Introduction to Modern Photogrammetry

Edward M. Mikhail and James S. Bethel
Purdue University

J. Chris McGlone
Carnegie Mellon University

John Wiley & Sons, Inc.
New York / Chichester / Weinheim / Brisbane / Singapore / Toronto

Acquisitions Editor *Wayne Anderson*
Marketing Manager *Katherine Hepburn*
Senior Production Editor *Michael Farley*
Senior Designer *Maddy Lesure*
Production Management Services *Publication Services*

This book was set in *Times Roman* by *Publication Services* and printed and bound by *Hamilton Printing*. The cover was printed by *Phoenix Color*.

The book is printed on acid-free paper.

Library of Congress Cataloging in Publication Data:
Edward M. Mikhail, James S. Bethel, and J. Chris McGlone
Introduction to Modern Photogrammetry
Mikhail.—
p. cm.
Includes bibliographic references.

L.C. Call no. Dewey Classification No. L.C. Card No.

ISBN 0-471-30924-9

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Chapter 10

Analysis of Multispectral and Hyperspectral Image Data

David Landgrebe
School of Electrical and Computer Engineering,
Purdue University, West Lafayette, IN

10.1 INTRODUCTION, BACKGROUND, AND HISTORY

The beginning of the space age is generally defined by the launching of Sputnik in October 1957. Along with many other effects, this event caused people to consider how to use space-based technology for practical purposes. An application of immediate interest was the observation of the Earth from space and, in particular, the observation of the Earth's atmospheric conditions. The first Earth observational satellite, TIROS 1, was launched on April 1, 1960. The newly formed U.S. National Aeronautics and Space Administration then began to launch additional observational satellites at the rate of about two per year.

It was not long before the use of satellites to monitor the land, in addition to the atmosphere, was considered (Landgrebe, 1997). Early in the 1960s, engineers began to develop techniques for monitoring the Earth's land resources, both renewable ones such as those of agriculture and forestry, and nonrenewable ones such as those of geologic interest. The initial questions to be addressed were what kind of land resource information would be most useful and what kind of sensor system and analysis approach should be used to obtain it. Imagery similar to photographs, analyzed using standard photo interpretation technique, was considered. However, it was quickly recognized that photographic cameras were not a good sensor choice because the data (imagery) would have to be transmitted to the Earth electronically. Photo interpretation was also problematical due to the large quantities of such imagery and the cost of manual methods. The advantage that the space vantage point provides is the ability to monitor large areas in a nearly instantaneous fashion, thus potentially leading to very large amounts of imagery.

The resources of the land are characterized by a much more detailed nature, requiring much higher spatial resolution than that needed for atmospheric monitoring. Since data volume increases as the square of spatial resolution, very large quantities of data would indeed need to be analyzed economically in order to obtain useful information on land resources from imagery. This immediately suggested some type of automated or computer-assisted analysis, rather than entirely manual methods. However, straightforward computer image analysis methods did not seem a good choice. To be able to identify a field of corn, for example, image analysis methods would require spatial resolution high enough to identify individual plants as corn by their leaf structure, etc., as we humans do it.

A more fundamental look at the problem suggested that perhaps the required identification could be carried out based on the spectral reflectance properties of the different materials. The idea was to identify materials by measuring the energy emanating from individual pixels as a function of wavelength, then using the emerging computer-implemented pattern recognition technology to label each pixel as to its contents. The advantage of this approach is that it makes very efficient use of whatever spatial resolution is used. The contents of each pixel can be identified individually, rather than having to use a collection of pixels to label an object as the smallest item in the scene to be labeled.

This approach, labeling a given pixel based on the distribution of its electromagnetic energy as a function of wavelength, became known as the *multispectral approach*. Research to test and develop this approach during the 1960s used aircraft data with 12 to 18 spectral bands, located in the visible, reflective infrared, and thermal infrared portions of the electromagnetic spectrum.

