LARS Information Note 071373

Multispectral Image Partitioning

T.V. Robertson, K.S. Fu
and P.H. Swain

The Laboratory for Applications of Remote Sensing

Purdue University, West Lafayette, Indiana

1973

LARS Information Note 071373

Multispectral Image Partitioning†

by

T. V. Robertson, P. H. Swain, and K. S. Fu

TR-EE 73-26

August 1973

† This work was supported by the NASA Contract NAS 5-21773. K. S. Fu was supported by the NSF Grant GK-3672.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

A multispectral image is a representation of the reflectance of a scene in several regions of the electromagnetic spectrum. Presented here is an algorithm that partitions a digitized multispectral image into parts that correspond to objects in the scene. The algorithm is applicable to images in which the objects of interest are regions that contrast with their surroundings and exhibit some form of internal regularity.

A theoretical model of an image made up of arbitrarly shaped objects is presented and a criterion function is defined that is minimized by good partitions. A partitioning algorithm is developed that divides an image into successively smaller rectangles and produces a partition that tends to minimize the criterion function.

The algorithm is applied to images produced by multispectral scanners mounted in an aircraft and in the ERTS-1 satellite. It is shown that classifying agricultural areas in the partitioned image gives results of comparable accuracy and smaller required storage than an algorithm

that classifies image points individually. The partitioning algorithm is also used to extract urban areas from an ERTS image, to approximately isolate the lung in a chest radiograph, and to partition a digitized photograph of a girl.

CHAPTER 1

INTRODUCTION

In a multispectral image, a target is represented by a finite number of image points.* For each image point there are measurements (gray levels) representing the reflectance of the target in one or several regions (channels) of the electromagnetic spectrum. The gray levels are quantized, often to either 64 or 256 levels. Multispectral images can be produced by multispectral scanners [1], flying-spot scanners [2], and by digitizing continuous images such as photographs [3] and radiographs [4]. Figure 1 shows a 3-channel multispectral image produced by an airborne multispectral scanner flying over an agricultural target. Each channel of this image has been quantized to 16 levels for line printer display.

Targets are represented by multispectral images so digital computers can be used to process target information. For example, medical single-channel images have been processed by computer for automatic chromasome identification [5] and diagnosis of chest X-rays [4]. In remote sensing

---

* In this report we consider only digital images; that is images stored as arrays of numbers. This restricts our use of the term "multispectral image" which is sometimes used in the literature to include continuous images that are stored, for example, as photographs.

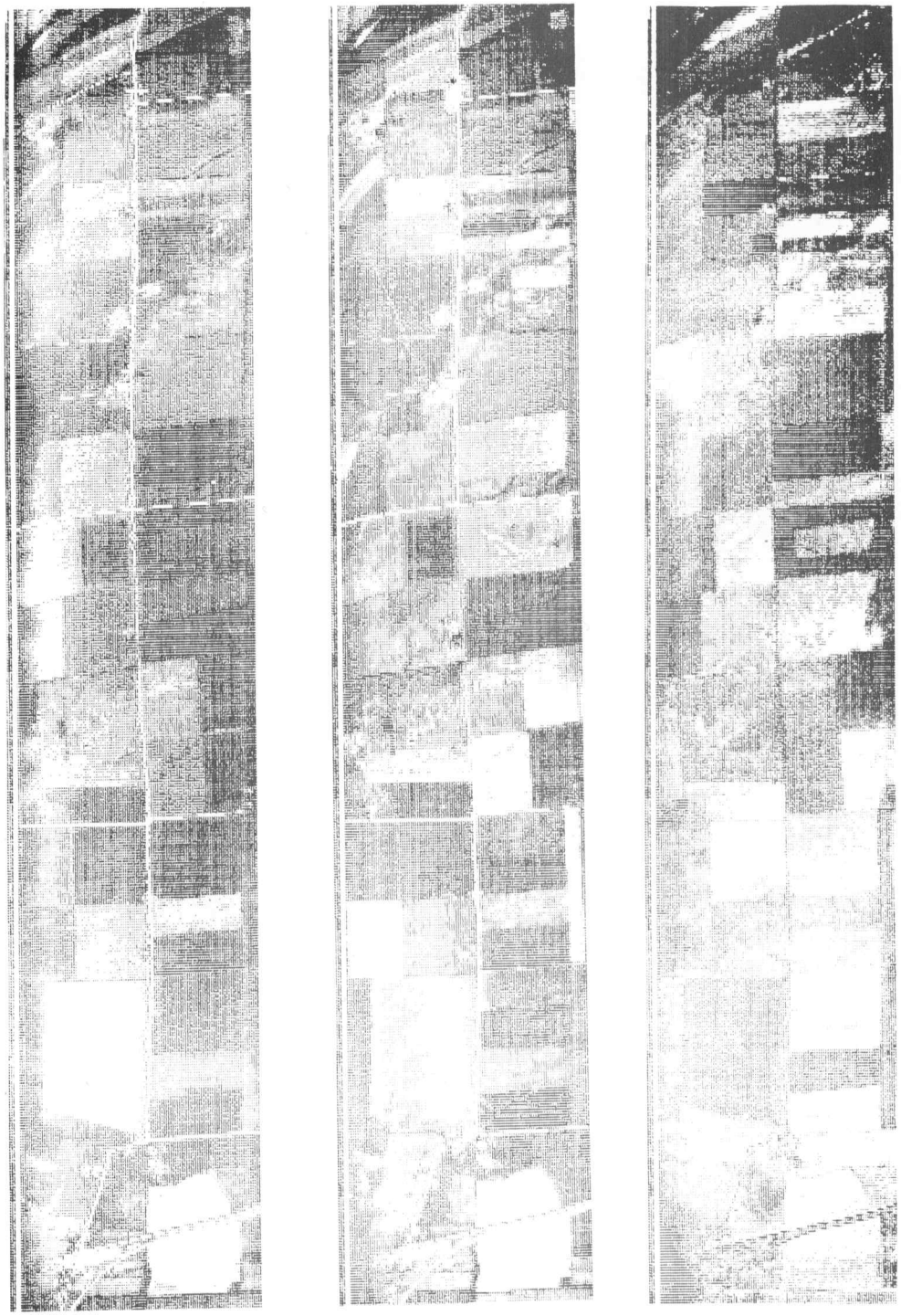


Figure 1. A Multispectral Image

applications, computer algorithms have been developed for several types of multispectral image processing, including classification [6], coding [7], registration [8], change detection [9], and display [10].

1.1 Multispectral Image Partitioning

In Figure 1 one can see regions, which we will call objects, that correspond to target entities such as agricultural fields, forests, bodies of water, and roads. Objects are characterized by internal regularity and contrast with surroundings, and they often contain many image points. The goal of image partitioning is to divide an image into parts that approximate objects. RIMPAR, the Recursive IMage PARtitioning algorithm presented in this report, approximates each object by one or more rectangular blocks of image points. Figure 2 shows an image partitioned by RIMPAR.

Image partitioning can be thought of as transforming the original image, which is a point-by-point (PP) description of a target, to an arrangement-of-objects (AO) description. An AO description is often better than a PP description as an input to processing algorithms, for two basic reasons.

First, more information about a target entity is available from a collection of points associated with the entity, than from an individual point associated with an entity. This fact has been exploited by "sample" classification algorithms [11] that make one classification decision for

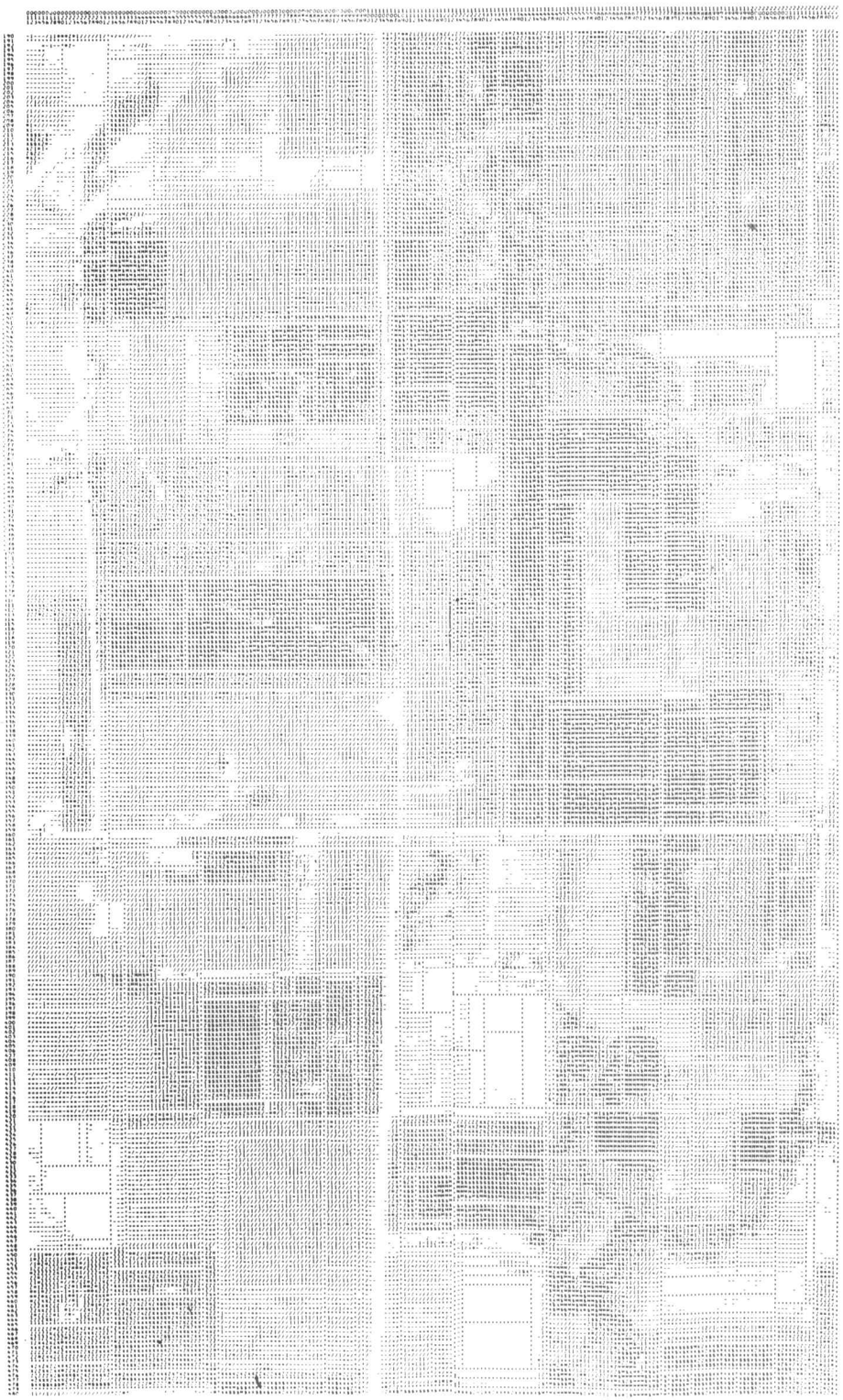


Figure 2. An Image Partitioned by RIMPAR

each group of image points. For several data sets, sample classification has given performance superior to classifiers that make one decision per point [12,6]. The potential advantages of sample classification are especially great when class probability densities differ in shape but exhibit a high degree of overlap [12, pp. 198-199]. Classifying objects instead of points also allows the measurement and use of spatial characteristics such as size, shape, and texture. These have been found to be useful in classification [13,14].

The second basic advantage of an AO description is that an AO description is often more compact than a PP description. This savings in storage space or transmission speed occurs if objects contain enough points so that specifying the locations and essential properties of the objects takes fewer bits than specifying the collection of individual point properties. Classification results can similarly be stored more economically using an AO description. For example, if 4 bytes specify an object location and 1 byte specifies a class label, then the classification results of a 100,000 point image with 10,000 objects would require 100,000 bytes in PP format, but only 50,000 bytes in AO format.

1.2. Related Work in Image Partitioning

An image partitioning algorithm must search for objects. Intuitively, objects have two basic characteristics (we are not considering line drawings):

1) They exhibit an internal regularity.

2) They contrast with their surroundings.

Objects do not exhibit these characteristics in a deterministic or absolute sense because of irregularities due to noise. Detecting these characteristics in a noisy image is a central problem in image partitioning. Another difficulty is that object regularity may exist not only as relatively constant gray level, but also as relatively constant texture or spatial pattern. For example, forests appear as textured objects in aircraft imagery [14]; and in satellite images, urban areas of constant land use can be textured [15].

Two basic approaches to image partitioning have appeared in the literature: the boundary-finding approach, which attempts to exploit object contrast, and the region-finding approach, which uses object regularity.

The boundary-finding approach has two steps. First points along the boundaries of objects are found, then the complete boundaries are derived from the boundary points. Among the techniques used to detect boundary points are local gradient [16], template matching [17], two-dimensional function fitting [18], clustering [19], and gradients estimated from variable-sized neighborhoods [20]. Only the last of these does not assume that objects have nearly constant gray levels. None of these techniques finds boundary points with zero error, so to form complete object boundaries, heuristics using a priori information must be used to fill in missing boundary points and eliminate superfluous boundary

points. Unfortunately effective heuristics seem to be available only when object shape is restricted. For example in [16] and [21], boundary finding is successful mainly because all boundaries are known a priori to be straight line segments.

Previously investigated region-finding methods have followed two steps. First the image is divided into elementary regions, then regions are merged according to a set of merging rules. In [22] elementary regions are regions of constant gray level, and the merging rules are heuristics based on what the final objects should look like. This method is difficult to use if objects do not have constant gray levels. Another region-finding method, which we will call the LBLOCK algorithm, appears in [23] and [24]. Here elementary regions are defined by a regular rectangular grid superimposed upon the image. The merging rule is to merge statistically similar, adjacent regions. This method works well only if elementary regions can be made both small enough to allow good boundary approximation and large enough to preserve the spatial characteristics of the objects of which they are a part.

1.3. A Comparison of Clustering and Image Partitioning

Clustering [6,25] and image partitioning are both methods of grouping data. In fact, minimizing intragroup scatter,which is the basis for the partition criterion used in Chapter 2, is essentially equivalent to a criterion that

has been used in clustering [25]. However, spatial considerations make clustering and image partitioning different. Because an object can be textured, the points within an object might not form a compact cluster in the measurement space. Also, because there can be several instances of a particular target entity in a single image, nonadjacent objects might be nearly identical in measurement space. Another difference is that in image partitioning, the existance of a partition that completely separates objects is guaranteed. However in clustering, if we seek underlying classes with overlapping density functions, the classes can never be completely separated.

1.4. Summary of Work Presented Here

In Chapter 2 we present a model for an idealized, continuous image made up of objects. A G-regular partition is defined as a partition in which every block (piece of the partitioned image) is either an object or part of a single object. Next we define a criterion function $V_G(P)$ for a partition P of an ideal image. It is shown that $V_G(P)$ is a minimum if and only if P is a G-regular partition. Next we present an algorithm that partitions an ideal image into successively smaller rectangles. At each step in the algorithm, a block is subdivided unless

(1) the size of the block (smallest side) is less than twice a parameter MINSIZE; or

(2) all the points within the block are from a single object.

An error $\Delta V_G(P_f)$ is defined for the final partition P_f of the algorithm. We show that for any image I and any $\varepsilon>0$, there are values of MINSIZE for which $\Delta V_G(P_f)<\varepsilon$.

In Chapter 3 we discuss the application of the results of Chapter 2 to real images with a finite number of points. Estimation with finite sample size, choosing MINSIZE, and programming considerations are among the topics considered.

In Chapter 4 the algorithms used to classify partition blocks are discussed.

In Chapter 5 we present the results of experiments designed to test how the parameters of the partitioning algorithm affect the final partition. We also show the results of applying the partitioning algorithm to the classification of agricultural areas in aircraft and satellite remotely-sensed images. The algorithm is also used to extract cities from a satellite-sensed image, to isolate the lung from a chest radiograph, and to partition an image of a human face.

CHAPTER 2

A PARTITIONING ALGORITHM

In this chapter we first present a model for an image containing objects. In this model each object is characterized by the expected value of the reflectance of points within the object. For a given object we assume that the expected point reflectance value is constant for all subsets of the object. We also assume that two adjacent objects differ in at least their expected point reflectance values. The concepts of internal regularity and external contrast, mentioned in Chapter 1 as characteristics of objects, are thus represented by parameters of distributions underlying the reflectance values of the objects. In Chapter 3 we discuss the problem of estimating expected point reflectance values.

This image model is valid for images in which objects can be considered to have constant gray levels with zero-mean noise added. System noise in sensors can be characterized as zero-mean noise. Texture consisting of fixed proportions of various gray levels in a random spatial distribution can also be considered zero-mean "noise."

The images of this chapter will be considered to be continuous. This assumption allows us to derive some theoretical results that give an indication of the algorithm's performance for digital images.

