ABSTRACT

Interpreting and Integrating Landsat Remote Sensing Image and Geographic Information System by Fuzzy Unsupervised Clustering Algorithm

by
Gwotsong Philip Chen

Due to the resolution of *Landsat* images and the multiplicity of the terrain, it is improper to assign each pixel in an image to one of a number of land cover types by using the conventional remote sensing classification method. This is also known as the hard partition method. The concept of the fuzzy set provides the means to resolve this problem. This paper presents a two-pass-mode fuzzy unsupervised clustering algorithm.

In the first passing, the cluster mean vectors which represent the geographic attributes or the land cover types are derived. In the second passing, the concept of fuzzy set is used. The cluster mean vectors which are obtained in the first passing are used to derive the membership function. The grade of memberships of each pixel to the land cover types are obtained according to the distance from the pixel to each cluster mean vector. The output of this algorithm can be used as the input of the Geographic Information System.

INTERPRETING AND INTEGRATING LANDSAT REMOTE SENSING IMAGE AND GEOGRAPHIC INFORMATION SYSTEM BY FUZZY UNSUPERVISED CLUSTERING ALGORITHM

by
Gwotsong Philip Chen

A Thesis
Summitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science
Department of Computer and Information Science
May 1992

APPROVAL PAGE

Interpreting and Integrating Landsat Remote Sensing Image and Geographic Information System by Fuzzy Unsupervised Clustering Algorithm

by
Gwotsong Philip Chen

5/11/92

Dr. Frank Y. Shih, Thesis Adviser
Assistant Professor of Computer
and Information Science, NJIT

BIOGRAPHICAL SKETCH

Author: Gwotsong Philip Chen

Degree: Master of Science in Computer Science

Date: May, 1992

Date of Birth:

Place of Birth:

Undergraduate and Graduate Education:

- Master of Science in Computer Science, New Jersey Institute of Technology, Newark, NJ, 1992
- Bachelor of Science in Earth Sciences, National Cheng Kung University, Tainan, Taiwan, Republic of China, 1986

Major: Computer and Information Science

This thesis is dedicated to
my parents for their love and support

ACKNOWLEDGEMENTS

It is my great pleasure to thank my advisor, Dr. Frank Y. Shih, for his guidance in thesis preparation. I especially appreciate his intellectually challenging questions which initiated many meaningful research directions. The author also would like to thank Mr. Stefan Winz for his help in editing many drafts of my thesis.

Special thanks go to my parents, Mr. Sheng-hsun Chen and Ms. Ling-yu Lin. Without their love and financil support, I could not have completed my Master degree.

Missing Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the last two to three decades, the Geographic Information System (*GIS*) has been greatly developed for different purposes. The applications of the *GIS* include urban planning, environmental supervising, water resources [1] [2], wildlife habitat protection [3], land property management [4], etc. At first, researchers who worked on the *GIS* spent a great deal of time in collecting data. Then they used the digitizer to input the data into the computer point by point. These procedures were very subjective and time-consuming. After the remote sensing data and the scanner were used in the *GIS*, these problems no longer existed. Instead, classification becomes the major barrier in using these two new techniques in the *GIS*.

It is well known that within a map there are several major land cover types, such as vegetation, urban area, water body, bare land, clear cut, etc. In the conventional methods, before the raw *GIS* data is input into the computer, it has already been classified manually. It is then input into computer layer by layer according to its land cover types. However, if we adopt the data from the satellite remote sensing, such as *Landsat*, SPOT, etc., before its input, we cannot manually classify them. Multispectral remote sensing data, ***Landsat***, for example, collect different emitted and reflected intensities of an object among several different wavelength intervals. As the wavelength interval changes, the spectral feature of the object will be distinct. Since every single object has its own unique spectral feature, different land cover types or geographic attributes can be identified according to their spectral feature. This is the basic idea of multispectral image classification.

Many algorithms have been developed to resolve the problem of classification. Multispectral classification methods can be categorized into two major groups: supervised or unsupervised [5] [6] [7]. The difference between these two groups is whether or not the identity of land cover types to be specified as a class is a priori knowledge. In the supervised classifications, including the parallelepiped classification algorithm, the minimum-distance to means classification algorithm, and the maximum likelihood classification algorithm, the identity and location of some of the land cover types are known a priori through a combination of field work. In unsupervised classification, the clustering analysis algorithm is widely used. Because all these classification methods are used to assign an unknown pixel to one of a number of classes, they are also considered hard partition methods [8].

There is an assumption in applying the hard partition, i.e., every pixel is represented as a homogeneous area. In the case of ***Landsat* Thematic Mapper** (***TM***) multispectral image, there are seven bands. Band 6 has a resolution of 120 x 120 m^2. The other six bands have a resolution of 30 x 30 m^2. This means that each pixel represents a ground area of 900 m^2 or 14400 m^2. Obviously, it is not necessarily true that a pixel contains a unique land cover type or geographic attribute. In order to offset this defect, we have to introduce the fuzzy set concept [9] [10] in remote sensing image classification.

Several research works have been done in remote sensing image classification using fuzzy set concept. Wang [8] [11] [12] proposed a fuzzy supervised classification method which is based on the maximum likelihood classification algorithm with the fuzzy mean and fuzzy covariance matrix replacing the conventional mean and covariance matrix. Cannon *et. al.* [13] [14] developed a fuzzy *c*-means clustering algorithm (***FCM***) to perform unsupervised classification on ***TM*** image. Foody [15] applied the fuzzy *c*-means algorithm to present the vegetation continua of remote sensing data. Zenzo *et. al.* [16] developed a fuzzy relaxation algorithm for contextual classification. Kent and Mardia [17] used fuzzy membership models to perform spatial classification. However, the method proposed by these reports require large amount of memory space in implementation. ***FCM***, for example, needs 5 to 7 Mbytes of memory to store all of its matrices. The method proposed in this paper is implemented in the personal computer with 1 Mbyte memory.

This paper is organized as follows: The unsupervised clustering algorithm used in finding the mean vectors of several major land cover types and generating a hard partition map is described in the next chapter. Next, the fuzzy unsupervised clustering algorithm used in this study is discussed. The description of the test image is presented in chapter 4. Experiments and results are described in chapter 5, and conclusions are presented in the last chapter.

CHAPTER 2

THE UNSUPERVISED CLUSTERING ALGORITHM

The clustering algorithm used here is modified from the two-pass-mode clustering algorithm [5]. It will pass through the registered multispectral image data twice. During the first passing, it will automatically generate the cluster mean vectors. On the second passing, each pixel will be assigned to a cluster which represents a single land cover type in order to produce the hard partition map.

Before further discussion of the algorithm, there are several parameters which must be introduced first.

B, the number of bands used during processing which will define the dimension of spectral space. (i.e., if three bands are used, it is a three dimensional spectral space.)

C_{max}, the maximum number of clusters which will be defined in the processing.

r, a distance in spectral space between the gray level vector of a pixel and the current cluster mean vector. It is defined as follows:

$$r = \sqrt{\sum_{i=1}^{B} (MEAN_i(k) - P_i)^2} \qquad (1)$$

where $1 \le k \le C_{max}$. $MEAN_i$ is the mean value of the *i*th band and P_i is the gray level of the pixel in the *i*th band.

R, a radius in spectral space used to decide whether a new cluster mean vector should be added or not. If the distance between the pixel's gray level vector and current cluster mean vector, r, then a new cluster is added. In certain situations, R is also used to make the decision of cluster merging.

d, a distance in spectral space between two distinct cluster mean vectors, is defined as follows:

$$r = \sqrt{\sum_{i=1}^{B} (MEAN_i(k_1) - MEAN_i(k_2))^2} \qquad (2)$$

where $1 \le k_1, k_2 \le C_{max}$ and $k_1 \ne k_2$.

D, a radius in spectral space used to decide whether or not two distinct clusters should be merged. If the distance between two distinct cluster mean vectors, d, is less than or equal to D, these two clusters will be merged.

N, the number of pixels to be evaluated before cluster merging.

Pass 1: Cluster Mean Building

First, designate the gray level vector of the first pixel, which is composed of the gray level values of the pixel in the chosen bands, as the initial cluster mean vector of the first cluster. Then read the gray level vector of the second pixel and calculate the distance between this gray level vector and the initial cluster mean vector, **r**. If **r** is greater than **R**, then create a new cluster whose cluster mean vector is equal to that gray level vector. If **r** is less than or equal to **R**, the cluster mean vector must be recalculated by the following equation:

$$MEAN_i(k)_{new} = \frac{MEAN_i(k)_{old} \times n(k) + P_i \times 1}{n(k) + 1} \qquad (3)$$

where *n(k)* is the total number of pixels accumulated in the *k*th cluster which will be incremented by one after this calculation. The total number of pixels evaluated (n_{total}) also will be incremented by one. Then read the next pixel's gray level vector.

When n_{total} is greater than **N**, or *k*, representing the accumulated cluster number, is equal to $\mathbf{C_{max}}$, this repeating processing will stop evaluating individual pixels. The cluster merging process will be activated to remove unnecessary clusters. Equation (2) will be used to calculate the distance between two cluster mean vectors, **d**. The chosen decision radius is dependent upon which condition activates the cluster merging process. If it is because n_{total} is greater than **N**, **D** will be chosen as the decision radius. Otherwise, **R** will be the decision radius. If **d** is less than or equal to the decision radius, these two cluster will be merged. The cluster mean vector of the new cluster is define as follows:

$$MEAN_i(k) = \frac{MEAN_i(k_1) \times n(k_1) + MEAN_i(k_2) \times n(k_2)}{n(k_1) + n(k_2)} \quad (4)$$

where $1 \leq k, k_1, k_2 \leq C_{max}$ and $k_1 \neq k_2$. The total number of pixels in the new cluster, $n(k)$, is the sum of $n(k_1)$ and $n(k_2)$. After the cluster merging process is completed, if the above condition resulted in n_{total} being greater than N, set n_{total} equal to zero; otherwise leave n_{total} unchanged. Then continue to evaluate individual pixels and accumulate the clusters until n_{total} is greater than N or k is equal to C_{max} again.

After all the individual pixels are evaluated, a certain number of clusters are built. In the second passing, these cluster mean vector will be used as the feature vectors of land cover types to determine the attribute of each pixel.

Pass 2: Pixel Classification

Pass 2 is used to classify each pixel into one of those clusters which are built in pass 1. The minimum-distance to means method is adopted. When each pixel is evaluated, equation (1) is employed to compute the spectral space between the pixel and each cluster. After all distances are determined, the cluster with minimum distance will be chosen, i.e., we will assign the land cover type of that cluster to that pixel. This is because similar objects will have similar spectral features. By using the minimum-distance to means method will assure those pixels with similar spectral features will converge together.

After pass 2, the hard partition map is generated. Each cluster mean vector represents a land cover type or a geographic feature according to its spectral attribute. However, it is not the best solution in interpreting the *Landsat* images. As we have depicted in the previous section, it is quite unlikely that a square area of 30 x 30 m^2 or 120 x 120 m^2 has the unique geographic feature. But using the unsupervised cluster analysis algorithm alone can not interpret the pixel with mixed components. Next, we will introduce how the concept of fuzzy set is used to resolve this problem.

CHAPTER 3

THE FUZZY UNSUPERVISED CLUSTERING ALGORITHM

Fuzzy set algorithm provides a partial membership concept in dealing with more complex geographic situations, such as mixture cover within a single pixel. The basic idea of fuzzy set is to consider that the spectral attribute of every pixel is contributed to by several land cover types, i.e., each geographic attribute has a certain extent of contribution to the pixel's spectral attribute. The answer of the question is not yes or no, but is how much. In a fuzzy representation for remote sensing image analysis, land cover types can be defined as fuzzy sets, and pixels as set elements [11]. The pixel can be illustrated by a group of membership grades which are used to indicate the extent to which the pixel belongs to certain land cover types.

Let I be the set of integers and I^B be the B-dimensional vector space over integers. Let **X** be the finite subset of $\mathbf{I}_B$, $X = \{x_{1,1}, x_{1,2}, \ldots, x_{1,B}, x_{2,1}, x_{2,2}, \ldots, x_{2,B}, \ldots, x_{A,1}, x_{A,2}, \ldots, x_{A,B}\}$ and for every $x_{a,b} \in [0,255]$. For an integer **c**, $2 \le c \le C_{max}$, U_{ia} is the grade of membership of the element x_a in the *i*th fuzzy subset where $1 \le i \le c$. U_{ia} also can be written as $U_i(x_a)$ which will satisfy the following conditions:

$$\sum_{i=1}^{c} U_{ia} = 1 \qquad \text{for all } a \qquad (5)$$

$$\sum_{a=1}^{A} U_{ia} > 0 \qquad \text{for all } i \qquad (6)$$

$$U_{ia} \in [0,\ 1] \qquad \text{for all } i,\ a \qquad (7)$$

Equation (5) defines the sum of each individual element's grade of membership in all fuzzy subset is equal to 1. The sum of the membership grades of all elements in any individual fuzzy subset is greater than 0. This means there is no fuzzy subset without any membership. Equation (7) defines all membership grades are in the interval between [0, 1].

In the viewpoint of multispectral image analysis, the finite subset **X** in I_B can be interpreted as the set of pixels in B-dimensional spectral space. The integer **c** represents the number of cluster to be used. So far, we have discussed the basic idea of the fuzzy set algorithm. However, the fuzzy membership function which is used to derive the grade of membership has not been discussed yet. The membership function must satisfy equation (5) to (7). As we have discussed before, the spectral characteristics of a pixel determine its position

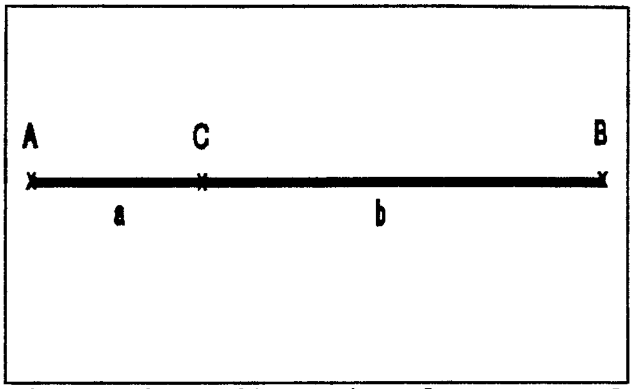


Figure 1 B-dimensional spectral space. A, B represent two cluster mean vectors in B-dimensional spectral space. C is an arbitrary pixel in this spectral space. a and b are the distances from C to A and B, respectively.

in the spectral space. The more a pixel contains a class, the closer the pixel is to the mean vector of that class. Fig. 1 shows two cluster mean vectors, A and B, in a B-dimensional spectral space. C is an arbitrary pixel in such a spectral space. *a* and *b* are the distances from C to A and B respectively. In this simplified case, the membership functions of C, U_{AC} and U_{BC}, can be defined as follows:

$$U_{AC} = \frac{b}{a + b} \tag{8}$$

$$U_{BC} = \frac{a}{a + b} \tag{9}$$

As we can see from the equations above, the grade of membership is the inverse ratio of the spectral distance. Equation (8) and (9) also satisfy the conditions which are showed in equation (5) to (7).

Next, let's further our discussion to three cluster mean vectors in B-dimensional spectral space. Three different methods in finding the membership function have been derived in this paper. Fig. 2 and 3 illustrate the relationship between the cluster mean vectors and an arbitrary pixel in B-dimensional spectral space. Later we will give an example to examine these three methods.

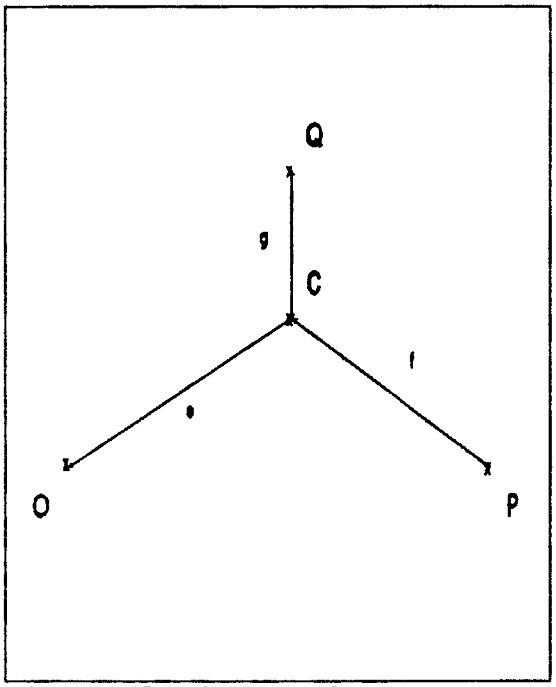


Figure 2 Three cluster mean vectors in B-dimensional spectral space. *e*, *f* and *g* are the distances from an arbitrary pixel to these three cluster mean vector.

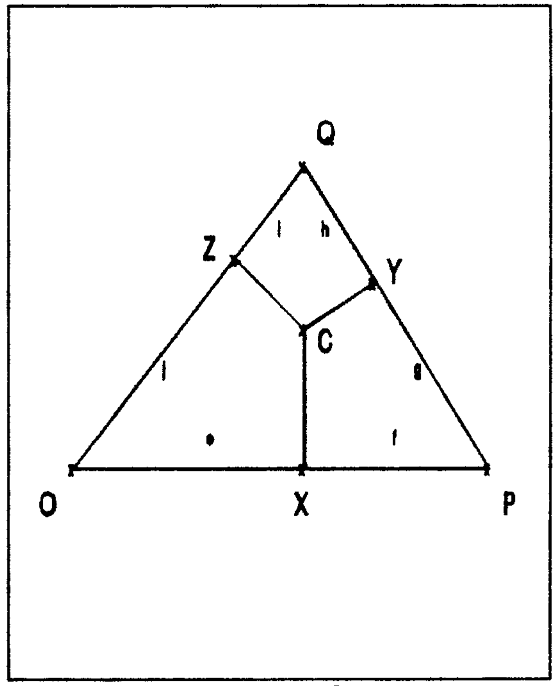


Figure 3 Three cluster mean vectors in B-dimensional spectral space. CX, CY and CZ are the normal vectors of OP, PQ and QO.

O, P and Q represent the cluster mean vectors while C is an arbitrary pixel in this B-dimensional spectral space. In Fig. 2, *e*, *f* and *g* are the distances from C to O, P and Q respectively. In Fig. 3, CX, CY and CZ are the normal vectors of OP, PQ and QO respectively. X is the intersection point of vector CX and OP. *e* is the distance between O and X.

Method 1:

From the previous discussion of the two cluster mean vectors circumstance, we can revise equation (8) and (9) to acquire the membership function for the three or more cluster mean vectors situation. The modification of equation (8) and (9) are shown in equation (10) and (11).

$$U_{AC} = 1 - \frac{a}{a + b} . \tag{10}$$

$$U_{BC} = 1 - \frac{b}{a + b} . \tag{11}$$

In the three cluster mean vectors state, Fig. 2, for example, we can rewrite the equation as follows:

$$U_{OC} = (1 - \frac{e}{e + f + g}) / 2 \tag{12a}$$

$$U_{PC} = (1 - \frac{f}{e + f + g}) / 2 \tag{12b}$$

$$U_{QC} = \left(1 - \frac{g}{e + f + g} \right) / 2 \,. \tag{12c}$$

As we can see, all the membership functions U_{OC}, U_{PC} and U_{QC} are divided by 2. This is because the membership functions of a single pixel must fulfill the condition mentioned in equation (5). If there are *c* cluster mean vectors in B-dimensional spectral space, the general form of the membership function will be as follows:

$$U_{ia} = \left(1 - \frac{r_i}{\sum_{i=1}^{c} r_i}\right) / (c - 1) \tag{13}$$

where r_i is the spectral distance from an arbitrary pixel's gray level vector to any cluster mean vector.

