB MAP

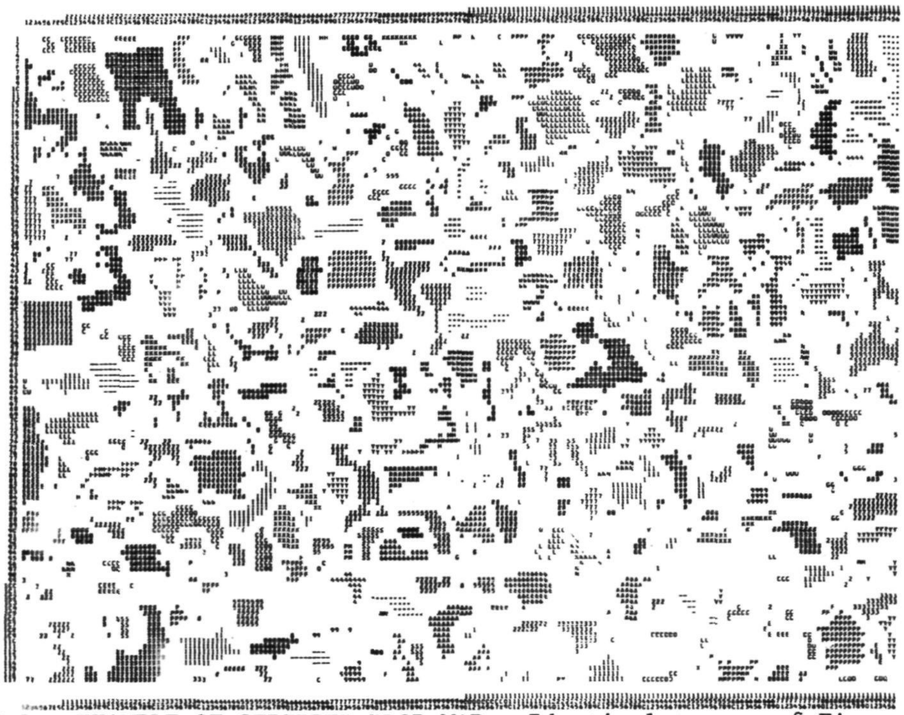

FIGURE 3. EXAMPLE OF STRIPPED BLOB MAP. Identical to map of Figure 1, except that pixels on boundary between blobs have been changed.

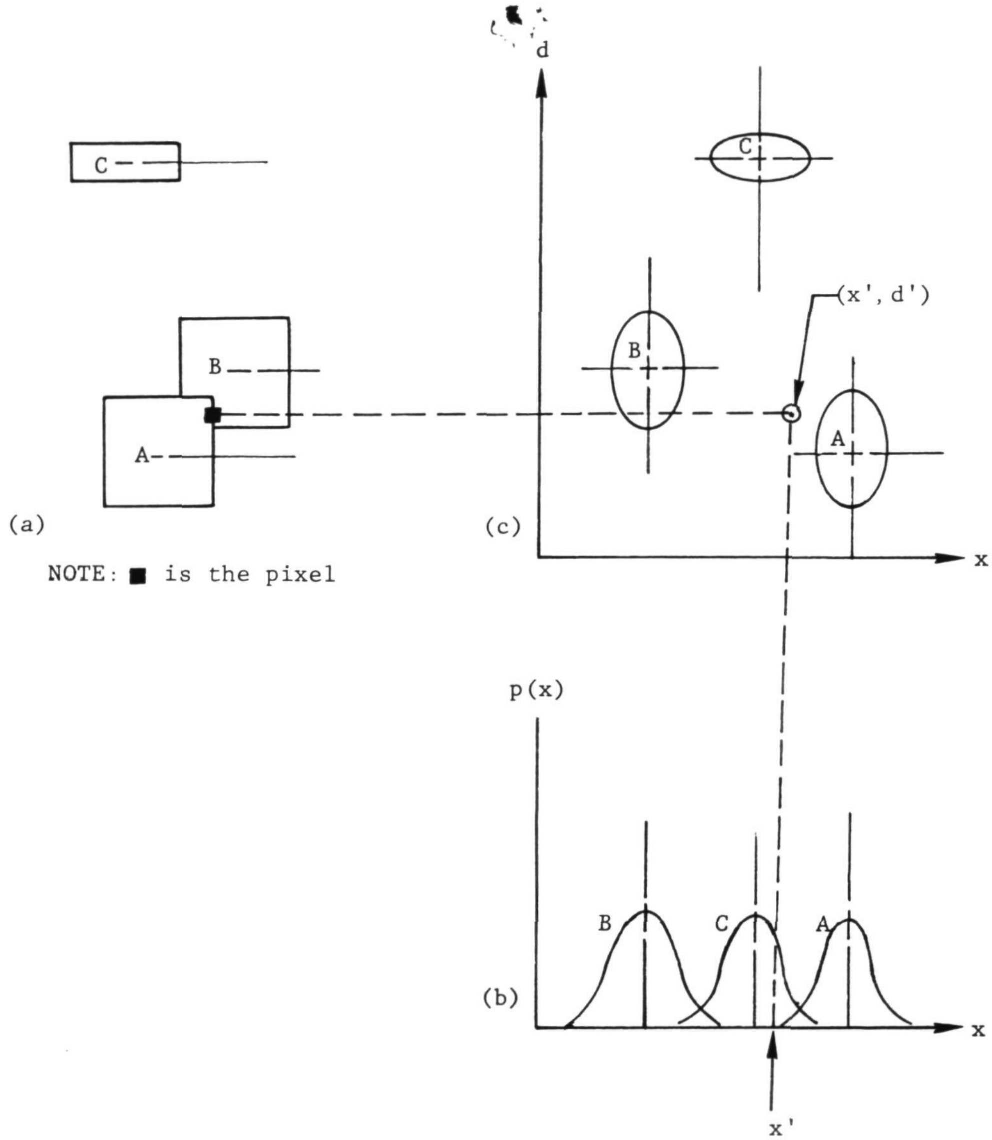

FIGURE 4. ILLUSTRATION OF BLOB TREATMENT OF BOUNDARY PIXELS

(a) Shows a pixel on the boundary of fields of classes A and B, with the nearest field of class C some distance away.

(b) Hypothetical probability density functions of the classes A, B, and C, with the pixel value, x', falling between A,B and in the range of class C.

(c) The hypothetical joint probability density of (vertical) distance, d, and pixel value, x. The pixel in question is represented by the point (x',d').