2.1. Notation and Definitions

An image I is an infinite set of points in a plane that is surrounded by a closed curve C of finite length, such that any set J surrounded by C satisfies $J \supseteq I$. Note that in our definition "image" refers to a set of points. The gray levels associated with an image will be discussed separately below. The essential properties of our "image" are that it has an infinite number of points per unit area, and that it is simply connected. A subimage of I is an image J such that $J \supseteq I$. From this point on in this paper, we will assume the all point sets under discussion are images.

A partition P of an image I is a finite set of images $\{I_1, I_2, \ldots, I_L\}$ such that

$$I = \bigcup_{i=1}^{L} I_i$$

and for $j \neq i$,

$$I_j \cap I_k = \emptyset,$$

where $\emptyset$ is the empty set. Each $I_j \in P$ will be called a block of P.

The area of an image J will be denoted $|J|$. Two subimages of I, J_1 and J_2, are said to be adjacent if $J_1 \cup J_2$ is an image, and $J_1 \cap J_2 = \emptyset$.

Horizontal and vertical will refer to a set of perpendicular coordinate axes that are fixed with respect to I. The size of an image $J \subseteq I$ is the minimum of the horizontal and vertical extent of J.

A gray-level function $g(\cdot)$ is a function whose domain is an image and whose range is a bounded interval on the real line. We will use $g(X)$ to stand for the gray level at a point $X \varepsilon I$. For a given X, $g(X)$ will be considered a random variable whose distribution depends on X. A gray-level vector $G(\cdot)$ is a vector of gray-level functions: $G(X) = (g_1(X), g_2(X), \ldots, g_N(X))$, where each $g_i(\cdot)$ is a gray-level function.

Consider an image J. Let $E[\cdot]$ be expected value. We will use the following notation:

$$M_{g_i}(J) = E[g_i(X) \,|\, X \varepsilon J]$$

$$M_G(J) = \begin{bmatrix} M_{g_1}(J) \\ M_{g_2} \\ \vdots \\ M_{g_N}(J) \end{bmatrix}$$

We call $M_G(J)$ the mean vector of J. Also let

$$S_{g_i}^2(J) = E[(g_i(X) - M_{g_i}(X))^2 \,|\, X \varepsilon J] \qquad (2.1)$$

$$Z_{g_i}^2(J) = E[g_i(X)^2 \,|\, X \varepsilon J]. \qquad (2.2)$$

An image J is G-regular if for any subimage $K \subseteq J$, $M_G(K) = M_G(J)$. A G-regular image is "homogeneous" with respect to G in the sense that the mean values of the gray-level functions $\{g_i(\cdot), i=1, 2, \ldots, N\}$ are constant throughout the image.

A subimage J of I is G-distinct if J is G-regular, and if for any subimage K⊆I that is adjacent to J, K∪J is not G-regular. In other words, a G-distinct subimage is surrounded by subimages with different mean values of the N gray-level functions of G.

A partition P is G-regular if every block of P is G-regular; P is called G-optimal if every block in P is also G-distinct. Note that a G-optimal partition is necessarily G-regular, but a G-regular partition is not G-optimal if some pair of adjacent blocks have the same mean vectors. In Figure 3 examples of G-optimal, G-regular, and approximately G-regular partitions of a hypothetical image are shown. In each example the G-optimal partition is shown with dotted lines.

2.2. Properties of G-Regular Images

Theorem 2.1: If K⊆J and J is G-regular, then K is G-regular.

Proof: Assume K is not G-regular. Then there is a B⊂K such that

$$M_G(B) \neq M_G(K) \tag{2.3}$$

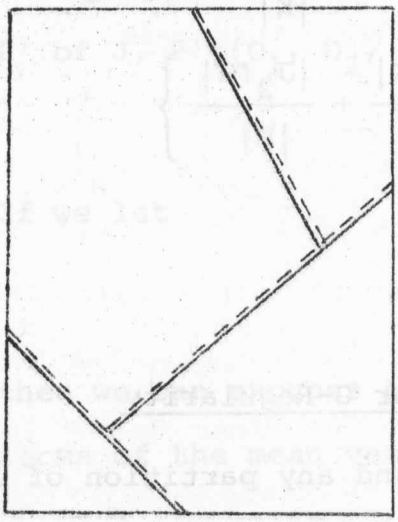
But B⊆K⊆J and J is G-regular, so

$$M_G(B) = M_G(K) = M_G(J). \tag{2.4}$$

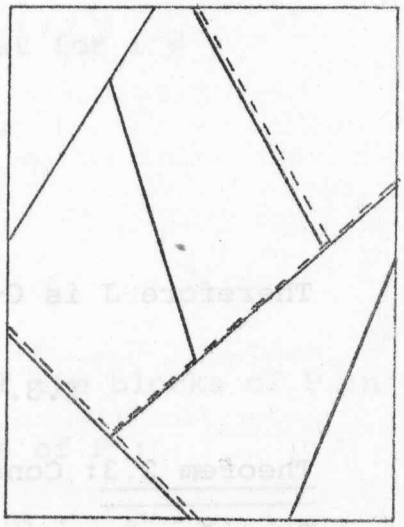
Since Eqn. 2.3 contradicts Eqn. 2.4, K must be G-regular.

Theorem 2.2: If J_1 and J_2 are adjacent and G-regular, and $M_G(J_1) = M_G(J_2)$, then $J = J_1 \cup J_2$ is G-regular.

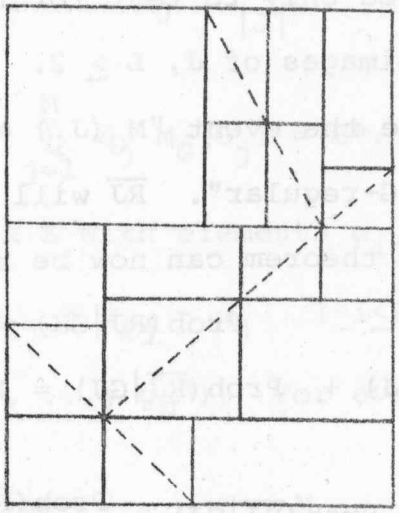
Proof: Consider any subimage K⊆J. Then



G-Optimal Partition



G-Regular Partition



Approximately G-Regular Partition

Figure 3 Partition Types

$$M_G(K) = \frac{|J_1 \cap K|}{|K|} M_G(J_1) + \frac{|J_2 \cap K|}{|K|} M_G(J_2)$$

$$= M_G(J_1) \left\{ \frac{|J_1 \cap K|}{|K|} + \frac{|J_2 \cap K|}{|K|} \right\}$$

$$= M_G(J_1)$$

Therefore J is G-regular.

2.3. The Mean Test for G-Regularity

Theorem 2.3: Consider an image J and any partition of J, $P = \{J_1, J_2, \ldots, J_L\}$, $L \geq 2$. If G* is a constant vector and $M_G(J_i) = G^*$, $1 \leq i \leq L$, then Prob(J is G-regular) = 1. The proof of this theorem will show that to determine if an image J is G-regular, we need only to test for the equality of the mean vectors of L subimages of J, $L \geq 2$.

Proof: Let GJ be the event "$M_G(J_i) = G^*$, $1 \leq i \leq L$", and RJ be the event "J is G-regular". $\overline{RJ}$ will stand for "J is not G-regular". The theorem can now be restated as:

$$\text{Prob}(RJ \mid GJ) = 1 \qquad (2.5)$$

Since $\text{Prob}(RJ \mid GJ) + \text{Prob}(\overline{RJ} \mid GJ) = 1$, the theorem holds if and only if

$$\text{Prob}(\overline{RJ} \mid GJ) = \frac{\text{Prob}(\overline{RJ}, GJ)}{\text{Prob}(GJ)} = 0 \qquad (2.6)$$

$\text{Prob}(GJ) \neq 0$ because we assume the existance of G-regular images, so the theorem holds if and only if $\text{Prob}(\overline{RJ}, GJ) = 0$. Now we will derive a matrix equation expressing the joint event $(\overline{RJ}, GJ)$, and show that the probability of satisfying this equation is zero.

If J is not G-regular, then there must be a partition P' of J, $P'=\{O_1, O_2, \ldots, O_M\}$, such that for $i \neq j$

$$M_G(O_i) \neq M_G(O_j).$$

If we let

$$c_{ij} = \frac{|J_i \cap O_j|}{|J_i|}$$

then we can express the mean vectors of the blocks of P in terms of the mean vectors of the blocks of P':

$$M_G(J_i) = \sum_{j=1}^{M} c_{ij} M_G(O_j)$$

Now if $M_G(J_i) = G^*$, $1 \leq i \leq L$, then $M_G(J)=G^*$, so if we let

$$d_j = \frac{|O_j|}{|J|}$$

then

$$\sum_{j=1}^{M} d_j M_G(O_j) = G^*.$$

Consider the matrix E with elements e_{ij} defined by

$$e_{ij} = c_{ij} - d_j \quad 1 \leq i \leq L, \; 1 \leq j \leq M.$$

Let $G^* = (g_1^*, g_2^*, \ldots, g_N^*)$. For every element $g_k(\cdot)$ of $G(\cdot)$ we can write

$$\sum_{j=1}^{M} e_{ij} M_{g_k}(O_j) = \sum_{j=1}^{M} c_{ij}(O_j) - \sum_{j=1}^{M} d_j M_{g_k}(O_j) = g_k^* - g_k^* = 0$$

if and only if GJ. Thus GJ can be expressed by

$$\begin{bmatrix} E \end{bmatrix} \begin{bmatrix} M_{g_k}(O_1) \\ M_{g_k}(O_2) \\ \vdots \\ M_{g_k}(O_M) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad 1 \leq k \leq N \qquad (2.7)$$

We now consider, for any k, solution vectors of the form $(M_{g_k}(O_1), M_{g_k}(O_2), \ldots, M_{g_k}(O_M))$ that satisfy Eqn. 2.7. The set of solution vectors for Eqn. 2.7 is called the "null space" of the matrix E [26]. The dimension q of the null space of E is the number of columns of E (=M) minus the rank of E. We assume that the rank of E is greater than zero, so that q is less than M.

For a given E, we assume that $(M_{g_k}(O_1), M_{g_k}(O_2), \ldots, M_{g_k}(O_M))$ is random vector, continuously distributed in a bounded region of M-space. (This assumption holds only if $\overline{RJ}$, for if J is G-regular, then all the vector elements are equal, and the equation is satisfied). Since the solutions of Eqn. 2.7 occupy a subspace of dimension $q<M$, the solution vectors occur with zero probability. Thus Eqn. 2.7 holds with zero probability, and the theorem is proved.

2.4. Uniqueness of the G-optimal Partition

Theorem 2.4: The G-optimal partition P* of I is unique.

Proof: Assume there are two partitions $P^* = \{I_1, I_2, \ldots, I_M\}$ and $P' = \{J_1, J_2, \ldots, J_L\}$ that are both G-optimal and $P^* \neq P'$. Since $P^* \neq P'$, for some k there are adjacent blocks J_i and J_j such that

$$I_k = (J_i \cap I_k) \cup (J_j \cap I_k)$$

Since I_k is G-regular

$$M_G(I_k) = M_G(J_i \cap I_k) = M_G(J_j \cap I_k). \qquad (2.8)$$

J_i and J_j are also G-regular so

$$M_G(J_i \cap I_k) = M_G(J_i)$$

$$M_G(J_j \cap I_k) = M_G(J_j)$$

But if P' is G-optimal then $M_G(J_i) \neq M_G(J_j)$. This contradicts Eqn. 2.8, so there can be only one G-optimal partition.

2.5. Partition Criterion

We assume that the blocks in the G-optimal partition P* of I correspond to the objects in I. Therefore a good partition of I is one that closely approximates P*. In this subsection we present a criterion function that is minimized by good partitions.

2.5.1 Criterion Definition

Consider the G-optimal partition of I, $P^* = \{O_1, O_2, \ldots, O_M\}$, an arbitrary partition $P = \{I_1, I_2, \ldots, I_L\}$, and a gray-level function $g(\cdot)$.

We first define a criterion $V_g(P)$ for the single gray-level function $g(\cdot)$:

$$V_g(P) = \sum_{i=1}^{L} \frac{|I_i|}{|I|} S_g^2(I_i) \qquad (2.9)$$

Recall that the $S_g^2(I_i)$'s are the variances of the blocks in the partition P. A block variance tends to be small if the block contains a single object; but a block that overlaps an object boundary or contains several objects will have relatively high variance. Since in Eqn. 2.9 the block variances are weighted by the block areas, $V_g(P)$ will tend to be small when most of

the largest blocks contain only a single object; in other words, when P is approximately g-regular.

For a gray-level vector $G(\cdot)$ we define

$$V_G(P) = \sum_{j=1}^{N} V_{g_j}(P) \tag{2.10}$$

We also define a partition error $\Delta V_g(P)$ for $g(\cdot)$,

$$\Delta V_g(P) = V_g(P) - V_g(P^*) \tag{2.11}$$

and a partition error for $G(\cdot)$,

$$\Delta V_G(P) = V_G(P) - V_G(P^*) = \sum_{j=1}^{N} \Delta V_{g_i}(P) \tag{2.12}$$

2.5.2 Criterion Properties

Theorem 2.5: Consider the G-optimal partition of I, $P^* = \{O_1, O_2, \ldots, O_M\}$, and an arbitrary partition of I, $P = \{I_1, I_2, \ldots, I_L\}$. Then

$$\Delta V_g(P) = \sum_{i=1}^{M} \sum_{j=1}^{L} \frac{|O_i \cap I_j|}{|I|} (M_g(O_i) - M_g(I_j))^2. \tag{2.13}$$

Proof: From set theory,

$$O_i = \bigcup_{j=1}^{L} (I_j \cap O_i).$$

Since $I_j \cap I_k = \emptyset$, $j \neq k$,

$$(I_j \cap O_i) \cap (I_k \cap O_i) = \emptyset, \quad j \neq k.$$

Therefore

$$|O_i| = \sum_{j=1}^{L} |I_j \cap O_i|$$

and

$$V_g(P^*) = \sum_{i=1}^{M} \frac{|O_i|}{|I|} S_g^2(O_i)$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{L} \frac{|O_i \cap I_j|}{|I|} S_g^2(O_i).$$

Now

$$V_g(P) = \sum_{j=1}^{L} \frac{|I_j|}{|I|} S_g^2(I_j)$$

so

$$V_g(P) - V_g(P^*) = \sum_{j=1}^{L} \left(\frac{|I_j|}{|I|} S_g^2(I_j) - \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I|} S_g^2(O_i)\right)$$

$$= \sum_{j=1}^{L} \frac{|I_j|}{|I|} \left(S_g^2(I_j) - \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} S_g^2(O_i)\right). \tag{2.14}$$

The probability density function $f(g(X)|X\varepsilon I_j)$ is given by

$$f(g(X)|X\varepsilon I_j) = \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} f(g(X)|X\varepsilon O_i). \tag{2.15}$$

We can write (See Eqns. 2.1 and 2.2)

$$S_g^2(I_j) = Z_g^2(I_j) - M_g^2(I_j)$$

$$Z_g^2(I_j) = \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} Z_g^2(O_i) \tag{2.16}$$

$$S_g^2(O_i) = Z_g^2(O_i) - M_g^2(O_i) \tag{2.17}$$

From Eqns. 2.14, 2.16, and 2.17

$$V_g(P) - V_g(P^*) = \sum_{j=1}^{L} \frac{|I_j|}{|I|} \left\{ \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} Z_g^2(O_i) - M_g^2(I_j) \right.$$

$$\left. - \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} (Z_g^2(O_i) - M_g^2(O_i)) \right\}$$

$$= \sum_{j=1}^{L} \frac{|I_j|}{|I|} \left\{ \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g^2(O_i) - M_g^2(I_j) \right\} \tag{2.18}$$

From Eqn. 2.15 we have

$$M_g(I_j) = \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g(O_i) \tag{2.19}$$

and so

$$\sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g^2(O_i) - M_g^2(I_j) = \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g^2(O_i) - \left(\sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g(O_i)\right)^2$$

$$= \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g^2(O_i) - 2\left(\sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g(O_i)\right) \cdot \left(\sum_{k=1}^{M} \frac{|O_k \cap I_j|}{|I_j|} M_g(O_k)\right) + \left(\sum_{k=1}^{M} \frac{|O_k \cap I_j|}{|I_j|} M_g(O_k)\right)^2 \cdot \left(\sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|}\right) \qquad (2.20)$$

The last equation follows from

$$\sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} = 1$$

Using Eqn. 2.19 we simplify Eqn. 2.20 to

$$\sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} M_g^2(O_i) - M_g^2(I_j)$$

$$= \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} \left\{ M_g^2(O_i) - 2 M_g(O_i) M_g(I_j) + M_g(I_j)^2 \right\}$$

$$= \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} (M_g(O_i) - M_g(I_j))^2 .$$

Now we can rewrite Eqn. 2.18

$$V_g(P) - V_g(P^*) = \sum_{j=1}^{L} \frac{|I_j|}{|I|} \sum_{i=1}^{M} \frac{|O_i \cap I_j|}{|I_j|} (M_g(O_i) - M_g(I_j))^2$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{L} \frac{|O_i \cap I_j|}{|I|} (M_g(O_i) - M_g(I_j))^2 .$$

Theorem 2.5 can be interpreted as follows: The partition error $\Delta V_g(P)$ depends only upon the regions where blocks overlap more than one object. Note that if a block I_j contains only one object O_k, then $M_g(I_j) - M_g(O_k) = 0$ and no contribution is made to the error. Also, non-overlapping object-block pairs do not contribute to the error because if O_i and I_j do not overlap, then $|O_i \cap I_j| = 0$. The partition error is large when the area of object-block overlap is large, and also when the differences between the means of overlapping blocks (Eqn. 2.19) and the means of the objects they overlap are large.

