LARS Information Note 122969
Purdue University

G. Santner

R

A Cluster Approach to Finding Spatial Boundaries in Multispectral Imagery

by

Arthur G. Wacker

Introduction

The term multispectral imagery (or simply multispectral data) is used to describe pictorial data where two or more superimposed images of a scene,as it appears when viewed through different spectral "windows," are available for processing. Multispectral imagery is of paramount importance in the remote sensing field since most of the data obtained in remote sensing is of this type. At LARS most of the multispectral imagery available originates from a multispectral scanner which is simply an airborne instrument designed for the express purpose of obtaining such images.[1] Usually because of computational and data handling problems involved, multispectral images are processed in digital form.

The researcher in remote sensing is generally interested in using multispectral images to classify regions of a scene into categories of interest. Regardless of the classification procedure used, the researcher is almost immediately faced with the problem of delineating spatial boundaries separating the region containing the classes of interest. For convenience such regions will be referred to as fields. For example, in a supervised learning situation, such as LARSYSAA,[2, 3] it is necessary to obtain training samples for each of the classes of interest. Before this can be done fields must be delineated, at least roughly. Similarily, in evaluating the performance of any categorizor (supervised or non-supervised) test fields whose classification is known must be delineated. Generally boundary information is only contained implicitly in the data. That is, boundaries are indicated only by changes in the data values and no data element is specifically identified as a boundary element.

Presently at LARS the task of locating boundaries in multispectral data is accomplished by using gray-scale printouts in conjunction with aerial photography. These aids enable the researcher to orient himself with respect to the data and to select training and test fields. Locating boundaries from gray-scale printouts is complicated by the fact that in general boundaries that show up clearly in one channel (i.e., spectral band) may not show up at all in another channel. Thus the user is faced with the rather unweildly task of looking simultaneously at several gray-scale printouts. It is clear that an automatic digital technique, which utilizes the multispectral nature of the data to extract boundary information, would be of considerable value to the researcher in remote sensing.

Locating boundaries is important for reasons other than the selection of training and test samples. The possibility that data compression may be achieved for pictorial data by utilizing boundaries, or equal intensity contours (essentially boundaries), has received considerable attention in the television industry. The problem under consideration here is essentially the multi-dimensional version of this problem. Certainly data compression possibilities are greater in the multi-dimensional than the one dimensional case. A use closely allied to data compression is information storage and retrieval. Thus for example in classifying a scene, rather than storing the classification of each data point it may be adequate to store the boundaries of each field and its classification. Another application in which boundary finding algorithms may be of considerable future importance is in conjunction with the recently developed "per field classification" scheme.[6] The basic idea involved here is that a whole field is classified at once rather than classifying each data point individually. In this method a field is assigned to the class whose distribution is "nearest" (in some sense) to the emperical field distribution. This method appears to provide better "field classification accuracy" than can be obtained on a point by point basis, at least for some data sets. Obviously boundary information must be available to carry out a per field classification. It should be pointed out that some of the above applications, in particular the per field classification scheme, require that the boundaries delineating the fields must be closed boundaries.

The algorithm described in this section is one approach to the problem of locating boundaries in multispectral, digitized pictorial data. Many boundary finding algorithms exist for one dimensional data; it appears, however, that boundary finding algorithms for the multispectral case are essentially nonexistent. The basic principle upon which this boundary finding algorithm depends is clustering. A survey of the methods and techniques of clustering may be found in Ball.[4] Since clustering techniques are largely heuristic the boundary finding algorithm is also largely heuristic. It is anticipated that the algorithm presented should work reasonably well for data sets that are approximately Gaussian and where Euclidean distance is a good metric in the observation space. In its present form the boundaries found by the algorithm are not necessarily closed.

General Approach

Similarity between vectors is of basic importance in attempting to find boundaries. This is readily evident since the existence of a boundary is a consequence of the dissimilarity of vectors from neighboring image resolution element (IRE). Consequently the following simple apprach suggests itself. Consider an IRE from the ith row and jth column and the associated L dimensional vector $X_{ij} = (X_{1ij}, X_{2ij}, \ldots, X_{Lij})$. Let Y be a vector from a neighboring IRE. If X_{ij} and Y are sufficiently alike, according to some similarity measure, then no boundary exists between the corresponding IRE's. If they are not sufficiently alike then a boundary does exist. To implement such a scheme it is only necessary to define a suitable measure of similarity and establish a "similarity threshold". Vectors from adjacent IRE's, whose measure of similarity falls below the "similarity threshold," are considered to be boundary IRE's.

A disadvantage of the above approach is that it is too local and makes no use of the fact that for typical sampling rates boundaries usually have some spatial extent (at least several IRE's). In effect the above approach is analogous to viewing a portion of a photograph through a very small window and attempting to decide if a boundary in the photograph passes through the window. Graininess in the photo (noise) might easily be mistaken for a boundary, while a rather gradual boundary would not be discernible. It is obvious that a larger window would enable the boundary to be more clearly identified. Eventually increasing the window size further does not aid appreciably (if at all) in discerning a boundary. This is because a boundary is essentially a local, or perhaps more correctly, a quasi-local property. It would appear that the proper approach to boundary finding should be a quasi-local approach.

The above considerations have led to a clustering approach to finding boundaries. In this approach boundaries are determined not on the basis of the similarity of individual vectors, but rather on the basis of the similarity of small groups of vectors, where all vectors under consideration have been drawn from a reasonably small region of the scene. The grouping of the vectors is determined by clustering the vectors in the observation space. The details of the complete procedure, which is referred to as the Boundary Finding Algorithm, is discussed in the ensuing section.

Description of the Boundary Finding Algorithm

In the algorithm to be described the scene under consideration is partitioned into square regions called *Boundary Cells*. Each of the Boundary Cells contains a moderate number of IRE's, and the union of the Boundary Cells represents essentially the whole scene. Boundaries are determined separately for each Boundary Cell and the union of these boundaries yields the boundaries for the whole scene. It is convenient to discuss the Boundary Finding Algorithm in two parts. The first part is concerned with the clustering procedure itself, while the second part is concerned with the method whereby the results of the clustering procedure are used to establish which of the IRE's are boundary IRE's. For convenience the first part of the Boundary Finding Algorithm will be referred to as the *Clustering Algorithm* and the second part as the *Edge Finding Algorithm*.

