Unsupervised Classification

Unsupervised classification (commonly referred to as *clustering*) is an effective method of partitioning remote sensor image data in multispectral feature space and extracting land-cover information. Compared to supervised classification, unsupervised classification normally requires only a minimal amount of initial input from the analyst. This is because clustering does not normally require training data.

Unsupervised classification is the process where numerical operations are performed that search for *natural groupings* of the spectral properties of pixels, as examined in multispectral feature space. The clustering process results in a classification map consisting of *m* spectral classes. The analyst then attempts *a posteriori* (after the fact) to assign or transform the *spectral* classes into thematic *information* classes of interest (e.g., forest, agriculture). This may be difficult. Some spectral clusters may be meaningless because they represent mixed classes of Earth surface materials. The analyst must understand the spectral characteristics of the terrain well enough to be able to label certain clusters as specific information classes.

Hundreds of clustering algorithms have been developed. Two examples of conceptually simple but not necessarily efficient clustering algorithms will be used to demonstrate the fundamental logic of unsupervised classification of remote sensor data:

- clustering using the *Chain Method*
- clustering using the *Iterative Self-Organizing Data Analysis Technique* (ISODATA).

Chain method

The *Chain Method* clustering algorithm operates in a *two-pass mode* (i.e., it passes through the multispectral dataset two times).

Pass #1: The program reads through the dataset and sequentially builds clusters (groups of points in spectral space). A mean vector is then associated with each cluster.

Pass #2: A minimum distance to means classification algorithm is applied to the whole dataset on a *pixel-by-pixel basis* whereby each pixel is assigned to one of the mean vectors created in *pass 1*. The first pass, therefore, automatically creates the cluster signatures (class mean vectors) to be used by the minimum distance to means classifier.

Pass 1: Cluster building

During the first pass, the analyst is required to supply four types of information:

1. R, is radius distance in spectral space used to determine when a new cluster should be formed (e.g., when raw remote sensor data are used, it might be a set at 15 brightness value unit.)

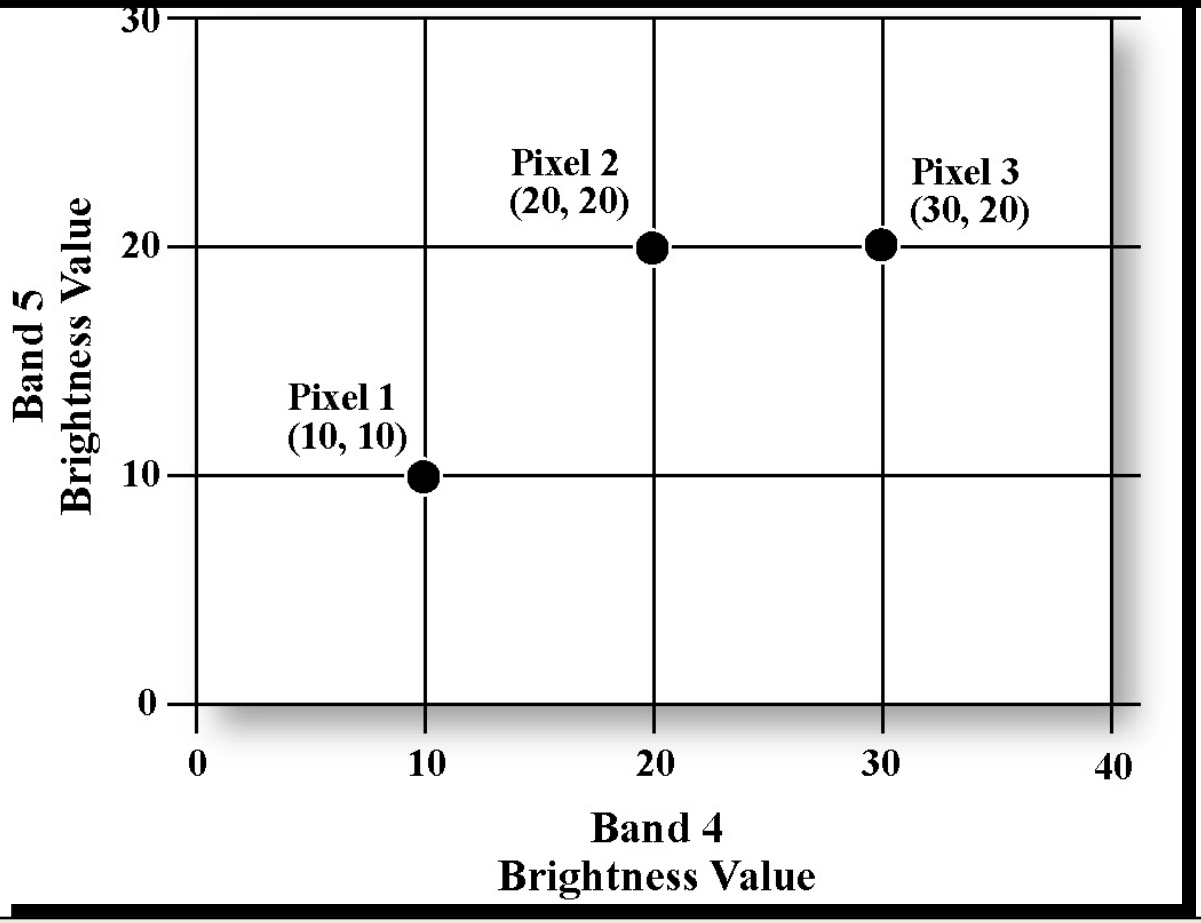
2. C, a spectral space distance parameter used when merging clusters (e.g., 30 units) when N is reached.