Theorem 2.6: $\Delta V_g(P) = 0$ if and only if P is g-regular.

Proof: First assume $\Delta V_g(P) = 0$. From Eqn. 2.13 this implies $|O_i \cap I_j| (M_g(O_i) - M_g(I_j))^2 = 0$, $1 \leq i \leq M$, $1 \leq i \leq L$. Therefore if $(O_i \cap I_j) \neq \emptyset$, we must have $M_g(O_i) = M_g(I_j)$.

Consider a block I_j of P, and any subimage $K \subseteq I_j$. We can write

$$K = \bigcup_{i=1}^{M} O_i \cap K \qquad (2.21)$$

For each i, $O_i \cap K \neq \emptyset$ implies $O_i \cap I_j \neq \emptyset$ and $M_g(O_i) = M_g(I_j)$. Since $(O_i \cap K) \subseteq O_i$, $M_g(O_i \cap K) = M_g(O_i) = M_g(I_j)$, $1 \leq i \leq M$, when $O_i \cap K \neq \emptyset$. Therefore each term on the right side of Eqn. 2.21 has the same mean $M_g(I_j)$, so $M_g(K) = M_g(I_j)$, and I_j is g-regular. Since every block I_j of P is g-regular, P is g-regular.

Now assume P is g-regular. Consider each $|O_i \cap I_j| \cdot (M_g(O_i) - M_g(I_j))^2$, $1 \leq i \leq M$, $1 \leq j \leq L$. We will show if

$|O_i \cap I_j| \neq 0$, then $M_g(O_i) = M_g(I_j)$.

If $|O_i \cap I_j| \neq 0$ then $(O_i \cap I_j) \neq \emptyset$. We can write

$$O_i = \bigcup_{j=1}^{L} O_i \cap I_j \qquad 1 \leq i \leq M \tag{2.22}$$

For every nonempty term on the right side of Eqn. 2.22 we have $(O_i \cap I_j) \subseteq O_i$ so

$$M_g(O_i \cap I_j) = M_g(O_i) \qquad 1 \leq i \leq M,\ 1 \leq j \leq L \tag{2.23}$$

We can also write

$$I_j = \bigcup_{i=1}^{M} I_j \cap O_i \qquad 1 \leq j \leq L$$

and since each I_j is g-regular, if $(I_j \cap O_i) \neq \emptyset$,

$$M_g(O_j \cap I_j) = M_g(O_i), \qquad 1 \leq i \leq M, \qquad 1 \leq j \leq L. \tag{2.24}$$

From Eqns. 2.23 and 2.24 we have if $(O_i \cap I_j) \neq \emptyset$. then

$$M_g(O_i) = M_g(I_j) \qquad 1 \leq i \leq M, \qquad 1 \leq j \leq L.$$

Therefore $\Delta V_g(P) = 0$.

Theorem 2.7: $\Delta V_G(P) = 0$ if and only if P is G-regular.

Proof: From Eqn. 2.12 we see that $\Delta V_G(P) = \sum_{j=1}^{N} \Delta V_{g_j}(P) = 0$ if and only if $\Delta V_{g_j}(P) = 0$, $1 \leq j \leq N$. (Note from Eqn. 2.9 that each $\Delta V_{g_j}(P) \geq 0$). From Theorem 2.6, $\Delta V_{g_j}(P) = 0$ if and only if P is g_j-regular, $1 \leq j \leq N$. Therefore $\Delta V_G(P) = 0$ if and only if P is G-regular.

From Eqn. 2.13 it is clear that $\Delta V_g(P) \geq 0$, and so $\Delta V_G(P) \geq 0$. Therefore from the preceeding two theorems it follows that $V_G(\cdot)$ is minimized by the G-optimal partition P^*, and also by any G-regular partition. The algorithm presented in the next section produces a partition P_f that tends to

minimize $V_G(P_f)$, so P_f is approximately G-regular. A G-regular partition is sub-optimal, but is a simple matter (in the ideal case) to transform a G-regular partition to the G-optimal partition, by merging adjacent blocks that have identical mean vectors. Therefore a partition that is approximately G-regular can be made approximately G-optimal by merging adjacent blocks having identical mean vectors.

2.6. Partitioning Algorithm for Ideal Images

Figure 4 is a flow chart of the basic RIMPAR algorithm. The algorithm is recursive in the sense that every image J (except when J=I) that is divided into two parts, is a subimage of a previously divided image.

RIMPAR continues to subdivide blocks until the block under consideration is either too small or G-regular. The question of G-regularity is decided in the algorithm by the mean test of Theorem 2.3 . In deriving the algorithm convergence properties (Theorem 2.9), we assume that there are no errors in deciding G-regularity of a subimage J by testing the equality of the mean vectors of an arbitrary partition of J into two parts. In Chapter 3 we discuss modifications to the basic algorithm for use when G-regularity cannot be decided with zero error. The main modification is that several partitions of J are tried, and the mean vectors of one of the trial partitions are compared statistically. These trial partitions are generated by (K_D-1) horizontal and (K_D-1) vertical, equally-spaced lines. In both the basic and the practical algorithm, we do not allow the formation of trial partitions with blocks smaller than MINSIZE.

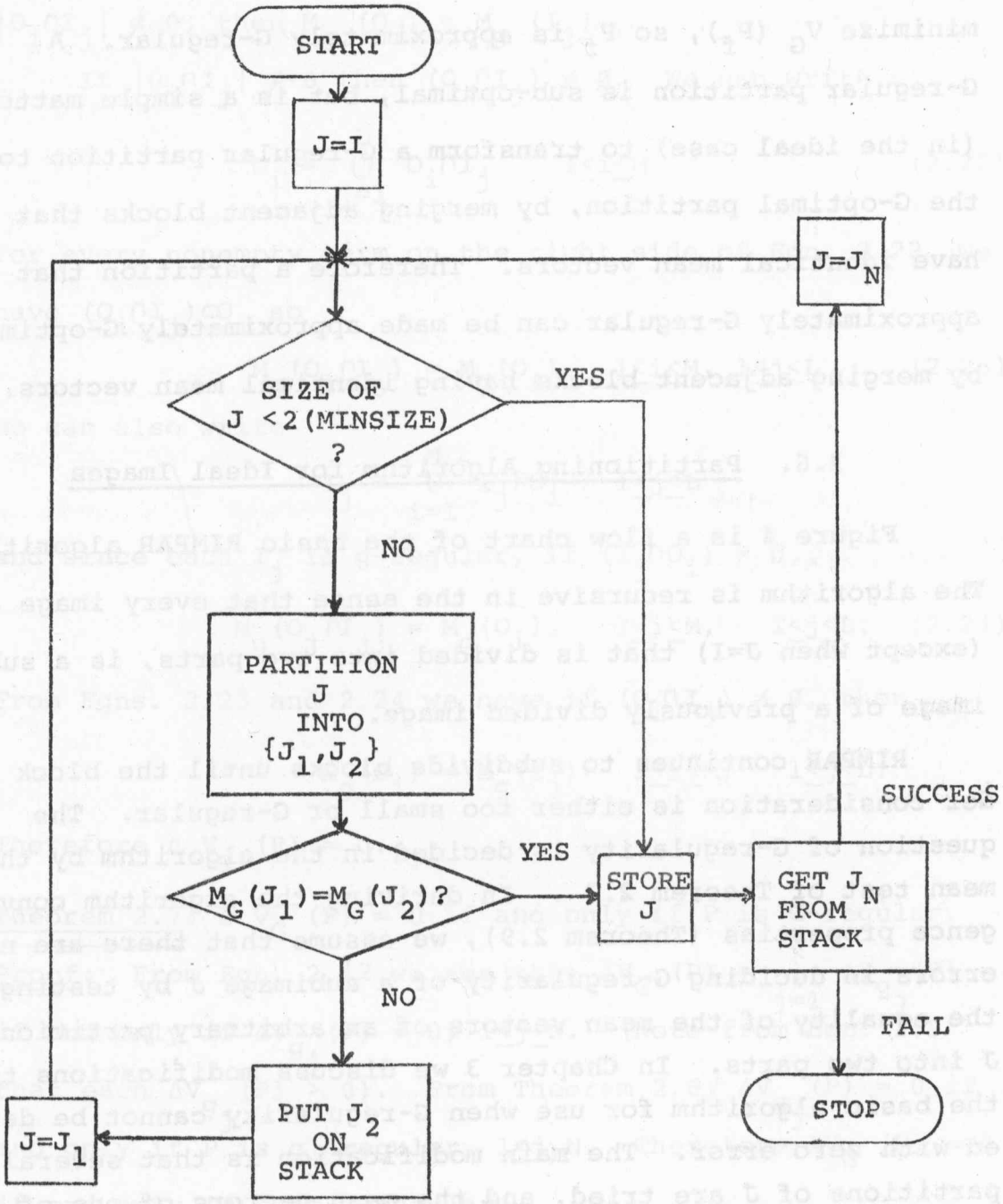


Figure 4. Basic RIMPAR Flow Chart

RIMPAR generates a sequence of partitions $P_0, P_1, \ldots, P_f$; where P_0 is the original unpartitioned image and P_{i+1} is obtained from P_i by dividing one of the blocks of P_i into two parts. In Figure 5 we show some of the members of such a sequence of partitions. The block boundaries are shown as horizontal or vertical lines of apostrophes.

The following theorem shows that each member P_{i+1} of the sequence of partitions is an improvement over the preceeding member P_i.

Theorem 2.8: $V_G(P_{i+1}) < V_G(P_i)$, $0 \le i \le f-1$.

Proof: Consider a partition of I, $P_i=\{I_1, I_2, \ldots, I_{L-1}, J\}$, where J is the block that is partitioned into $\{J_1, J_2\}$ to produce P_{i+1} from P_i. Let $g(\cdot)$ be an arbitrary gray-level function of $G(\cdot)$.

$$V_g(P_i) = \sum_{i=1}^{L-1} \frac{|I_i|}{|I|} S_g^2(I_i) + \frac{|J|}{|I|} S_g^2(J)$$

and

$$V_g(P_{i+1}) = \sum_{i=1}^{L-1} \frac{|I_i|}{|I|} S_g^2(I_i) + \frac{|J_1|}{|I|} S_g^2(J_1) + \frac{|J_2|}{|I|} S_g^2(J_2).$$

Therefore

$$V_g(P_i)-V_g(P_{i+1}) = \frac{|J|}{|I|} S_g^2(J)-\frac{|J_1|}{|I|} S_g^2(J_1)-\frac{|J_2|}{|I|} S_g^2(J_2)$$

$$=\frac{1}{|I|}\left\{|J_1|Z_g^2(J_1)+|J_2|Z_g^2(J_2)-(|J_1|M_g(J_1)+|J_2|M_g(J_2))^2(\frac{1}{|J|})\right.$$

$$\left.-|J_1|Z_g^2(J_1)+|J_1|M_g^2(J_1)-|J_2|Z_g^2(J_2)+|J_2|M_g^2(J_2)\right\}$$

$$=\frac{1}{|I||J|}\left\{|J|(|J_1|M_g^2(J_1)+|J_2|\ M_g^2(J_2))-(|J_1|M_g(J_1)+|J_2|M_g(J_2))^2\right\}$$