To describe the Clustering and Edge Finding Algorithms we focus our attention on an arbitrary Boundary Cell. We define a Clustering Cell to be a square area, centered on, but slightly larger than a Boundary Cell. Figure 1 depicts the general relationship. The vectors associated with IRE's in the Clustering Cell are clustered in the observation space, that is, a "natural" grouping for these vectors is found in the observation space. Two input parameters are of prime importance in controlling the clustering procedure. The first of these is the maximum number M_m of modes (groups) permitted, and the second is a threshold T. The algorithm first finds the "best" grouping of the vectors into M_m modes. A pairwise measure of separatin is then computed for each pair of modes. Let S_{ij} be this measure for modes i and j. If $S_{ij} \geq T$ for all i, j = 1, 2, - -, M; ij then the M_m modes are taken to be distinct and the clustering procedure is terminated. If one or more $S_{ij} < T$ then all modes are not distinct and the two modes corresponding to the lowest S_{ij} are merged yielding $M_m - 1$ modes. The clustering procedure is then repeated with $M_m - 1$ modes. In this manner the number of modes is successively reduced until $M(1 \leq M \leq M_m)$ distinct modes are found. Thus the clustering procedure establishes how many groups of vectors there are, up to some specified maximum M_m, and assigns each vector in the Clustering Cell to one of these groups. Consequently after a Clustering Cell has been processed by the Clustering Algorithm, there is available a spatial array of the type depicted in Figure 2a. Here each number in the array represents the group to which the corresponding IRE has been assigned.

The manner in which a set of vectors is clustered is outlined below. The procedure is essentially that suggested by Swain and Fu[5] with some modifications.

1) Intialization

Let $X_1, X_2, \ldots, X_N$ be N L-dimensional vectors from a Clustering Cell. If the maximum number of modes permitted is M_m, the M_m initial mode centers are generated as Follows:

Compute the sample mean of the N vectors according to

$$\mu = \frac{1}{N} \sum_{i=1}^{N} X_i$$

and the sample variance for each dimension

$$\sigma_j^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_{ij} - m_j)^2 \qquad j = 1, 2, \ldots, L$$

Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_L)$ and $\mu = (\mu_1, \mu_2, \ldots, \mu_L)$. Consider the real line intervals $\gamma_i = [\mu_i - \sigma_i, \mu_i + \sigma_i]$, $i = 1, 2, \ldots, L$. The cartesian product $\gamma_1 X \gamma_2 X \ldots X \gamma_L$ defines a rectangular parallelepiped in the observation space which should contain most of the vectors from the Clustering Cell. The M_m initial mode centers are chosen to be uniformly spaced along a diagonal of this rectangular parallelepiped. Accordingly the mode centre for the kth mode is

$$m_k = \mu + \sigma\left(\frac{2(k-1)}{(M_m-1)} - 1\right) \qquad k = 1, 2, \ldots, M_m$$

Initially none of the vectors are assigned to any mode.

2) Mode Assignment

Determine the Euclidean distance from each vector to each mode center. Assign each vector to the mode with the nearest mode center.

3) Mode Migration

If step 2 did not change the mode assignment of any of the N vectors go to step 4; otherwise, replace the old mode centers by the means of the vector clusters resulting from step 2, and then go to step 2.

4) Distinctness Test

A new set of modes has been tenatively established. Test these modes for distinctness using the method suggested by Swain and Fu.[5] This test assumes that the modes are Gaussian and that the variables are independent. If all the modes are distinct clustering is complete. If all the modes are not distinct merge the two least distinct modes. Suppose modes i and j have mode centers m_i and m_j and contain n_i and n_j vectors respectively, are the least distinct modes. The merged mode center is given by

$$m = \frac{n_i m_i + n_j m_j}{n_i + n_j}$$

If after merger the number of modes is greater than one go to step 2; if only one mode remains clustering is complete.

The Edge Finding Algorithm is responsible for processing the Clustered Array (i.e., Figure 2a) generated by the clustering algorithm and detecting which of the IRE's are boundary IRE's. To do this the Clustered Array is scanned in both the vertical and horizontal direction to establish which of the IRE's qualify as boundary IRE's according to the definition of a boundary IRE outlined below. Let C_{ij} be the mode to which $(\text{IRE})_{ij}$ (i.e., the IRE in the ith row and jth column of the Clustering Cell) has been assigned; that is, C_{ij} is the number appearing in the ith row and jth column of the Clustered Array. Let k be an input parameter to the Edge Finding Algorithm which we will refer to as the Correlation Distance. A vertical boundary exists between $(\text{IRE})_{ij}$ and $(\text{IRE})_{i,j+1}$ in case $(\text{IRE})_{ij}$ and the first (k-1) IRE's left of $(\text{IRE})_{ij}$ all belong to one mode, and $(\text{IRE})_{i,j+1}$ and the first (k-1) IRE's to the right of $(\text{IRE})_{i,j+1}$ all belong to another mode. If this condition is satisfied then $(\text{IRE})_{ij}$ and $(\text{IRE})_{i,j+1}$ are vertical boundary IRE's. Horizontal boundary IRE's are found in an analogous manner by scanning the clustered array in the vertical direction. On the line printer output the symbol I is used to designate a vertical boundary IRE, a minus sign designates a horizontal boundary IRE, while an IRE that qualifies as both a horizontal and vertical boundary is designated by an asterisk.

The main reason for introducing the correlation distance k is to insure that if an IRE is assigned to a different mode then its 8 immediate neighbors, then none of these IRE's are necessarily interpreted as a boundary IRE. Usually such IRE's have been assigned to the wrong mode by the clustering algorithm although occasionally they may represent legitimate boundary IRE's. The need for some sort of spatial smoothing becomes evident if one considers that in the absence of any smoothing (i.e., k=1) a single isolated IRE would result in five boundary IRE's. The clutter produced by a relatively few isolated IRE's would be intolerable.

A disadvantage of using a correlation distance is that the boundaries of fields narrower than k IRE's will not be detected. To remove this undesirable condition a narrow field detector has been added to the Edge Finding Algorithm. The narrow field detector simply relaxes slightly the condition an IRE must satisfy to qualify as a boundary IRE.

To describe the operation of the narrow field detector assume that in scanning the ith row of the Clustered Array one encounters a sequence of $k_1 < k$ adjacent IRE's all assigned to the same mode. This sequence of k_1 IRE's will be considered to originate from a narrow vertical field in case there exists a "similar" sequence of k_1 IRE's in either the (i-1)st or (i+1)st row (or both). For the purposes of this description two sequences are "similar" if the IRE's in one sequence are shifted by not more than one column with respect to the IRE's in the other sequence. All elements from a narrow vertical field are designated by the symbol X on the line printer output. Narrow horizontal fields are detected in an anagous way by scanning the clustered array in the vertical direction. Figure 2b shows the boundaries for the clustered array of Figure 2a.

It is apparent that because of the manner in which a boundary element is defined it is not possible to decide whether any IRE within one correlation distance of the edge of a Clustering Unit is a boundary IRE. This means that if each IRE is to be considered as a possible boundary IRE, then adjacent Clustering Cells must be chosen to provide a certain amount of overlap.

Specifically, they must overlap by at least 2k IRE's. Also note that this means it is not possible to decide if an IRE is a boundary IRE if the IRE in question is nearer the edge of a scene than k IRE's.

Performance and Determination of Parameters for the Boundary Finding Algorithm

The parameters that must effectively be specified for the Boundary Finding Algorithm are as follows.