Method 2:

Due to the inverse ratio relationship between the grade of membership and the distance, the reciprocal of the distance between the pixel's gray level vector and the cluster mean vector are used in this method. The larger the distance, the smaller its reciprocal. Fig. 2, for example, the membership functions are

$$U_{OC} = \frac{\frac{1}{e}}{\frac{1}{e} + \frac{1}{f} + \frac{1}{g}} \qquad (14a)$$

$$U_{PC} = \frac{\frac{1}{f}}{\frac{1}{e} + \frac{1}{f} + \frac{1}{g}} \qquad (14b)$$

$$U_{QC} = \frac{\frac{1}{g}}{\frac{1}{e} + \frac{1}{f} + \frac{1}{g}} \qquad (14c)$$

It is easy for us to derive the general membership function formula for *c* cluster mean vectors in B-dimensional spectral space. The membership function, Equation (15), is defined as follows:

$$U_{ia} = \frac{\frac{1}{r_i}}{\sum_{i=1}^{c} \frac{1}{r_i}} \qquad (15)$$

Now, we can verify this reciprocal method with the two cluster mean vector circumstance:

$$U_{AC} = \frac{\frac{1}{a}}{\frac{1}{a} + \frac{1}{b}} = \frac{\frac{1}{a}}{\frac{a + b}{a \times b}} = \frac{b}{a + b} \tag{16a}$$

$$U_{BC} = \frac{\frac{1}{b}}{\frac{1}{a} + \frac{1}{b}} = \frac{\frac{1}{b}}{\frac{a + b}{a \times b}} = \frac{a}{a + b} \tag{16b}$$

which obviously also satisfy equation (8) and (9).

Method 3:

The third method uses the distance between cluster mean vectors to define the membership functions. In Fig. 3, the three cluster mean vectors case, we can disassemble it into two two cluster mean vector situations, QO and QP, for example. Find the ratio of U_{OC} and U_{QC}, and also the ratio between U_{PC} and U_{QC}. Then combine and find the ratio among U_{OC}, U_{PC} and U_{QC} by their common term, U_{QC}. After broken up into two two cluster mean vectors situations, the membership functions can be easily derived according to equation (8) and (9).

As shown in Fig. 3, X, Y and Z are the intersection points of vector OP and CX, PQ and CY, and QO and CZ, respectively. Or, we can say that X is the projection of C on OP. So the ratio between U_{OC} and U_{QC} can be specified as follows:

$$U_{OC} : U_{QC} = \frac{i}{i + j} : \frac{j}{i + j} = i : j \tag{17a}$$

And the ratio between U_{PC} and U_{QC} are described as follow:

$$U_{QC} : U_{PC} = \frac{g}{h + g} : \frac{h}{h + g} = g : h \tag{17b}$$

So the combined ratio of Equation (17a) and (17b), i.e., the ratio of the grade of the membership function is

$$U_{OC} : U_{PC} : U_{QC} = (i \times g) : (h \times j) : (g \times j) \tag{18}$$

In order to satisfy the fuzzy set constraint, the membership function can be obtained as follows:

$$U_{OC} = \frac{g \times i}{g \times i + h \times j + g \times j} \tag{19a}$$

$$U_{PC} = \frac{h \times j}{g \times i + h \times j + g \times j} \tag{19b}$$

$$U_{QC} = \frac{g \times j}{g \times i + h \times j + g \times j} \tag{19c}$$

The sum of each term in equation (18) is used as the denominator in equation (19a), (19b) and (19c). This will assure that the sum of the membership grades is equal to 1. Now, let's concentrate our discussion in the *c* cluster mean vectors situation. In a *c* cluster mean vectors model, each mean vector is connected to the others. One of these cluster mean vectors will be chosen to be the common vertex which is similar to Q in the three cluster mean vectors' example. Let cluster mean vector number 1 be the chosen common vertex. The projection of any pixel, C, to the connecting line between the common vertex and other cluster mean vectors will divide that line into two portions. Let d_{i0} be the distance between the projection and the common vertex, and d_{i1} be the distance between the projection and the other cluster mean vector where $2 \leq i \leq C_{max}$. So the ratio among the grade of membership will be as follows:

$$U_{1C} : U_{2C} : \ldots : U_{iC} : \ldots : U_{nC} =$$

$$(d_{21} \times d_{31} \times \ldots \times d_{i1}) : (d_{20} \times d_{31} \times \ldots \times d_{i1}) :$$

$$\dots : (d_{21} \times \dots \times d_{(i-1)1} \times d_{i0} \times d_{(i+1)1} \times \dots \times d_{i1}) :$$

$$\dots : (d_{21} \times \dots \times d_{(n-1)1} \times d_{n0}) \tag{20}$$

Let S be the sum of all terms in that ratio. So the general form of the membership function except for the common vertex is shown as follows:

$$U_{ia} = \frac{d_{21} \times \dots \times d_{(i-1)1} \times d_{i0} \times d_{(i+1)1} \times \dots \times d_{n1}}{S} \tag{21}$$

The membership function of the common vertix is shown as follows:

$$U_{1a} = \frac{d_{21} \times \dots \times d_{n1}}{S} \tag{22}$$

We will examine these three methods by the following examples which are illustrated in Fig. 4 and Fig. 5. Both figures show three cluster mean vectors, O, P and Q, and a pixel, C, in a two- dimensional spectral space. The gray level values, the cluster means and the distance in the spectral space are disclosed in the figures. And the result, i.e., the grades of membership which are obtained by using the methods discussed above, is summarized in Table 1.

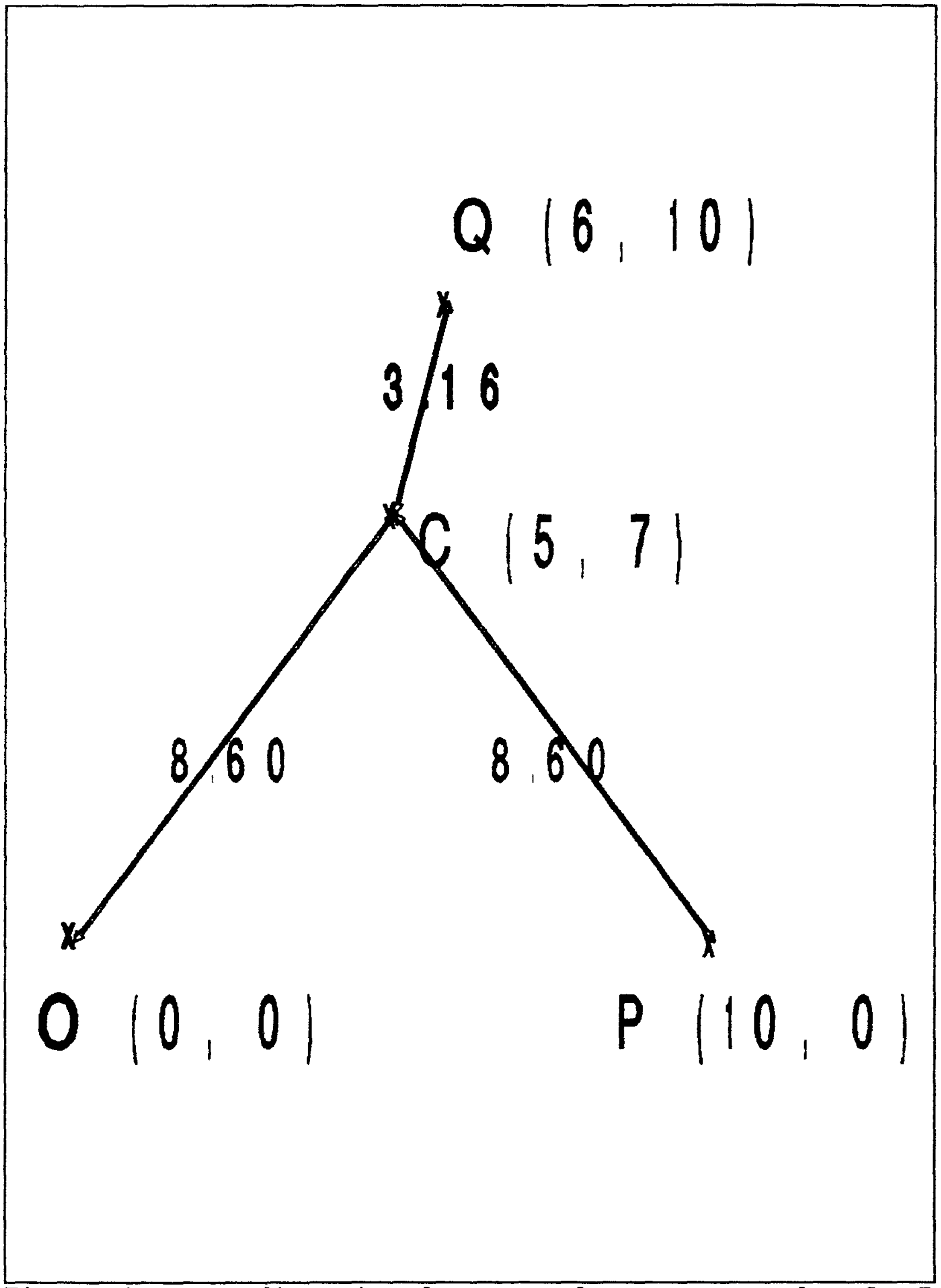


Figure 4 A two-dimensional spectral space example. O, P and Q are the cluster mean vectors. C is the pixel's gray level vector. 8.60, 8.60 and 3.16 are the distances from C to O, P and Q respectively.

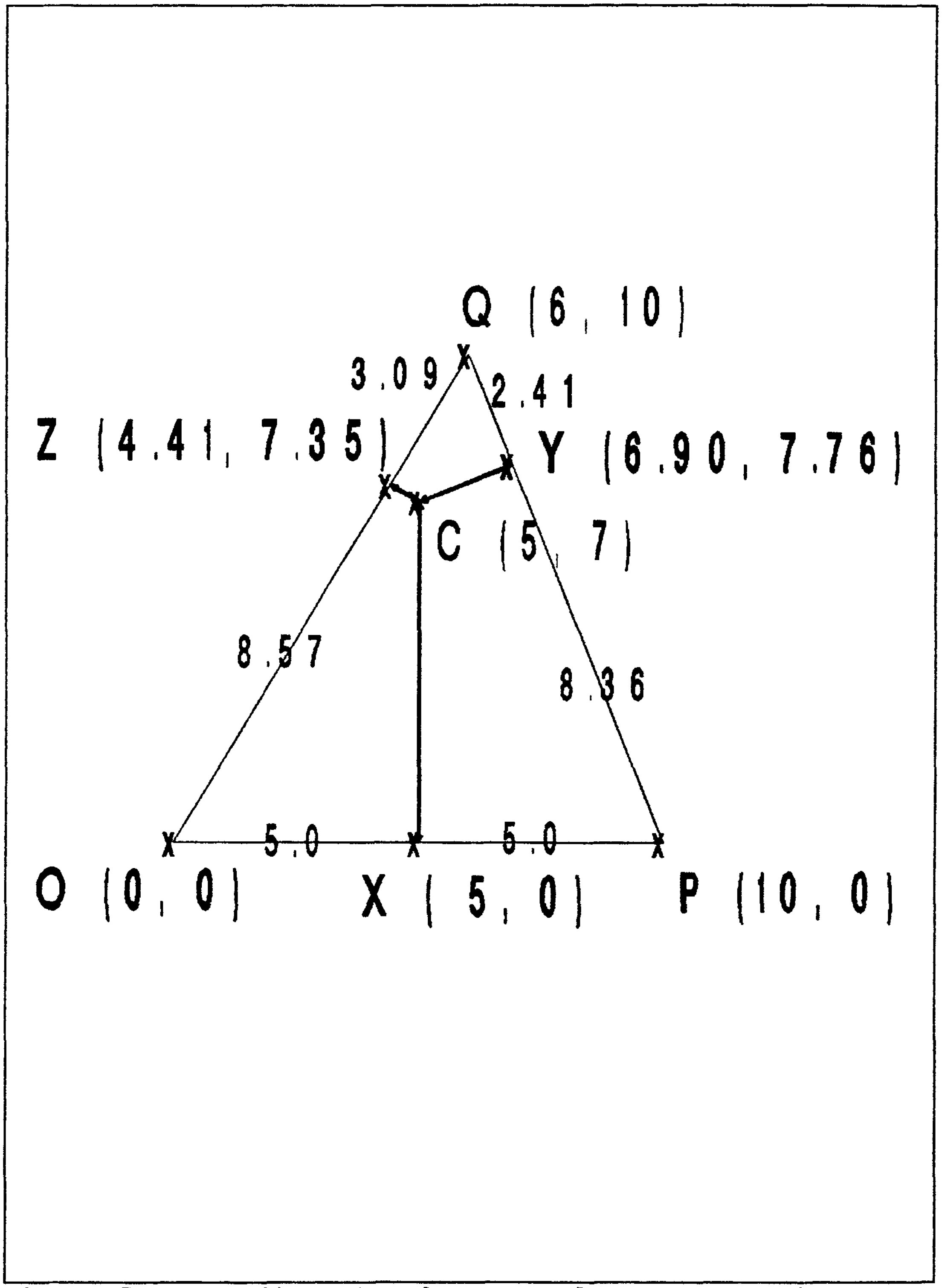


Figure 5 A two-dimensional spectral space example. O, P and Q are the cluster mean vectors. C is the pixel's gray level vector. CX, CY and CZ are the normal vectors of OP, PQ and QO respectively.

Table 1 The result of computing the grade of membership by using the methods which are demonstrated in equation (12a) - (12c), (14a) - (14c) and (19a) - (19c). The gray level values, the cluster mean vectors and the distances are shown in Fig. 4 and Fig. 5.

	THE GRADE OF MEMBERSHIP		
	U_{OC}	U_{PC}	U_{QC}
METHOD 1	0.2888	0.2888	0.4224
METHOD 2	0.2118	0.2118	0.5764
METHOD 3	0.2187	0.1748	0.6065

Different viewpoints of the relationship between the location of the pixel in the spectral space and the geographic attribute have been used in deriving the membership function. The three methods discussed use different techniques to obtain the membership functions under different considerations. However, method 2 is employed in implementing the fuzzy unsupervised clustering algorithm. From Fig. 4, we find that the length ratio of OC to QC is about 2.7. The grades of membership which are obtained by method 1 do not satisfy this ratio relationship. On the other hand, according to the location of the pixel in Fig. 4 and Fig. 5, U_{OC} and U_{PC} should be the same. But, U_{OC} and U_{PC} which are calculated by method 3 are not unique.

The fuzzy unsupervised clustering algorithm proposed in this paper will utilize the first passing of the unsupervised clustering algorithm which is presented in previous section. Then, during the second passing, instead of using the pixel classification method of the unsupervised clustering algorithm, the second fuzzy method is used. After using this algorithm, several maps are generated. Each map represents one single geographic attribute. Each pixel of the map is contributes its grade of membership to that geographic attribute.

CHATPER 4

TEST IMAGE

The testing image data is composed of several binary files which are the seven bands' data of the Landsat satellite remote sensing ***TM***. Table 2 lists the seven spectral bands of ***TM*** of Landsat, along with a brief summary of the intended principal applications of each [6].

The image size of the testing image is 320 x 200 which comprises four 320 x 50 bytes' binary files with 8 bytes of file header, i.e, the length of a file is 16808 bytes. Each pixel is represented by one byte which can depict an integer between 0 to 255. There is no priori knowledge about this research area but only those gray level data are contained in those files.

Table 2 Thematic Mapper Spectral Bands (Adapted from Lillesand, Keifer, *Remote Sensing and Image Interpretation*, 2nd ed., John Wiley & Sons, New York, 1987)

Band	Wavelength (μm)	Nominal Spectral Location	Principle Applications
1	0.45 - 0.52	Blue	Designed for water body penetration, making it useful for coastal water mapping . Also useful for soil/vegetation discrimination, forest type mapping, and cultural feature identification.
2	0.52 - 0.60	Green	Designed to measure green reflectance peak of vegetation for vegetation discrimination and vigor assessment. Also useful for culture feature identification.
3	0.63 - 0.69	Red	Designed to sense in a chlorophyll absorption region aiding in plant species differentation. Also useful for culture feature identification.
4	0.76 - 0.90	Near-infrared	Useful for determining vegetation types, vigor, and biomass content, for delineating water bodies, and for soil moisture discrimination.
5	1.55 - 1.75	Mid-infrared	Indicative of vegetation moisture content & soil moisture. Also useful for differentiation of snow from clouds.
6	10.4 - 12.5	Thermal-infrared	Useful in vegetation stress analysis , soil moisture discrimination , and thermal mapping applications.
7	2.08 - 2.35	Mid-infrared	Useful for discrimination of mineral and rock types . Also sensitive to vegetation moisture content.

CHAPTER 5

EXPERIMENT AND RESULT

According to Table 2, different bands can be combined to identify various land cover types. Three of the seven bands in Landsat *TM* which are Bands 2, 3 and 4, are employed in this study. Also three sets of **R** and **D** are used to test the proposed algorithm. They are 15 and 15, 15 and 20, and 15 and 25.

First, the unsupervised clustering classifica^Xxon is implemented. Scatterplots of the cluster mean vectors using those three sets of **R** and **D** are displayed in Figs. 6, 7, 8, 9, 10 and 11.
The mean vector values for these three sets of testing radii are summarized in Tables 3, 4 and 5, respectively.

The cluster labeling is according to the locations of their cluster mean vectors in this three-dimensional spectral space. Many references provide the information about the spectral attributes of certain kinds of land cover types in remote sensing [5] [6] [7] [18] [19]. Fig. 13, for example, scatterplots of their cluster means are displayed in Figs. 8 and 9. 8 clusters are extracted when the decision radii are **R** = 15 and **D** = 20. In the band 2 to band 3 scatterplot, Fig. 8, the 8 clusters lie on a diagonal extending from the origin of

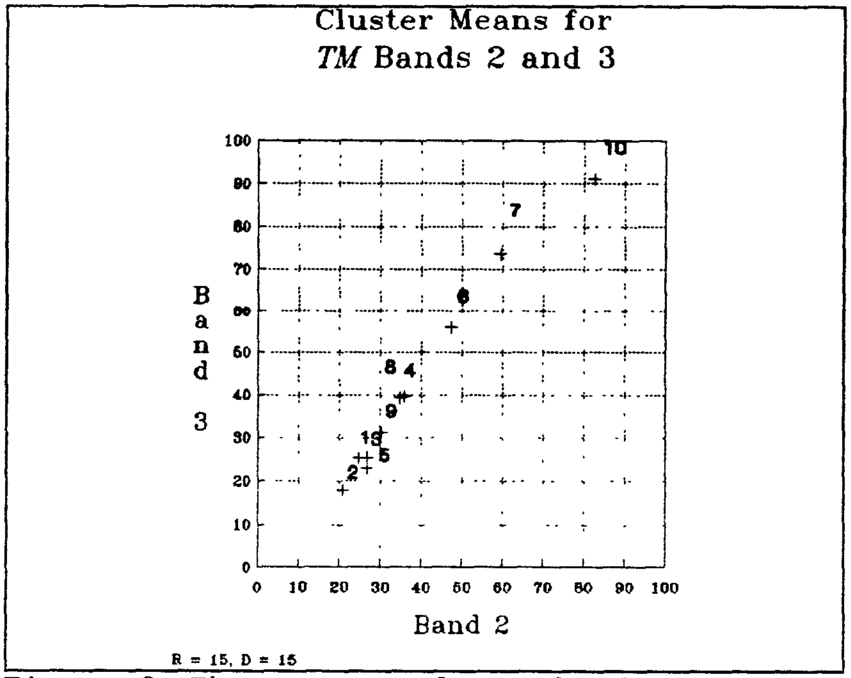


Figure 6 The scatterplot of cluster mean vectors using Bands 2 and 3 where **R** = 15 and **D** = 15. The mean vector values are summarized in Table 3.

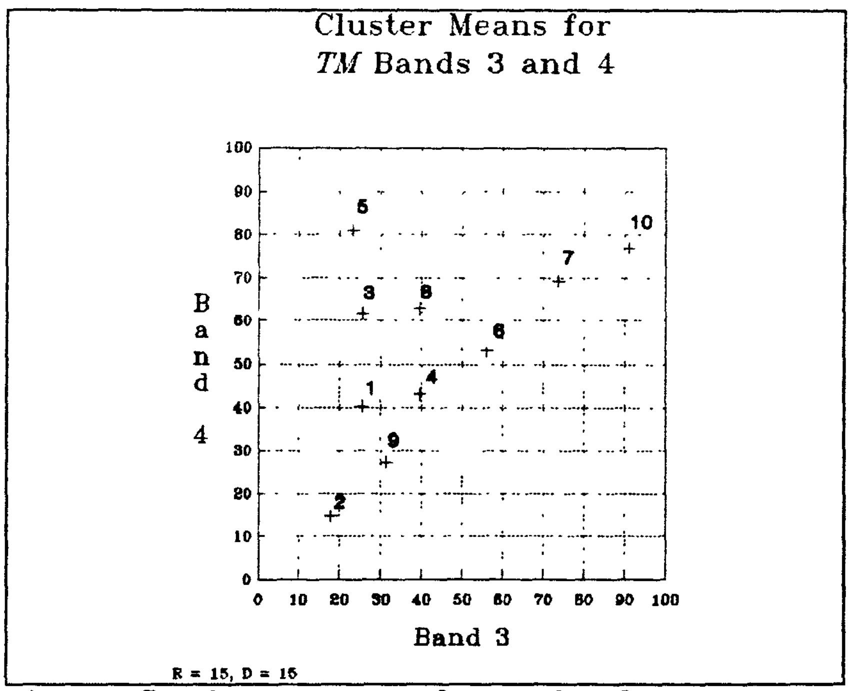


Figure 7 The scatterplot of cluster mean vectors using Bands 3 and 4 where **R** = 15 and **D** = 15. The mean vector values are summarized in Table 3.