3. N, the number of pixels to be evaluated between each major merging of the clusters (e.g., 2000 pixels)

4. Cmax, the maximum number of clusters to be identified by the algorithm (e.g., 20 clusters)

These can be set to default values to avoid human interaction.

Start at the origin of multispectral dataset (i.e. line 1, column 1)

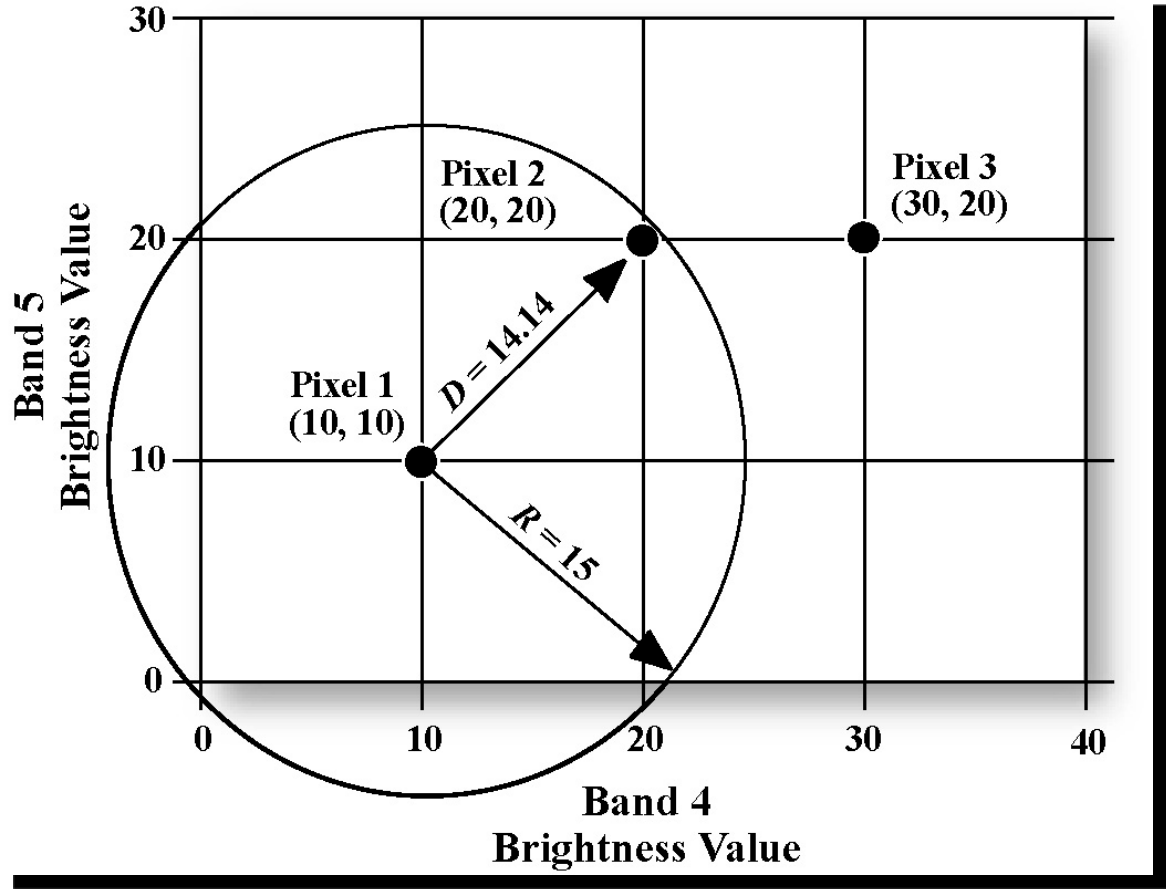


Original brightness values of pixels 1, 2, and 3 as measured in Bands 4 and 5 of the hypothetical image data. We take them as:

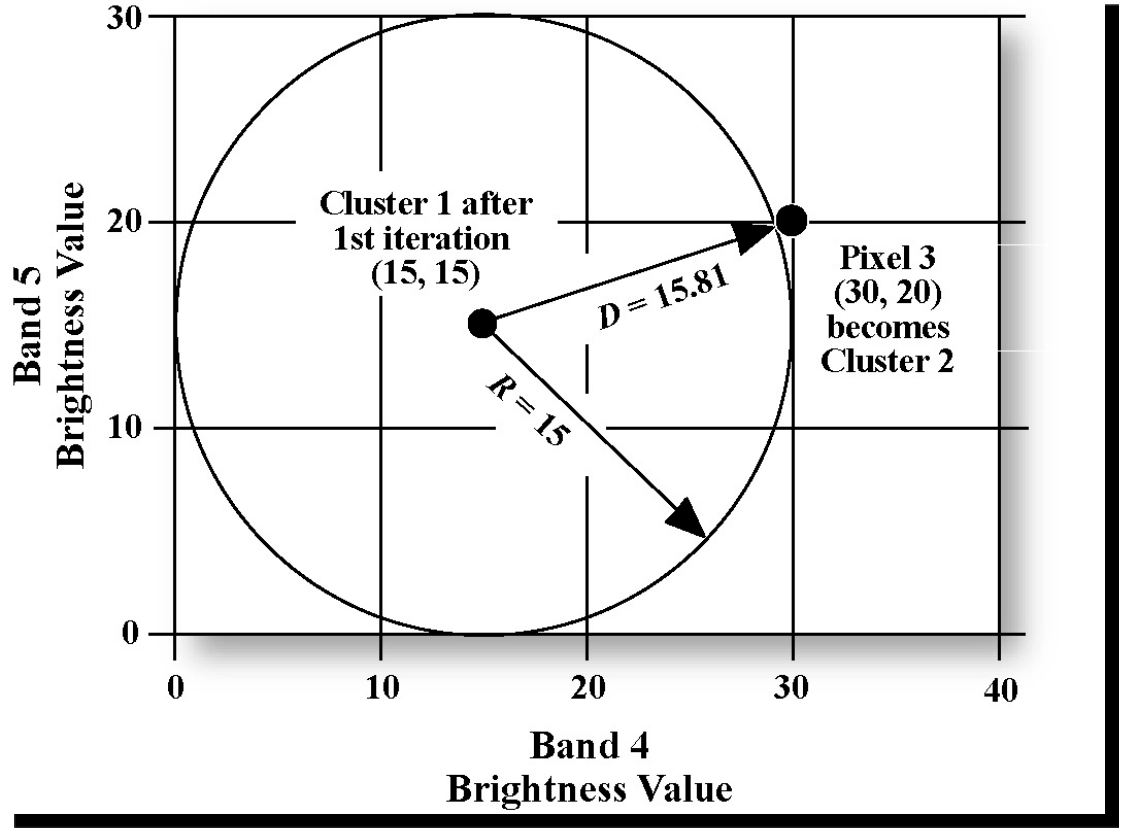
Mean data vector of cluster #1 (M1={10,10})