1) Clustering Cell Size (w_c x w_c IRE's)

2) Boundary Cell Size (w_b x w_b IRE's)

3) Maximum number of modes permitted (M_m)

4) Number of Channels (L)

5) Correlation Distance (k)

6) Threshold (T)

Two factors are important in establishing "best" parameter values. One is the processing time, and the second is the performance of the Boundary Finding Algorithm. The latter factor is perhaps the more important factor, but at present no method exists to quantitatively evaluate performance. Thus about all that can be done is to visually examine the output from the Boundary Finding Algorithm and make a relative, qualitative, evaluation concerning the algorithms performance. For different combinations of parameters, where performance is judged to be essentially the same, processing time becomes the prime criterion.

Experimentally it is readily verified that the vast majority of the processing time is accounted for by the Clustering Algorithm. Consequently the processing time is intimately linked to the time required to cluster the vectors in a Clustering Cell. Unfortunately this time is a random variable because it depends on the number of iterations required to cluster a set of vectors. In determining the boundaries for a reasonable amount of data many

Clustering Cells are processed. It is therefore reasonable to talk about an average number of iterations per Clustering Cell and consequently an average processing time per IRE. In practice the average processing time can be approximately determined by measuring the time required to find the boundaries for a section of data which is large enough to contain many Clustering Cells.

The large number of parameters involved in the Boundary Finding Algorithm makes it difficult to carry out a completely comprehensive investigation regarding the effect of all of the parameters for a wide range of parameter values. Fortunately some of the parameters affect performance in a fairly predictable manner; others are relatively independent and can therefore be set rather easily. Some combinations of parameters lead to exhorbitant processing times with little apparent benefit and can be readily set from a practical point of view. More specifically, a threshold value of about 1 or slightly lower seems to be a good theoretical choice. The validity of this choice is born out in practice for the multispectral scanner data and consequently most of the results presented will use a threshold value of 1. In the interests of minimizing processing time the correlation distance k should be chosen as small as is practical. For the multispectral scanner data a choice of k=2 usually provides adequate smoothing of the Clustered Array. Consequently this value is used for all the results presented. Finally the Boundary Cell will always be assumed to be 2k IRE's per side smaller than the Clustering Cell. For a given size Clustering Cell this is the largest permissible size for the Boundary Cell. Hence, this choice minimizes processing time since the amount of overlap of adjacent Clustering Cells is minimized. There are cases in which it is not desirable to choose the size of the Boundary Cell in the manner suggested. In particular if for some reason it is desirable to work with a very small Boundary Cell (say 1x1 or 2x2 IRE's) then a Clustering Cell chosen on the previous basis will contain very few samples. This small sample size will cause poor variance estimates in the Clustering Algorithm and a consequent deterioration in the Algorithms performance.

The remaining three parameters namely, Clustering Cell Size, Maximum Number of Modes Permitted, and the Number of Channels used are very much interrelated. To arrive at reasonable values for these parameters a section of multispectral scanner data that was obtained from the area shown in Figure 3 was processed using various combinations of these parameters. For each case considered the average processing time per IRE was experimentally determined. Figure 4 and Figure 5 summarize these results. Note that the given processing times are essentially based on the assumption that all the data of interest is in core storage. In other words the given times must be increased to allow for any required data manipulation. This additional time can be very significant but is negligible if the data is available in a form that necessitates very little data manipulation prior to processing. The computations were carried out on an IBM System/360 Model 44.

The results shown in Figure 4 demonstrate the effect of varying the size of the Clustering Cell and the number of channels used to cluster the data. For these results the threshold was fixed at 1.00 and the maximum number of modes was fixed at 2. Note that for fixed but arbitrary L the processing time increases as the size of the Clustering Cell decreases. In fact a close check shows that this increase in processing time is essentially all accounted for by variation in the overlap between adjacent Clustering Cells. This together with the essentially linear variation in processing time with L indicates that the average number of interations required to cluster the vectors from a Clustering Cell is essentially independent of both the size of the Clustering Cell and the number of dimensions used. This suggests that to minimize processing time large Clustering Cells should be used. This result is however based on the assumption that the maximum number of modes permitted (M_m) is fixed. Clearly M_m should increase monotonically with the size of a Clustering Cell. Examining times in Figure 5 demonstrates how processing time is effected by M_m and T. Considering the computations involved one would expect the processing time to be approximately proportional to $M_m (M_m-1)$. Figure 5 bears out this expectation. Thus M_m is such a dominant factor in controlling processing time that from the practical point of view it is imperative that M_m be relatively small. This means that relatively small Clustering Cells should be used. For small Clustering Cells M_m equal to two provides quite reasonable performance.

Asthetic reasons also indicate the desirability of using small Clustering Cells. When the Clustering Algorithm fails then all the boundaries within the particular Clustering Cell being processed tend to be in error. This results in a "granulation" of the output which is not nearly as evident or as objectionable for small Clustering Cells.

Further examination of Figure 4 reveals that performance improves only slightly as the number of dimensions L is increased above 3. Processing time increases in essentially a linear fashion with L. It is worth mentioning, that comparatively speaking, the results using only one channel are suprisingly good. Note that adding a channel does not always improve the ease with which a boundary can be located. In fact sometimes the reverse is true. This is clearly the case when the boundary in question is not evident in the added channel. Under these circumstances the additional channel certainly does not help and may in fact hinder the clustering procedure. The added channel may however result in the detection of some other boundary not previously detected, and overall performance may improve.

The above considerations suggest that permitting a maximum of only 2 modes (or possibly 3), using a Cluster Cell of about 10x10 IRE's, and utilizing 3 or 4 channels of data represents a reasonable compromise for the present multispectral scanner data. No method exists for choosing the "best" subset of data channels from those available. It seems reasonable however to choose them so they are spectrally as different as possible (i.e., spread out over the spectrum).

It is frequently of interest to know what the boundaries would look like if the threshold is varied. It is, in general, not easy to achieve this without multiple passes through the data. An exception occurs when the maximum number of modes is restricted to 2. Since this in an important practical case its implementation will now be discussed. Recall that during the clustering procedure a pairwise measure of separation is computed for each pair of modes. Let S be this measure of separation for the 2 mode case. Rather than compare S with a single threshold T, as was previously the case, we now compare S with a sequence of thresholds $T_1 > T_2 > \ldots > T_r$. The lowest of these thresholds is chosen so that if $S < T_r$ then almost certainly only one mode is present and no boundaries exist. If $S > T_r$ then it is assumed that 2 modes are present and boundaries do exist. The degree of confidence attached to the boundaries depends upon how many of the T_i's are exceeded by S. This is logical since the larger the value of S, the more separable are the modes, and the more reliable are the resulting boundaries. Different line printer symbols can be used to reflect this degree of confidence. Thus in effect a multi-threshold map is obtained in a single pass.