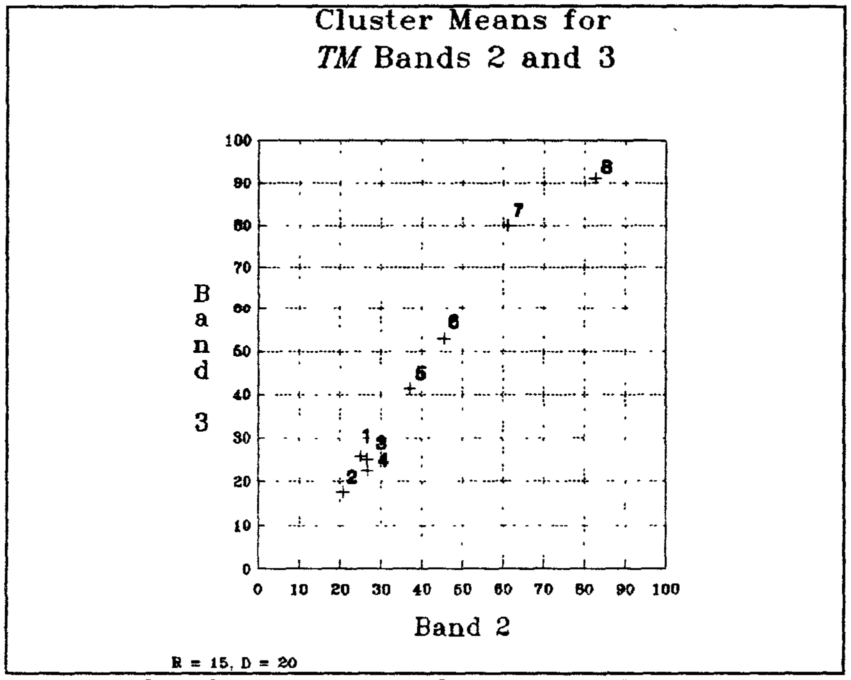


Figure 8 The scatterplot of cluster mean vectors using Bands 2 and 3 where **R** = 15 and D = 20. The mean vector values are summarized in Table 4.

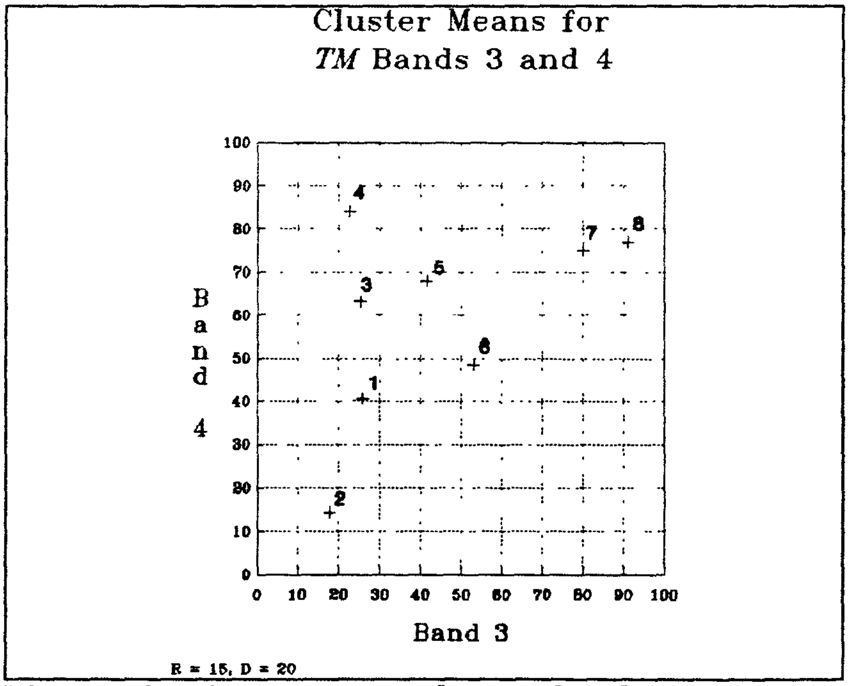


Figure 9 The scatterplot of cluster mean vectors using Bands 3 and 4 where **R** = 15 and D = 20. The mean vector values are summarized in Table 4.

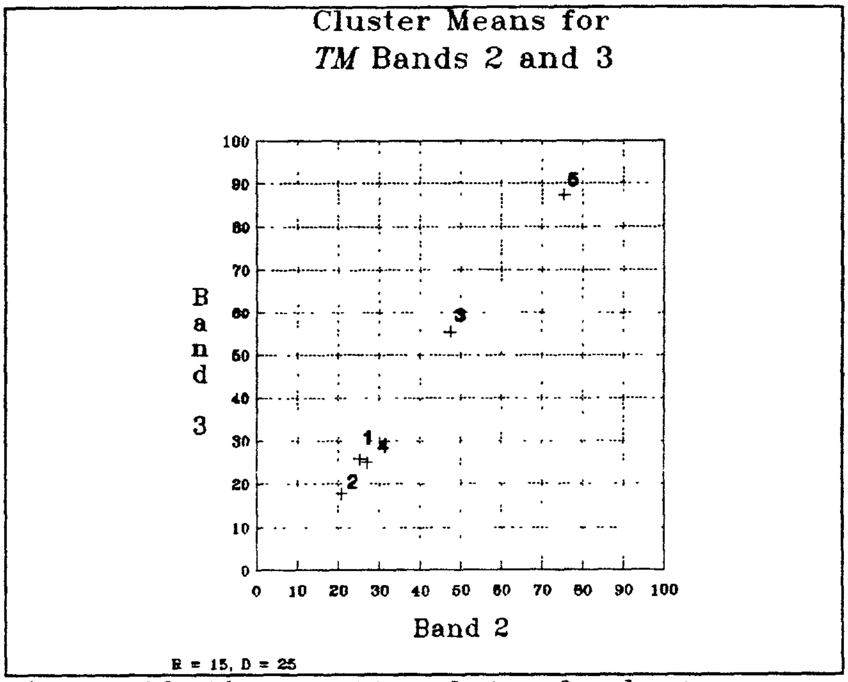


Figure 10 The scatterplot of cluster mean vectors using Bands 2 and 3 where R = 15 and D = 25. The mean vector values are summarized in Table 5.

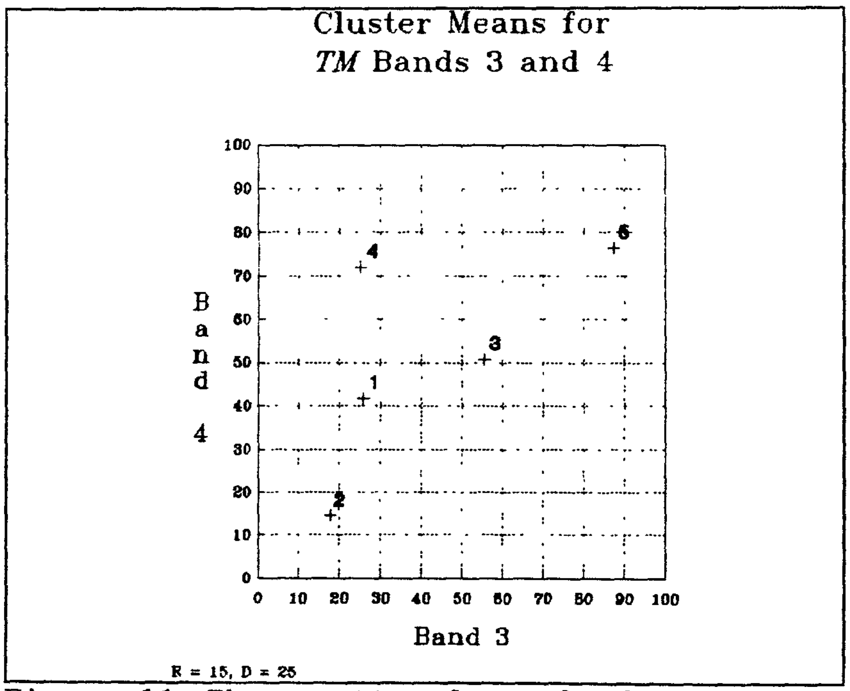


Figure 11 The scatterplot of cluster mean vectors using Bands 3 and 4 where R = 15 and D = 25. The mean vector values are summarized in Table 5.

Table 3 Cluster Mean Values of *TM* Bands 2, 3 and 4. Ten clusters have been built when **R** = 15 and **D** = 15. The picture of this table is shown in Fig. 12.

cluster	Number of Pixels	Mean Vector Band 2	Band 3	Band 4	Class Description	Color Assignment
1	42326	24.61	25.32	40.38	Commercial 1	Blue
2	284	20.61	17.85	14.68	Water	Green
3	4438	26.62	25.44	61.60	Forest 1	Cyan
4	1994	35.61	39.64	43.17	Residential 1	Red
5	24	26.61	23.07	80.84	Forest 2	Magenta
6	0	47.33	56.10	53.04	Residential 2	Brown
7	0	59.33	73.67	69.33	Commercial 2	Light Gray
8	4813	34.67	39.33	62.67	Park	Dark Gray
9	10103	30.17	31.24	27.19	Water & wetland	Light Blue
10	18	82.50	91.00	77.00	Commercial 3	Light Green

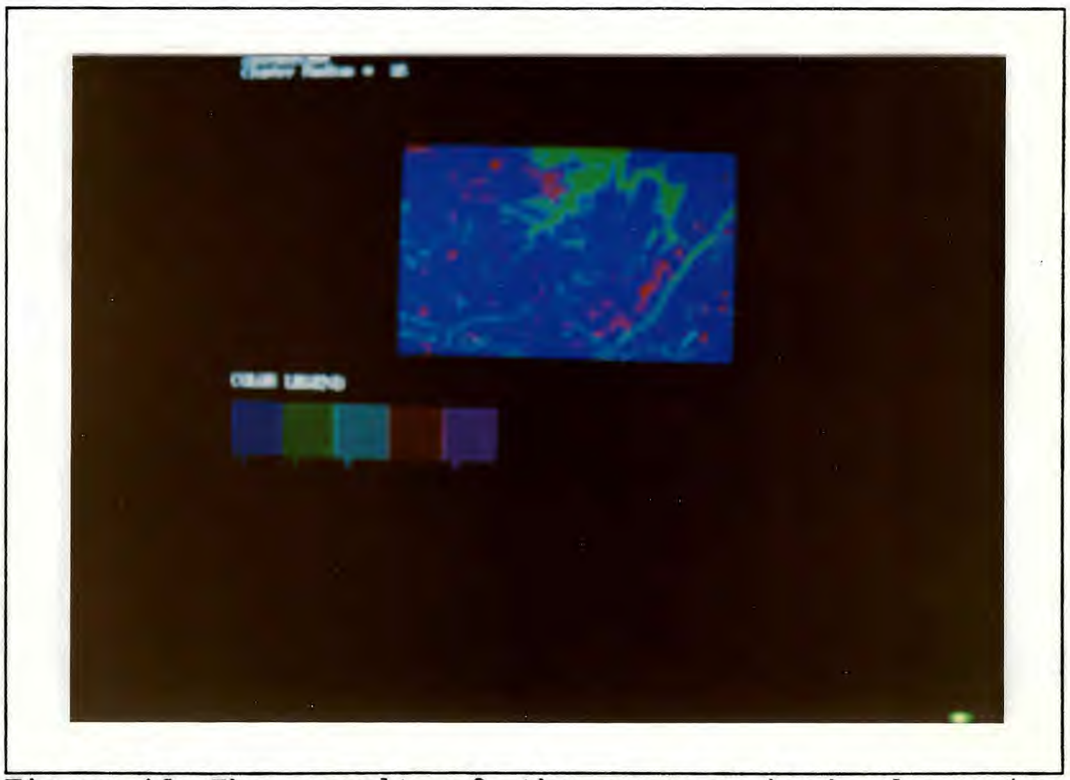


Figure 12 The result of the unsupervised clustering algorithm in Landsat *TM* by using **R** = 15 and **D** = 15. The categories of this classification are shown in Table II.

Table 4 Cluster Mean Values of *TM* Bands 2, 3 and 4. When R = 15 and D = 20, eight clusters is built. The picture of this table is shown in Fig. 13.

Cluster	Number of Pixels	Mean Vector Band 2	Mean Vector Band 3	Mean Vector Band 4	Class Description	Color Assignment
1	48292	24.95	25.74	40.54	Commercial 1	Blue
2	6701	20.59	17.75	14.33	Water	Green
3	5398	26.52	25.10	63.20	Forest 1	Cyan
4	71	26.63	22.59	84.04	Forest 2	Red
5	2406	37.00	41.50	68.00	Park	Magenta
6	1087	45.36	53.00	48.52	Residential	Brown
7	27	61.00	80.00	75.00	Commercial 2	Light Gray
8	18	82.50	91.00	77.00	Commercial 3	Dark Gray

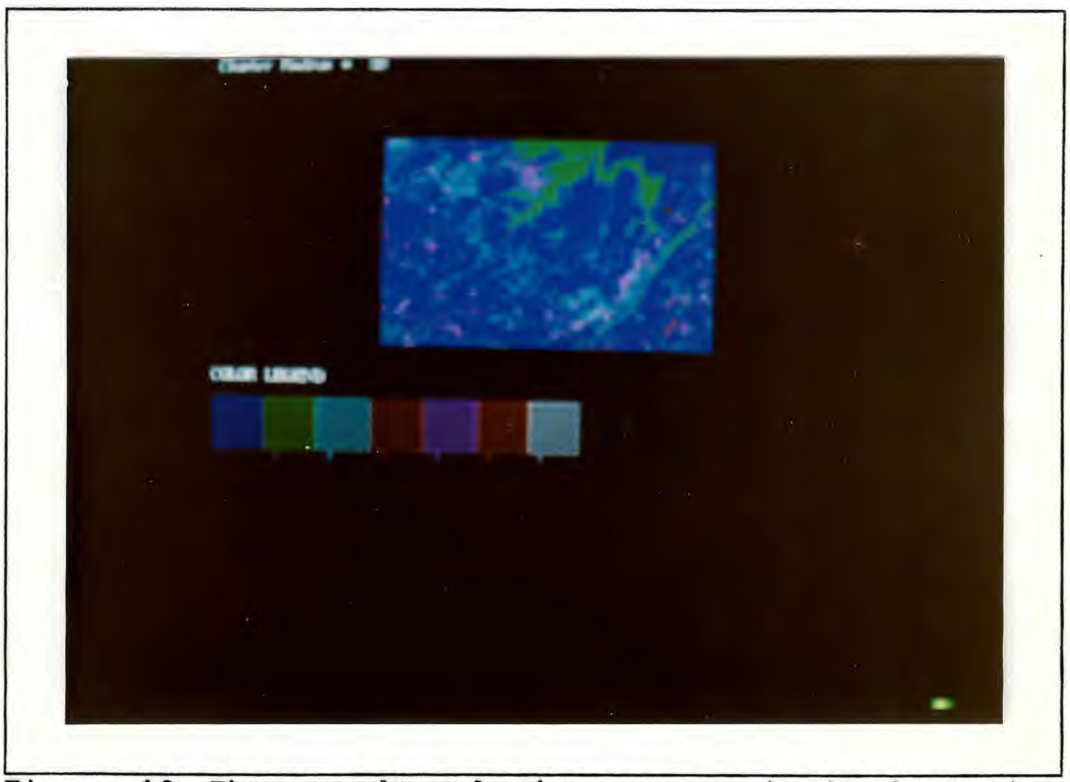


Figure 13 The result of the unsupervised clustering algorithm in *TM* by using R = 15 and D = 20. The categories of this classification are shown in Table III.

Table 5 Cluster Mean Values of *TM* Bands 2, 3 and 4. When **R** = 15 and **D** = 25, five clusters will be built. The picture of this table is shown in Fig. 14.

Cluster	Numbers of Pixels	Mean Vector Band 2	Band 3	Band 4	Class Description	Color Assignment
1	51912	25.03	25.71	41.67	Commercial 1	Blue
2	7243	20.61	17.78	14.46	Water	Green
3	796	47.23	55.51	50.65	Residential & park	Cyan
4	4025	27.00	25.00	72.00	Forest	Red
5	24	75.33	87.33	76.33	Commercial 2	Magenta

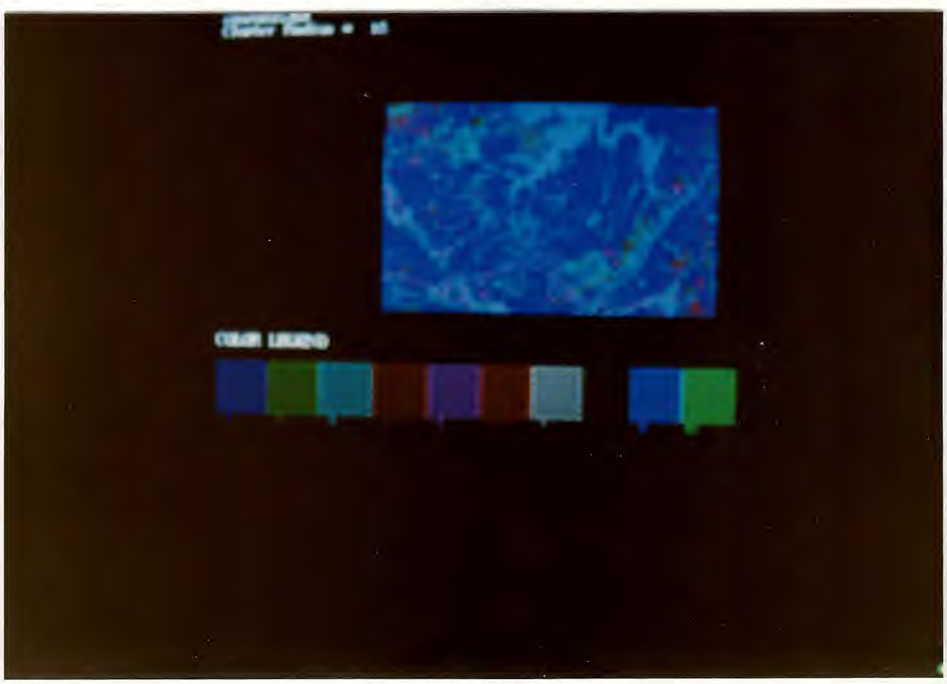


Figure 14 The result of the unsupervised clustering algorithm in Landsat *TM* by using **R** = 15 and **D** = 25. The categories of this classification are summarized in Table IV.

the plot. In Fig. 9, the shape of the distributions of the 8 clusters looks like a triangle.

Cluster 2 is a very distinct cluster. In Landsat *TM* images, water is the unique land cover type such that its band refelectance intensity values always decrease as the band number increases, except for Band 6. Therefore, it is not difficult to assign cluster 2 to the class of water. Clusters 3 and 4 have a high reflectance in the near-infrared (Band 4) with low reflectance in the red (Band 3) due to chlorophyll absortion. These two cluster were both assigned to the forest class.

Cluster 3 was assigned as residential and park land use. It lies between the commercial and forest area. Since the residential housing and park is composed of a mixture of vegetated and nonvegetated surface. Three clusters were associated with commercial land use. Clusters 4 and 5 reflected high amounts of both red and near-infrared energy, as commercial land used composed of concrete and bare soil often does. Cluster 1 was also assigned as commercial area where vegetation is more abundant.

After generating the cluster mean vectors, the proposed fuzzy unsupervised clustering algorithm is used to obtain the fuzzy map for the individual land cover type which can be used as the input file of a geographic information system. In the previous experiment, three sets of cluster mean vectors have been generated. Now, we will compute each pixel's membership grades to the cluster mean vectors. The dimension of the membership grade vector is depended on the numbers of cluster mean vectors. If 10 cluster mean vectors is used, then each pixel shall have ten membership grades and the sum of these membership grades is equal to 1. Tables 6, 7 and 8 disclose the distribuation of the membership grades to each cluser mean vetors. The membership map are shown in Figs. 15(a) - 15(j), 16(a) - 16(h) and 17(a) - 17(e).