Mean data vector of cluster #2 (M2={20,20})

Mean data vector of cluster #3 (M3={30,20})



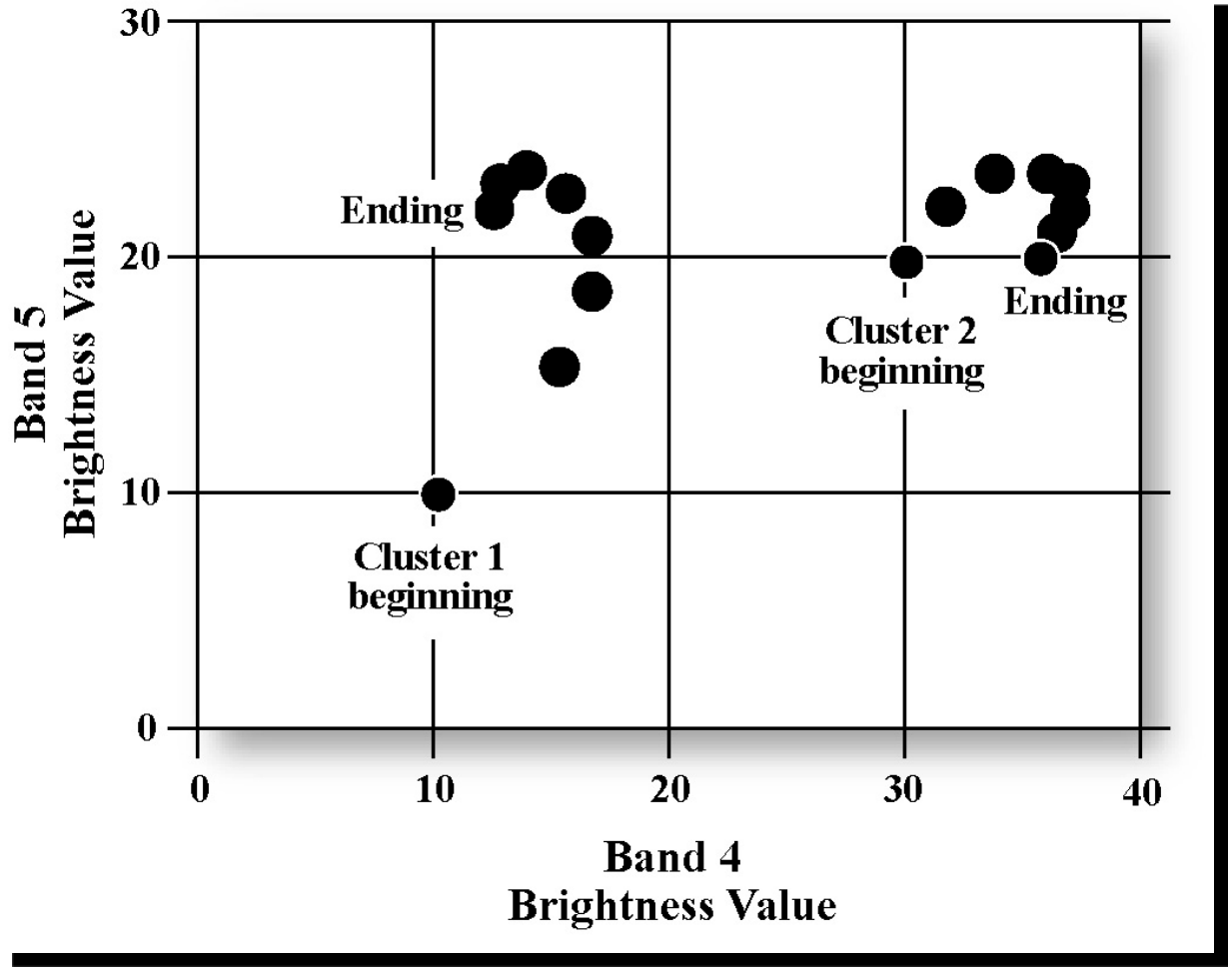
The distance (D) in 2-dimensional spectral space between pixel 1 (cluster 1) and pixel 2 (potential cluster 2) in the first iteration is computed and tested against the value of R=15, the minimum acceptable radius. In this case, D does not exceed R. Therefore, we merge clusters 1 and 2 as shown in the next illustration.