Figure 6 compares the output of the algorithm modified to accommodate three levels of boundaries corresponding to three thresholds $T_1 > T_2 > T_3$ with a gray-scale printout and aerial photograph. The most distinct boundaries are those for which the measure of separation $S \geq T_1$. These are indicated by the symbols I, -, *, and X as before. For $T_2 \leq S < T_1$ the boundaries are moderately distinct and a plus sign is used to represent all such boundaries. For $T_3 \leq S < T_2$ the boundaries are indistinct and are represented by a period. If $S < T_3$ one mode exists and there are no boundaries. Figure 7 compares the output for one, two, and three levels of boundaries. The values of T_1, T_2, and T_3 used to obtain Figure 6 and Figure 7 were 1.00, 0.85, and 0.70 respectively.

The examples depicted above have involved agricultural type of fields. Figure 8 is an example involving completely different type of terrain. Essentially the same parameters were used to obtain Figure 6 and Figure 8. The parameters differ only in that different channels were used and that in Figure 8 an asterisk is used to represent all of the most distinct boundary IRE's.

Closure

We have argued heuristically that clustering should provide a good approach to the problem of locating spatial boundaries in multispectral scanner data. The basis of this argument is that typically boundaries have some spatial extent and that clustering provides one possible way of using this spatial information. The fact that reasonable performance can be achieved is clearly demonstrated by the experimental results presented. Experimentally we have also established reasonable values for the input parameters to the Boundary Finding Algorithm.

No claim of superiority is made for the particular clustering Algorithm used. The algorithm should perform reasonably well for data sets where modes are approximately Gaussian and Euclidean distance is a good metric in the observation space.

REFERENCES

1. R. A. Holmes and R. B. MacDonald, "The Physical Basis of System Design for Remote Sensing in Agriculture," Proceedings IEEE; Vol. 57, pp. 629-639, April 1969.

2. K. S. Fu, D. A. Landgrebe, and T. L. Phillips, "Information Processing of Remotely Sensed Agricultural Data," Proceedings IEEE; Vol. 57, pp. 639-653, April 1969.

3. D. A. Landgrebe and Staff of Laboratory for Agricultural Remote Sensing, "LARSYS AA, A Processing System for Airborne Earth Resources Data." LARS Information Note 091968, Purdue University.

4. G. H. Ball, "Data Analysis in the Social Sciences: What About the Details?" 1965 Fall Joint Computer Conference, AF/PS, Vol. 27, pp. 533-539, Spartan Books, Washington, D. C., 1965.

5. P. H. Swain and K.S. Fu., "On the Applications of Nonparametric Techniques to Crop Classification Problems," 1968 National Electronics Conference Proceedings, Vol. 24, pp. 14-19.

6. Teddy Huang, "Per Field Classifier for Agricultural Applications," LARS Information Note 060569, Purdue University.

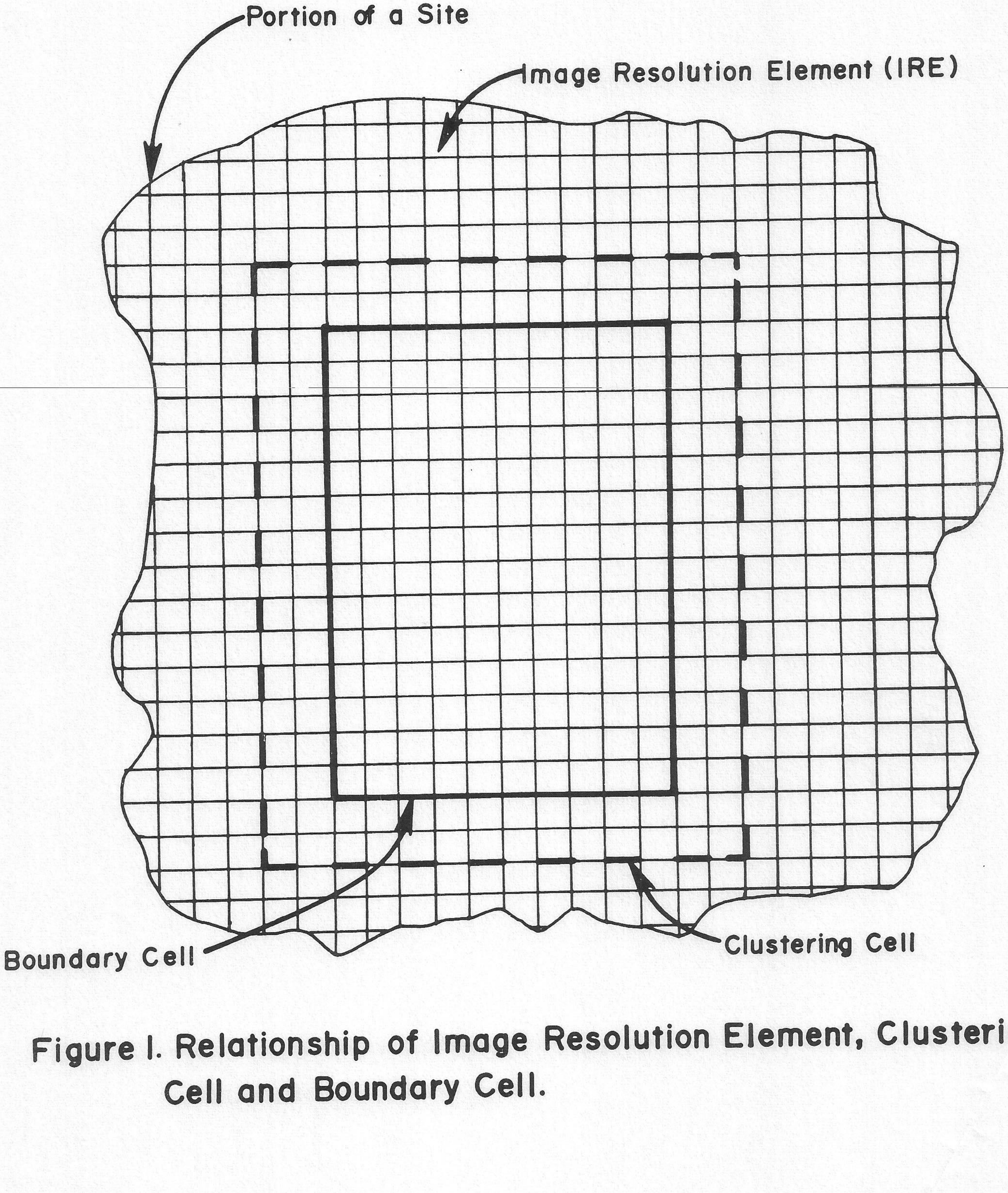


Figure I. Relationship of Image Resolution Element, Clustering Cell and Boundary Cell.

```
422222222222222222222221222111
222222222222222222222222221111
222222222222222222222222222221
222222222222222222222222222221
222224221222222222212111111111
333222444444244444444444444444
111111244444444444444444444444
111111244444444444444444444444
111111244444444444444444344444
111111444444444444444444444444
111111244444444444444444444444
111111243433333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111133333333333333333333333
111111133333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111333333333333333333333333
111111133333333333333333333333
```

Clustered Array

a.

```
                         -
   X - --- ------ -XXXXXXX
X*-IX- ---------- -
 --  X     -
    X
    X
   II
    X - ------------------
    X---------------------
   I*- -
   II
   II
   II
   II
   II
   II
   II
   II
   II
   II
   I*
    *I
    *I
   I*
   II
```

Boundaries

b.