Compare Tables 7 and 4, we will find that the distribution of the membership grades is quite reasonable. When we use the R = 15 and D = 20 as the decision radii, eight cluster mean vectors are extracted, i.e., each pixel will have eight membership grade. In cluster 8, for example, most of the membership grades are concentrated in the range between 0 to 9. From Table 4, we know that there are only 18 pixels in this cluster. In the other words, these pixels are much near to this cluster mean vector in comparison with the distances to the other cluster mean vectors.

In Table 9, several pixels are selected to show the difference between the hard partition method and the fuzzy approach in remote sensing image classification. These pixels are located in the first row of the test image. Pixels 1, 10,

Table 6 The Membership Grades Distribution of Fuzzy Unsupervised Clustering Algorithm with **R** = 15 and **D** = 15. Each row represents a cluster. The total number of pixels is 64000.

Cluster	Mean Vector			The Grade of Membership (%)									
NO.	Band2	Band3	Band4	00-09	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-100
1	24.61	25.32	40.38	8148	12367	15389	12393	8358	4576	2206	426	137	0
2	20.61	17.85	14.68	45797	11734	1951	554	495	2666	546	198	49	10
3	26.62	25.44	61.60	42259	13886	3912	2015	1021	504	269	99	35	0
4	35.61	39.64	43.17	37607	22382	2578	864	356	133	62	15	3	0
5	26.61	23.07	80.84	62344	989	271	167	95	56	41	30	4	3
6	47.33	56.10	53.04	63328	516	89	35	20	6	3	3	0	0
7	59.33	73.67	69.33	63976	18	3	1	1	1	0	0	0	0
8	34.67	39.33	62.67	51202	12433	291	46	17	7	2	1	1	0
9	30.17	31.24	27.19	25931	32526	4379	661	226	139	85	42	7	1
10	82.50	91.00	77.00	63997	1	0	0	0	0	1	1	0	0

R = 15, D = 15

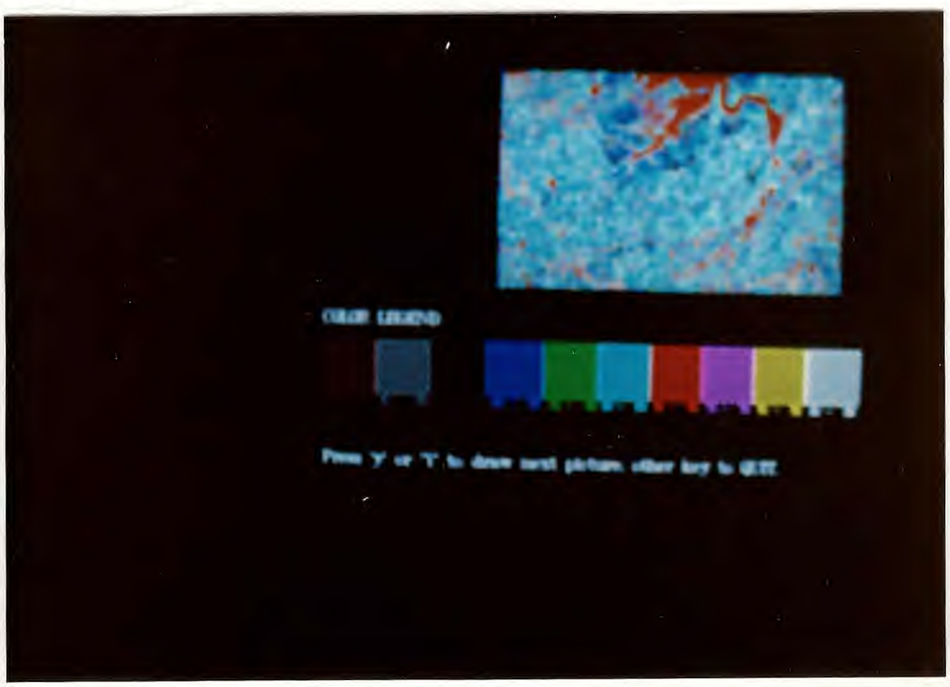


Figure 15(a) The Fuzzy Membership Map of *Commercial 1*. Figs. 15(a) - 15(i) show the fuzzy membership maps. See Table V for detail pixel'x membership grades distribution.

R = 15, D = 15

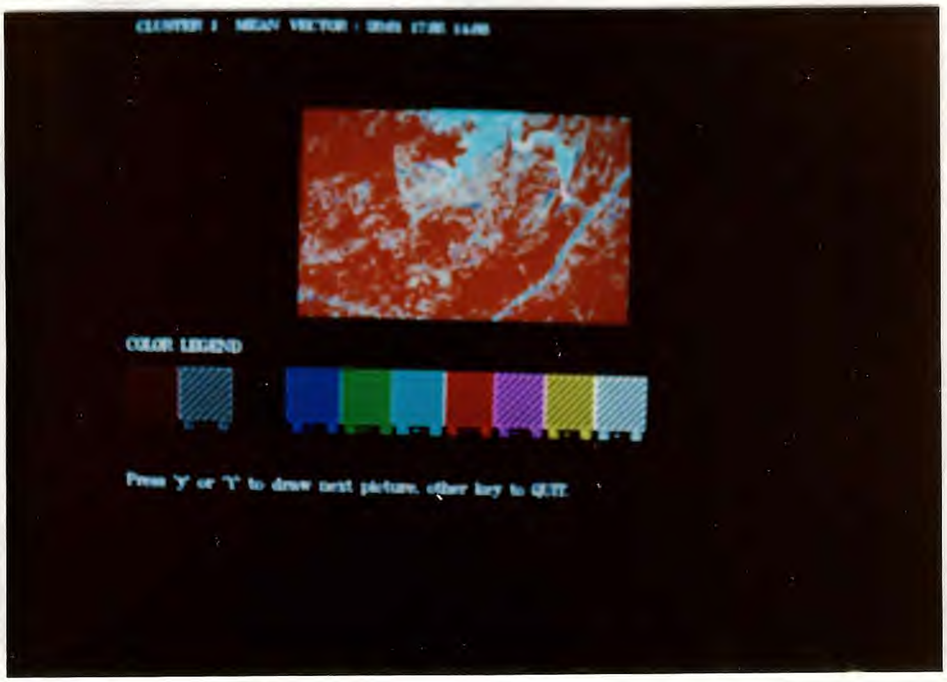


Figure 15(b) The Fuzzy Mambership Map of *Water*.

R = 15, D = 15

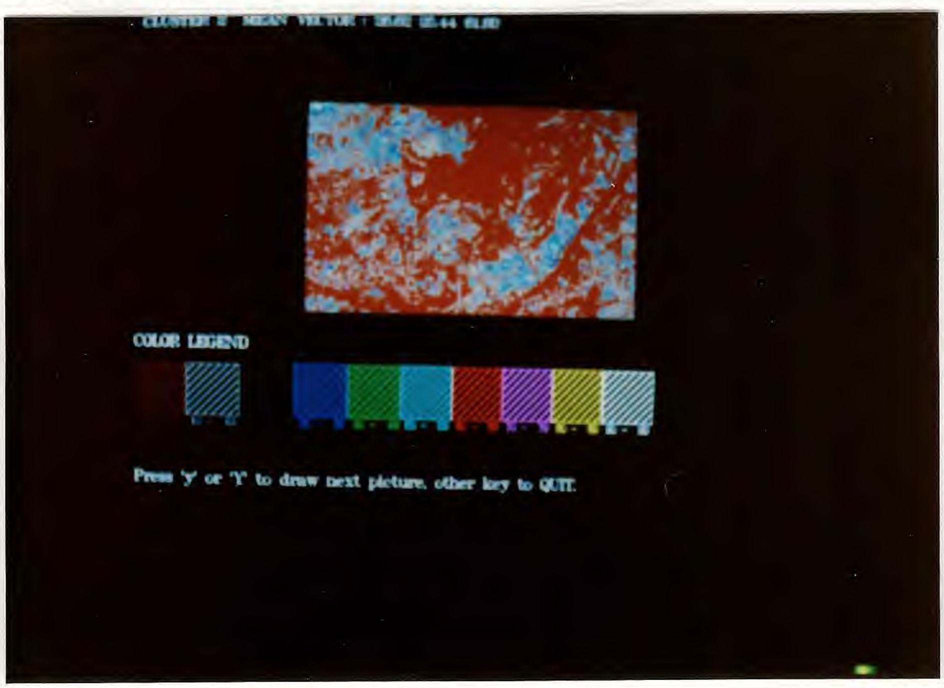


Figure 15(c) The Fuzzy Membership Map of *Forest 1*.

R = 15, D = 15

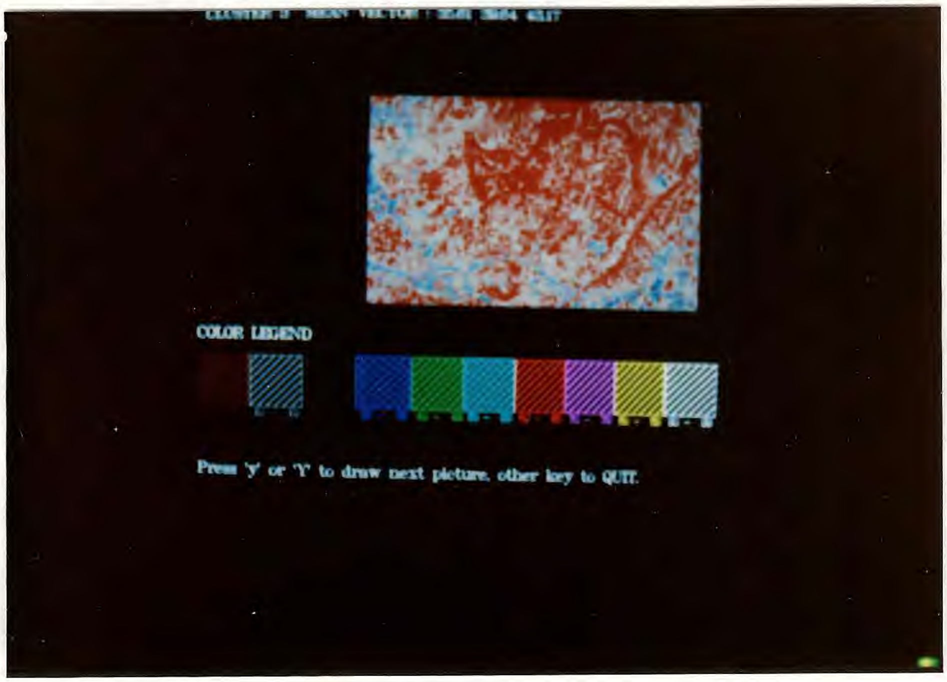


Figure 15(d) The Fuzzy Membership Map of *Residential 1*.

R = 15, D = 15

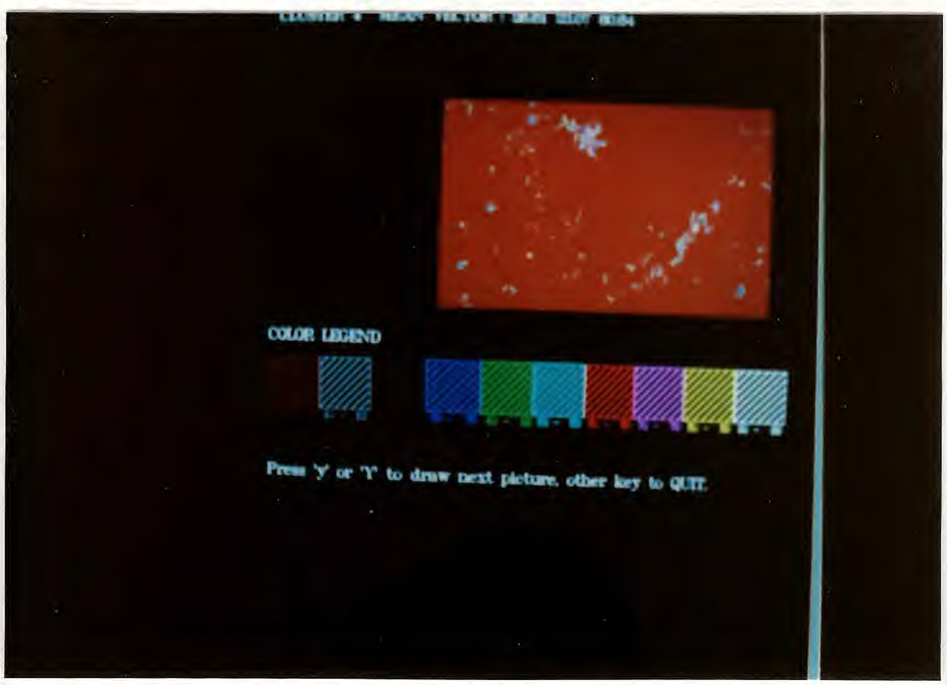


Figure 15(e) The Fuzzy Membership Map of *Forest 2*.

R = 15, D = 15

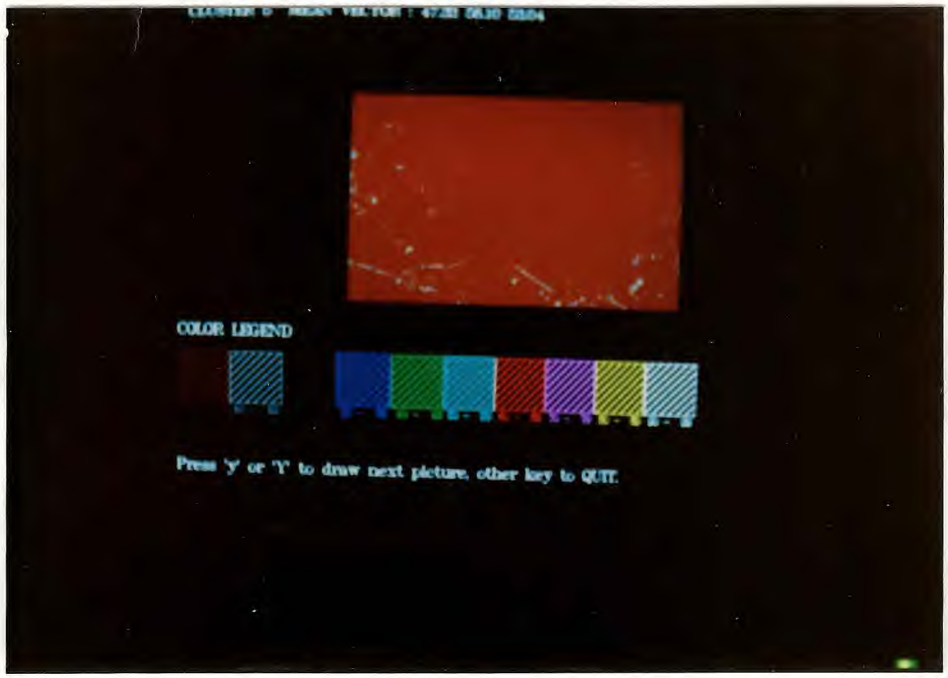


Figure 15(f) The Fuzzy Membership Map of *Residential 2*.

R = 15, D = 15

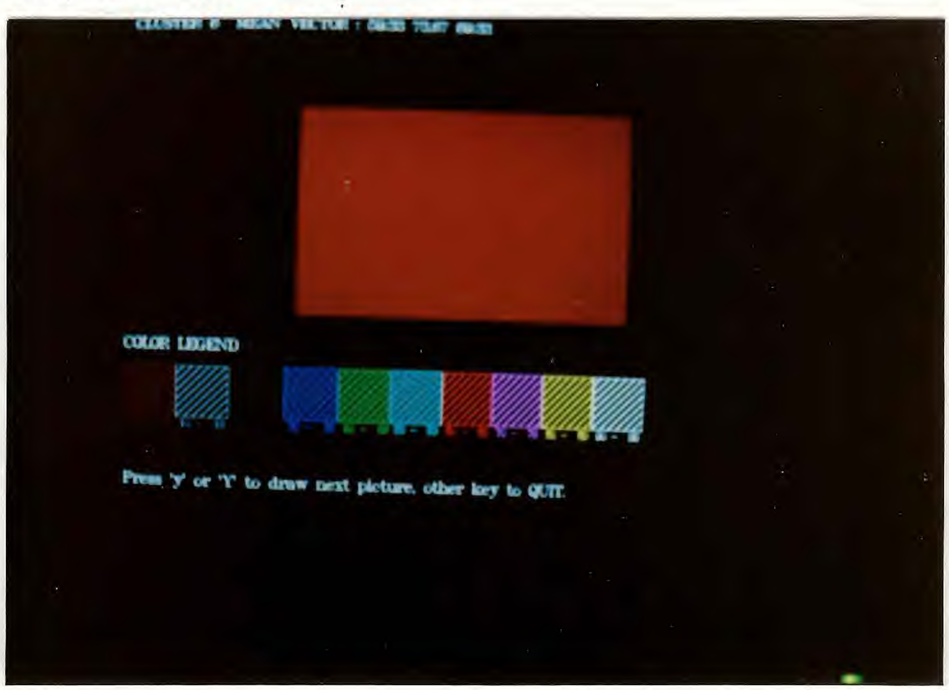


Figure 15(g) The Fuzzy Membership Map of *Commercial 2*.

R = 15, D = 15

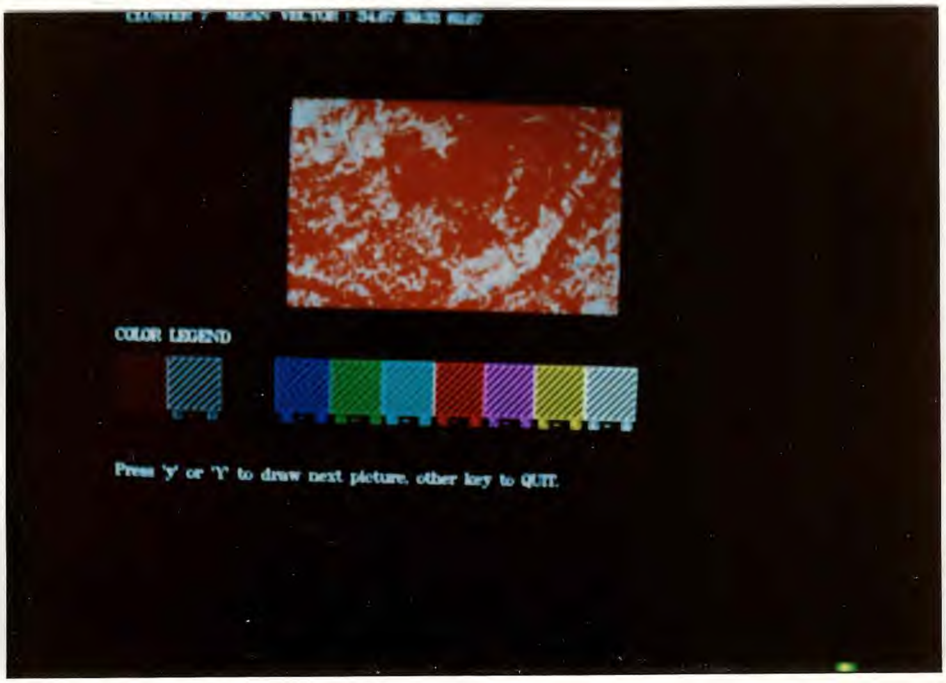


Figure 15(h) The fuzzy Membership Map of *Park*.

R = 15, D = 15

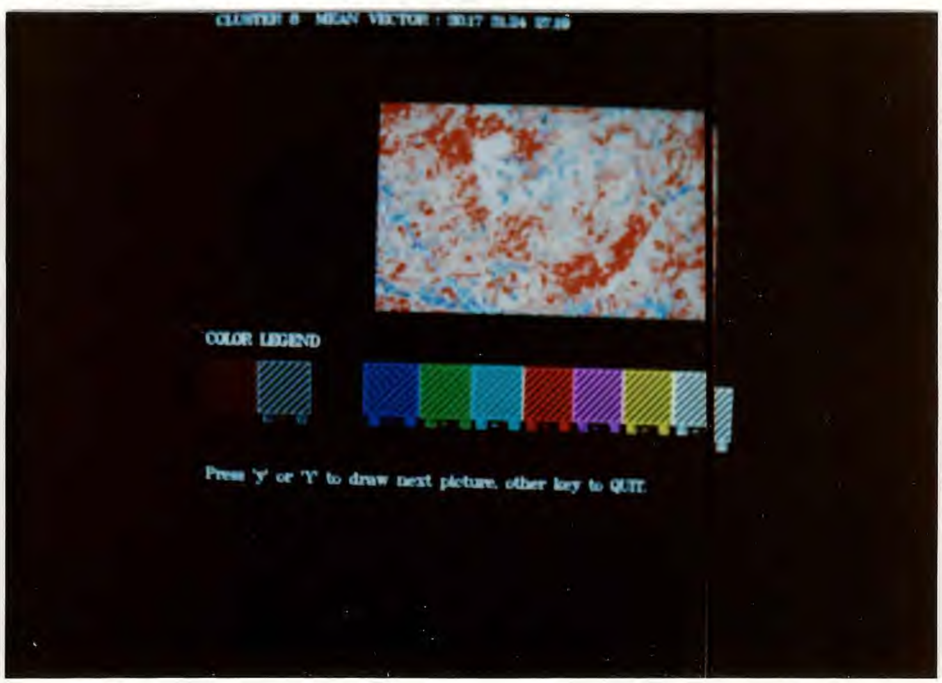


Figure 15(i) The Fuzzy Membership Map of *Water & Wetland*.