Pixels 1 and 2 now represent cluster #1. Note that the location of cluster 1 has migrated from 10,10 to 15,15 after the first iteration. Now, pixel 3 distance (D=15.81) is computed to see if it is greater than the minimum threshold, R=15. It is, so pixel location 3 becomes cluster #2.

Mean data vector of cluster #1 (M1={15,15})
Mean data vector of cluster #2 (M3={30,20})

This process continues until all 20 clusters are identified. Then the 20 clusters are evaluated using a distance measure, C (not shown), to merge the clusters that are closest to one another.



How clusters migrate during the several iterations of a clustering algorithm. The final ending point represents the mean vector that would be used in *phase 2* of the clustering process when the minimum distance classification is performed.

Pass 2: Assignment of pixels to clusters

In pass 2, Assignment of pixels to one of the Cmax clusters is made using Minimum Distance Classification logic. The final cluster mean data vectors are used in minimum distance to means classification algorithm to classify all the pixels in the image into one of the Cmax clusters.

The analyst usually produces a cospectral plot display to document where the clusters in three-dimensional feature space.

It is then necessary to evaluate the location of the clusters in the image, label them if possible, and see if any should be combined.

It is usually necessary to combine come clusters. This is where an intimate knowledge of the terrain is critical.

ISODATA Clustering

The *Iterative Self-Organizing Data Analysis Technique* (ISODATA) represents a comprehensive set of *heuristic* (rule of thumb) procedures that have been incorporated into an iterative classification algorithm. Many of the steps incorporated into the algorithm are a result of experience gained through experimentation.

The *ISODATA* algorithm is a modification of the *k*-means clustering algorithm, which includes a) merging clusters if their separation distance in multispectral feature space is below a user-specified threshold and b) rules for splitting a single cluster into two clusters.

• *ISODATA* is iterative because it makes a *large number of passes through the remote sensing dataset* until specified results are obtained, instead of just two passes.

• *ISODATA* does not allocate its initial mean vectors based on the analysis of pixels in the first line of data the way the two-pass algorithm does. Rather, an initial arbitrary assignment of all C_{max} clusters takes place along an *n*-dimensional vector that runs between very specific points in feature space. The region in feature space is defined using the mean, μ_k, and standard deviation, s_k, of each band in the analysis. This method of automatically seeding the original C_{max} vectors makes sure that *the first few lines of data do not bias the creation of clusters*.

ISODATA is self-organizing because it requires relatively little human input. A sophisticated ISODATA algorithm normally requires the analyst to specify the following criteria:

- C_{max}: the maximum number of clusters to be identified by the algorithm (e.g., 20 clusters). However, it is not uncommon for fewer to be found in the final classification map after splitting and merging take place.
- *T*: the maximum percentage of pixels whose class values are allowed to be *unchanged* between iterations. When this number is reached, the ISODATA algorithm terminates. Some datasets may never reach the desired percentage unchanged. If this happens, it is necessary to interrupt processing and edit the parameter.