Figure 2. Clustered Array and Boundaries.

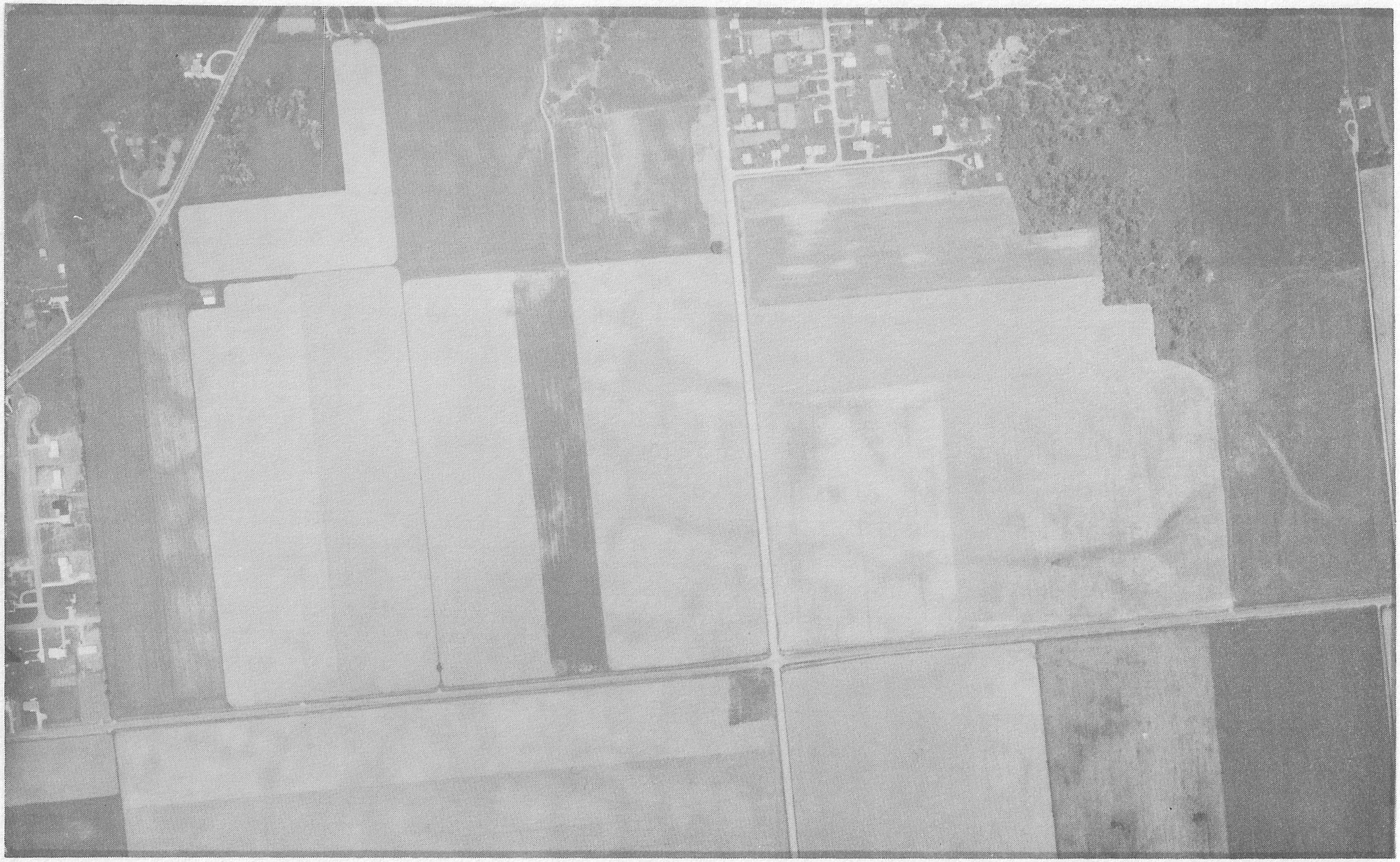


Figure 3. Aerial photograph of test area.