41

R = 15, D = 15

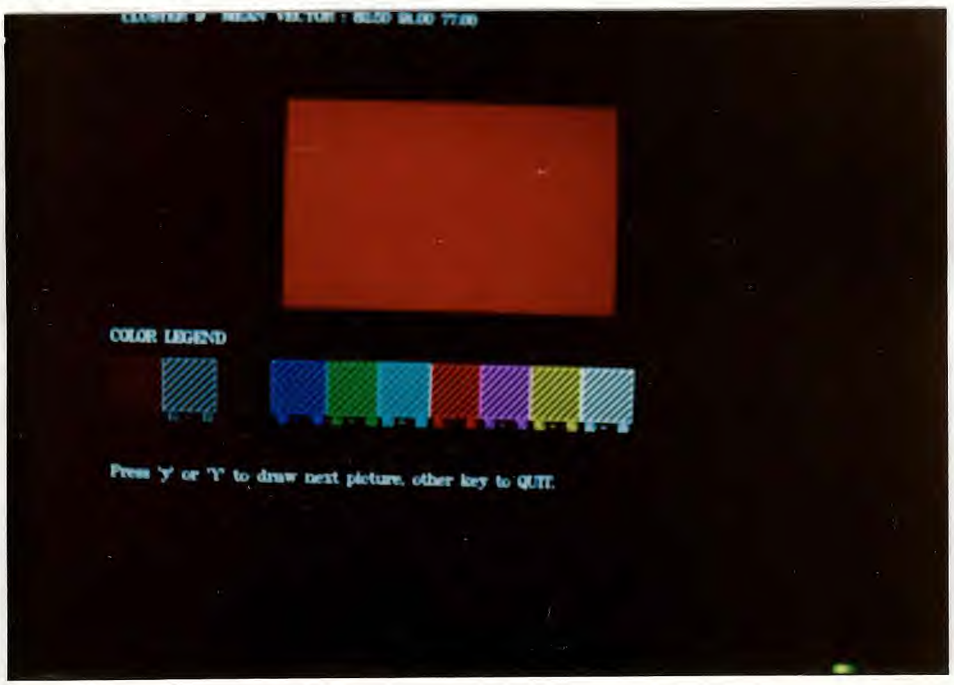


Figure 15(j) The Fuzzy Membership Map of *Commercial 3*.

Table 7 The Membership Grades Distribution of Fuzzy Unsupervised Clustering Algorithm with **R** = 15 and **D** = 20. Each row represents a land cover type. The total number of pixels is 64000.

Cluster	Mean Vector			The Grade of Membership (%)									
NO.	Band2	Band3	Band4	00-09	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-100
1	24.95	25.74	40.54	5421	5279	9873	12780	12380	9597	5695	2299	564	112
2	20.59	17.75	14.33	36009	16728	5065	1636	513	406	2270	1134	211	28
3	26.52	25.10	63.20	17474	35175	6178	2354	1358	786	433	181	61	0
4	26.63	22.59	84.04	60728	2676	271	146	71	33	42	21	9	3
5	37.00	41.50	68.00	45681	18001	278	26	11	1	2	0	0	0
6	45.36	53.00	48.52	55001	7864	765	209	88	39	18	9	7	0
7	61.00	80.00	75.00	63971	24	2	2	0	0	0	0	0	1
8	82.50	91.00	77.00	63992	6	0	0	0	0	0	2	0	0

R = 15, D = 20

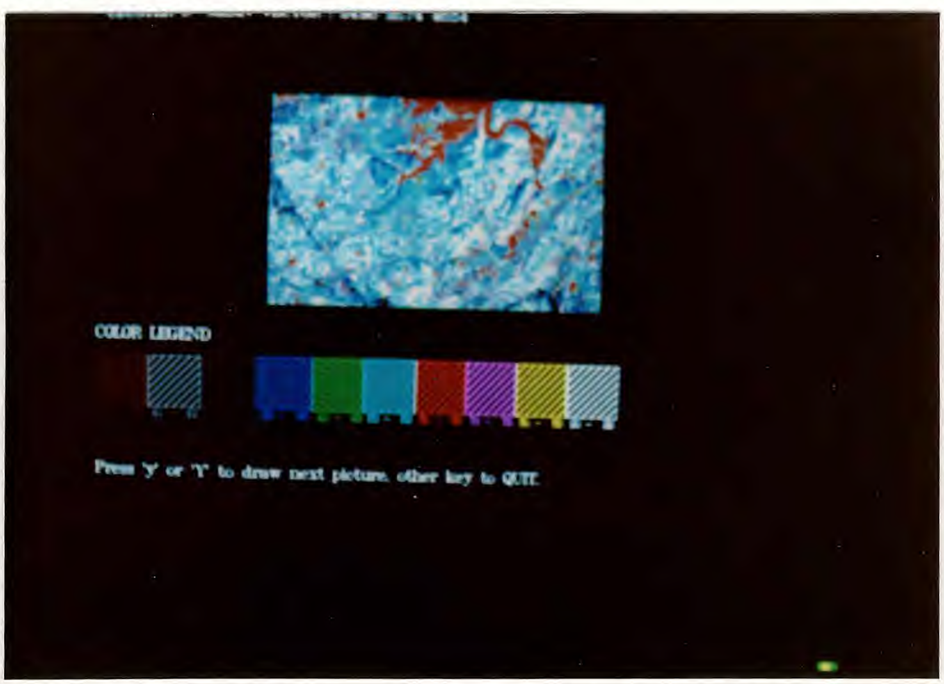


Figure 16(a) The Fuzzy Membership Map of *Commercial 1*. Figs. 16(a) - 16(h) show the fuzzy membership maps. See Table VI for detail pixel's membership grades distribution.

R = 15, D = 20

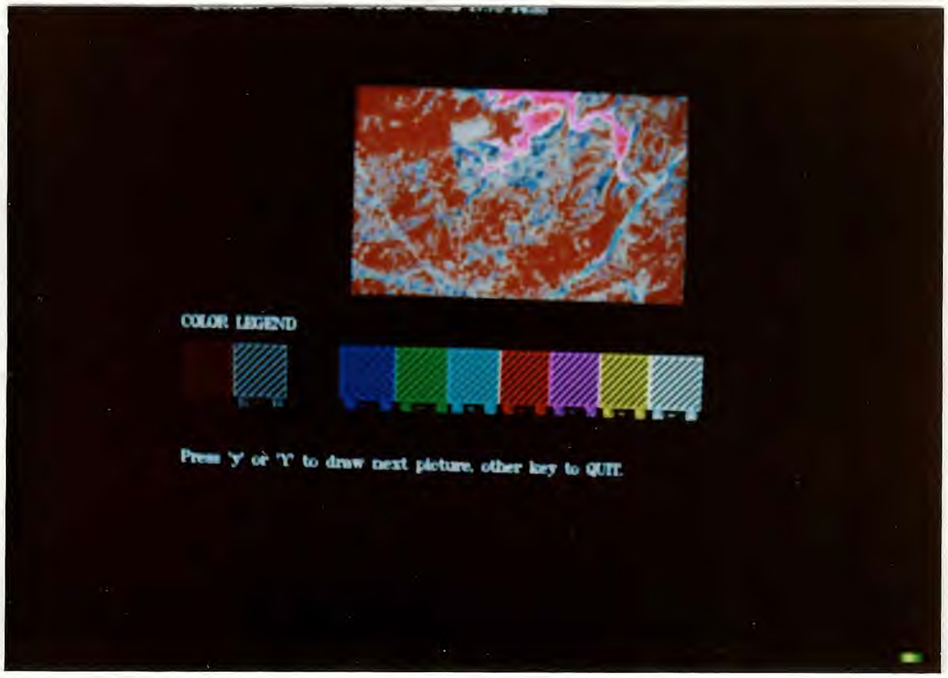


Figure 16(b) The Fuzzy Membership Map of *Water*.

R = 15, D = 20

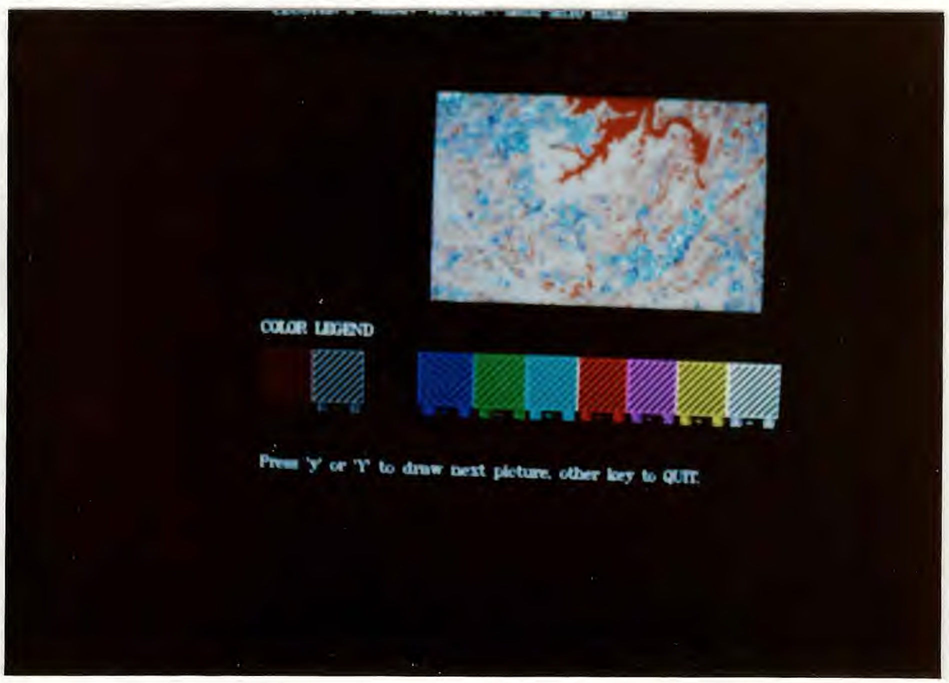


Figure 16(c) The Fuzzy Membership Map of *Forest 1*.

R = 15, D = 20

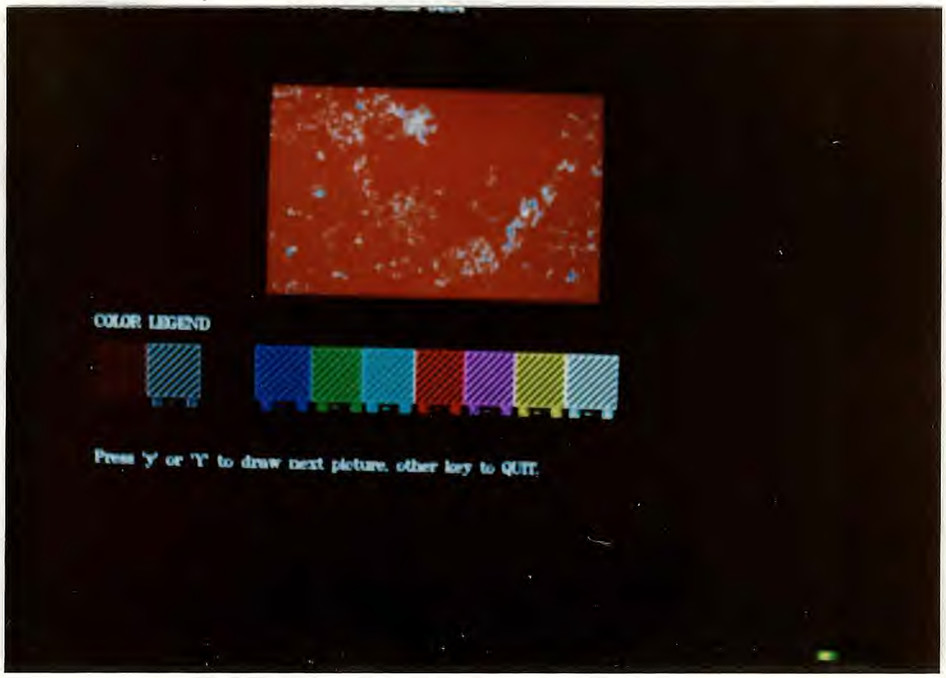


Figure 16(d) The Fuzzy Membership Map of *Forest 2*.

R = 15, D = 20

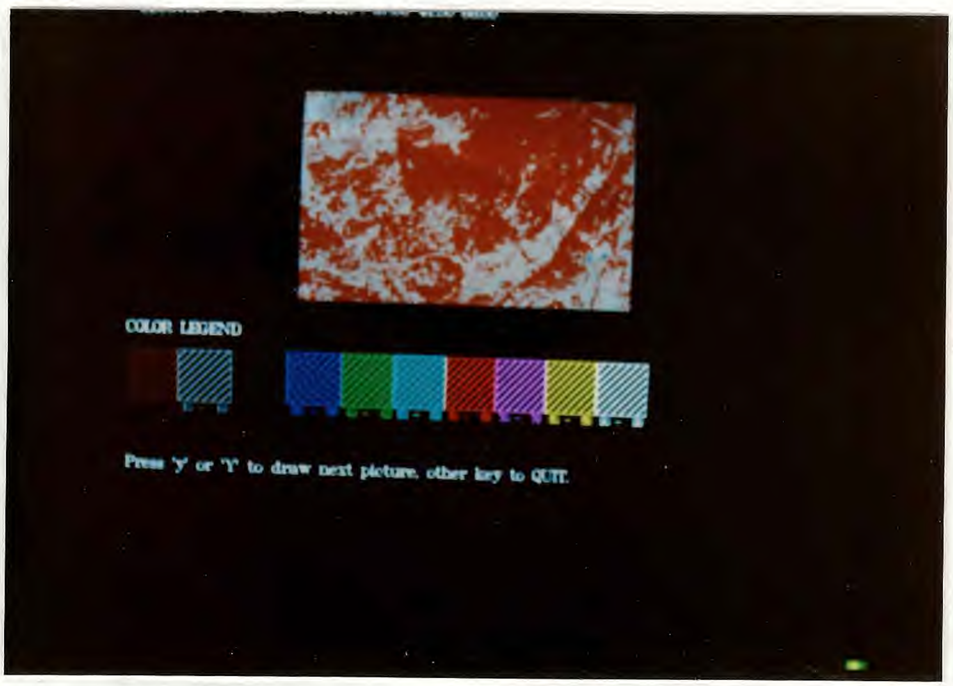


Figure 16(e) The Fuzzy Membership Map of *Park*.

R = 15, D = 20

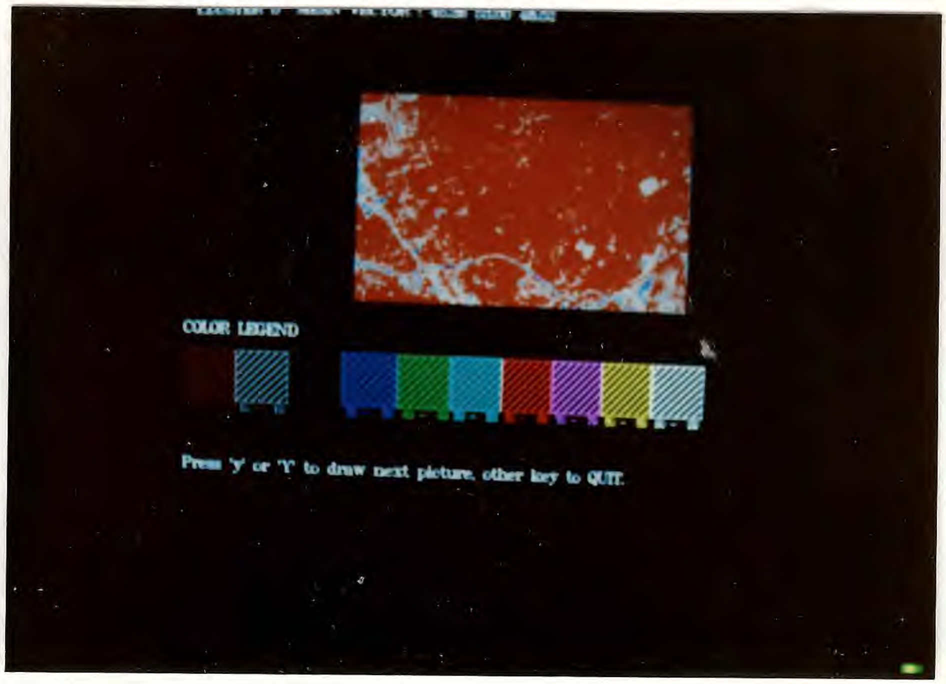


Figure 16(f) The Fuzzy Membership Map of *Residential*.

R = 15, D = 20

Figure 16(g) The Fuzzy Membership Map of *Commercial 2*.

R = 15, D = 20

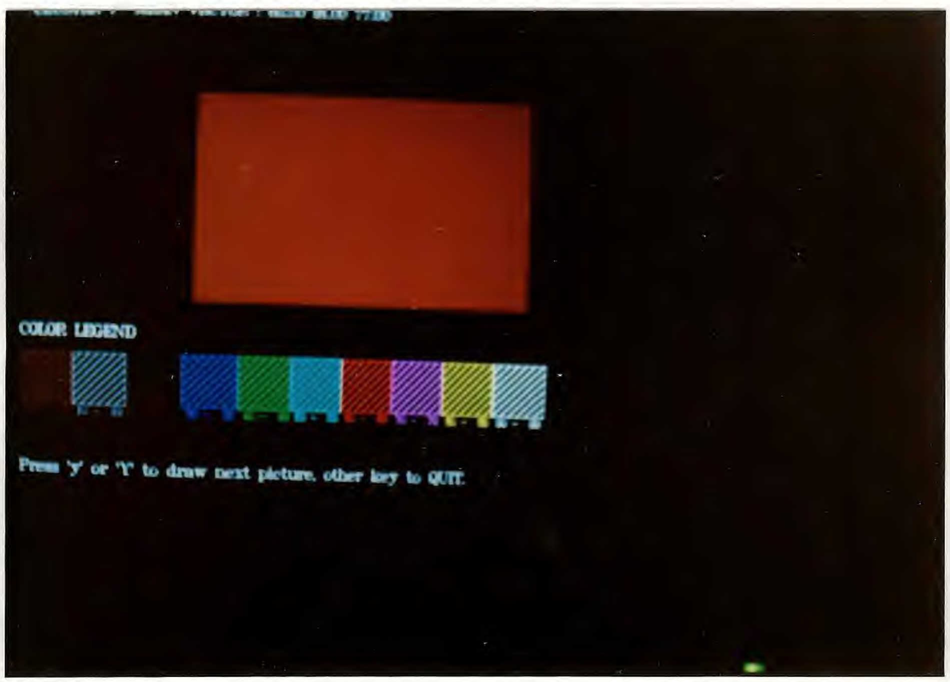


Figure 16(h) The Fuzzy Membership Map of *Commercial 3*.

Table 8 The Membership Grade Distribution of Fuzzy Unsupervised Clustering Algorithm with **R** = 15 and **D** = 25. Each row represents a land cover type. The total number of pixels is 64000.

Cluster NO.	Mean Vector			The Grade of Membership (%)									
	Band2	Band3	Band4	00-09	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-100
1	25.03	25.71	41.67	2461	3314	3605	8322	11776	12910	10947	7684	2784	221
2	20.61	17.78	14.46	13239	32461	8714	3454	1450	580	389	2935	715	63
3	47.23	55.51	50.65	31951	29333	1932	476	173	72	33	21	9	0
4	27.00	25.00	72.00	19350	34468	5485	2168	889	619	449	367	172	25
5	75.33	87.33	76.33	63948	45	3	1	1	0	1	1	0	0

R = 15, D = 25

Figure 17(a) The Fuzzy Membership Map of *Commercial 1*. Figs. 17(a) - 17(e) show the fuzzy membership maps. See Table VII for detail pixel's membership grades distribution.