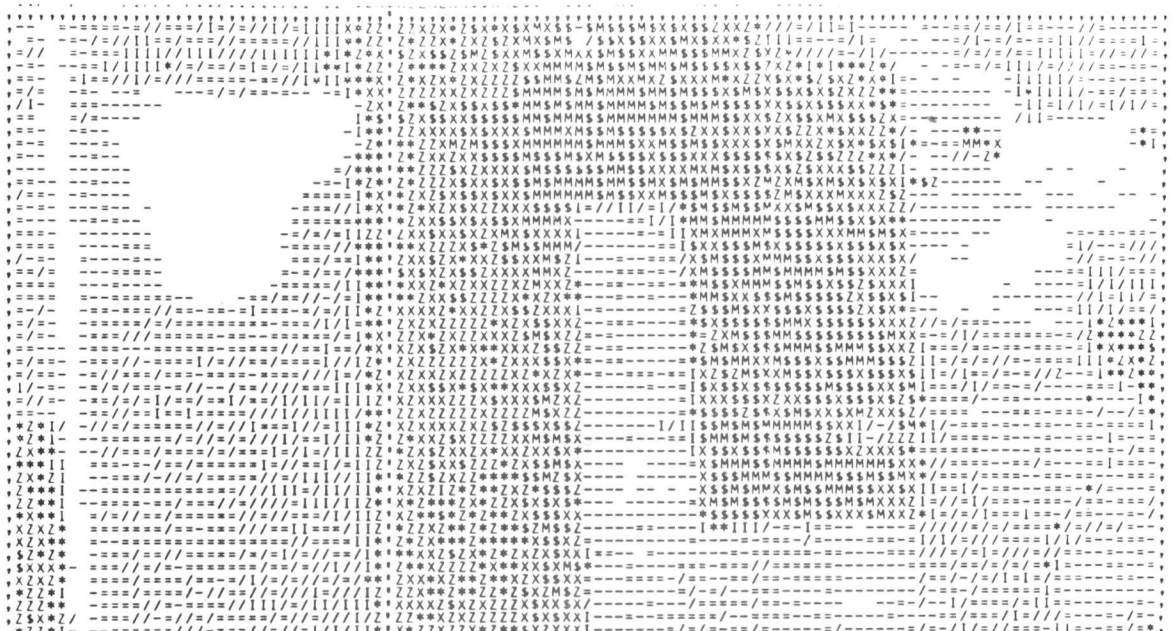


Figure 5a. Sequence of Partitions: P_1

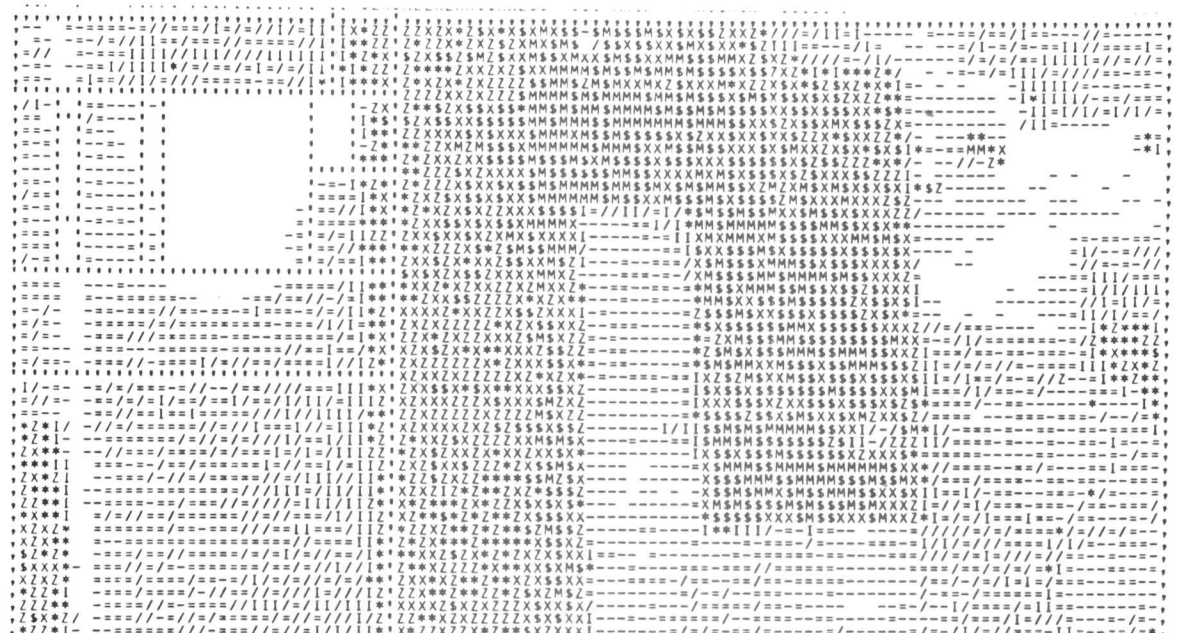


Figure 5b. Sequence of Partitions: P_{15}

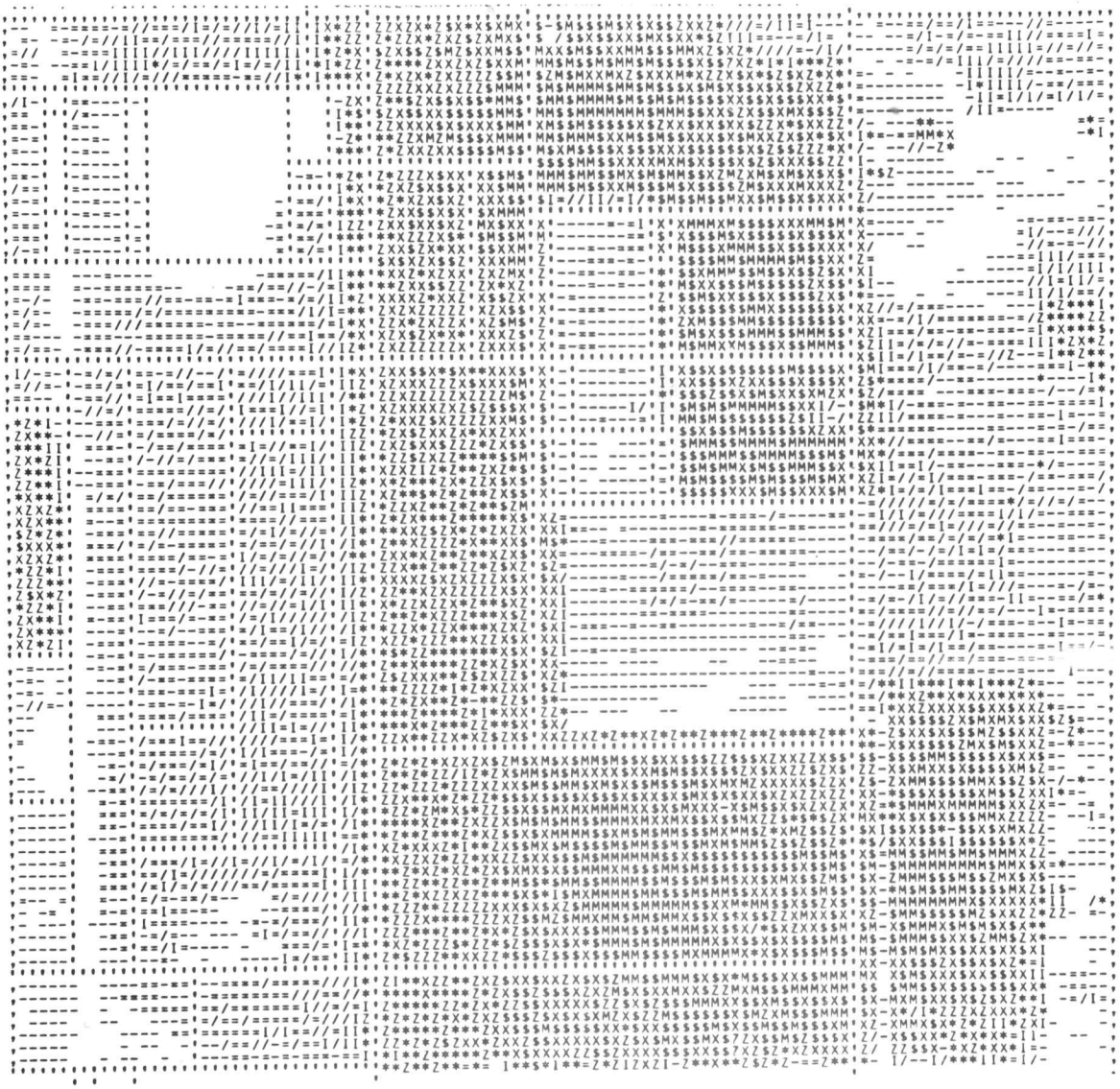


Figure 5c. Sequence of Partitions: P_{56}

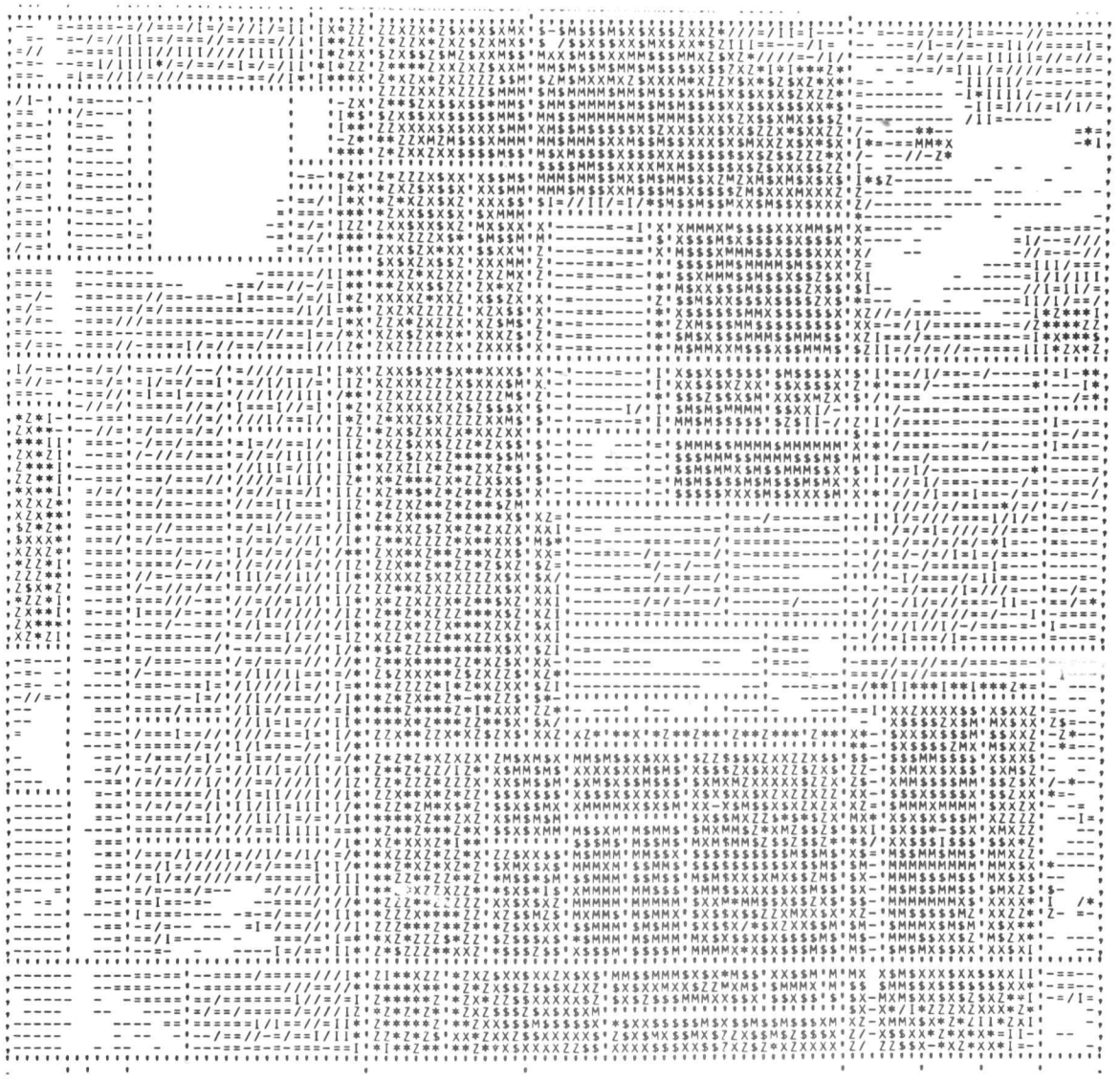


Figure 5d. Sequence of Partitions: P_{111}

By using $|J| = |J_1| + |J_2|$, the above reduces to

$$V_g(P_i)-V_g(P_{i+1})= \frac{|J_1||J_2|}{|I|\ |J|} (M_g(J_1)-M_g(J_2))^2. \tag{2.25}$$

Therefore for every $g(\cdot)$ of $G(\cdot)$, $V_g(P_i)-V_g(P_{i+1}) \geq 0$; and therefore $V_G(P_i)-V_G(P_{i+1}) \geq 0$. $V_G(P_i)=V_G(P_{i+1})$ if and only if $M_G(J_1) = M_G(J_2)$. Therefore, since we assume that RIMPAR decides to partition J, $M_G(J_1)\neq M_G(J_2)$ and $V_G(P_i)> V_G(P_{i+1})$.

When RIMPAR reaches the box labeled "STORE J" in Figure 4, J is accepted as one of the blocks of the final partition P_f. J is accepted for P_f only if the size of J is less than 2x MINSIZE (no acceptable trial partition can then be made), or if $M_G(J_1)=M_G(J_2)$. The algorithm is structured so that at any point in the sequence of algorithm steps, all parts of I that are not included in the already-accepted blocks of P_f are either in J, or on the stack. Therefore no part of I is excluded from testing by the algorithm, and every block I_j in P_f satisfies either

(1) the size of I_j is less than 2(MINSIZE); or

(2) I_j is G-regular.

The following theorem shows how these properties allow partition error to be controlled by MINSIZE.

Theorem 2.9: Consider any image I and any $\epsilon>0$. There exist MINSIZE values such that RIMPAR produces a final partition P_f satisfying $\Delta V_G(P_f)<\epsilon$.

Proof: From Eqn. 2.12 we see that $\Delta V_G(P_f) = \sum_{j=1}^{N} \Delta V_{g_j}(P_f)$ so the theorem holds if $\Delta V_{g_j}(P_f) < \varepsilon/N$, $1 \le j \le N$. For an arbitrary $g(\cdot)$ in $G(\cdot)$ we have

$$\Delta V_g(P_f) = \sum_{i=1}^{M} \sum_{j=1}^{L} \frac{|O_i \cap I_j|}{|I|} (M_g(O_i) - M_g(I_j))^2 .$$

We assume that for $1 \le i \le M$ and $1 \le j \le L$, since gray level values are bounded,

$$(M_g(O_i) - M_g(I_j))^2 < D$$

where D is some constant (an upper bound). Therefore

$$V_g(P_f) < \frac{D}{|I|} \sum_{i=1}^{M} \sum_{j=1}^{L} |O_i \cap I_j| .$$

Now $\sum_{i=1}^{M} \sum_{j=1}^{L} |O_i \cap I_j|$ is just the area of the blocks of P_f that overlap boundaries of objects (objects are blocks in P^*). The object boundaries have a finite total arc length S_T. The smallest side of a block in P_f is MINSIZE (see the algorithm constraints). Therefore the number of overlapping blocks is less than or equal to (S_T / MINSIZE), and since the area of each block is less than $2(\text{MINSIZE})^2$,

$$\sum_{i=1}^{M} \sum_{j=1}^{L} |O_i \cap I_j| < \frac{S_T \cdot 2(\text{MINSIZE})^2}{\text{MINSIZE}} \qquad (2.26)$$

and

$$\Delta V_g(P_f) < \frac{DS_T 2}{|I|} (\text{MINSIZE}) .$$

There if we set

$$\text{MINSIZE} \le \frac{|I|}{2DS_T N} \epsilon \qquad (2.27)$$

then $\Delta V_g(P_f) < \varepsilon/N$ and therefore $\Delta V_G(P_f) < \varepsilon$.

It is interesting to note that to insure that the right hand side of Eqn. 2.26 decreases with MINSIZE, the size of overlapping blocks must decrease more slowly with MINSIZE than the area of overlapping blocks. Therefore RIMPAR must limit block size rather than block area.

CHAPTER 3

IMPLEMENTING RIMPAR

In this chapter we discuss application considerations that arise at four major points where the image model and theoretical results of Chapter 2 depart from practical situations:

(1) In real images adjacent objects may not differ in mean reflectance. In section 3.1 we discuss generalized mean vectors that distinguish between objects that differ only in second or higher moments of their distributions, or in spatial pattern.

(2) The mean test of Theorem 2.3 assumes that both the number of possible mean vectors and the number of possible object boundaries are infinite. Due to spectral and spatial quantization, both of these assumptions are violated in practice. However the chance of the mean test failing by finding a G-regular partition of a non-G-regular subimage decreases as the number of trial partitions increases. In section 3.2.3 we discuss how to choose the best of a number of trial partitions.

(3) In practice, mean vectors can only be estimated from the available finite number of gray level vectors. In section 3.2.2 we discuss a statistical test that is used to decide from these estimates if two mean vectors are equal.

(4) In Theorem 2.9 we demonstrated the convergence of RIMPAR in an ideal situation. In section 3.3 we interpret this theorem and the role of MINSIZE for real images.

Throughout this chapter we consider real images which contain a finite number of points. To apply the results of Chapter 2 to real images, we will refer to the "number of points" of an image instead of image "area". Thus in this chapter, $|J|$ will denote the number of points in J.

3.1. Sufficient Gray-Level Vectors

In order for RIMPAR to produce a partition that is meaningful, the blocks of the G-optimal partition P* should correspond to areas of the target we wish to consider as single entities. A gray-level vector G for which P* satisfies this goal is called sufficient. In other words, G is sufficient if for any adjacent areas H_1 and H_2 in I such that H_1 and H_2 represent different target entities, $M_G(H_1) \neq M_G(H_2)$.

The first logical candidate for gray-level functions is the set $G_1(\cdot)=(g_1(\cdot), g_2(\cdot), \ldots, g_N(\cdot))$ corresponding to the reflectance in the N channels of the multispectral image. In many applications $G_1(\cdot)$ or even a subset of $G_1(\cdot)$ may be sufficient. However, if for adjacent areas H_1 and H_2, $M_{g_i}(H_1)=M_{g_i}(H_2)$, $1 \leq i \leq N$, then $G_1(\cdot)$ is not sufficient. This would be the case if H_1 and H_2 had identical means in all channels, but different variances in some channels. To handle such situations we propose sets of gray-level function $G_k(\cdot)=(g_1^k(\cdot), g_2^k(\cdot), \ldots, g_N^k(\cdot))$ where $g_i^k(X)$ is an unbiased estimate of $E[(g_i(Y))^k / Y \in N(X)]$ and N(X) is a small neighborhood of X, consisting perhaps of X and the eight nearest neighbors of X. If any H_1 and H_2 have gray-level distributions that differ in at least one of the first r moments of at least one of the first N channels, then $G(\cdot)=(G_1(\cdot), G_2(\cdot), \ldots, G_r(\cdot))$ is sufficient.

The gray-level vector can be further generalized to distinguish between adjacent target areas that differ in either

pattern or dependence among neighboring points. The procedure in these cases is to define gray-level functions with mean values that reflect these differences.

3.2. Testing G-Regularity

Since a real image has a finite number of image points, we can only estimate the mean vector for any subimage J. Therefore testing G-regularity cannot be done with zero error, and it is desirable to find a test that tends to minimize testing errors.

Recall that to determine if J is G-regular, we partition J into $\{J_1, J_2\}$ and ask "$M_G(J_1)=M_G(J_2)$?" In this subsection we discuss the implementation of this test.

3.2.1 Partitioning the Gray-Level Vector

To determine $M_G(J_1)\neq M_G(J_2)$ it is sufficient to determine $M_{g_i}(J_1)\neq M_{g_i}(J_2)$ for some $g_i(\cdot)\varepsilon G(\cdot)$. Therefore it is not necessary to use all gray-level functions to decide $M_G(J_1)\neq M_G(J_2)$. If we partition $G(\cdot)$ into subvectors $(G_1(\cdot), G_2(\cdot), \ldots, G_p(\cdot))$, then we can test for G-regularity as shown in Figure 6. This testing scheme has two advantages:

(1) All gray level functions need not be evaluated for each J.

(2) Testing subvectors instead of individual gray-level functions allows the use of correlation between channels to improve estimates.