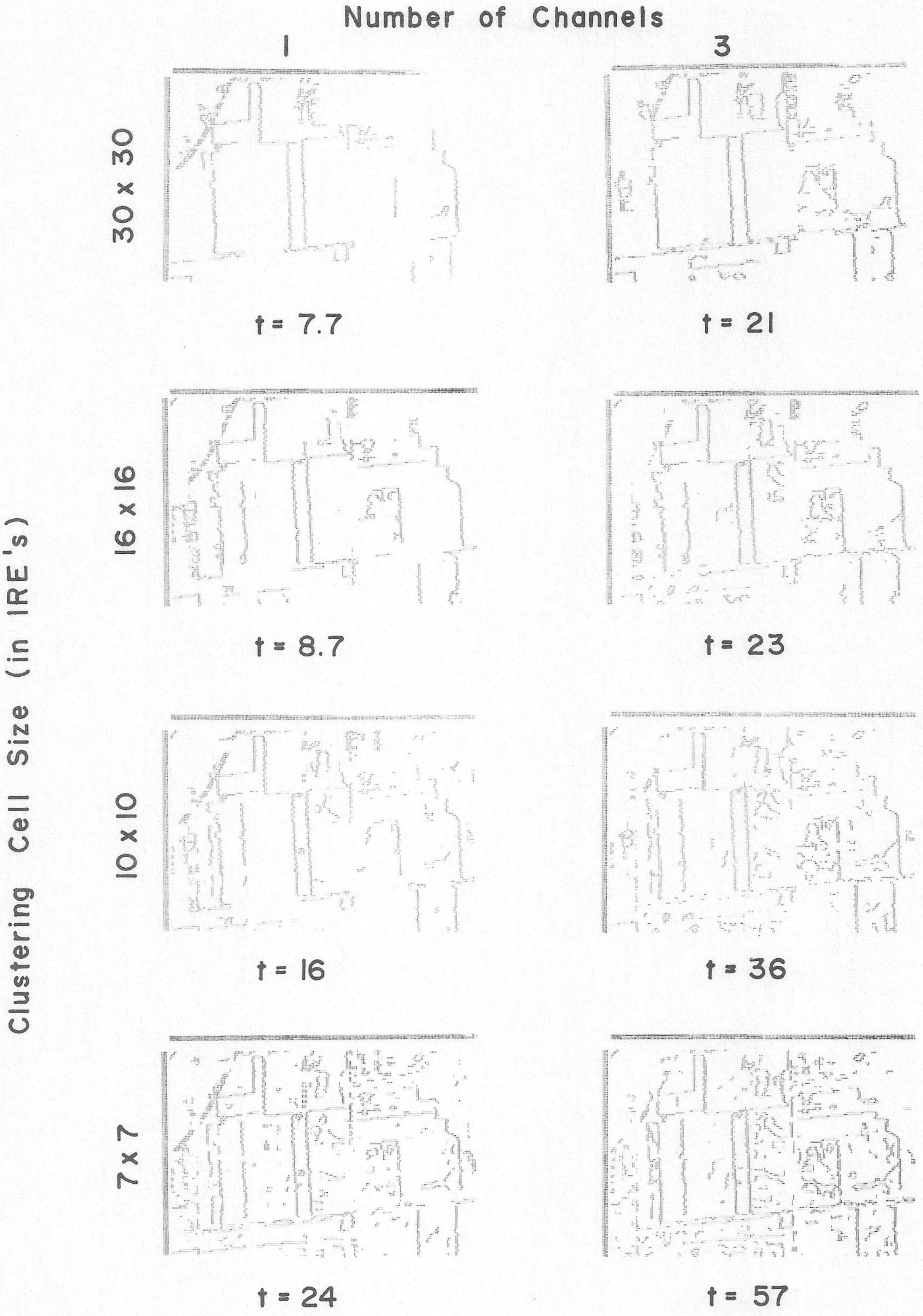


Figure 4. Effect of Clustering Cell size and number of channels on performance and average processing time t (ms/IRE) for Mm=2, K=2 and T=1.00.

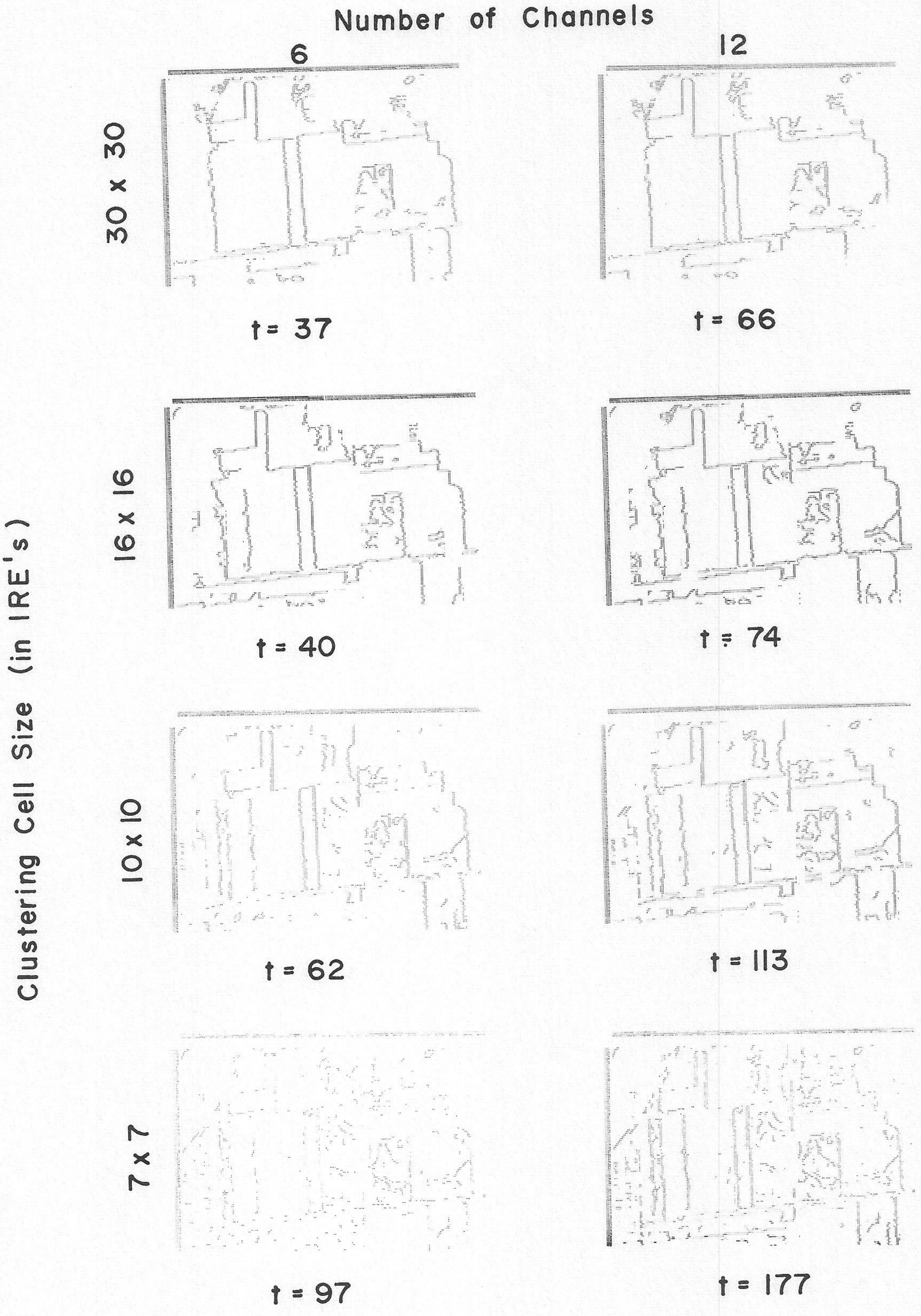


Figure 4. Continued.