The type of sensor needed for this approach is one that measures the energy intensity in each of a number of wavelengths at once. The first satellite to carry such a sensor was Landsat 1, launched in July 1972. Landsat 1 carried an instrument known as MSS (for multispectral scanner). The MSS collected image data with pixels 80 meters across in four spectral bands per pixel with a signal-to-noise ratio sufficient to support a 6-bit data system (i.e., $2^6 = 64$ shades of gray in each band).

For some years, this low number of bands per pixel was the primary factor limiting the performance of multispectral sensors. Sampling the spectrum at only four locations was useful for rather simple problems, but did not provide the detail needed to discriminate between more subtle classes. A second-generation system, known as *Thematic Mapper*, was first launched in 1984 and served as the primary multispectal space sensor through the 1980s and 1990s. It sampled the spectrum at seven locations with pixels 30 meters across and had an 8-bit (256 shades of gray) data system; a significant improvement, but still quite limiting.

Advances in sensor technology have now made much more complete sampling of the spectrum possible. Sensor systems now sample the spectrum in hundreds of locations with signal-to-noise ratios justifying 10 to 12-bit data systems. This much more complex data moves the key problem of information extraction from sensor limitations to the analysis process. Thus, it is very important to have a thorough understanding of such complex data and adequately powerful and sophisticated algorithms for analyzing it.

10.2 STATISTICAL PATTERN RECOGNITION AND CLASSIFICATION

The analysis of a multispectral image data set is based on spectral processing techniques rather than image processing techniques. That is, the advantage of data that has

been collected in a number of spectral bands is that it is possible to label pixels individually. This is because it is possible to discriminate between different materials in a scene based on the difference in the spectral response of the various materials, rather than spatial characteristics, such as the physical shapes of scene elements. Being able to label pixels individually makes very efficient use of the spatial resolution of the sensor. Spatial resolution is one of the most expensive parameters of data collection.

Analysis of multispectral data begins with viewing the measurements made on each pixel as an N-dimensional vector, where N is the number of spectral bands sensed. The most common product of such analysis is a thematic map, which is a digital image of the scene in which each pixel in the scene is labeled with one of the desired classes of the data set.

The analysis process may be viewed as a mapping from points in an N-dimensional vector space to a one-dimensional set of M labels specified by the analyst. The way this can best be accomplished is via a method called *pattern recognition* (Swain and Davis, 1978; Richards and Jia, 1999; Fukunaga, 1990). To see how this mapping might be done, consider a case where measurements are made in only two spectral bands for each pixel, and so $N = 2$. Assume there are three classes in the data set, $M = 3$. In two-dimensional space, the data might look as shown in Fig. 10-1. Here, the abscissa value of a pixel indicates the magnitude of the response in band 1 and the ordinate indicates the response of the pixel in band 2. The classification problem then comes down to dividing this two-dimensional space into three nonoverlapping regions so that any outcome is uniquely associated with one of the three classes.

10.2.1 Discriminant Functions

A very powerful way to accomplish this partitioning is based upon so-called *discriminant functions*. Assume that one can determine M functions of the N-dimensional vector $\boldsymbol{X}$, $\{g_1(\boldsymbol{X}), g_2(\boldsymbol{X}), \ldots, g_M(\boldsymbol{X})\}$, such that the value of $g_i(\boldsymbol{X})$ is larger than all the others when $\boldsymbol{X}$ is from class i. Then a decision rule to accomplish the desired mapping from data to classes would be

> Let ω_i denote the ith class.
>
> Then, decide measured (vector) value $\boldsymbol{X}$ is in class i ($\boldsymbol{X} \in \omega_i$) if and only if $g_i(\boldsymbol{X}) \geq g_j(\boldsymbol{X})$ for all $j = 1, 2, \ldots, M$.

Note that this rule is particularly easy to implement in computer code. Given the discriminant functions, to classify a new pixel, the algorithm has only to calculate the magnitude of the M discriminant functions and select the largest one.