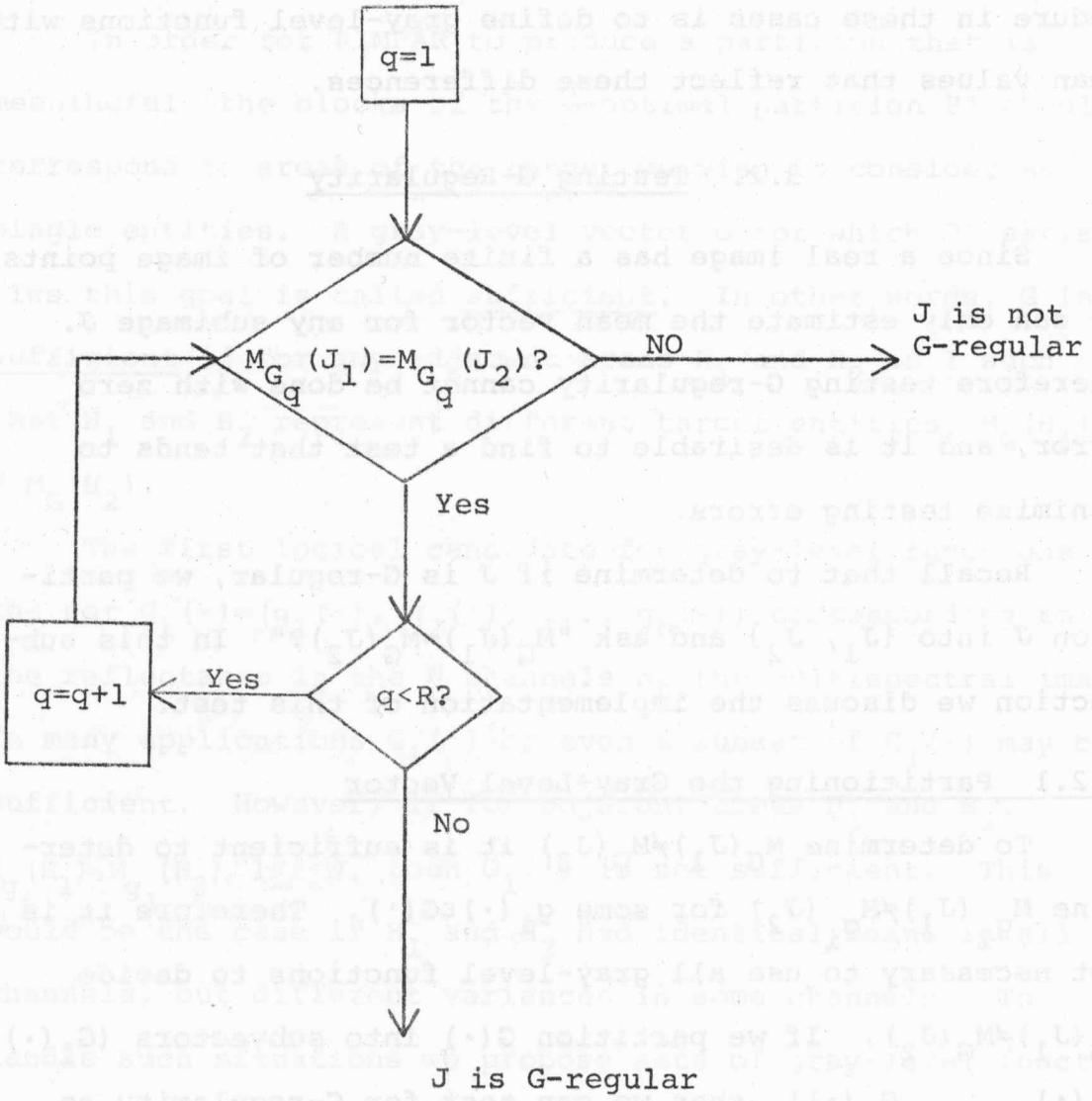


Figure 6. G-Regularity Test

3.2.2 A Test Statistic

We calculate an estimate $\hat{M}_{G_i}(J_k)$ of $M_{G_i}(J_k)$ by

$$\hat{M}_{G_i}(J_k)=\frac{1}{|J_k|}\sum_{X\varepsilon J_k} G_i(X) \quad k=1,2 \tag{3.1}$$

Throughout this section $|K|$ stands for the number of points in any image K. Let

$$S=\frac{1}{|J|-2}\{\sum_{X\varepsilon J_1}(G_i(X)-\hat{M}_{G_i}(J))(G_i(X)-\hat{M}_{G_i}(J_1))^T+\sum_{X\varepsilon J_2}(G_i(X)-\hat{M}_{G_i}(J_2))$$

$$(G_i(X)-\hat{M}_{G_i}(J_2))^T\}$$

and

$$T^2=\frac{|J_1||J_2|}{|J|}(\hat{M}_{G_i}(J_1)-\hat{M}_{G_i}(J_2))^T S^{-1}\cdot(\hat{M}_{G_i}(J_1)-\hat{M}_{G_i}(J_2)) \tag{3.2}$$

The statistic T^2 has been used [27] to test the equality of the mean vectors of two multivariate Gaussian distributions in the following way: the two mean vectors are considered equal (at significance level α) if

$$T^2<\frac{(|J|-2)r}{(|J|-r-1)}F_{r,\ |J|-r-1}(\alpha) \tag{3.3}$$

where r is the number of gray-level functions in $G_i(\cdot)$, and $F_{r,|J|-r-1}(\alpha)$ is the upper $100\alpha\%$ point of the F distribution with r and $|J|-r-1$ degrees of freedom. In our application we have no guarantee that the distributions $f(G_i(X)/X\varepsilon J_1)$ and $f(G_i(X)/X\varepsilon J_2)$ are multivariate Gaussian, but the distributions are probably close enough to Gaussian to allow the use of T^2 with some success. Using Eqn. 3.3 instead of comparing T^2 to a fixed threshold has the following advantages:

(1) Eqn. 3.3 takes into account the number of gray level vectors used to estimate the mean vectors. Since estimation with small sample size is unreliable, a small $|J|$ increases the threshold to maintain a constant probability of false rejection of the equality hypothesis.

(2) The threshold of Eqn. 3.3 also reflects the dependence of T^2 on the number of channels r.

3.2.3 Efficiency of Partitions

If it is found that $M_{G_i}(J_1) \neq M_{G_i}(J_2)$, then J is partitioned into J_1 and J_2. Although any partition $\{J_1, J_2\}$ is suitable for the mean test, some partitions of J have advantages to offer in later steps of RIMPAR. For example, if J_1 happens to be G-regular, then J_1 will not be partitioned later. To evaluate various partitions of J (with respect to $G_i(\cdot)$) we define the partition <u>efficiency</u> $\eta_{G_i}(J_1, J_2)$ by

$$\eta_{G_i}(J_1,J_2)=\frac{|J_1||J_2|}{|J|}(M_{G_i}(J_1)-M_{G_i}(J_2))^T(M_{G_i}(J_1)-M_{G_i}(J_2)). \tag{3.4}$$

The partition efficiency indicates the improvement of the partition criterion due to partitioning J into $\{J_1, J_2\}$. This can be shown by referring to Eqn. 2.25 and noting that

$$V_{G_i}(P_j)-V_{G_i}(P_{j+1})=\frac{1}{|I|}\eta_{G_i}(J_1,J_2)\ ,$$

when P_{j+1} is derived from P_j by replacing J by $\{J_1, J_2\}$.

To achieve partitions with high efficiency, RIMPAR tries several partitions of each J, and picks the partition with the highest efficiency. The trial partitions are generated by (K_D-1) equally-spaced horizontal lines, and (K_D-1) equally-spaced vertical lines. Here K_D is the same as in Sec. 2.6. The test for G-regularity is applied only to the most efficient partition of J.

3.2.4 Storage and Retrieval of Subimages

In the computer implementation of the G-regularity test, much processing time is used to retrieve the gray levels of J, and various subimages of J, for use in calculating the $\eta_{G_i}(J_1,J_2)$ and T^2. The following steps are taken to minimize retrieval time.

(1) The total image I is initially stored on tape. Before partitioning, I is transferred to a disc that allows almost direct access to any line of gray-levels.

(2) If at any point in RIMPAR we have $|J| <$ MAXCOR, then the gray-levels of J are transferred to core. All subimages of J can then be retrieved from core instead of disc. Arrays in the FORTRAN program are dynamically allocated to maximize MAXCOR, the amount of core available for image storage. Typically 150k bytes are available for this purpose on an IBM 360/67 with 256k total available core.

(3) As J is retrieved from either core or disc, the sums and sums of squares of the gray-levels of blocks of J formed by $2(K_D-1)$ equally-spaced horizontal and vertical lines are calculated. These partial sums are stored in core and are used to calculate the $\eta_{G_i}(J_1, J_2)$'s for the $2(K_D-1)$ trial partitions, and T^2 for the most efficient partition. The points of each J are thus read only once.

3.3. Choice of MINSIZE

In Theorem 2.9 the convergence of RIMPAR was shown for the case of continuous images in which G-regularity could be decided with zero error. The role of MINSIZE in this theorem was to prevent the expenditure of too much effort in approximating arbitrary object boundaries by horizontal and vertical line segments. This theorem demonstrates the basic ability of RIMPAR to successfully partition real images, but real images present circumstances that differ from those in the theorem statement.

Because real images are spatially quantized there is a degenerate G-regular partition---the partition in which each block is a single image point. Thus the size of the smallest partition block has a natural lower limit in a digitized image.

Since the T^2 test allows for statistical uncertainty, we prevent the partitioning of not only noisy G-regular blocks, but also blocks that are almost G-regular because of boundary

overlap. Therefore the T^2 test controls to some extent the degree of boundary overlap.

Thus the role of MINSIZE is diminished when real images are considered. However, there is a definite need for constraining partition block size in some images. This need occurs when object definition is ambiguous because the objects of interest are composed of smaller objects. In these cases, if MINSIZE is set to the smallest block size that gives reasonable estimates of the object means, the breaking up of textured or compound objects into their component parts can be prevented.

CHAPTER 4
CLASSIFICATION TECHNIQUES

Several classification algorithms are used in experiments described later in this report, so in this chapter we discuss the details of these algorithms. When probability density functions are involved, we assume they are Gaussian for computational simplicity. Previous work in classification has shown that this assumption gives good results for many data sets [12].

Two main types of classifiers are discussed: supervised classifiers and unsupervised classifiers. Within each of these types there are per-point classifiers and sample classifiers. We refer to the unit of data that is classified with one decision as a data unit. In per-point classifiers the data unit is a single image point X, and in sample classifiers the data unit is a set Q of image points.

4.1. Supervised Classification

The basic procedure in supervised classification can be stated as follows: We are given a set of data units, (called a training set) each labeled as belonging to one of m classes of interest $\omega_1, \omega_2, \ldots, \omega_m$. Using the gray level vectors associated with these data units, we design a classifier that assigns a class label to any unknown data unit.

4.1.1 Supervised Per-Point Classification

In per-point classification we classify a single image point X on the basis of its gray level vector G(X). (We use here the notation of section 2.1.) The classification rule used is an approximation to the Bayes maximum likelihood decision rule, assuming a 0-1 loss function and equal class prior probabilities [6]. To implement this rule we assume the probability density functions

$$f(G(X)/X\varepsilon\omega_i), \quad 1\leq i\leq m$$

are Gaussian with mean vectors M_i, $1\leq i\leq m$, and covariance matrices K_i, $1\leq i\leq m$. Since the M_i's and K_i's are unknown, we estimate them from the training set $\{T_{ij}, 1\leq i\leq m, 1\leq j\leq t_i\}$. each T_{ij} is the jth point known to belong to ω_i. The estimates for the M_i's and K_i's are

$$\hat{M}_i=\frac{1}{t_i}\sum_{j=1}^{t_i} G(T_{ij})$$

$$\hat{K}_i=\frac{1}{t_i-1}\sum_{j=1}^{t_i} (G(T_{ij})-\hat{M}_i)(G(T_{ij})-\hat{M}_i)^T.$$

The classification rule is: decide a point X is from ω_j if

$$d_j(X)\geq d_i(X), \quad 1\leq i\leq m, \quad i\neq j;$$

where

$$d_j(X)=-\log[\det(\hat{K}_j)]-(G(X)-\hat{M}_j)^T\hat{K}_j^{-1}(G(X)-\hat{M}_j). \qquad (4.1)$$

(Throughout this report, det(A) will stand for the determinant of a matrix A, and A^T will stand for the transpose of a matrix A).

4.1.2 Supervised Sample Classification

In sample classification we classify a set Q of image points on the basis of the set of gray level vectors $\{G(X) \mid X \in Q\}$. The data units are classified using a statistical distance measure, the approximate Bhattacharyya distance [12] for Gaussian density functions. The training set is $\{R_{ij},\ 1 \le i \le m,\ 1 \le j \le t_i\}$, where R_{ij} is the jth set of image points known to belong to ω_i. The training set is characterized by mean vectors M_i, $1 \le i \le m$, and covariance matrices K_i, $1 \le i \le m$, which are estimated by

$$\hat{M}_i = \frac{1}{P_i} \sum_{j=1}^{t_i} \sum_{X \in R_{ij}} G(X)$$

$$\hat{K}_i = \frac{1}{P_i - 1} \sum_{j=1}^{t_i} \sum_{X \in R_{ij}} (G(X) - \hat{M}_i)(G(X) - \hat{M}_i)^T$$

where

$$P_i = \sum_{j=1}^{t_i} (\text{Number of points in } R_{ij}).$$

For the set of image points Q to be classified we estimate

$$\hat{M}_Q = \frac{1}{q} \sum_{X \in Q} G(X) \qquad (4.2)$$

$$K_Q = \frac{1}{q-1} \sum_{X \in Q} (G(X) - M_Q)(G(X) - M_Q)^T$$

where q=number of points in Q. The decision rule is: decide Q is from ω_j if

$$B_j(Q) < B_i(Q) \quad 1 \le i \le m, \quad i \ne j,$$

where

$$B_i(Q)=\frac{1}{8}(\hat{M}_Q-\hat{M}_i)^T\hat{K}^{-1}(\hat{M}_Q-\hat{M}_i)+\frac{1}{2}\text{Ln}\frac{\det(\hat{K})}{[\det(\hat{K}_Q)]^{\frac{1}{2}}[\det(\hat{K}_i)]^{\frac{1}{2}}} \tag{4.3}$$

$$\hat{K}=\frac{\hat{K}_Q+\hat{K}_i}{2}$$

4.2. Unsupervised Classification

In unsupervised classification no training set is provided. In our application, the unsupervised classification problem is: given a set F of data units, partition F into m classes such that the data units associated with any given class are "similar" to each other, and relatively "different" from the data units associated with other classes. This type of unsupervised classification is also called clustering.

A clustering algorithm that tends to minimize the differences between members of the same class is shown in Figure 7 [12]. This algorithm is used for both per-point and sample classification.

4.2.1 Unsupervised Per-Point Classification

In per-point classification, the class centers are mean vectors $\hat{M}_i$, $1\leq i\leq m$. The class centers are initialized to the gray level vectors of m arbitrary image points in the set F to be clustered. The class membership of a point X is decided as follows: X is in ω_j if

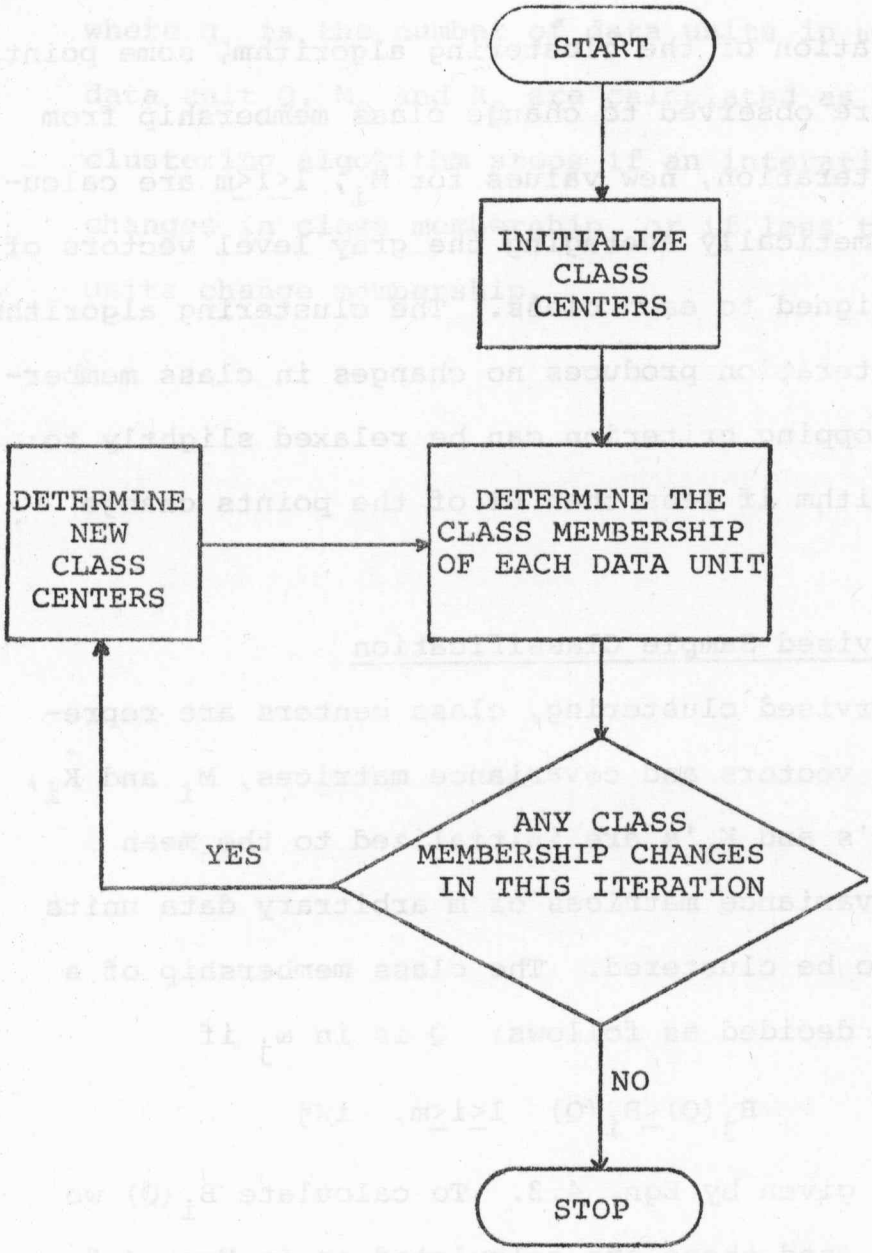


Figure 7. Clustering Algorithm

$$(\hat{M}_j - X)^T (\hat{M}_j - X) \leq (\hat{M}_i - X)^T (\hat{M}_i - X) \quad 1 \leq i \leq m, \quad i \neq j.$$

If, in an iteration of the clustering algorithm, some points in the set F are observed to change class membership from the previous iteration, new values for $\hat{M}_i$, $1 \leq i \leq m$ are calculated by arithmetically averaging the gray level vectors of the points assigned to each class. The clustering algorithm stops if an interation produces no changes in class membership. This stopping criterion can be relaxed slightly to: stop the algorithm if less than P% of the points change membership.