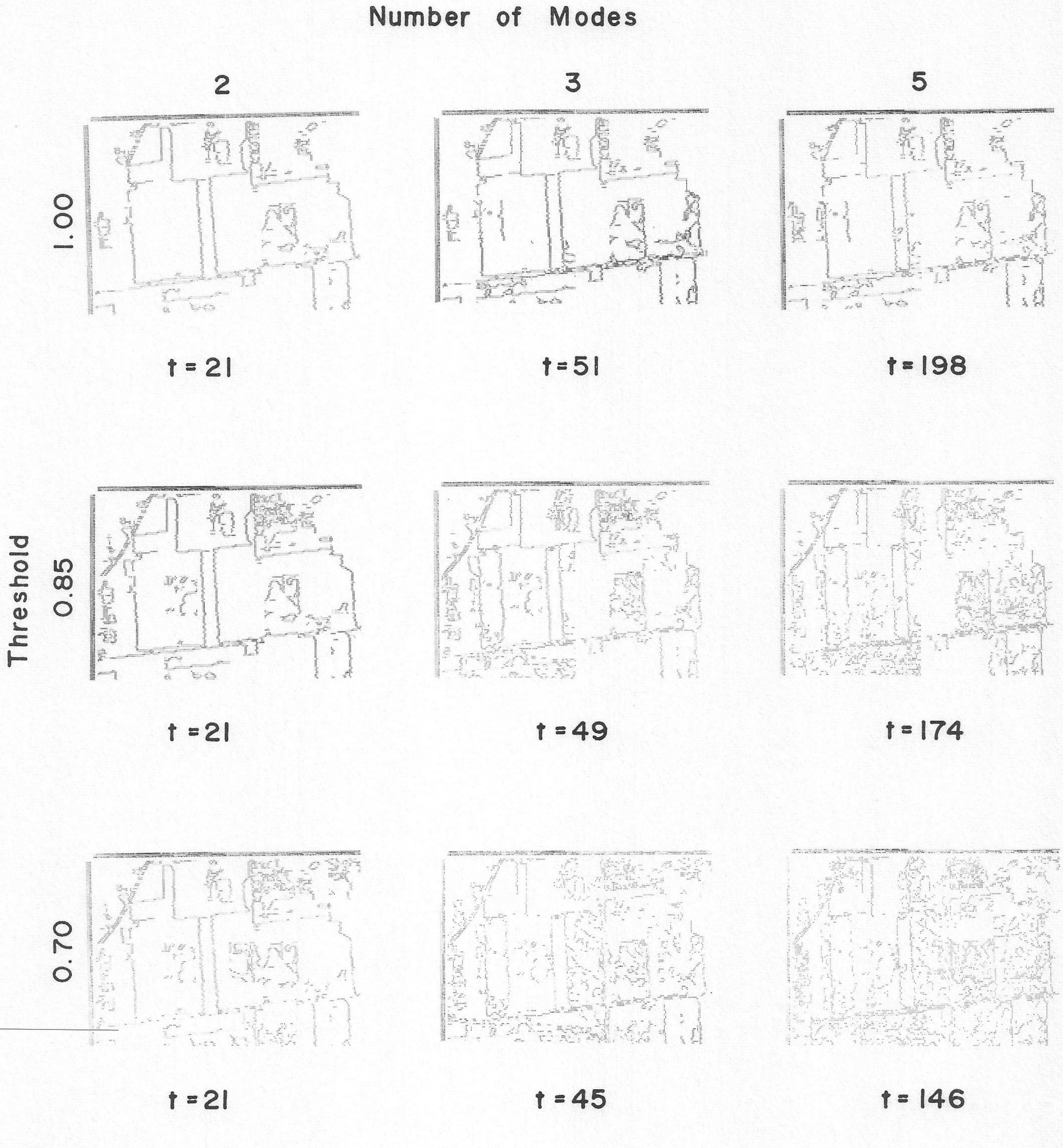


Figure 5. Effect of Threshold and maximum number of modes on performance and average processing time t (ms/IRE) for $w_c = 30$, $k = 2$ and $L = 3$.

BOUNDARIES WITH THRESHOLD VARIATION

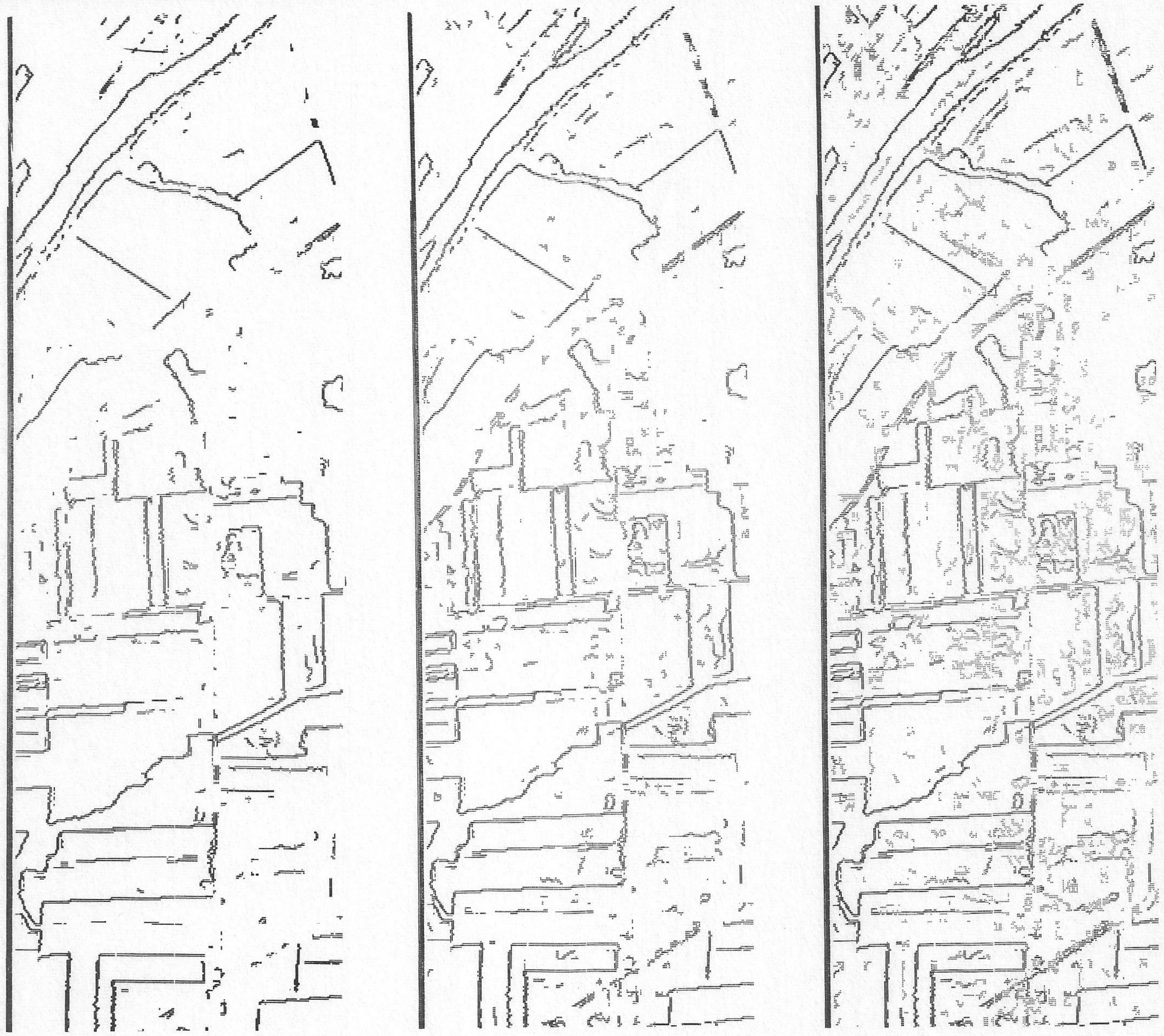
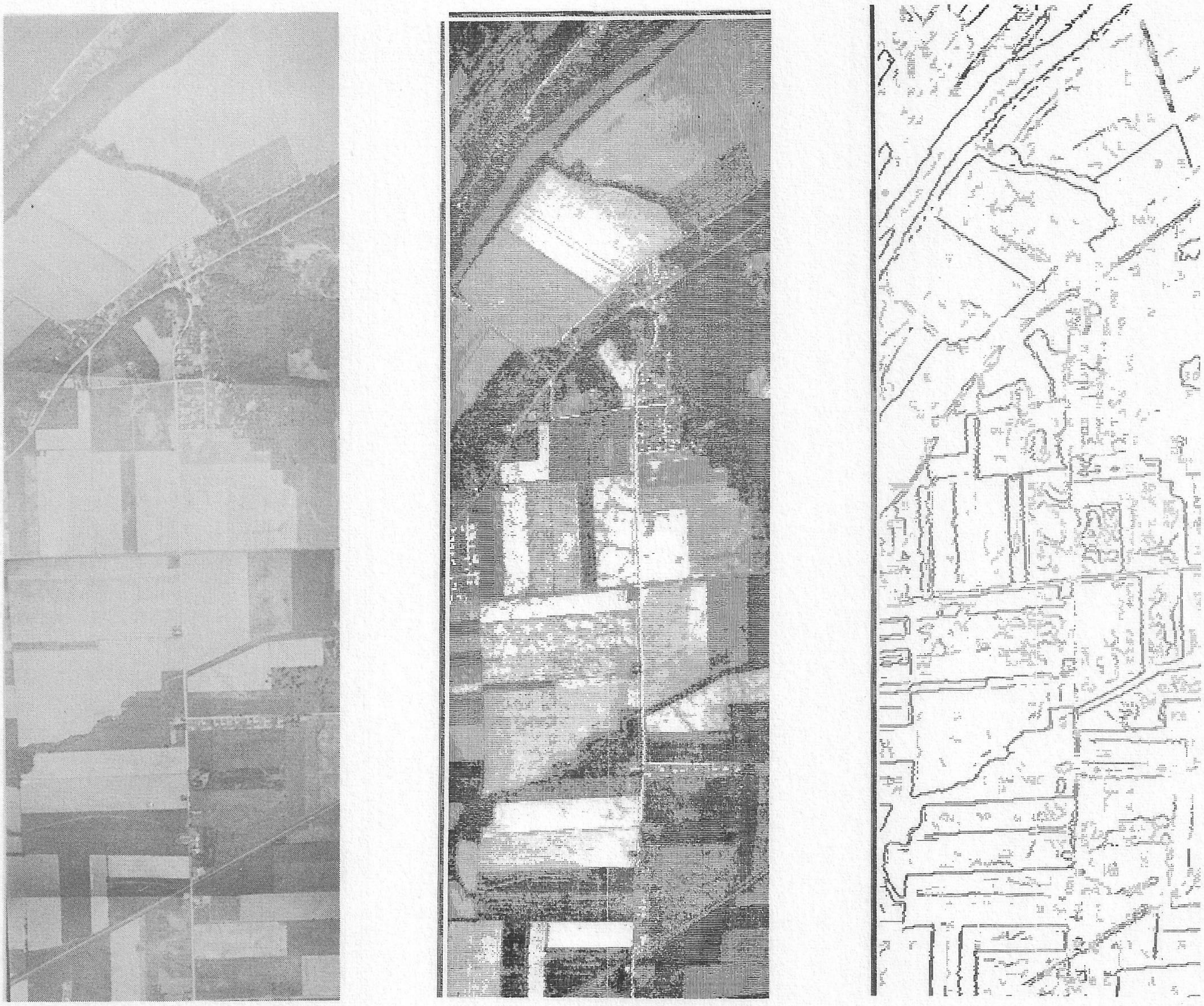