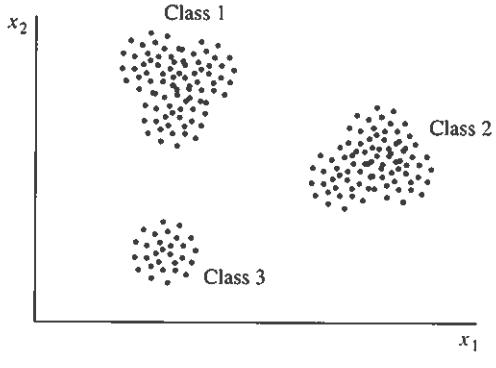

Figure 10-1 Two-dimensional data plotted for three hypothetical classes.

But how does one determine what the discriminant functions should be? In answering this, one must note that the spectral response for any given material is best described as a distribution. Any given material in a scene will

- exist in a number of different states
- be observed through many different columns of atmosphere
- be illuminated and observed from a number of different angles
- have a number of different topographic positions
- be near to any number of other kinds of materials

All of these and a number of other variables have an effect on the spectral response. Some of these effects tend to be diagnostic of the material, but some are not. Thus the spectral response expected from a given material will not be a single point in the N-dimensional feature space referred to above. It will be a distribution in this space.

If this distribution can be accurately described in terms of a probability density function, then the density function itself can be used as a discriminant function for that class. The value of a probability density function at any point is a quantitative measure of how likely that value is to occur. Thus, if one has the density function for all M classes, assignment of a data vector to a particular class is a matter of evaluating the value of all M densities at a given point to see which one indicates the greatest likelihood. This type of analysis scheme is referred to as *maximum likelihood classification.*

This leads to another question, namely how to determine the class probability density functions for a practical circumstance. One usually begins with the data set on the one hand and one or more classes to be identified on the other. The most effective way for the user to specify what is desired of the analysis process is to label some examples of each class in the data set to be analyzed. These samples are called *training samples,* or *design samples*, and this phase of the analysis process is the most critical part of analyzing a data set.

For optimal performance, the list of classes must be

- *Of informational value.* Obviously, the user must specify at some point the classes about which information is desired.
- *Separable.* There is no reason to specify classes that cannot be discriminated based upon the spectral features at hand.
- *Exhaustive.* There must be a logical class to which every pixel in the scene can be assigned.

Note that the user imposes the first of these three requirements, while the latter two are conditions determined by the data. The user's desires and the properties of the data must be brought together in the analysis process. This part of the task is not a trivial one, as these three conditions must be met simultaneously. It is often not trivial to be able to specify, even by direct example in the data set, what the limits of each desired class are intended to be, to specify the limits so that the classes turn out to be adequately separable, and at the same time to anticipate all of the spectral classes that are present in the data set at hand. Often, a good deal of practice and skill are needed, and there are many possible algorithms (Landgrebe, 1999) and tools that can aid in the procedure. Thus we will need to spend a good deal of time studying this process.

10.2.2 Training a Classifier

A wide variety of methods have been devised to train classifiers. Some do not appear to involve estimating the class density function, although they nearly always amount

to that. One of the most common schemes is to model each class density function in terms of one or more Gaussian probability density functions. The Gaussian density function in one dimension is given by

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp\left[\frac{-(x-\mu_i)^2}{2\sigma_i^2}\right] \tag{10-1}$$

where x is the variable of the density and corresponds to the measured value of radiance of the pixel, ω_i designates class i, μ_i indicates the mean or average value of x for class i, and σ_i^2 indicates the variance of x around μ_i. In the usual case of more than one dimension ($N > 1$), though ω_i remains a scalar, x and μ_i become vectors of dimension N, and σ_i^2 becomes an N-dimensional matrix.