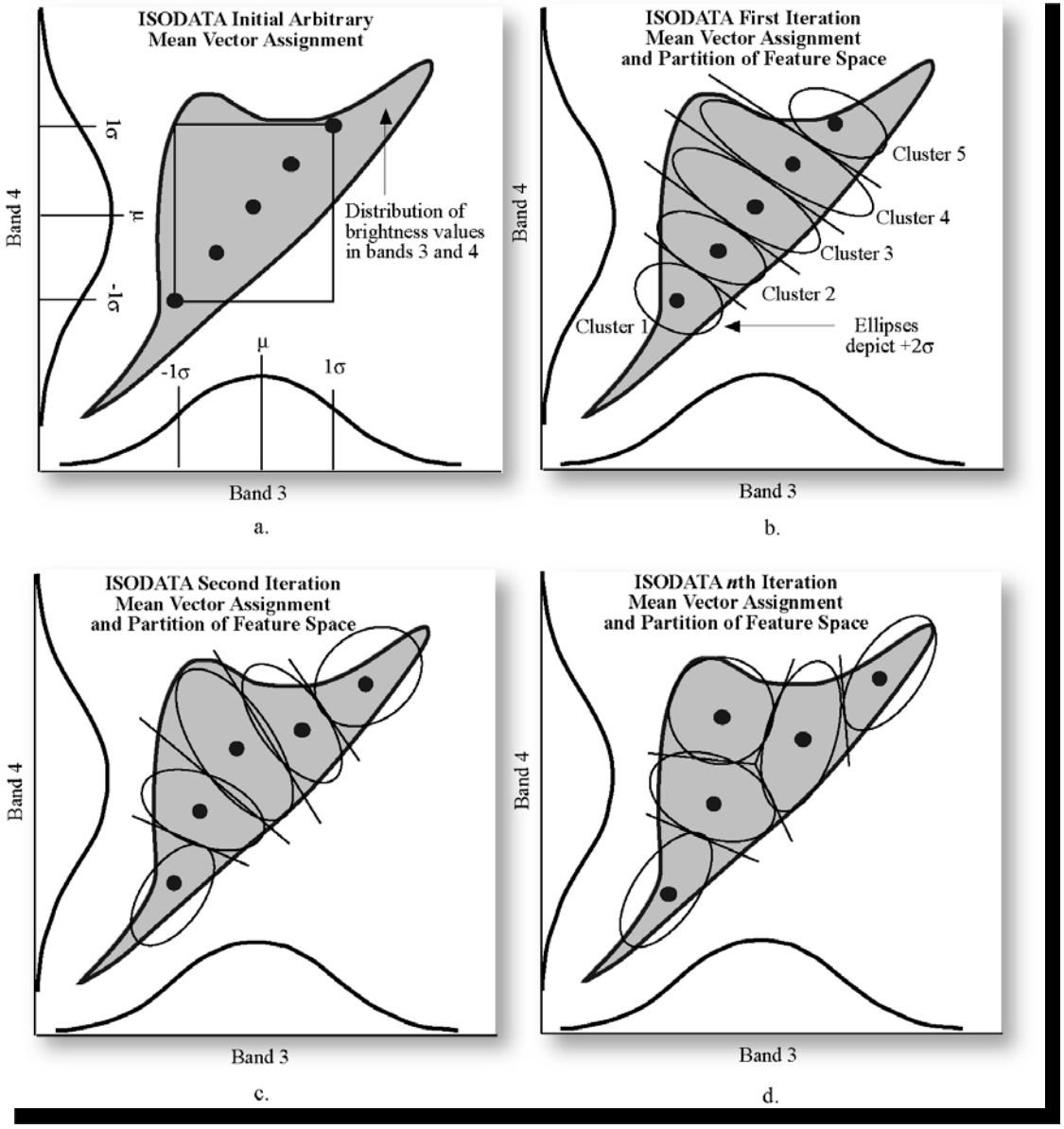
M: the maximum number of times ISODATA is to classify pixels and recalculate cluster mean vectors. The ISODATA algorithm terminates when this number is reached.

Minimum members in a cluster (%): If a cluster contains less than the minimum percentage of members, it is deleted and the members are assigned to an alternative cluster. This also affects whether a class is going to be split (see maximum standard deviation). The default minimum percentage of members is often set to 0.01.