Figure 6.

MACHINE-DRAWN BOUNDARIES

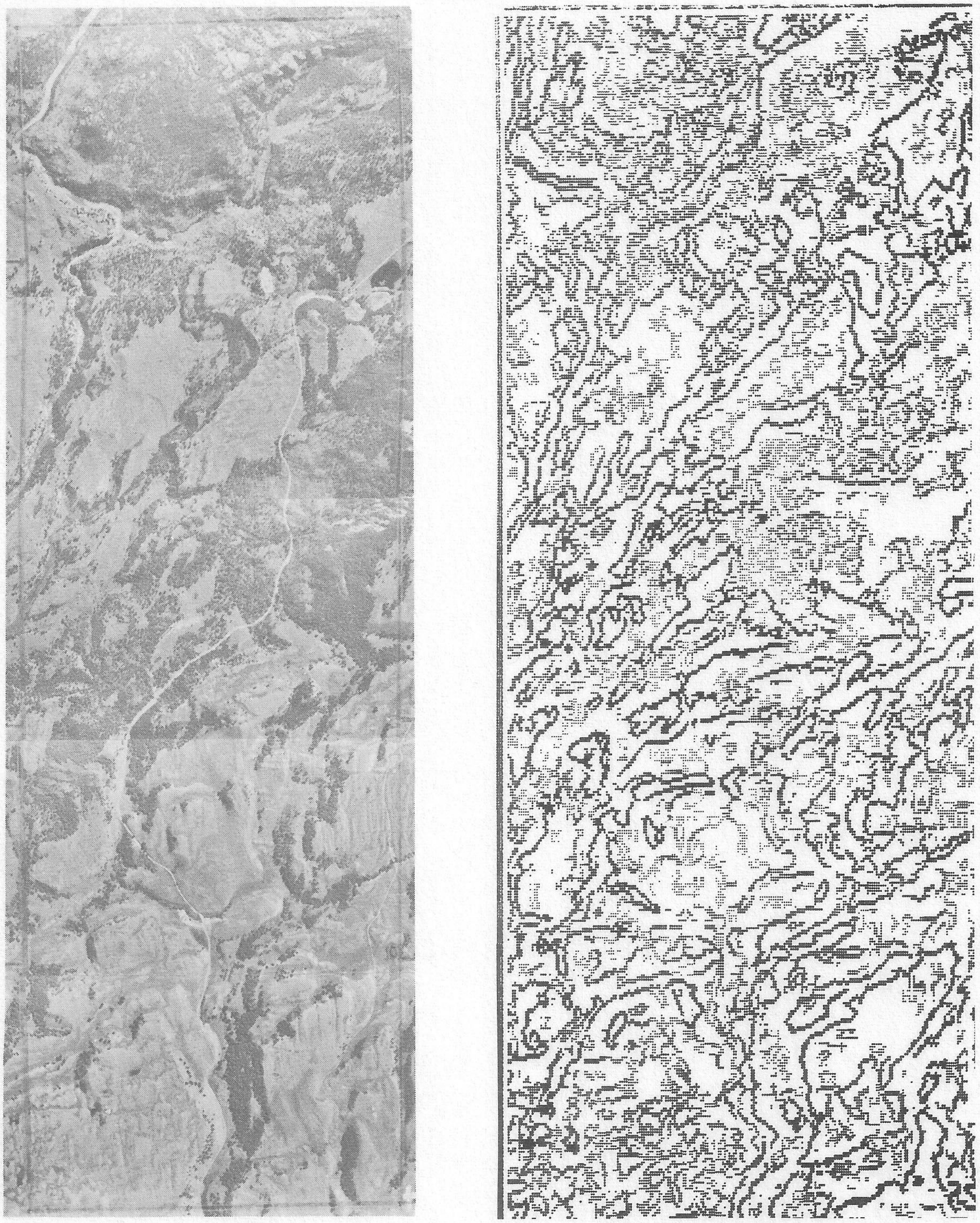


Panchromatic Air Photo

.62 -.66µm

Boundaries

Figure 7.



Panchromatic Air Photo

Boundaries

Figure 8. Boundaries for a Non-Agricultural type scene.