R = 15, D = 25

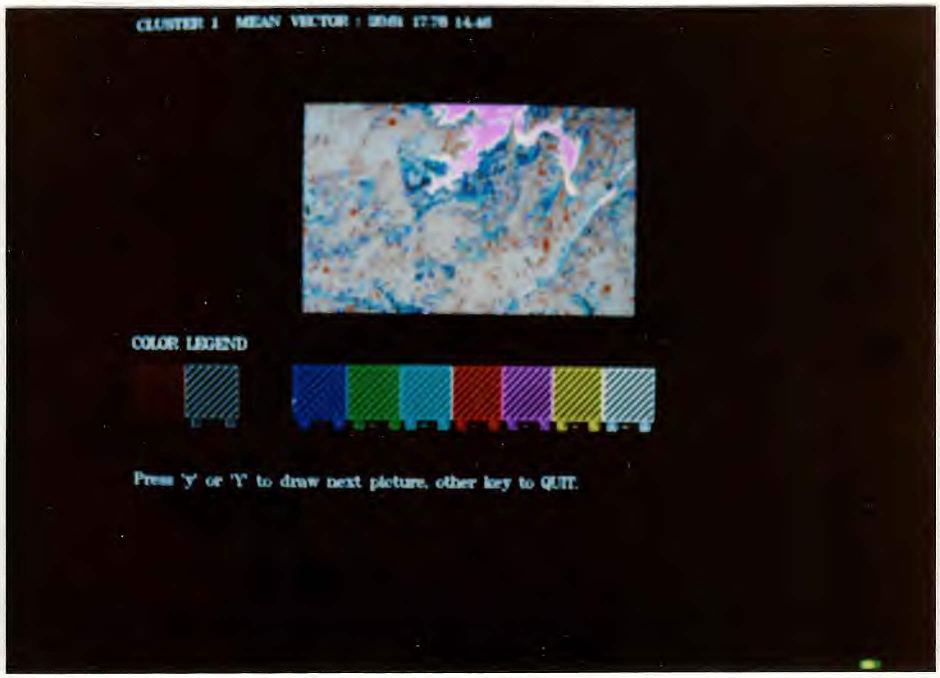


Figure 17(b) The Fuzzy Membership Map of *Water*.

R = 15, D = 25

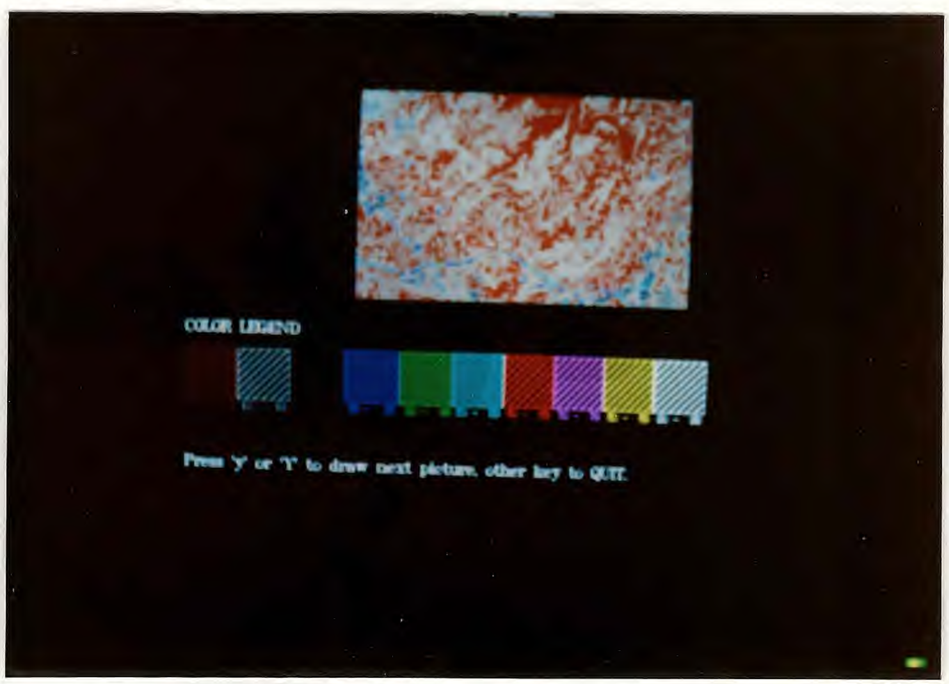


Figure 17(c) The Fuzzy Membership Map of *Residential & Park*.

R = 15, D = 25

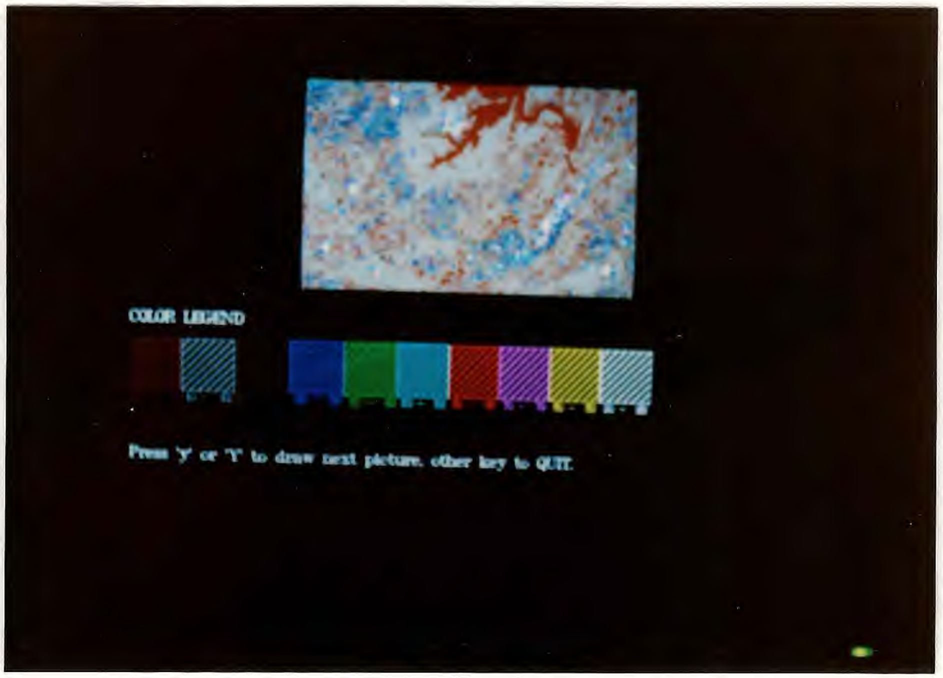


Figure 17(d) The Fuzzy Membership Map of *Forest*.

R = 15, D = 25

Figure 17(e) The Fuzzy Membership Map of *Commercial 2*.

80, 110, 160, 170 and 240 are chosen. In the hard partition method, pixels 110, 160 and 170 are classified in the cluster of *Water*. Pixels 1, 240 are assigned to the cluster of *Commercial 1*. Pixel 80 is classified in the cluster of *Forest 1*. From the viewpoint of the fuzzy membershtp representation, the class assigned to the pixel has the majority of the membership, i.e., its grade of membership is larger, in most case. However, every predefined land cover type will show its membership to this pixel.

On the other hand, the hard partition may misclassify the pixel. Pixel 110, for example, it is classified in the cluster of *Water* when the unsupervised clusering algorithm is used. But, in fact, it does not have the special spectral attribute of water in which the band reflectance intensity value will decrease as the band number increase. However, in fuzzy membership representation approach, we can explain this phenomenon. According to the concept of the mixed pixel, pixel 110 contains several geographic attributes in which *Water* and *Commercial 1* have the major contribution to its spectral characteristic. If this pixel is divided into numerous small portions, *Water* occupies one third of the area. Roughtly, about one fourth of the portions is belong to the land cover type of *Commercial 1*.

Table 9 This table shows the difference between the hard partition and the fuzzy approach. See Table II for the land cover classes.

Pixel	Gray Level			Hard Part. Class	The Grade of Membership							
NO.	Band2	Band3	Band4	Description	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8
1	26	28	44	*Commercial 1*	0.5423	0.0726	0.1190	0.0572	0.0780	0.0724	0.0331	0.0255
80	31	33	54	*Forest 1*	0.1910	0.0718	0.2430	0.0979	0.1801	0.1246	0.0527	0.0388
110	21	19	26	*Water*	0.2416	0.3396	0.1047	0.0683	0.0794	0.0840	0.0454	0.0371
160	20	17	10	*Water*	0.0916	0.6644	0.0542	0.0395	0.0452	0.0503	0.0296	0.0250
170	20	16	10	*Water*	0.0947	0.6523	0.0565	0.0411	0.0468	0.0519	0.0307	0.0259
240	21	22	32	*Commercial 1*	0.3627	0.2020	0.1154	0.0702	0.0836	0.0859	0.0445	0.0357

CHAPTER 6

CONCLUSION

Image classification becomes more important and more urgent in the application of remote sensing. It will help people understand the content of a remote sensing image. The picture itself is not only showing a colorful scene but also containing a lot of informations. Without an appropriate classification algorithm, these informations can not be interpreted correctly. The proposed two-pass-mode fuzzy unsupervised clustering algorithm provides a simple and straightforward way in interpreting the meaning of each pixel in the image without the complicate computation. And the output of this algorithm can be the input of the geographic information system, directly.

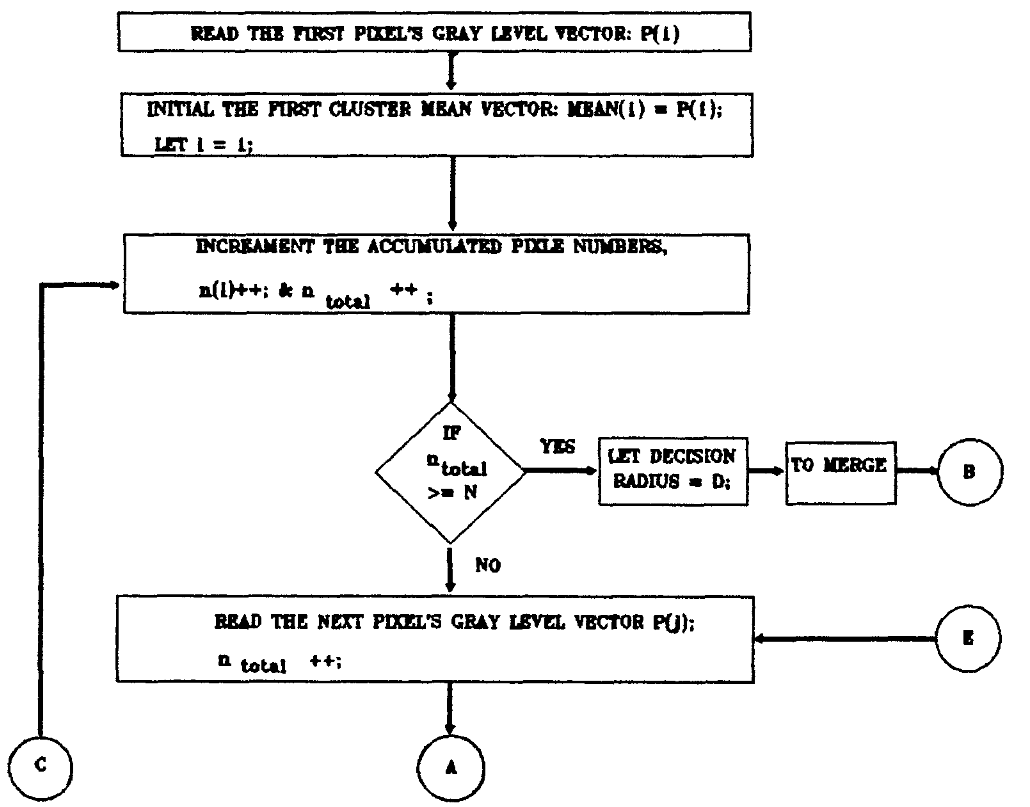
In our proposed algorithm, there are two important factors which will affect the number of clusters generated during the first passing, i.e., the decision radii, **R** and **D**. The samller **R** and **D** are, the more clusters will be formed. But it also costs more time and memory space in processing the data. On the contrary, if their are too large, it will increase the probability of misclassification. We suggest that R's and D's values should be between 15 - 20 and 15 - 30, respectively.

Interpreting and integrating ***Landsat*** remote sensing image and geographic information system is a difficult task. It needs the combination of knowledge from remote sensing, image processing, statistics, geographic information system, etc. We propose a possible method to implement the interpretation and integration. Expected results have been achieved. However, if we can combine this method with a spectral knowledge base in cluster labeling, then it is very possible to generate an automatic ***Landsat*** image interpretation system.

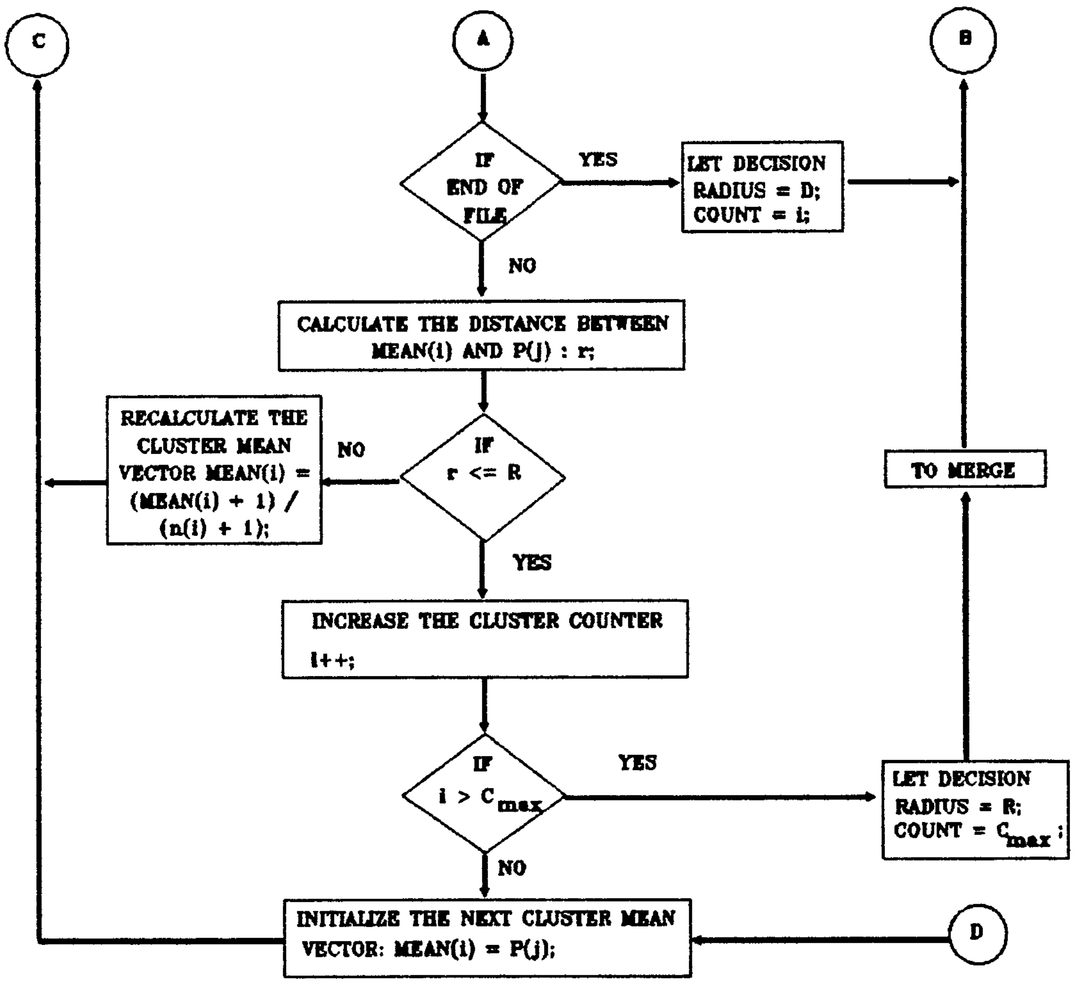
APPENDIX I

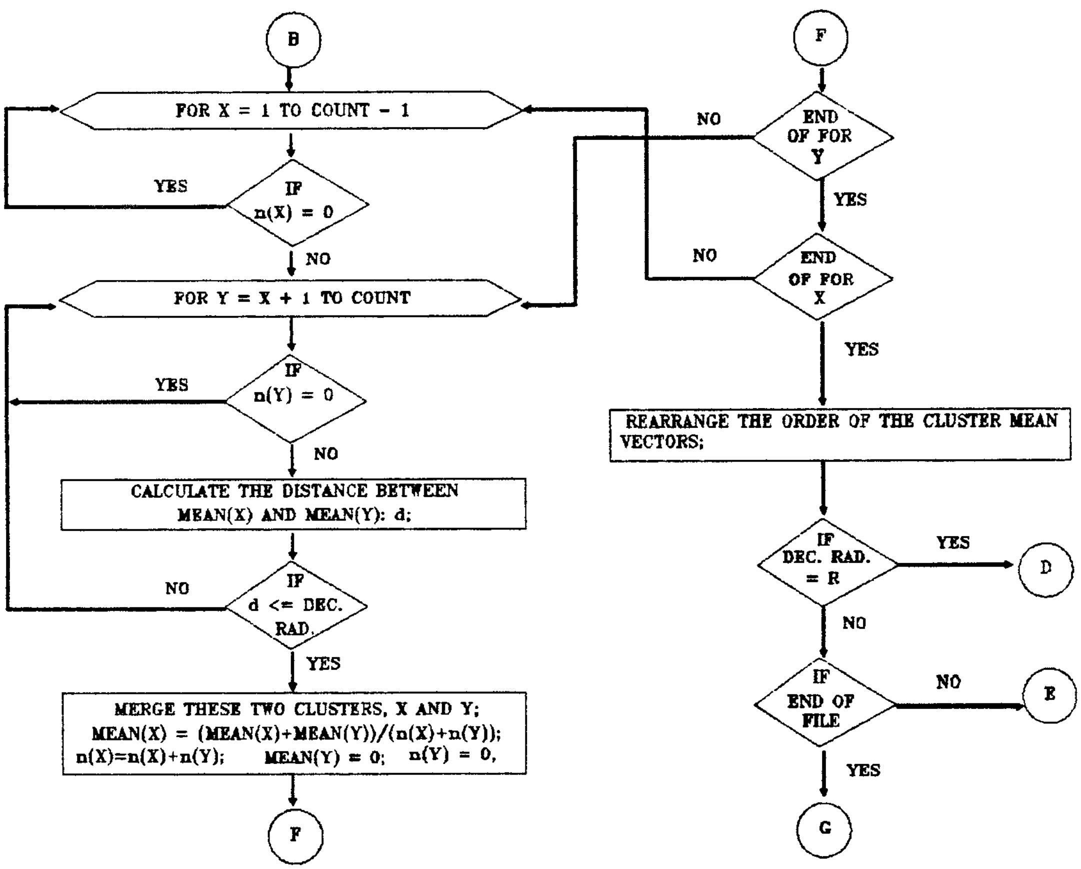
FLOW CHART OF THE UNSUPERVISED CLUSTERING ALGORITHM (PASS 1)