4.2.2 Unsupervised Sample Classification

In unsupervised clustering, class centers are represented by mean vectors and covariance matrices, $\hat{M}_i$ and $\hat{K}_i$, $1 \leq i \leq m$. The $\hat{M}_i$'s and $\hat{K}_i$'s are initialized to the mean vectors and covariance matrices of m arbitrary data units in the set F to be clustered. The class membership of a data unit Q is decided as follows: Q is in ω_j if

$$B_j(Q) \leq B_i(Q) \quad 1 \leq i \leq m, \quad i \neq j$$

where $B_i(Q)$ is given by Eqn. 4.3. To calculate $B_i(Q)$ we need $\hat{M}_Q$ and $\hat{K}_Q$, and these are calculated as in Eqn. 4.2. If in an iteration of the clustering algorithm some data units in the set F are observed to change class membership, new values for M_i and K_i, $1 \leq i \leq m$ are calculated as follows:

$$\hat{M}_i = \frac{1}{q_i} \sum_{Q \in \omega_i} \hat{M}_Q$$

$$\hat{K}_i = \frac{1}{q_i} \sum_{Q \in \omega_i} \hat{K}_Q$$

where q_i is the number of data units in ω_i, and for each data unit Q, $\hat{M}_Q$ and $\hat{K}_Q$ are calculated as in Eqn. 4.2. The clustering algorithm stops if an interation produces no changes in class membership, or if less than P% of the data units change membership.

CHAPTER 5
EXPERIMENTAL RESULTS

The results of this chapter are divided into several sets. In the first set we study the effects of the RIMPAR parameters SLEV, K_D, MINSIZE, and the number of partitioning channels. In the second set we investigate some alternative methods for using multiple partitioning channels. In the third set of experiments we use RIMPAR followed by supervised sample classification to classify agricultural areas in aircraft and satellite images. In the fourth set we use RIMPAR followed by unsupervised classification to isolate urban areas in a satellite image. In the last set of experiments, RIMPAR followed by unsupervised sample classification is studied as a method for isolating the lung in a digitized chest radiograph, and head outline and facial features in a digitized photograph of a girl.

These experiments were performed at the Laboratory for Applications of Remote Sensing (LARS), Purdue University. The computer used was an IBM 360/67 time-shared computer. The computer times reported for the experiments are stated in seconds of virtual CPU time. In the reported experiments, input to and output from the author's programs, and comparisons with conventional algorithms, were

greatly facilitated by LARSYS [30], a software system developed at LARS for the analysis of remotely sensed data.

In refering to the data sets used in our experiments we will use LARS run numbers such as "72032803". Runs at LARS are stored in digital form as lines amd columns. A convenient way to refer to a subset of a run is LINES(first line,last line,line increment), COLS(first column,last column, column increment). For example, one of the images we discuss is LARS RUN(69002901), LINES(860,1660,2), COLS(1, 221,2). In each run there are measurements in various spectral bands (channels), and these channels will be refered to by number. In Table 6 we list the wavelengths associated with the channel numbers and other information about the data sets we consider. In general, partitioning and classification are carried out using separate channel sets, so we will refer to the channel set used for partitioning as PSET(C_1,C_2, ... , C_N), and the channel set used for classification as CSET(C_1,C_2, ... C_M). For example if channels 7 and 8 are used to partion an image, and channels 5,7,9, and 12 are used to classify the image, we will write PSET(7,8), CSET(5,7,9,12).

Supervised classification is used in some of the experiments. In these instances classification accuracy can be measured by comparing the results of the classification algorithm to areas of known classification. The percentage of correctly classified points is calculated two ways. Overall percent correct is 100X(number of

correctly classified image points)/(total number of image points of known classification). Another classifier performance measure is percent correct by class. To calculate this percentage, the overall percent correct is calculated for each class, then the individual class performances are arithmetically averaged. When areas of known classification are used to train the classifier, they are called training fields. Other areas of known classification are called test fields. We report training and test field accuracy separately and use the notation TRAIN(overall percent correct / percent correct by class), TEST(overall percent correct / percent correct by class). For example TRAIN(99.3/80.0), TEST(96.4/80.7). Training field accuracy shows how well the measurements and the classifier structure are able to characterize and distinguish the classes of interest. Test field accuracy indicates how well this characterization generalizes to non-training data.Note that for a give classification experiment, overall percent correct and percent correct by class can by quite different if each class is not represented by nearly the same number of image points of known classification.

5.1 PARAMETER EFFECTS

In these experiments we measure the effects various parameter values have on partitioning. The test image used in this section is LARS RUN(71053900), LINES(190,540,2), COLS(10,322,2). Partition quality in these experiments is measured by the training field classification accuracy

obtained when RIMPAR is followed by a supervised sample classifier. The classification program also contains a supervised per-point classifier for use with single-point partition blocks and with blocks with so few points that the estimated covariance matrix is singular. In the latter case classification is carried out by classifying the block mean vector as if it were a single gray level vector. Classification accuracy indicates partion quality because a poor partition contains blocks that overlap object boundaries. Since the classification is based upon statistics calculated from the points in a block, these overlapping blocks, which contain points from at least two objects, will have many or all of their points misclassified. Classification into 5 classes was performed. The classes of interest were corn, soybeans, forage, forest, and water. Twenty-five training fields were used, with a total of 1387 image points. The range of points per training field is 2 to 190. The classification channel set used in this section is CSET(5,7,9,12). Classification with a supervised per-point classifier gave training field performance TRAIN(97.5/98.0).The relatively high performance indicates that the channel set is good and that the assumption of Gaussian statistics used in the classifier is reasonable. For use in some of the experiments of this section, subsets of the 12 available channels were evaluated using a feature selection algorithm [31]. The channel sets selected for their ability to separate the 5 classes as represented by the

training fields are (9), (7,8), (7,8,12), and (7,8,11,12). Preliminary experiments indicated that PSET(7,8), K_D=20, MINSIZE=1, SLEV=.01 give good partition accuracy, so these parameter values were a basis around which the parameter values were changed.

5.1.1 SLEV

In this section the α of Eqn. 3.3 is called SLEV. SLEV is the significance level in the T^2 test used to decide if two mean vectors are equal. SLEV is related to the probability of false rejection of the equality hypothesis. Therefore high SLEV values cause relatively many mean vector pairs to be considered equal, and low SLEV values cause relatively few mean vector pairs to be considered equal.

In Table 1 we show the results of using several SLEV values. The partions produced for all SLEV values seem fairly accurate, and SLEV=.01 gave the best results. For SLEV=.005 some of the small training fields for the class water were not isolated from their surrounding and therefore were erroneously classified. For SLEV=.1 objects were apparently subdivided, leading to decreased accuracy in the estimation of block statstics. Since SLEV=.01 gives good classification results with relatively few blocks, this seems to be a good choice for the parameter value.

5.1.2 K_D

In partioning each subimage J, K_D horizontal and K_D vertical partitions are tried. A relatively large K_D will therefore allow some partition lines to better approximate object

Table 1. Effect of SLEV

SLEV	Classification Accuracy	Number of Partition Blocks	Partitioning Time (sec)	$V_g(P)$ (Channel 7)
.100	TRAIN(95.6/95.7)	2686	451	55.7
.050	TRAIN(99.2/96.6)	1918	425	74.6
.010	TRAIN(99.8/99.8)	1164	389	102.
.005	TRAIN(99.7/92.0)	976	376	106.

MINSIZE=1 K_D=20 PSET(7,8) CSET(5,7,9,12)

boundaries, but at a cost of increased partioning time. In Table 2 the effect of changing K_D is shown. These results show that partitioning time is roughly proportional to K_D. The best accuracy occurs with K_D=20, and there is some degradation for K_D=10 and K_D=15. The inaccuracies here were due to the small water areas. With K_D=5 good accuracy is obtained, but since this goes against the trend set by the other experiments, this is probably due to the chance coincidence of a trial partition line and object boundary. To insure good object boundary fit, K_D=20 seems to be a good choice. In other images a lower K_D value may be used to decrease computation time if boundary fit is not judged to be important enough to warrant the extra computation time.

5.1.3 Number of Partitioning channels

In these experiments the channel sets found by the feature selection algorithm were used in partiioning the test image. The results are shown in Table 3. The best results were obtained with the set (7,8). The fact performance went down when a third channel was added probably reflects the difference in classification channel requirements , for which the feature selection algorithm was designed, and partitioning channel requirements.

5.1.4 MINSIZE

Recall from the discussion of section 3.1 that the main function of MINSIZE in practical image partitioning is to prevent the splitting of textured objects. Since good results were obtained with MINSIZE=1 in this section, this conjecture

Table 2. Effect of K_D

K_D	Classification Accuracy	Number of Partition Blocks	Partitioning Time (Sec)	$V_g(P)$ (Channel 7)
5	TRAIN(98.8/98.6)	1029	164	120.
10	TRAIN(98.6/85.4)	1156	238	119.
15	TRAIN(99.4/84.0)	1199	318	93.2
20	TRAIN(99.8/99.8)	1164	389	102.

MINSIZE=1 SLEV=.01 PSET(7,8) CSET(5,9,7,12)

Table 3. Effect of PSET

PSET	Classification Accuracy	Number of Partition Blocks	Partitioning Time (Sec)	$V_g(P)$ (Channel 7)
9	TRAIN(96.6/76.6)	1043	180	120.
7,8	TRAIN(99.8/99.8)	1164	389	102.
7,8,12	TRAIN(97.9/83.1)	1354	657	88.5
7,8,11,12	TRAIN(98.2/98.7)	1444	956	79.8

MINSIZE=1 K_D=20 SLEV=.01 CSET(5,7,9,12)

seems to be supported. Some experiments were run, however, to check the effect of increasing MINSIZE. The results are shown in Table 4. Accuracy is observed to increase slightly with MINSIZE=2. This increase is due to the tendency of RIMPAR to make a classification results image relatively homogeneous. Since classification accuracy is measured assuming homogeneous fields, the performance evaluation is somewhat biased. Apparently MINSIZE=2 does not prevent the location of the small water areas, but some of these areas are not found with MINSIZE=3. We conclude here that MINSIZE=1 is a good value to use, unless there are textured objects.

5.2 Methods of Using Multiple Partioning Channels

Since partitioning time goes up rapidly with the number of channels used, we present here two alternatives to the multivariate T^2 test. We used the same test image and classification test procedure here as in the previous section. The first alternative method uses a sequential G-regularity test as shown in Figure 6, with q=1 and R=4. In other words 4 channels were used, and a univariate T^2 test was applied sequentially to the 4 channels. It was found that with this technique SLEV=.01 produced too many partition blocks, judging from the results of the previous section. Apparently many subimages that should be considered G-regular were different enough in at least one of the channels to be divided by the algorithm. To offset this tendency, we used SLEV=.005 .

Table 4. Effect of MINSIZE

MINSIZE	Classification Accuracy	Number of Partition Blocks	Partitioning Time (Sec)	$V_g(P)$ (Channel 7)
1	TRAIN(99.8/99.8)	1164	389	102.
2	TRAIN(100./100.)	789	319	169.
3	TRAIN(99.7/92.0)	604	266	211.

K_D=20 SLEV=.01 PSET(7,8) CSET(5,7,9,12)

The second method investigated is called classify-then-partition. To use this technique the image was first classified by a supervised per-point classifier. The classification results were then used to generate a single-channel image by associating gray levels with class labels. This association was carried out such that the gray levels were about equally spaced throughout the 0-255 gray level range of the normal data. To measure the accuracy of the partition produced by this method, the partition blocks generated were used to retrieve block statistics from the original image. The blocks were then classified using these statistics and the training field accuracy was computed. SLEV=.1 was used in partitioning the classified image. SLEV=.01, which was found to be a good value in the last section, was found to give too many blocks in this case. A possible explanation for this performance is that the single-channel image produced from the classified image has an unusually high variance because of the way gray levels were assigned to class labels.

A comparison of the 3 multichannel partitioning methods is shown in Table 5. The accuracy in all cases is good, indicating that the 4 channels of information was used about equally well by all methods. The classify-then-partition method is much faster then the other two. This method is clearly the best approach when the purpose of partitioning is to get reduced classification results storage. However, since classification in this case is carried out per-point, the advantages of sample classification are not available

Table 5. Comparison of Multichannel Methods

Method	Classification Accuracy	Number of Partition Blocks	Partitioning Time (Sec)
Multivariate T^2 SLEV=.01	TRAIN(98.2/98.7)	1440	956
Sequential T^2 SLEV=.005	TRAIN(96.8/97.5)	1808	729
Classify-then-Partition SLEV=.10	TRAIN(97.8/99.1)	1038	215

MINSIZE=1 K_D=20 PSET(7,8,11,12) CSET(5,7,9,12)

with this method.To classify blocks using their gray levels one of the other two techniques must be used. It appears that the sequential method is a good choice in these cases. A disadvantage to the sequential method is that corelation between channels is not used. The advantage of this method is relatively small processing time. As the number of channels increases, the multivariate T^2 test goes up about quadratically in computation time, but the sequential test can be expected to increase only linearly in processing time.

5.3 Classifying Agricultural Areas

In this section we describe experiments in which RIMPAR followed by sample classification is used to classify agricultural area in 3 aircraft images and two satellite images. In Table 6 a description of the test images is given. The classification results were evaluated by computing training and test field accuracy. The characteristics of these fields are given in Table 7. In Figure 8 we show a digital display photograph of one of the satellite test images. The image shown in Figure 2 is an example of an aircraft image.

In each image, partitioning was carried out using the two channels chosen as best by the feature selection algoritm discussed earlier.The RIMPAR parmeters used in all cases were MINSIZE=1, K_D=20, and SLEV=.01. Classification was performed using the best set of 4 channels as chosen by the feature selection algorithm for the aircraft data, and PSET(2,3) and CSET(2,3) were used for the satellite data.

Table 6. Test Image Data

Run Number	Test Image	Data Source	Channel Spectral Bands
69002901	LINES(860,1660,2) COLS(1,221,2) Purdue Flt. Ln. 24	Aircraft Altitude 2400 feet Date 6/25/69	1/0.40-0.44 7/0.66-0.72 2/0.46-0.48 8/0.72-0.80 3/0.52-0.55 9/0.80-1.00 4/0.55-0.58 10/1.00-1.40 5/0.58-0.62 11/1.50-1.80 6/0.62-0.66 12/2.00-2.60
66000600	LINES(61,812,2) COLS(1,217,2) Purdue Flt. Ln. C1	Aircraft Altitude 2600 feet Date 6/28/66	1/0.40-0.44 7/0.55-0.58 2/0.44-0.46 8/0.58-0.62 3/0.46-0.48 9/0.62-0.66 4/0.48-0.50 10/0.66-0.72 5/0.50-0.52 11/0.72-0.80 6/0.52-0.55 12/0.80-1.00
71053900	LINES(200,1036,2) COLS(1,221,2) Corn Blight Flt. Ln. 210	Aircraft Altitude 5000 feet Date 8/13/71	1/0.46-0.49 7/0.61-0.70 2/0.48-0.51 8/0.72-0.92 3/0.50-0.54 9/1.00-1.40 4/0.52-0.57 10/1.50-1.80 5/0.54-0.60 11/2.00-2.60 6/0.58-0.65 12/9.30-11.7
72032803A	LINES(381,500,1) COLS(998,1307,1) ERTS 101716093 Ill.	Satellite Altitude 580 miles Date 8/9/72	1/0.50-0.60 2/0.60-0.70 3/0.70-0.80 4/0.80-1.10
72032803B	LINES(680,770,1) COLS(1070,1380,1) ERTS 101716093 Ill.	Satellite Altitude 580 miles Date 8/9/72	1/0.50-0.60 2/0.60-0.70 3/0.70-0.80 4/0.80-1.10

Table 7. Training and Test Field Data

Run	Classes of Interest	Points Per Field Range/Average	Number of Train. Field Points	Number of Test Field Points
69002901	Corn, Soybeans, Wheat, Forage, Forest, Water	(9-754)/221	2727	5237
66000600	Corn, Soybeans, Wheat, Oats, Clover, Alfalfa, Bare Soil	(56-1363)/410	4459	13562
71053900	Corn, Soybeans, Forage, Forest, Water	(2-360)/64	2727	5237
72032803A	Corn, Soybeans, Other (Other Vegetation)	(2-112)/18	850	4842
72032803B	Corn, Soybeans, Other (Other Vegetation)	(2-228)/18	1309	1409

Channel 2

Channel 3

Figure 8. Satellite Test Image

In Table 8 we compare the RIMPAR classification results with results obtained with the per-point classifier using the same classification channels. Overall, the accuracy of the two classifiers is about equal. RIMPAR performs significantly better in Run 72032803B, Evidently in this image the class characteristics are such that sample classification is better than per-point classification. In Run 69002901 the per-point classifier performs more accurately than RIMPAR. In this test image errors made by RIMPAR in finding object boundaries were apparently great enough to offset any advantage gained by sample classification. In the other test images, the two classifiers were about equal in accuracy.