Maximum standard deviation (σ_{max}): When the standard deviation for a cluster exceeds the specified maximum standard deviation and the number of members in the class is greater than twice the specified minimum members in a class, the cluster is split into two clusters. The mean vectors for the two new clusters are the old class centers ±1s. Maximum standard deviation values between 4.5 and 7 are typical.

Split separation value: If this value is changed from 0.0, it takes the place of the standard deviation in determining the locations of the new mean vectors plus and minus the split separation value.

Minimum distance between cluster means (C): Clusters with a weighted distance less than this value are merged. A default of 3.0 is often used.

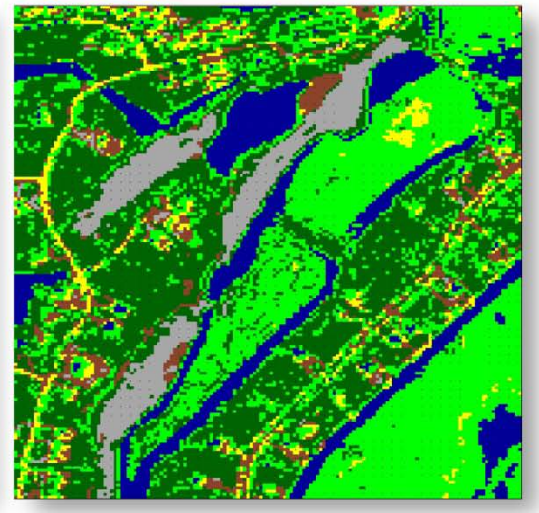


a) ISODATA initial distribution of five hypothetical mean vectors using ±1σ standard deviations in both bands as beginning and ending points. b) In the first iteration, each candidate pixel is compared to each cluster mean and assigned to the cluster whose mean is closest in Euclidean distance. c) During the second iteration, a new mean is calculated for each cluster based on the actual spectral locations of the pixels assigned to each cluster, instead of the initial arbitrary calculation. This involves analysis of several parameters to merge or split clusters. After the new cluster mean vectors are selected, every pixel in the scene is assigned to one of the new clusters. d) This split–merge–assign process continues until there is little change in class assignment between iterations (the T threshold is reached) or the maximum number of iterations is reached (M).

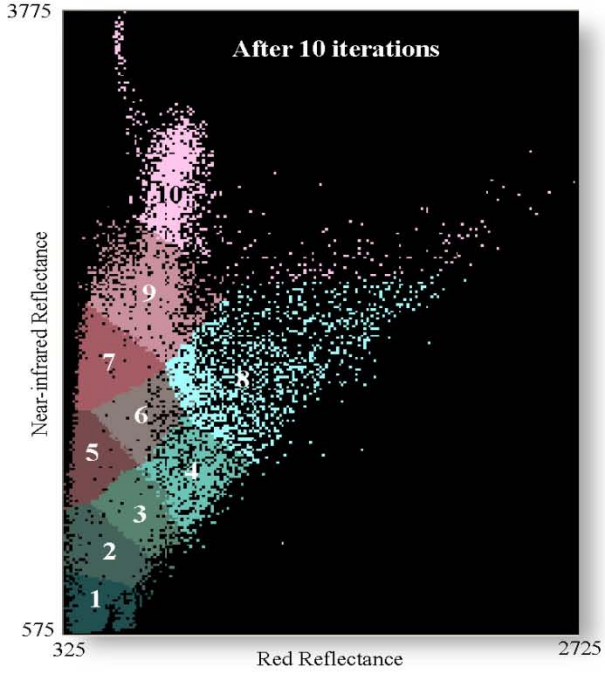
Classification Based on ISODATA Clustering

Information Classes Derived from an ISODATA Unsupervised Classification Using 10 Iterations and 10 Mean Vectors of an Area Near North Inlet, SC

a. Color composite of HyMap data.