FLOW CHART
THE UNSUPERVISED CLUSTERING ALGORITHM



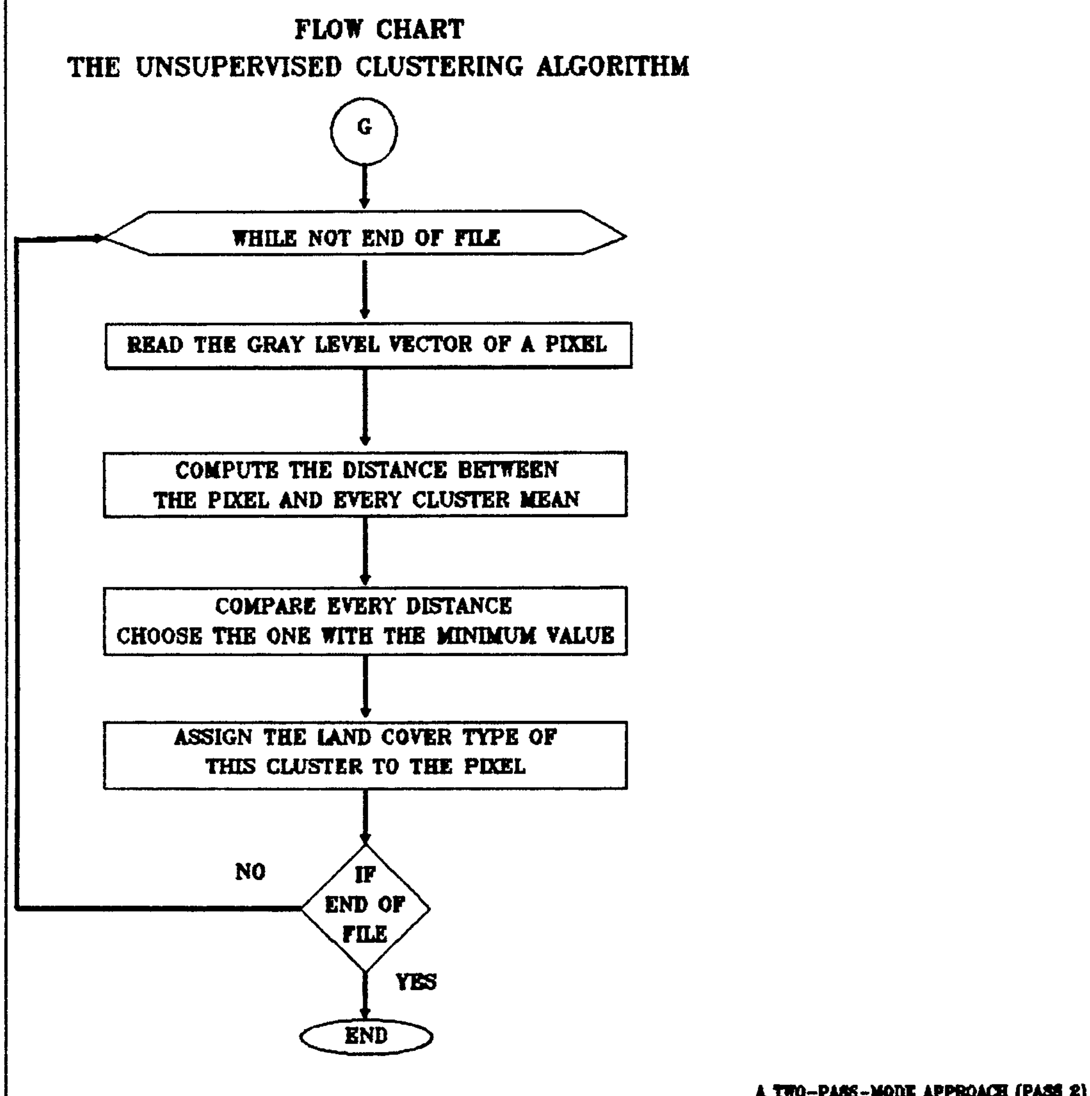
A TWO-PASS-MODE APPROACH (PASS 1)


C
A
B
IF
END OF
FILE
YES
LET DECISION
RADIUS = D;
COUNT = i;
NO
CALCULATE THE DISTANCE BETWEEN
MEAN(i) AND P(j) : r;
RECALCULATE THE
CLUSTER MEAN
VECTOR MEAN(i) =
(MEAN(i) + 1) /
(n(i) + 1);
NO
IF
r <= R
TO MERGE
YES
INCREASE THE CLUSTER COUNTER
i++;
IF
i > C max
YES
LET DECISION
RADIUS = R;
COUNT = C max ;
NO
INITIALIZE THE NEXT CLUSTER MEAN
VECTOR: MEAN(i) = P(j);
D


B
FOR X = 1 TO COUNT - 1
YES
IF n(X) = 0
NO
FOR Y = X + 1 TO COUNT
YES
IF n(Y) = 0
NO
CALCULATE THE DISTANCE BETWEEN MEAN(X) AND MEAN(Y): d;
NO
IF d <= DEC. RAD.
YES
MERGE THESE TWO CLUSTERS, X AND Y;
MEAN(X) = (MEAN(X)+MEAN(Y))/(n(X)+n(Y));
n(X)=n(X)+n(Y); MEAN(Y) = 0; n(Y) = 0,
F
F
NO
END OF FOR Y
YES
NO
END OF FOR X
YES
REARRANGE THE ORDER OF THE CLUSTER MEAN VECTORS;
IF DEC. RAD. = R
YES
D
NO
IF END OF FILE
NO
E
YES
G

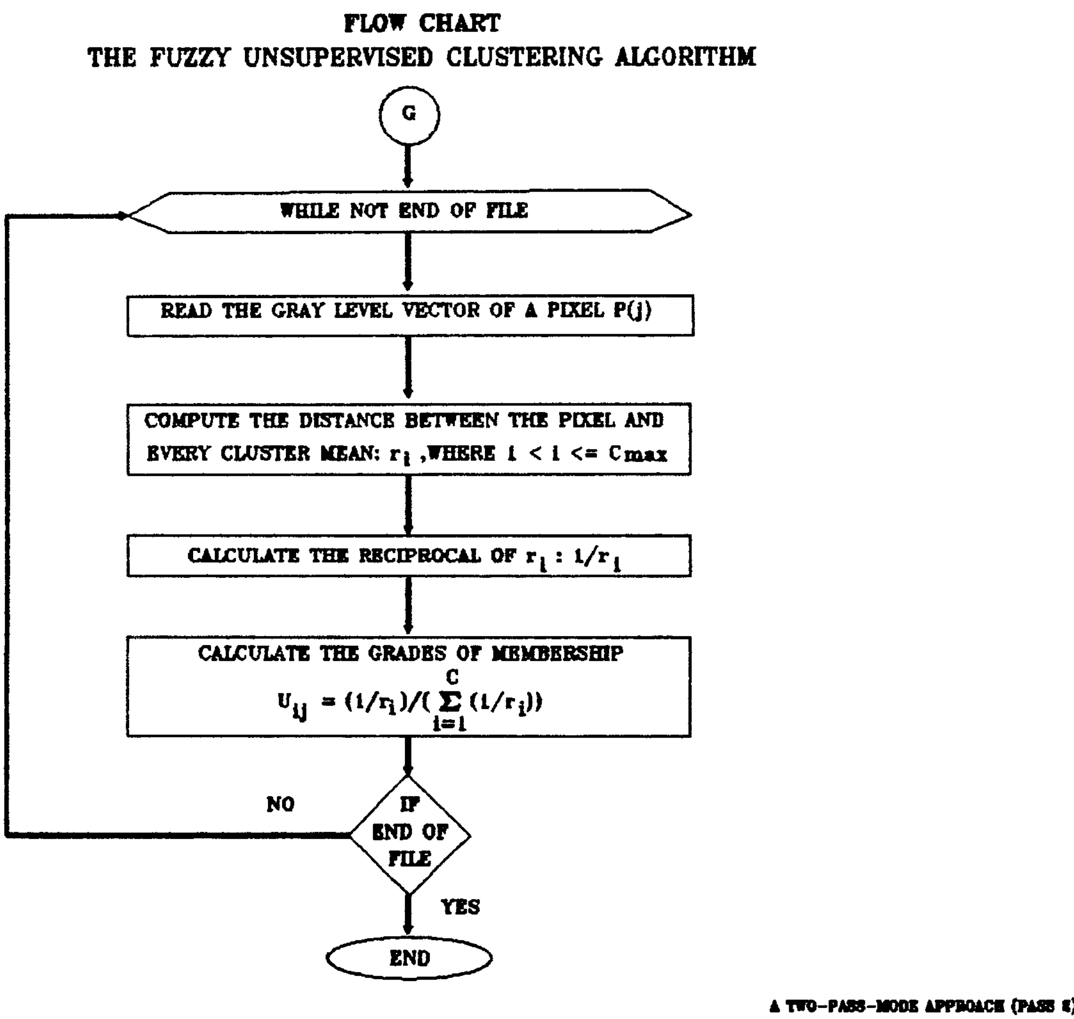
APPENDIX II

FLOW CHART OF THE UNSUPERVISED CLUSTERING ALGORITHM (PASS 2)