Per-point classification is clearly much faster than RIMPAR in these experiments. However for particular images RIMPAR might perform faster than indicated here. For example if a degradation in boundary fit were acceptable,we could set K_D to a small value and decrease processing time. It is also possible to decrease partitioning time by using only one channel.

In results storage RIMPAR is significantly better than the per-point approach. The storage numbers are given in bytes (8 bits). In the calculation of these numbers we assumed one byte is required to store a class label (for either an image point or a block), and 4 bytes are required to store a block location (one byte each for the 4 coordinates needed to specify the location of 2 block corners).

Table 8. Comparison of Per-Point and RIMPAR Classifiers

Run	Classification Accuracy RIMPAR/Per-Point	Processing Time RIMPAR/Per-Point	Results Storage RIMPAR/Per-Point
69002901	TRAIN(95.6/97.0)/TRAIN(98.2/98.9) TEST(76.7/69.7)/TEST(78.5/78.9)	1214/100	15630/44000
66000600	TRAIN(98.7/98.6)/TRAIN(95.1/95.7) TEST(79.9/77.0)/TEST(78.5/72.8)	967/95	9535/40280
71053900	TRAIN(99.3/99.3)/TRAIN(98.6/99.0) TEST(95.4/80.9)/TEST(93.2/85.2)	1145/105	13950/46509
72032803A	TRAIN(83.5/80.1)/TRAIN(85.6/85.6) TEST(82.6/65.7)/TEST(81.3/65.7)	753/81	14125/36000
72032803B	TRAIN(88.5/85.8)/TRAIN(80.7/69.6) TEST(74.0/67.6)/TEST(71.8/65.2)	615/67	11635/27900

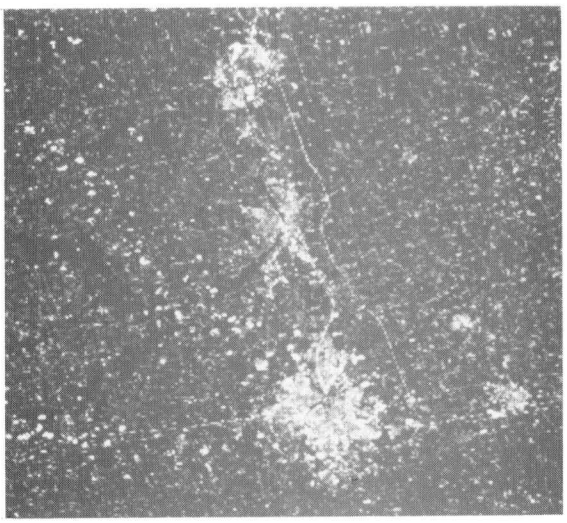
MINSIZE=1 K_D=20 SLEV=.01 PSET(Best 2)* CSET(Best 4)*

*As determined by feature selection algorithm. For Runs 72032803A and B PSET(2,3) and CSET(2,3) were used.

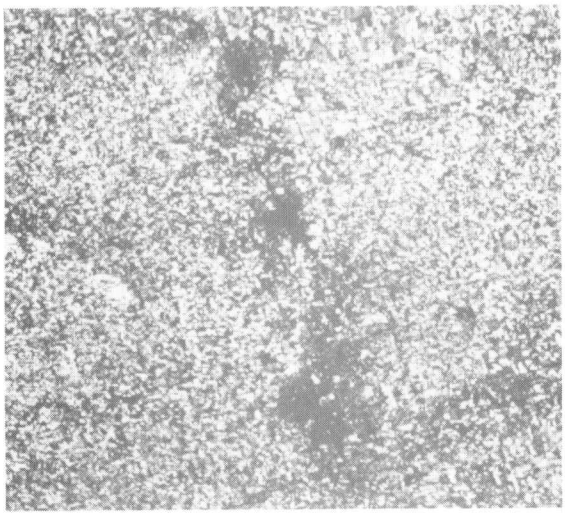
5.4 Isolating Urban Areas

Next we performed some experiments in the unsupervised classification of ERTS-1 imagery. The test image, shown in Figure 9 , is LARS RUN(73032801), LINES(887,1751, 3), COLS(393,1357,3). There are 5 relatively large cities in this image, and they appear as light colored objects in channel 2. The 3 largest cities are, from top to bottom, Janesville, Wisconsin; Beloit, Wisconsin; and Rockford, Illinois. A smaller city, Belvedere, Illinois, appears to the right of Rockford, and above Belvedere is Poplar Grove, Illinois. The goal of these experiments was to isolate these cities from the rest of the image. This isolation was accomplished by clustering the image and displaying the cluster classes as different gray levels. The cities were considered to be effectively isolated if they were represented exclusively by a single cluster class. Two methods using clustering were compared: clustering a PP image description and clustering an AO description produced by RIMPAR.

Figure 10 shows the result of clustering the PP description of the image into 5 cluster classes using channels 2 and 4. The clustering was carried out by the LARS program $NSCLAS, which uses the unsupervised per-point classifier described in Chapter 4. Due to software limitations only 22,770 out of the 93,058 image points were actually clustered. This subset of the available points was spatially evenly distributed throughout the image. The clustering algorithm

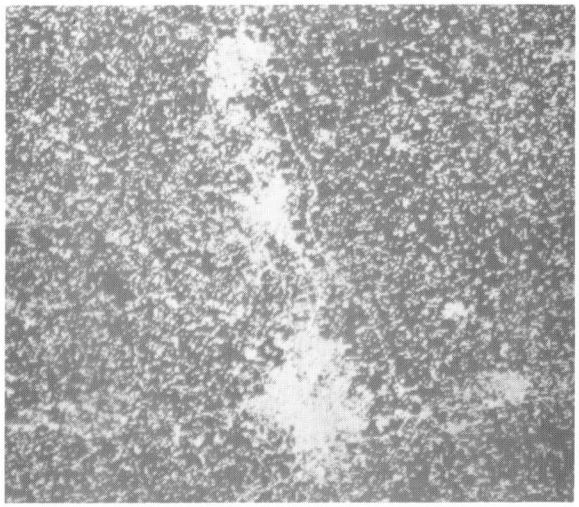


Channel 2

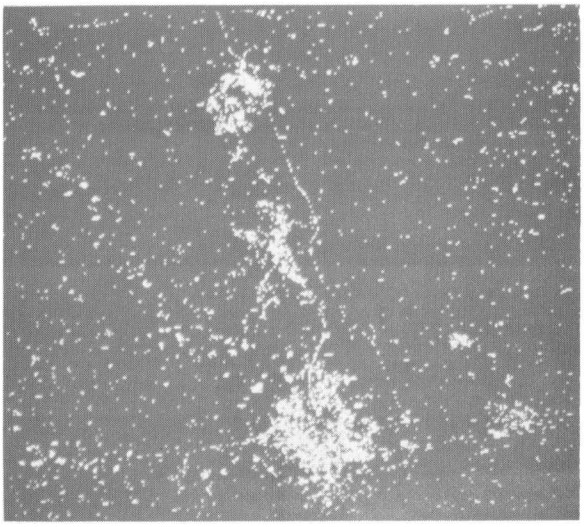


Channel 4

Figure 9. Satellite Image



5 Cluster Classes



Class 5 Shown as White

Figure 10. Per-point Clustered Image

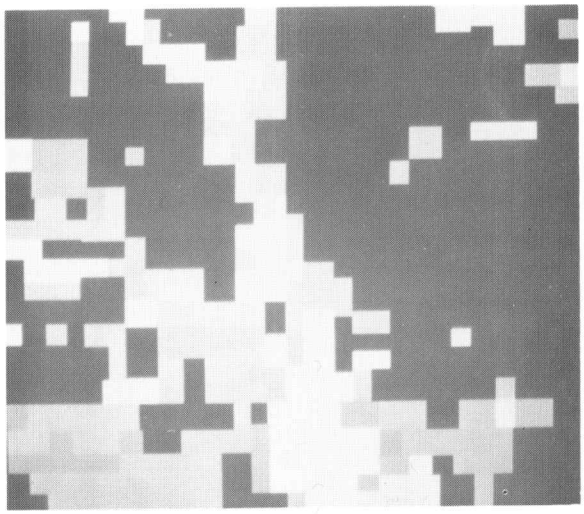
converged to zero class membership changes in 26 iterations and took 393 seconds of virtual CPU time. A subset of the clustered points was then used to train a supervised per-point classifier, and the entire image was then classified.

Visually the image shown at the top of Figure 10 seems to be a good representation of the original image. However, the human visual system does a lot of spatial integration in viewing such a picture. As shown in the lower image of Figure 10, the cluster class most nearly representing the cities consists of (1) separated points within the cities, and (2) many superfluous points outside the cities. Thus the image description stored in the computer, represented by Figure 10, does not specify the location of five major objects that represent cities. The cities are not found as single objects in the PP description because cities are not characterized only by the reflectance of their individual points, but also by their texture. Texture is not represented in a PP description. Although it might be possible to gain more spatial continuity within the cities by using fewer cluster classes, it is not likely that this approach would eliminate the superfluous points. As a final comment on this method, note that the classified image of Figure 10 requires about 93,058 bytes of storage, assuming one byte for each class label.

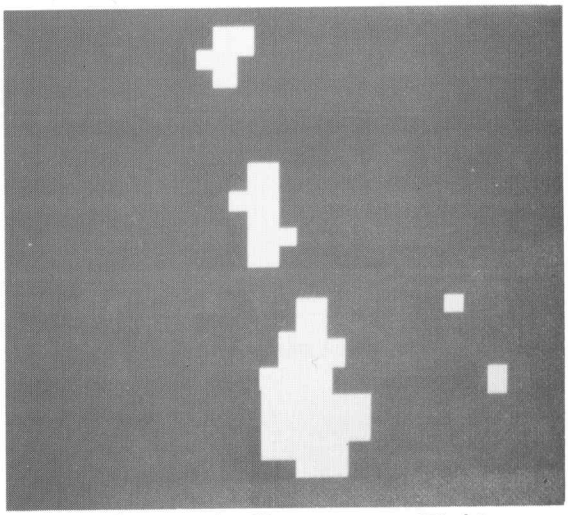
In the next experiment the image of Figure 9 was partitioned by RIMPAR using K_D=20, SLEV=.01, and PSET(2,4).

Since the cities are textured objects, it was necessary to use a large enough MINSIZE value so that the cities were not erroneously partitioned. Since the size of Poplar Grove in this image was about 10, and since blocks of size 10 seemed to characterize the texture of the cities well, we used MINSIZE=10. The image was partitioned into 418 blocks in 1030 seconds. The blocks were then clustered using CSET(2,4) into 5 cluster classes by the unsupervised sample classifier described in Chapter 4. The algoritnm converged to zero class membership change after 34 iterations and 206 seconds. The results are shown in Figure 11, and it is clear that the cities have been approximately isolated.

Although the city boundaries shown in Figure 11 are quite approximate, the image description represented by the figure can be very useful as an input to more detailed processing steps. If we are interested only in urban areas, the approximate location of cities can eliminate much of the image from consideration, and direct the application of algorithms that would be too time consuming to use on the entire image. Such reductions in processing time is important when vast quantities of data are involved, as with the ERTS-1 satellite. This idea of a hierarchy of processing steps could also be used within RIMPAR. For example, probably only a subset of the available image points are needed to produce Figure 11, and partitioning time could be considerably reduced if fewer image points were used.



5 Cluster Classes



Class 4 Shown as White

Figure 11. Partitioned Satellite Image

Clustering then displaying images has been previously used to determine how well an image represents a target [15]. This approach to evaluating image data was the original motivation for clustering the partition blocks produced by RIMPAR. However, clustering also tends to create approximately G-optimal partitions from approximately G-regular partitions, and thus can be considered an extension of the basic partitioning algorithm. An alternative and less time consuming approach would be to link adjacent blocks using the T^2 test, but this approach does not use the idea of global similarity that is inherent in clustering.

Recall that in clustering, there are no class identities associated with the cluster class numbers. Therefore to actually find cities, for example, not only must the city objects be isolated, but these objects must also be recognized as cities. The assumption made here and in later experiments is that if image objects are isolated, shape, size, and location information can be used to assign identities to the cluster class numbers.

5.5 Other Applications

We now investigate the use of RIMPAR to partition a digitized chest x-ray and a digitized photograph of human face. Each of these images has one channel, and they are both representative of pattern recognition applications that have received considerable interest.

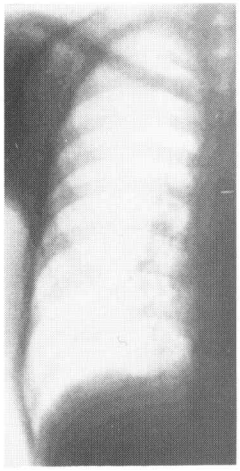
Research in the computer processing of chest radiographs has been aimed at the automatic diagnosis of chest diseases [4,32,33]. The algorithms that have been studied have two steps: First the organ of interest, for example heart or lung, is isolated in the image. Next texture or size measurements are automatically carried out on the isolated organ to determine the health of the organ. Since RIMPAR isolates objects, we will restrict ourselves to the problem of lung isolation in chest x-rays. In Figure 12 we show a digitized 132x256 point x-ray of the right lung region.

Chien and Fu [32] mention some of the difficulties involved in lung isolation:

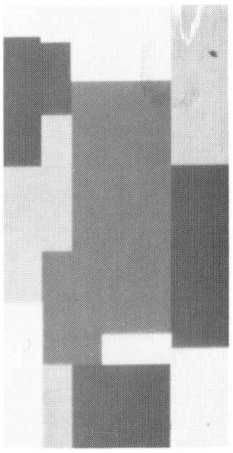
1) Lung edges are characterized only by a gradual gray level change.
2) Gray levels inside and outside the lung are often identical.
3) It is difficult to distinguish between lung boundaries and other boundaries, for example boundaries along ribs.

The basic problem is that the lung is a globally defined object; it is not well defined locally.

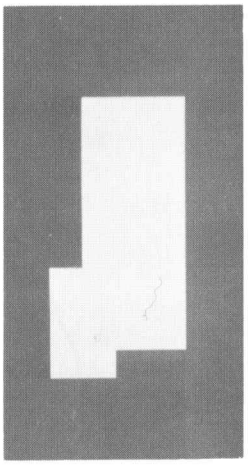
Several lung isolation techniques have been used. In [4] a local thresholding algorithm is presented. In this algorithm the entire x-ray is divided into overlapping blocks and the distributions of the gray levels in the blocks are examined. If a block distribution is bimodal, a boundary



Original Image



5 Cluster Classes



Class 2 Shown as
White

Figure 12. Partitioning a Chest Radiograph

segment is assumed to occur in the block. The characteristics of the two distribution modes are then used to distinguish between points on either side of the boundary segment. Blocks having unimodal distributions are considered inside or outside the lung, depending on a priori knowledge. The boundary segments found by this method are not connected, so various heuristics must be used to link and smooth the lung boundaries. The results shown in [4] indicate that part of the lung boundary is approximately found, but that upper and lower superfluous regions are included in the lung. The resulting lung description does not seem to be directly useable for disease recognition. No indication of computation time is given in [4], but the technique is probably time consuming.

In [33] regions of nearly constant gray level are detected using a simple threshold test. These regions are then linked together to form the lung region. The linking procedure involves a complicated set of heuristics, and the results presented in [33] show that lung boundaries are partially found, but the lung region is merged with other regions and the overall quality is poor. The algorithm reportedly takes about 300 seconds to process a chest radiograph.