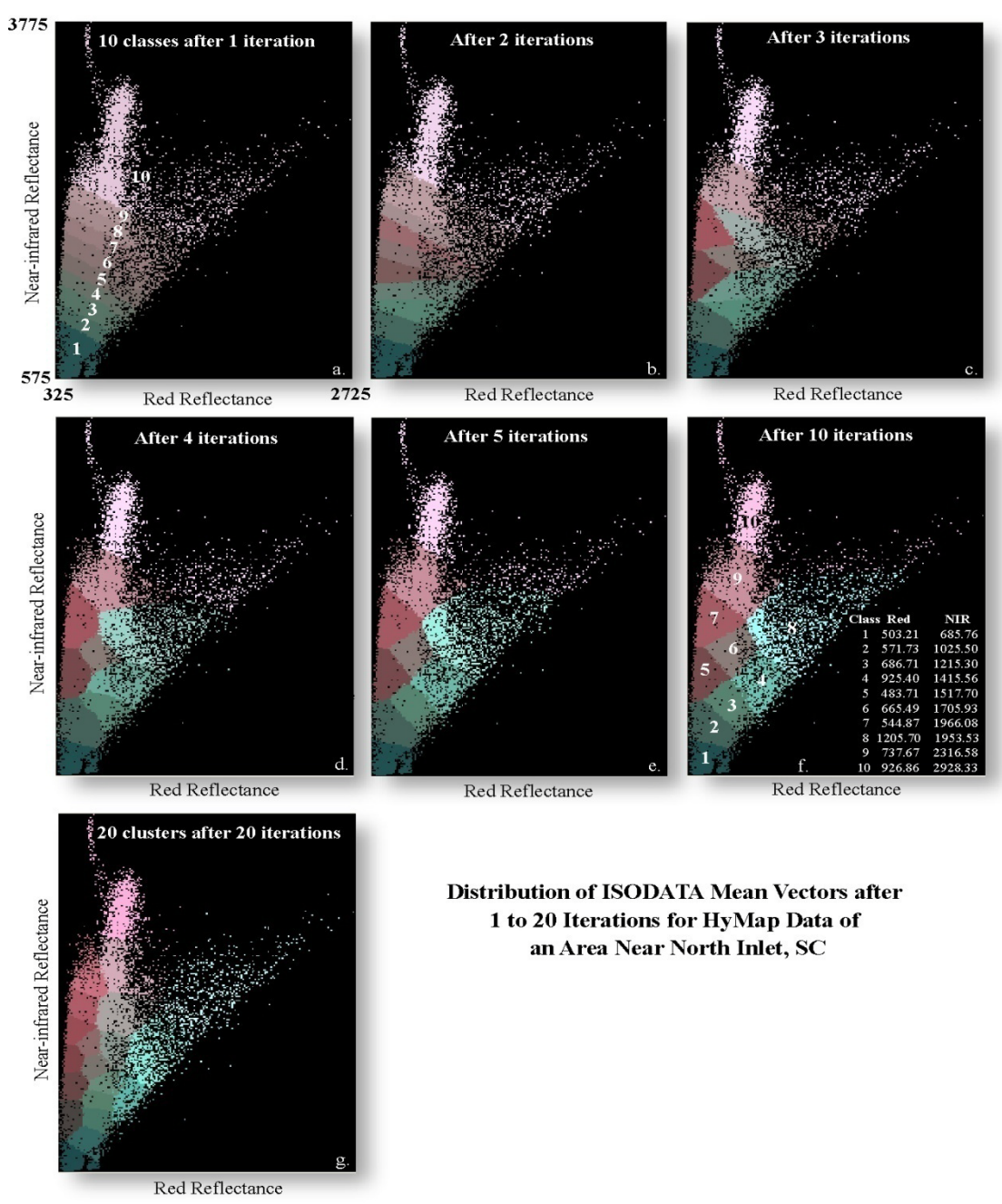
c. Classification map derived from 10 ISODATA clusters.



b. Final location of 10 ISODATA mean vectors after 10 iterations.

Class	Legend
1. Water	
2. Wetland	
3. Wetland	
4. Roof/asphalt	
5. Forest	
6. Wetland	
7. Forest	
8. Bare soil	
9. Forest	
10. Fairway	

ISODATA Clustering Logic



Distribution of ISODATA Mean Vectors after 1 to 20 Iterations for HyMap Data of an Area Near North Inlet, SC