APPENDIX III

FLOW CHART OF THE FUZZY UNSUPERVISED CLUSTERING ALGORITHM



APPENDIX IV

PROGRAM LISTINGS

```
/* CLUSTMN.C -- This program is used to implement the pass  *
 * 1 of the two-pass-mode unsupervised clustering algorithm *
 * Certain number of cluster mean vectors are generated in  *
 * this program.                                             *
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <graphics.h>
#include <math.h>
#include <conio.h>


#define SQUARE(a)        ((a) * (a))     /* define SQUARE */
#define CLUSTER 400      /* define the maximum cluster */


/*Function Prototype*/
void clustmt(void);
int clustmg(int n, int cmerge);
float cluster_mean(float xx, unsigned nx,
                   float yy, unsigned ny);
float cluster_dis(float xx, float yy, float zz,
                  unsigned char nx, unsigned char ny,
                  unsigned char nz);
float cluster_dis1(float xx, float yy, float zz,
                  float nx, float ny, float nz);


/* global variable declare */
float MEANx[CLUSTER], MEANy[CLUSTER], MEANz[CLUSTER];
unsigned count[CLUSTER];


/* Main Program */
main()
{
  clustmt();
  return(0);
}


/* Unsupervise Classification -- cluster analysis pass 1 */
void clustmt(void)
{
  FILE *bandx;
```

```
FILE *bandy;
FILE *bandz;
FILE *cmean;
char outputf[80];
char filex[15], filey[15], filez[15];
unsigned char bgx[8], bgy[8], bgz[8];
unsigned char bufx[320], bufy[320], bufz[320];
int i, j, k, l;
int ccont, pcont,x,y,z;
int Rad, Cmer;
float sqx, sqy, sqz, distance;

/* data about the output file & the decision radii R & D*/
printf("The Filename of Output File:\n");
gets(outputf);
printf("\nThe Decision Radii:\n");
printf("R = ");
scanf("%d", &Rad);
printf("\nD = ");
scanf("%d", &Cmer);
printf("\n");

/* mean & counter initialize */
for (i = 0; i <= CLUSTER - 1; i++)
{
  MEANx[i] = MEANy[i] = MEANz[i] = 0;
  count[i] = 0;
}
ccont = 0;            /* cluster counter */

/* read data & classify */
for(k= 1; k <= 4; k++)
{
  sprintf(filex,"a:locrav02.r0%d",k);
  sprintf(filey,"a:locrav03.r0%d",k);
  sprintf(filez,"a:locrav04.r0%d",k);
  printf("%s\n",filex);
  bandx = fopen(filex,"rb");
  bandy = fopen(filey,"rb");
  bandz = fopen(filez,"rb");

  /* Read the image file header. */
  fread((void *)bgx, sizeof(char), 8, bandx);
  fread((void *)bgy, sizeof(char), 8, bandy);
  fread((void *)bgz, sizeof(char), 8, bandz);

  /* Read the pixel */
  for (j = 0; j <= 49; j++)
  {
    fread((void *)bufx, sizeof(char), 320, bandx);
    fread((void *)bufy, sizeof(char), 320, bandy);
    fread((void *)bufz, sizeof(char), 320, bandz);
```

```
    /* put the pixel */
    for (i = 0; i <= 319; i++)
    {
      if((k == 1) && (j == 0) && (i == 0))
      {
        MEANx[0] = bufx[0];  count[0]++;
        MEANy[0] = bufy[0];
        MEANz[0] = bufz[0];
      }
      else
      {
        distance = cluster_dis(MEANx[ccont], MEANy[ccont],
                               MEANz[ccont], bufx[i],
                               bufy[i], bufz[i]);
        if(distance <= Rad)
        {
          MEANx[ccont] = cluster_mean(MEANx[ccont],
                                      count[ccont],
                                      bufx[i], 1);
          MEANy[ccont] = cluster_mean(MEANy[ccont],
                                      count[ccont],
                                      bufy[i], 1);
          MEANz[ccont] = cluster_mean(MEANz[ccont],
                                      count[ccont],
                                      bufz[i], 1);
          count[ccont]++;
        }
        else
        {
          if(ccont >= 399)
            ccont = clustmg(ccont, Rad);
          ccont++;
          count[ccont]++;
          MEANx[ccont] = bufx[i];
          MEANy[ccont] = bufy[i];
          MEANz[ccont] = bufz[i];
          printf("k = %d  j = %d  i = %d  c = %d  %.2f  %.2f
                  %.2f\n", k, j, i, ccont, MEANx[ccont],
                  MEANy[ccont], MEANz[ccont]);
        }
      }  /* end of else k=1; i, j = 0 */
    }    /* end of line iteration */
  }      /* end of one segment of file */

  fclose(bandx);
  fclose(bandy);
  fclose(bandz);

  /* cluster merge */
  ccont = clustmg(ccont, Cmer);
}
cmean = fopen(outputf,"w+");
```

```
  fprintf(cmean,"%d %d %d\n",Rad, Cmer, ccont);

  printf("cluster   band2   band3   band4     counter\n");
  for(i = 0; i <= ccont; i++)
  {
    printf("%5d    %5.2f   %5.2f   %5.2f  %10u\n",i,MEANx[i],
           MEANy[i],MEANz[i],count[i]);
   fprintf(cmean,"%5.2f %5.2f %5.2f %10u\n", MEANx[i],
           MEANy[i], MEANz[i], count[i] );
  }
  fclose(cmean);
  getch();
}


/* CLUSTMG -- merge the mean vectors if matrix counter is   *
 * greater than given or the accumulated pixels are greater *
 * than the limitation.                                     *
 */
int clustmg(int n,int cmerge)
{
  int i, j, merge;
  int x, y, pcont, l;

  for(j = 0; j <= n; j++)
  {
    if(count[j] != 0)
    {
      merge = 1;
      while(merge == 1)
      {
        merge = 0;
        for(i = 0; i <= n; i++)
        {
          if((count[i] != 0) && ( i != j) &&
            (cluster_dis1(MEANx[j], MEANy[j], MEANz[j],
                          MEANx[i], MEANy[i], MEANz[i])
                          <= cmerge))
            {
              merge = 1;
              MEANx[j] = cluster_mean(MEANx[j], count[j],
                                      MEANx[i], count[i]);
              MEANy[j] = cluster_mean(MEANy[j], count[j],
                                      MEANy[i], count[i]);
              MEANz[j] = cluster_mean(MEANz[j], count[j],
                                      MEANz[i], count[i]);
              count[j] += count[i];
              count[i] = 0;
            }
        } /* end of for i */
      }  /*end of while */
    }
```

```
  } /*end of for j*/

   /* rearrange the cluster */

   pcont = 0;
   for(i = 0; i <= n; i++)
     if(count[i] == 0)
       pcont++;

   x = 0;
   i = 0;
   while(x <= n)
   {
     if(count[i] == 0)
     {
       l = 1;
       while((count[i + l] == 0) && ((x + l - 1) <= n))
         l++;
       for(y = i + l; y <= n; y++)
       {
         MEANx[y - l] = MEANx[y];
         MEANy[y - l] = MEANy[y];
         MEANz[y - l] = MEANz[y];
         count[y - l] = count[y];
       }
       for(y = n - l + 1; y <= n; y++)
       {
         MEANx[y] = MEANy[y] = MEANz[y] = 0;
         count[y] = 0;
       }
       x = x + l;
     }
     printf("x = %d i = %d l = %d ccont = %d\n",x,i,l,n);
     x++;
     i++;
   }
  printf("count = %d pcont=%d\n",n, pcont);

  for (i = 0; i <= n - pcont; i++)
    printf("%.2f %.2f %.2f %u\n", MEANx[i], MEANy[i],
            MEANz[i], count[i]);

  n -= pcont;
  return(n);
}


/* claculate the mean */
float cluster_mean(float xx, unsigned nx,
                   float yy, unsigned ny)
{
  float mean;
```

```
  mean = ((xx * nx) + (yy * ny)) / (nx + ny);
  return(mean);
}


/* CLUSTER_DIS -- calculate the distance between the cluster*
 * mean vector and the gray level vector of the pixel.      *
 */
float cluster_dis(float xx, float yy, float zz,
                  unsigned char nx, unsigned char ny,
                  unsigned char nz)
{
  float distance;

  distance = sqrt(SQUARE(nx - xx) + SQUARE(ny - yy) +
                  SQUARE(nz - zz));
  return(distance);
}


/* CLUSTER_DIS1 -- calculate the distance between two mean  *
 * vectors                                                  *
 */
float cluster_dis1(float xx, float yy, float zz,
                   float nx, float ny, float nz)
{
  float distance;

  distance = sqrt(SQUARE(nx - xx) + SQUARE(ny - yy) +
                  SQUARE(nz - zz));
  return(distance);
}
```

```
/* CLUSTCL.C -- This program is used to implement the pass 2*
 * of the two-pass-mode unsupervised clustering algorithm.  *
 * Each pixel will be assigned to one of the clusters by    *
 * using the minimum-distance to means method.              *
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#include <conio.h>


#define SQUARE(a)       ((a) * (a))     /* define SQUARE */
#define CLUSTER 400     /* define the maximum cluster */


/* Function Prototype */
void clustcl(void);
float cluster_dis(float xx, float yy, float zz,
                  unsigned char nx, unsigned char ny,
                  unsigned char nz);


/* global variable declare */
float MEANx[CLUSTER], MEANy[CLUSTER], MEANz[CLUSTER];
unsigned count[CLUSTER];


/* Main Program */
main()
{
  clustcl();
  return(0);
}


/* CLUSTCL -- Pass 2. Determine the pixel's geographic      *
 * attribute.                                               *
 */
void clustcl(void)
{
  FILE *bandx;
  FILE *bandy;
  FILE *bandz;
  FILE *ptscl;
```

```
FILE *meanf;
FILE *accnt;
char clf[80], cmn[80], acc[80];
char filex[15], filey[15], filez[15];
unsigned char bgx[8], bgy[8], bgz[8];
unsigned char bufx[320], bufy[320], bufz[320];
unsigned count[CLUSTER], nus;
int i, j, k, ptc;
int dcl;
float smd;
int Rad, Cmer, ccont;


printf("Output Image File Name After Clustering Analysis
        :\n");
gets(clf);
printf("\nCluster Mean Vectors File Name :\n");
gets(cmn);
printf("\nOutput File Name of Accumulated Pixels in Each
        Cluster:\n");
gets(acc);
ptscl = fopen(clf,"w+");
meanf = fopen(cmn,"r+");
fscanf(meanf,"%d %d %d",&Rad, &Cmer, &ccont);
accnt = fopen(acc,"w+");
printf("\nInsert the original Image File in A:. \n");
printf("Press Any Key when Ready.........\n");
getch();

for(i = 0; i <= ccont; i++)
{
  fscanf(meanf,"%f %f %f %u", &MEANx[i], &MEANy[i],
         &MEANz[i], &nus);
  count[i] = 0;
}

for(k= 1; k <= 4; k++)
{
  sprintf(filex,"a:locrav02.r0%d",k);
  sprintf(filey,"a:locrav03.r0%d",k);
  sprintf(filez,"a:locrav04.r0%d",k);
  bandx = fopen(filex,"rb");
  bandy = fopen(filey,"rb");
  bandz = fopen(filez,"rb");

  /* Read the image file header. */
  fread((void *)bgx, sizeof(char), 8, bandx);
  fread((void *)bgy, sizeof(char), 8, bandy);
  fread((void *)bgz, sizeof(char), 8, bandz);

  /* Read the pixel */
  for (j = 0; j <= 49; j++)
```

```
    {
      fread((void *)bufx, sizeof(char), 320, bandx);
      fread((void *)bufy, sizeof(char), 320, bandy);
      fread((void *)bufz, sizeof(char), 320, bandz);

      /* put the pixel */
      for (i = 0; i <= 319; i++)
      {
     printf("k = %2d  j = %2d  i = %3d\n", k, j, i);
        smd = cluster_dis(MEANx[0], MEANy[0], MEANz[0],
                          bufx[i],bufy[i],bufz[i]);
        dcl = 0;
        for(ptc = 1; ptc <= ccont; ptc++)
          if(smd > cluster_dis(MEANx[ptc],  MEANy[ptc],
                               MEANz[ptc], bufx[i],
                               bufy[i],bufz[i]))
            dcl = ptc;
        count[dcl]++;
        fprintf(ptscl,"%d ",dcl);
      } /* end of for i */
    } /* end of for j */
    fclose(bandx);
    fclose(bandy);
    fclose(bandz);
  } /* end of for k */
  fclose(ptscl);

  for(i = 0; i <= ccont; i++)
    fprintf(accnt, "%u\n", count[i]);

  fclose(meanf);
  fclose(accnt);
}


/* CLUSTER_DIS -- Calculate the distance between the cluster*
 * mean vector and the gray level vector of the pixel.     *
 */
float cluster_dis(float xx, float yy, float zz,
                  unsigned char nx, unsigned char ny,
                  unsigned char nz)
{
  float distance;

  distance = sqrt(SQUARE(nx - xx) + SQUARE(ny - yy) +
                  SQUARE(nz - zz));
  return(distance);
}
```

```
/* CLUSTPL.C -- This program is used to plot the hard parti-*
 * tion image map.                                          *
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <graphics.h>

#define SQUARE(a)       ((a) * (a))     /* define SQUARE */

/*Function Prototype*/
void framedr(char *cmn);
void mapping(void);


/* Main Program */
main()
{
  int gdriver = VGA;
  int gmode = VGAHI;
  int w;

  initgraph(&gdriver,&gmode,"..\\bgi");
  setviewport(0,0,639,479,1);
  clearviewport();
  mapping();
  closegraph();
  return(0);
}


/* MAPPING -- Draw the map. */
void mapping(void)
{
  FILE *ptscl;
  FILE *meanf;
  char clf[80], cmn[80];
  int Rad, Cmer, ccont;
  unsigned accm[200];
  int i, j, co, px, py;
  int start_left = 160;
  int start_top = 100;
  unsigned  k;

  printf("Image File Name After Clustering Analysis :\n");
  gets(clf);
  printf("\nCluster Mean Vectors File Name :\n");
  gets(cmn);

  setviewport(0,0,639,479,1);
  clearviewport();
```

```
  framedr(cmn);
  ptscl = fopen(clf,"r+");
  meanf = fopen(cmn,"r+");
  fscanf(meanf,"%d %d %d",&Rad, &Cmer, &ccont);

  for(i = 0; i <= 199; i++)
    accm[i] = 0;

  for(j = 0; j <= 199; j++)
  {
    for(i = 0; i <= 319; i++)
    {
      fscanf(ptscl,"%d", &k);
      accm[k]++;
      co = (k % 15) + 1;
      px = start_left + i;
      py = start_top + j;
      putpixel(px, py, co);
    }
  }
  fclose(ptscl);

  fclose(meanf);
  getch();
}

/* FRAMEDR -- Draw the legend. */
void framedr(char *cmn)
{
  int k, co, x1, x2, y1, y2, p;
  int Rad, Cmer, ccont;
  FILE *meanf;
  char tt[3], label[40];
  int t_width,t_height;

  meanf = fopen(cmn,"r+");
  fscanf(meanf,"%d %d %d",&Rad, &Cmer, &ccont);
  settextstyle(TRIPLEX_FONT,HORIZ_DIR,0);
  settextjustify(LEFT_TEXT,TOP_TEXT);
  setusercharsize(1,2,1,2);
  outtextxy(0, 0, cmn);
  t_height = textheight("B");
  sprintf(label, "Cluster Radius = %3d", Cmer);
  outtextxy(0, t_height,label);

  settextstyle(TRIPLEX_FONT,HORIZ_DIR,0);
  settextjustify(LEFT_TEXT,TOP_TEXT);
  setusercharsize(1,2,1,2);
  setcolor(15);
  outtextxy(0,320,"COLOR LEGEND");
  setusercharsize(1,3,1,3);
  y1 = 350;
```

```
  y2 = 399;

  for(k = 0; k <= ccont; k++)
  {
    p = k + 1;
    sprintf(tt,"%3d",p);
    co = (k % 15) + 1;
    setcolor(co);
    x1 = 50 * k + 1;
    x2 = 50 * (k + 1);
    setfillstyle(SLASH_FILL, co);
    bar3d(x1, y1, x2, y2, 0, 0);
    outtextxy(x1,y2,tt);
  }

  fclose(meanf);
}
```

```
/* MEANDS.C -- This program is designed to plot the distri- *
 * bution of the cluster mean vectors which is generated by *
 * CLUSTMN.C.                                                *
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <graphics.h>
#include <math.h>
#include <conio.h>


#define CLUSTER 50


/*Function Prototype*/
void reading(char *fnm);
void clustms(void);


/* global variable declare */
float MEANx[CLUSTER], MEANy[CLUSTER], MEANz[CLUSTER];


/* Main Program */
main()
{
  int gdriver = VGA;
  int gmode = VGAHI;
  int w;
  char fnm[80];


  printf("Cluster Mean Vectors File Name :\n");
  gets(fnm);
  initgraph(&gdriver,&gmode,"..\\bgi");
  clustms();
  reading(fnm);
  getch();
  closegraph();
  return(0);
}


/* CLUSTMS -- Draw the frame. */
void clustms(void)
{
  int oldcolor;
  char outbuf[3];
  int numpt;
  int i;
```

```
  setviewport(0,0,639,479,1);
  clearviewport();
  oldcolor = getcolor();
  setlinestyle(SOLID_LINE,0,NORM_WIDTH);
  setcolor(15);
  rectangle(30,145,285,400);
  rectangle(349,145,604,400);
  setlinestyle(DASHED_LINE,0,NORM_WIDTH);

  /* draw the grid */
  for(i = 1;i <= 12; i++)
  {
    line(30,400 - i * 20,285, 400 - i * 20);
    line(349,400 - i * 20,604, 400 - i * 20);
    line(30 + i * 20,145,30 + i * 20, 400);
    line(349 + i * 20,145,349 + i * 20, 400);
  }

  /* draw the text*/
  setusercharsize(1,8,1,8);

        /* horizontal */
  settextjustify(CENTER_TEXT,TOP_TEXT);
  for(i = 0; i <= 6; i++)
  {
    numpt = i * 20;
    sprintf(outbuf,"%3d",numpt);
    outtextxy(30 + i * 40, 403, outbuf);
    outtextxy(349 + i * 40, 403, outbuf);
  }

        /* vertical */
  settextjustify(RIGHT_TEXT,CENTER_TEXT);
  for(i = 0; i <= 6; i++)
  {
    numpt = i * 20;
    sprintf(outbuf,"%3d",numpt);
    outtextxy(25,400 - i * 40, outbuf);
    outtextxy(344,400 - i * 40, outbuf);
  }
  setcolor(oldcolor);
}


/* READING -- Read the mean values & plot it on to the      *
 * screen.                                                  *
 */
void reading(char *fnm)
{
  int x, y, z;
  int Rad, Cmer, ccont;
  FILE *meanf;
```

```
  int i,co, th;
  char tt[3], label[40];
  unsigned cc;

  setcolor(15);
  meanf = fopen(fnm,"r+");
  fscanf(meanf,"%d %d %d",&Rad, &Cmer, &ccont);
  settextstyle(TRIPLEX_FONT, HORIZ_DIR, 0);
  setusercharsize(1,2,1,2);
  settextjustify(LEFT_TEXT,TOP_TEXT);
  outtextxy(0,0, fnm);
  sprintf(label, "Radius = %3d Merge = %3d Cluster = %3d",
          Rad, Cmer,ccont + 1);
  th = textheight("B");
  outtextxy(0, th, label);

  for(i = 0; i <= ccont; i++)
  {
    co = (i % 15) + 1;
    setcolor(co);
    sprintf(tt,"%3d",i);
    fscanf(meanf,"%f %f %f %u", &MEANx[i], &MEANy[i],
           &MEANz[i], &cc);
    x = 30 + MEANx[i] * 2;
    y = 400 - MEANy[i] * 2;
    putpixel(x, y, co);
    circle(x, y, 50);
    outtextxy(x + 1, y + 1, tt);

    x = 349 + MEANy[i] * 2;
    y = 400 - MEANz[i] * 2;
    putpixel(x, y, co);
    circle(x, y, 50);
    outtextxy(x + 2, y + 2, tt);
  }
}
```

```
/* FUZZY.C -- This program reads the image data from a: then*
 * produces the fuzzy membership mapping into e:. The fuzzy *
 * membership function used here is (1 - (ai/∑ai))/(n-1). ai*
 * is the distance between a to the ith mean and n here is  *
 * the number of the cluster mean vectors.                   *
 */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>


#define MAXMM   10    /* define the maximum cluster mean  */
                      /* vectors                          */
#define SQUARE(x)    ((x) * (x))   /* define the function */
                                   /* SQUARE              */


/* Function Prototype */
void fuzzy(void);
float cluster_dis(float xx, float yy, float zz,
                  unsigned char nx, unsigned char ny,
                  unsigned char nz);
unsigned cal_member(float dis, float sum);


/* Main Program */
main()
{
  fuzzy();
  return(0);
}


/* FUZZY -- Implement the fuzzy unsupervised clustering    *
 * algorithm.                                              *
 */
void fuzzy(void)
{
  FILE *bandx, *bandy, *bandz;
  FILE *meanf;
  FILE *fuz[10];
  char fm[80];
  char ffnam[10][20], filex[15], filey[15], filez[15];
  float MEANx[MAXMM], MEANy[MAXMM], MEANz[MAXMM];
  unsigned char bgx[8], bgy[8], bgz[8];
  unsigned char bufx[320], bufy[320], bufz[320];
  int radius, cmerge, count;    /* variable for radius,  */
                                /* merge & number of     */
                                /* clusters              */
  unsigned nones;
  float distance[MAXMM];      /* distance to mean vectors */
```

```
float inv_dis[MAXMM];
unsigned members[MAXMM];       /* membership for each   */
                               /* cluster               */
float dissum;
int i, j, k, ptc;

printf("Input the mean vector file name: \n");
gets(fm);
meanf = fopen(fm,"r+");
fscanf(meanf,"%d %d %d", &radius, &cmerge, &count);

for(i = 0; i <= count; i ++)
  fscanf(meanf, "%f %f %f %u", &MEANx[i], &MEANy[i],
         &MEANz[i], &nones);

for(i = 0; i <= count; i++)
{
  sprintf(ffnam[i],"e:\\test\\test%d.dat",i);
  fuz[i] = fopen(ffnam[i],"w+");
  fprintf(fuz[i],"%d %.2f %.2f %.2f\n", i, MEANx[i],
          MEANy[i], MEANz[i]);
}

fclose(meanf);

/* calculate the membership function */
for(k= 1; k <= 4; k++)
{
  sprintf(filex,"a:locrav02.r0%d",k);
  sprintf(filey,"a:locrav03.r0%d",k);
  sprintf(filez,"a:locrav04.r0%d",k);
  printf("%s",filex);
  bandx = fopen(filex,"rb");
  bandy = fopen(filey,"rb");
  bandz = fopen(filez,"rb");

  /* Read the image file header. */
  fread((void *)bgx, sizeof(char), 8, bandx);
  fread((void *)bgy, sizeof(char), 8, bandy);
  fread((void *)bgz, sizeof(char), 8, bandz);

  /* Read the pixel */
  for (j = 0; j <= 49; j++)
  {
    printf("k = %d j = %d\n",k,j);
    fread((void *)bufx, sizeof(char), 320, bandx);
    fread((void *)bufy, sizeof(char), 320, bandy);
    fread((void *)bufz, sizeof(char), 320, bandz);

    /* put the pixel */
    for (i = 0; i <= 319; i++)
    {
```

```
        dissum = 0;
        for(ptc = 0; ptc <= count; ptc++)
        {
          distance[ptc] = 0;
          distance[ptc] = cluster_dis(MEANx[ptc], MEANy[ptc],
                                     MEANz[ptc], bufx[i],
                                     bufy[i], bufz[i]);

          /* If the distance is equal to zero, let it be a *
           * small number.                                 *
           */
          if(distance[ptc] == 0) distance[ptc] = 0.00001;
          inv_dis[ptc] = 1 / distance[ptc];
          dissum += inv_dis[ptc];
        }
        for(ptc = 0; ptc <= count; ptc++)
        {
          members[ptc] = cal_member(inv_dis[ptc], dissum);
          fprintf(fuz[ptc],"%u ",members[ptc]);
        }
      }
    }
    fclose(bandx);
    fclose(bandy);
    fclose(bandz);
  }


  for(i = 0; i <= count; i++)
    fclose(fuz[i]);
}

/* CLUSTER_DIS -- Calculate the distance between the cluster*
 * mean vector and the gray level vector of the pixel.     *
 */
float cluster_dis(float xx, float yy, float zz,
                  unsigned char nx, unsigned char ny,
                  unsigned char nz)
{
  float distance;

  distance = sqrt(SQUARE(nx - xx) + SQUARE(ny - yy) +
                  SQUARE(nz - zz));
  return(distance);
}


/* CAL_MEMBER -- calculate the membership of individual     *
 * class                                                    *
 */
unsigned cal_member(float dis, float sum)
{
```

```
  unsigned member;
  float rmem;
  double intpart;

  rmem = dis * 10000 / sum;
  if(modf(rmem,&intpart) >= 0.5)
    member = rmem + 1;
  else
    member = rmem ;
  return(member);
}
```

```
/* FUZCL.C -- This program is to classify the fuzzy member- *
 * ship into percentage type. The membership file is        *
 * constructed in Fuzzy.c. This program is to find out the  *
 * distribution of the grade of membership.                 *
 */
#include <stdio.h>
#include <stdlib.h>


/* function prototype */
void fluzcl(void);


/Main Program */
main()
{
  fluzcl();
  return(0);
}


void fluzcl(void)
{
  FILE *discl;
  FILE *disot;
  char ipfl[80], oufl[80];
  int i, j, cl, k;
  unsigned fzcl[10], hh;
  float mx, my, mz;

  for(i = 0; i <= 9; i++)
    fzcl[i] = 0;

  printf("Input the Fuzzy Membership file:\n");
  gets(ipfl);
  printf("\noutput filename:\n");
  gets(oufl);

  discl = fopen(ipfl,"r+");
  fscanf(discl,"%d %f %f %f\n", &k, &mx, &my, &mz);

  printf("%d %f %f %f\n",k, mx, my, mz);

  for(i = 0; i <= 199; i++)
    for (j = 0; j <= 319; j++)
    {
      printf("%d %d\n", i, j);
      fscanf(discl," %u", &hh);

      cl = hh / 1000;
      printf("%u %d\n",hh, cl);
      fzcl[cl]++;
```

```
    }

  disot = fopen(oufl,"w+");
  fprintf(disot, "%d %f %f %f\n",k, mx, my, mz);

  for(i = 0; i <= 9; i++)
    fprintf(disot,"%d %u\n", i, fzcl[i]);

  fclose(discl);
  fclose(disot);
}
```

```
/* FUZPL.C -- This program is used to plot the percentage   *
 * distribution map or the fuzzy map.                         *
 */
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>


/* function prototype */
void fluzpl(void);
void framedr(void);


/* Main Program */
main()
{
  int gdriver = VGA;
  int gmode = VGAHI;
  int w;

  initgraph(&gdriver,&gmode,"..\\bgi");
  setviewport(0,0,639,479,1);
  clearviewport();
  fluzpl();
  getch();
  closegraph();
  return(0);
}


/* FLUZPL -- Plot the fuzzy map. */
void fluzpl(void)
{
  FILE *discl;
  int start_left = 160;
  int start_top = 100;
  char ipfl[80], oufl[80];
  char label[40];
  int i, j, cl, k, px, py;
  unsigned fzcl[10], hh;
  float mx, my, mz;

  for(i = 0; i <= 9; i++)
    fzcl[i] = 0;

  printf("Input the Fuzzy Membership file:\n");
  gets(ipfl);
  setviewport(0,0,639,479,1);
  clearviewport();

  framedr();
  discl = fopen(ipfl,"r+");
```

```
  fscanf(discl,"%d %f %f %f\n", &k, &mx, &my, &mz);

  settextstyle(TRIPLEX_FONT,HORIZ_DIR,0);
  settextjustify(LEFT_TEXT,TOP_TEXT);
  setusercharsize(1,2,1,2);
  sprintf(label,"CLUSTER %d  MEAN VECTOR : %.2f %.2f %.2f\n",
          k, mx, my, mz);
  outtextxy(0,5,label);

  for(i = 0; i <= 199; i++)
    for (j = 0; j <= 319; j++)
    {
      fscanf(discl," %u", &hh);

      cl = hh / 1000 + 6;
      px = start_left + j;
      py = start_top + i;
      putpixel(px, py, cl);
      fzcl[cl]++;
    }

  fclose(discl);
}


/* FRAMEDR -- Draw the legend */
void framedr(void)
{
  int k, co, k1, k2;
  int Rad, Cmer, ccont;
  FILE *meanf;
  char tt[8];
  int t_width,t_height;
  int x1, x2, y1, y2;

  settextstyle(TRIPLEX_FONT,HORIZ_DIR,0);
  settextjustify(LEFT_TEXT,TOP_TEXT);
  setusercharsize(1,2,1,2);
  setcolor(15);
  outtextxy(0,320,"COLOR LEGEND");
  setusercharsize(1,3,1,3);
  t_width = textwidth("000");
  t_height = textheight("000");
  y = 400;
  y1 = 350;
  y2 = 399;

  for(k = 0; k <= 9; k++)
  {
    k1 = k * 10;
    k2 = k1 + 9;
    sprintf(tt,"%3d - %3d",k1, k2);
```

```
    co = k + 6;
    setcolor(co);
    x1 = 50 * k + 1;
    x2 = 50 * (k + 1);
    setfillstyle(SLASH_FILL, co);
    bar3d(x1, y1, x2, y2, 0, 0);
    outtextxy(x1,y2,tt);
  }
}
```

REFERENCES

1. Rosenfeld, C. L. "Watershed Management: Fighting the Effects of Drought in West Africa." *Geo Info Systems* 2.3 (1992): 28-39.

2. Levy, K. "GIS Headlines Strategies for Improving Water Management in Newport News." *Geo Info Systems* 1.8 (1991): 46.

3. Randy S. K. "Satellite Imagery and GIS Help Protect Wildlife Habitat in Florida." *Geo Info Systems* 2.1 (1992): 37-42.

4. Emilio J. S. "Los Angeles County Begins Long-term Conversion of Assessor's Property Records." *Geo Info Systems* 2.1 (1992): 43-46.

5. Jenson, J. R. *Introductory Digital Image Processing -- A Remote Sensing Perspective.* New Jersey: Prentice-Hall, 1986.

6. Thomas, I. L., Benning, V. M., and Ching, N. P. *Classification of Remotely Sensed Images.* Bristol, England: IOP Publishing, 1987.

7. Lillesand, T. M., and Kiefer, R. W. *Remote Sensing & Image Interpretation.* 2nd ed. New York: John Wiley & Sons, 1987.

8. Wang, F. "Improving Remote Sensing Image Analysis Through Fuzzy Information Representation." *Photogrammetric Engineering & Remote sensing* 56.8 (1990): 1163-1169.

9. Zadeh, L. A. "Fuzzy Sets." *Information and Control* 8 (1965): 338-353.

10. Zadeh, L. A. "Probability Measures of Fuzzy Events." *Journal of Mathematical Analysis and Applications* 23 (1968): 421-427.

11. Wang, F. "Fuzzy Supervised Classification of Remote Sensing Images." *IEEE Transactions on Geoscience and Remote Sensing* GE-28.2 (1990): 194-201.

12. Wang, F. "Integrating GIS's and Remote Sensing Image Analysis Systems by Unifying Knowledge Representation Schemes." *IEEE Transactions on Geoscience and Remote Sensing* GE-29.4 (1991): 656-664.

13. Cannon, R. L., Dave, J. V., and Bezdek, J. C. "Efficient Implementation of the Fuzzy *c*-Means Clustering Algorithms." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.2 (1986): 248-255.

14. Cannon, R. L., Dave, J. V., Bezdek, J. C., and Trivedi, M. M. "Segmentation of a Thematic Mapper Image Using the Fuzzy *c*-Means Clustering Algorithm." *IEEE Transactions on Geoscience and Remote Sensing* GE-24.3 (1986): 400-408.

15. Foody, G. M. " A fuzzy Sets Approach to the Representation of Vegetation Continua from Remotely Sensed Data: An Example from Lowland Heath." *Photogrammetric Engineering & Remote sensing* 58.2 (1992): 221-225.

16. Zenzo, S. D., Bernstein, R., Degloria, S. D., and Kolsky, H. G. "Gaussian Maximum Likelihood and Contextual Classification Algorithm for Multicrop Classification." *IEEE Transactions on Geoscience and Remote Sensing* GE-25.9 (1987): 805-814.

17. Kent, J. T., and Mardia, K. V. "Spatial Classification using fuzzy membership models." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-10.5 (1986): 659-671.

18. Asrar, G. *Theory and Applications of Optical Remote Sensing.* New York: John Wiley & Sons, 1989.

19. Sabins, F. F. Jr., *Remote Sensing -- Principles and Interpretation.* 2nd ed. New York: W. H. Freeman and Company, 1986.