In [32] a coarse lung boundary is found by searching for 5 key points that each satisfy some local criteria, and together satisfy some global constraint. The results of this procedure seem to be good, but the algorithm is

reportedly time consuming. Also presented in [32] is a method for creating a more accurate lung outline from the coarse lung outline.

To investigate the use of RIMPAR to locate the lung we partitioned and clustered the image shown in Figure 12. An early version of RIMPAR was used that compares the T^2 statistic with a fixed threshold, which in this case was 10. In order to prevent the partitioning of the lung into subregions defined by the ribs, we set MINSIZE=20. By examining a line printer display of the image we determined that 20 was about the smallest sized block that contained both lung and rib areas and thus preserved the global spatial structure of the lung. RIMPAR produced 30 blocks in 45 seconds, then the blocks were clustered into 5 cluster classes. The clustering algorithm converged in 5 seconds after 3 iterations, giving a total processing time of 50 seconds. In Figure 12 we show the results of our procedure. In the upper right image only 4 of the 5 cluster classes are visible because the display histogramming algorithm assigned the same gray level to classes 4 and 5.

RIMPAR produced a straight-line approximation to the lung. The approximation obtained by RIMPAR is probably not as accurate as one obtained by the key-point method of [32], but the fact that most points outside the lung are excluded makes the RIMPAR result more useful for texture measurements than the results of [4] and [33]. If a more accurate

boundary were desired, the technique for producing an accurate boundary from an approximate boundary presented in [32] could perhaps be applied to RIMPAR's approximation.

The processing time required by RIMPAR is definitely less than the 300 seconds (on a similar computer) reported by [33], and probably less than the other two methods. Another interesting comparison is the amount of a priori information used. The use of such information is very useful in problem solving, but the more a priori information required by an algorithm, the less flexible the algorithm is likely to be. The a priori information required by RIMPAR was numerical-only the object size was used in determining the necessary parameters for this application. (The best number of cluster classes is, however, still open to question. It is not certain how this number depends on the application.) The other algorithms made heavy use of the basic structure of the lung to insure the global accuracy of the boundary. In RIMPAR it seems that we are able to characterize the concept of a global object in a less structural and more numerical sense.

The conclusion drawn from this single lung extraction experiment can only be tentative, but it appears that RIMPAR is useful in the analysis of chest radiographs.

With the eventual goal of automatic human face recognition, researchers have studied techniques for outlining facial features [34], obtaining head outlines [35], and locating face components [36]. In [34] a local gradient

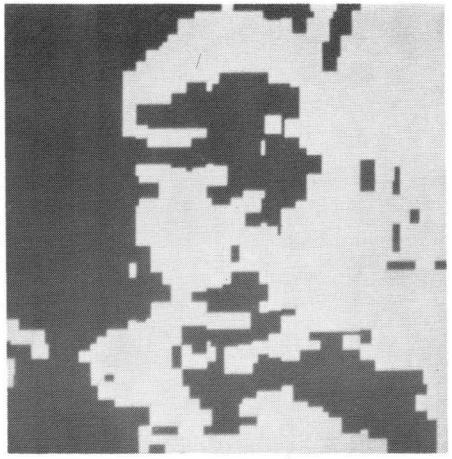
technique was used to locate points on facial boundaries. The result of this approach was, depending on a threshold, either too many superfluous boundary points or too few boundary points. Heuristics requiring boundary connectivity were used to make boundaries semi-complete. Although template matching to determine the existance of faces in images was reported as successful with these boundaries, no facial features were actually isolated. The problems encountered in [34] indicate that as in the lung extraction problem, global information is important in face extraction.

Global information concerning head shape was used successfully in [35] to link locally detected boundary points into complete head contours. Another feature of this work was the use of a small, less detailed version of the original image in the first processing steps. This hierarchy of processing steps, which Kelly [35] calls "planning", allows head outlines to be found very quickly, in around 10 seconds.

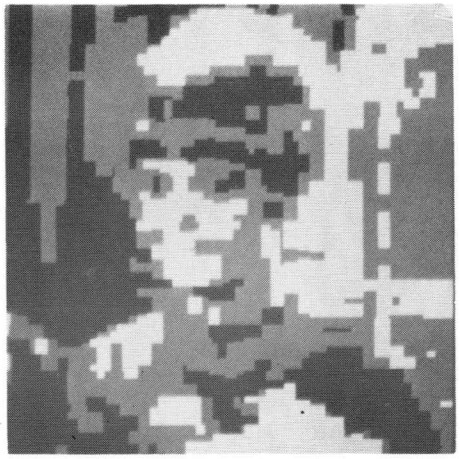
In [36] global information on the relative location of eyes, nose, mouth, and sides of the face are used in conjunction with local operators to successfully locate facial components. The processing time required by this algorithm is about 50 seconds.

To determine whether RIMPAR's ability to characterize global objects is useful in face partitioning, some experiments were run using the 256x256 digitized photograph of a girl's face shown in Figure 13.

Original Image

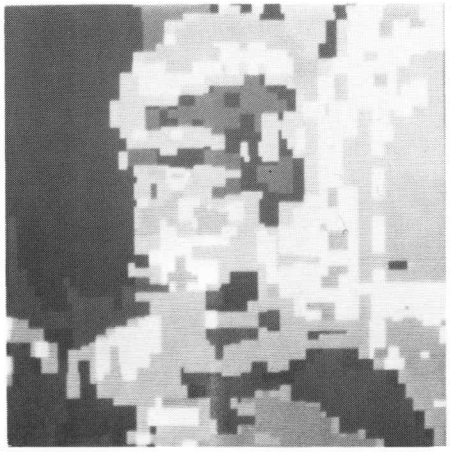


2 Cluster Classes

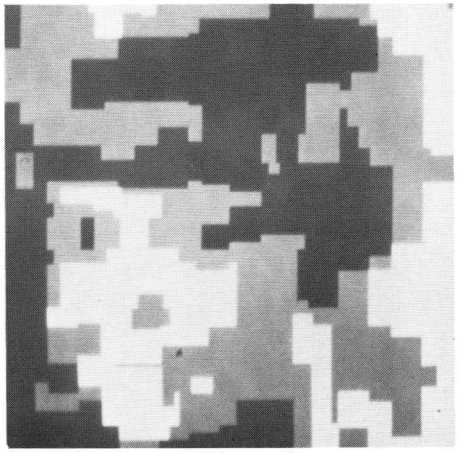


3 Cluster Classes

4 Cluster Classes



5 Cluster Classes



4 Classes (Detail)

Figure 13. Partitioning a Face

In order to investigate RIMPAR's performance in isolating facial detail and head outline, the fixed threshold version was used to partition the image of Figure 13. We set MINSIZE=2 so that the eyes and other small facial features could be found as single blocks. For clustering in into 5 classes, 36 iterations and 431 seconds were required for convergence to zero class membership change.

In spite of RIMPAR's restriction to horizontal and vertical boundaries, the details of the original image are fairly well preserved. This is especially evident if Figure 13 is viewed at a distance or intentionally blurred. The basic outlines of the face and the top of the head are visible in the 2 cluster class image. However in this case the face and head are merged with the background. Lighting effects also interrupt the continuity of the head and face. The location of the eyes is visible in all cases, the nose becomes evident with 3 cluster classes, and the mouth becomes visible with 4 cluster classes. With 5 cluster classes there appears to be too much detail.

The images shown in Figure 13 do not represent a complete face description. However, since the basic face features are well and simply defined, these images would probably make good inputs to feature location algorithms. For example Kelly [35] finds individual edge points before finding head contours. His procedure might be improved if the regions found by RIMPAR were used as an input. Fischler and Elschlager's [36] procedure might also be improved with a RIMPAR partitioned image as an input. The main deterrent

to accurate boundary finding in the RIMPAR experiments seems to be the lighting in the photograph. In [35] and [36] the image lighting was more direct to minimize shadows and this procedure would probably improve the RIMPAR result.

The computation time required to partition and cluster the image of Figure 13 seems excessive compared to the other algorithms discussed. However, this time could be significantly reduced if Kelly's planning idea were adapted. For example a small version of the face could first be partitioned to find the head outline, then the region within the head could be partitioned to find the facial features. Kelly reported that planning reduced computation by a factor of 40 in his experiments.

It is difficult to draw many conclusions from a single experiment, but RIMPAR appears to be potentially useful in face recognition. If the subject were well lighted and if planning were used, RIMPAR would probably be competative in quality and computation time to the algorithms in [35] and [36]. This is an interesting possibility because RIMPAR uses very little structural information compared to [35] and [36].

CHAPTER 6

CONCLUSION

6.1 Summary of Results

A multispectral image partitioning algorithm is presented that divides an image into successively smaller blocks until certain stopping criteria are met. The procedure has the basic advantages that globally-regular objects can be detected, and the size of samples used to estimate similarity between subimages is kept relatively large.

A theoretical, idealized image model is developed, and a partition criterion is presented that is minimized by a class of "good" image partitions. The partitioning algorithm is shown to converge for ideal images to a partition of arbitrarly small error (with respect to the partition criterion) in a finite number of steps.

The ideas from the idealized model are used to develop a practical partitioning algorithm.

In the supervised classification of agricultural areas RIMPAR is shown to give accuracy that is comparable to, and in some cases better than per-point classification. In these experiments per-point classification required only 8% to 11% of the processing time required by RIMPAR, but RIMPAR's classification results could be stored in only 24% to 42% of the space required by the per-point method.

RIMPAR was shown to perform well in isolating urban areas in a satellite image, a task at which the per-point method completely failed.

RIMPAR was used to partition a digitized chest radiograph and a digitized photograph of a girl's face. In these experiments it was demonstrated that RIMPAR can be used to extract global as well as local spatial structure.

6.2 Suggestions for Further Work

One aspect of RIMPAR that has not been experimentally investigated is the use of gray level vectors that are functions of the original image. This approach would be beneficial in images where adjacent objects differ mainly in spatial pattern.

The use of the sequential mean test, the classify-then-partition method, and small values of K_D have been shown to decrease RIMPAR's processing time. The details of applying these techniques in various applications should be investigated. Performing preliminary partitioning on a subset of the original image could also be incorporated into RIMPAR to improve speed.

Clustering was presented here as a way to merge adjacent and nonadjacent blocks that are statistically similar but it is possible to merge adjacent blocks before classification. In some cases this may lead to more accurate classification, but a more complicated block storage scheme would have to be implemented to handle nonrectangular partition blocks.

Only a small part of the spatial information contained in a partitioned image is used in sample classification. Other attributes such as size, shape, and spatial context could be used for classification, or for changing the partitioned image into a still more structured description, perhaps a linguistic description.

LIST OF REFERENCES

LIST OF REFERENCES

[1] R. A. Holmes and R. B. MacDonald, "The Physical Basis of System Design for Remote Sensing in Agriculture", Proc. IEEE, Vol. 57, No. 4, April 1969, pp. 629-639.

[2] E. A. Patrick, K. Henry, and J. Altman, "Computer Controlled Picture Scanning with Applications to Labeled Biological Cells", Comput. Biol. Med. Vol. 2, pp. 5-14.

[3] P. E. Anuta and R. B. MacDonald, "Crop Surveys from Multiband Satellite Photography Using Digital Techniques", Remote Sensing of Environment 2, 1971, pp. 53-67.

[4] R. N. Sutton and E. L. Hall, "Texture Measures for Automatic Classification of Pulmonary Disease", IEEE Trans. Comp., Vol. C-21, No. 7, July 1972, pp. 667-676.

[5] R. S. Ledley, H. A. Lubs, F. H. Ruddle, "Introduction to Chromasome Analysis", Comp. Biol. Med., Vol. 2, pp. 107-128.

[6] K. S. Fu, "On the Application of Pattern Recognition Techniques to Remote Sensing Problems", TR-EE 71-13, Purdue University, June 1971.

[7] P. J. Ready, "Multispectral Data Compression Through Transform Coding and Block Quantization", TR-EE 72-2, Purdue University, Lafayette, Indiana 1972.

[8] P. E. Anuta, "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques", IEEE Trans. Geo. Elec., Vol. GE-8, No. 4, October 1970, pp. 353-368.

[9] E. A. Smith and D. R. Phillips, "Automated Cloud Tracking Using Precisely Aligned Digital ATS Pictures", IEEE Trans. Comp., Vol. C-21, No. 7, July 1972, pp. 715-729.

[10] D. A. Landgrebe, R. M. Hoffer, R. E. Goodrick and Staff, "An Early Analysis of ERTS-1 Data", LARS Information Note 092972, Purdue University, Lafayette, Indiana.

[11] T. Huang, "Per Field Classifier for Agricultural Applications", LARS Information Note 060569, Purdue University, Lafayette, Indiana.

[12] A. G. Wacker and D. A. Landgrebe, "The Minimum Distance Approach to Classification", LARS Information Note 100771, Purdue University, Lafayette, Indiana.

[13] K. Shanmugan, R. M. Haralick, and R. Bosley, "Land Use Classification Using Texture Information in ERTS MSS Imagery", Tech. Rept. No. 2262-1, University of Kansas Center for Research, Inc., January, 1973.

[14] L. Sayn-Wittgenstein, "Patterns of Spatial Variation in Forests and other Natural Populations", Pattern Recognition, Vol. 2, No. 4, December 1970, pp. 245-253.

[15] R. Ellefsen, P. H. Swain, J. R. Wray, "Urban Land Use Mapping by Machine Processing of ERTS-1 Multispectral Data - A San Francisco Bay Area Example," LARS Information Note 032973, Purdue University, Lafayette, Indiana.

[16] K. K. Pingle, "Visual Perception By a Computer", in Automatic Interpretation and Classification of Images, A. Grasselli (ed), Academic Press, New York, 1969, pp. 277-284.

[17] J. M. S. Prewitt, "Object Enhancement and Extraction", in Picture Processing and Psychopictorics, B. S. Lipkin and A. Rosenfeld (eds.), Academic Press, New York, 1970, pp. 113-114.

[18] M. H. Hueckel, "An Operator Which Locates Edges in Digital Pictures", J. ACM, Vol. 18, No. 1, January 1971, pp. 113-125.

[19] A. G. Wacker, "A Cluster Approach to Finding Spatial Boundaries in Multispectral Imagery", LARS Information Note 122969, Purdue University, Lafayette, Indiana.

[20] A. Rosenfeld, M. Thurston, Y-H Lee, "Edge and Curve Detection: Further Experiments", IEEE Trans. Comp., Vol. C-21, No. 7, July 1972, pp. 677-715.

[21] A. K. Griffith, "Edge Detection in Simple Scenes Using A Priori Information", IEEE Trans. Comp., Vol. C-22, No. 4, April 1973.

[22] C. R. Brice and C. L. Fennema, "Scene Analysis Using Regions", Artifical Intelligence I, 1970, pp. 205-266.

[23] E. M. Rodd, "Closed Boundary Field Selection in Multispectral Digital Images", IBM Publication No. 320. 2420, January, 1972.

[24] J. N. Gupta, "Remotely Sensed Multi-Image Modelling", Ph.D. Thesis Proposal, School of Elec. Eng., Purdue University, Lafayette, Indian

[25] K. Fukanaga and W. L. G. Koontz, "A Criterion and an Algorithm for Grouping Data," IEEE Trans. Comp., Vol. C-19, No. 10, October 1970.

[26] G. Hadley, Linear Algebra, Addison Wesley, 1961, pp. 175-176.

[27] T. W. Anderson, An Introduction to Multivariate Statistical Analysis, John Wiley & Sons, Inc., New York, 1958, pp. 108-109.

[28] A. Hald, Statistical Theory with Engineering Applications, John Wiley & Sons, Inc., New York, 1952, p. 384.

[29] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1965, p. 246.

[30] LARSYS User's Manual, Laboratory for Applications of Remote Sensing, Purdue University, Lafayette, Indiana.

[31] P. H. Swain, T. V. Robertson, and A. G. Wacker, "Comparison of the Divergence and B-Distance in Feature Selection", LARS Information Note 020871, Laboratory for Applications of Remote Sensing, Purdue University, Lafayette, Indiana.

[32] Y. P. Chien and K. S. Fu, "On the Preprocessing of Radiographic Imagery", TR-EE 73-6, Purdue University, Lafayette, Indiana.

[33] C. A. Harlow and S. A. Eisenbeis, "The Analysis of Radiographic Images", IEEE Trans. Comp., Vol. C-22, No. 7, July 1973, pp. 678-689.

[34] T. Sakai, M. Nagao, and S. Fujibayashi, "Line Extraction and Pattern Detection in a Photograph", Pattern Recognition, Vol. 1, March 1969, pp. 233-248.

[35] M. D. Kelly, "Edge Detection in Pictures by Computer Using Planning", Machine Intelligence 6, B. Meltzer and D. Michie (eds.), American Elsevier, New York, 1971.

[36] M. A. Fischler and R. A. Elschlager, "The Representation and Matching of Pictorial Structures", IEEE Trans. Comp., Vol. C-22, No. 1, January 1973, pp. 67-92.