From probability theory (Cooper and McGillem, 1999), it is shown that

$$p(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)p(\omega_i)}{p(\boldsymbol{x})} \tag{10-2}$$

The notation used here is as follows:

$p(\omega_i|\boldsymbol{x})$ is the probability of class ω_i, given the measured value $\boldsymbol{x}$

$p(\boldsymbol{x}|\omega_i)$ is the probability of measured value $\boldsymbol{x}$, given class ω_i

$p(\omega_i)$ is the probability that class ω_i occurs in the data

$p(\boldsymbol{x})$ is the probability that measured value $\boldsymbol{x}$ occurs in the data

The relationship described by Eq. 10-2, known as *Bayes' Theorem*, is very useful in classification. It is the quantity on the left, the likelihood of class ω_i, given the measurement $\boldsymbol{x}$, that we seek to maximize by picking the correct class i, thus providing the minimum error rate. The term $p(\boldsymbol{x}|\omega_i)$, the probability of $\boldsymbol{x}$, given class ω_i, is available as a result of the training process. The quantity $p(\omega_i)$ is known as the *prior probability*, or the probability of the given class before the data is evaluated. The same relationship is valid if each of the quantities are density functions rather than discrete probabilities.

Since the denominator quantity, $p(\boldsymbol{x})$, is the same for all classes, the quantity in the numerator should be used as the discriminant function. Bayes' Theorem then ensures that the error rate will be a minimum. Such a minimum error rate classifier is known as a *Bayes classifier*. For calculation purposes, the discriminant function can be simplified a bit further, as follows.

In the N-dimensional case, the Gaussian probability density function is written in vector notation as

$$p(\boldsymbol{x}|\omega_i) = (2\pi)^{-N/2}|\Sigma_i|^{-1/2}\exp\left[-\frac{1}{2}(\boldsymbol{x}-\mu_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\mu_i)\right] \tag{10-3}$$

The Bayes rule classifier, also called the *minimum a posteriori error rule* would now be: Decide $\boldsymbol{x}$ is in class ω_i if and only if

$$p(\boldsymbol{x}|\omega_i)p(\omega_i) \geq p(\boldsymbol{x}|\omega_j)p(\omega_j) \text{ for all } j = 1, 2, \ldots, M \tag{10-4}$$

Now if $p(\boldsymbol{x}|\omega_i)p(\omega_i) \geq p(\boldsymbol{x}|\omega_j)p(\omega_j)$ for all $j = 1, 2, \ldots, M$, then it is also true that

$$\ln p(\boldsymbol{x}|\omega_i)p(\omega_i) \geq \ln p(\boldsymbol{x}|\omega_j)p(\omega_j) \text{ for all } j = 1, 2, \ldots, M \tag{10-5}$$

Thus, we may take the following as an equivalent discriminant function that requires substantially less computation time. (Note in this expression that we have dropped the

factor involving 2π since it would be common to all class discriminant functions and thus does not contribute to the discrimination.)

$$g_i(x) = \ln p(\omega_i) - \frac{1}{2}\ln |\Sigma_i| - \frac{1}{2}(x - \mu_i)^T\Sigma_i^{-1}(x - \mu_i)$$

or

$$2g_i(x) = \ln\left(\frac{p^2(\omega_i)}{|\Sigma_i|}\right) - (x - \mu_i)^T\Sigma_i^{-1}(x - \mu_i) \qquad (10\text{-}6)$$

For the same reason, we may drop the leading factor of two. Note also that the first term on the right in Eq. 10-6 only must be computed once per class, and only the last term must be computed for each measurement to be classified.

Thus, in this case, the determination of the needed discriminant functions has been reduced to estimating the mean vector μ_i and the covariance matrix Σ_i for each class.

10.2.3 Classes and Subclasses

The derivation of the discrimant functions described above assumes that the probability distribution for a given class is a Gaussian distribution, meaning that the class has a single mode with the familiar e^{-x^2} shape. This is often not a very good model for a class. It may not allow sufficient flexibility to the user. For example, in an agricultural area, the planting season of corn in the spring might have been interrupted by a rainy period, so that some of the corn was planted early and some late. Thus, later in the season the corn canopies might be found in two different states. The user, on the other hand, might not wish to distinguish between these two states, and simply desire one class called corn.

This circumstance can easily be handled by training two spectral classes for the user class "corn," allowing the classification to take place on this basis, then combining the two together after the classification. The two spectral classes would be referred to as subclasses of the class "corn." By using a varying number of subclasses for a desired user class, arbitrarily complex non-Gaussian distributions can be modeled successfully.

10.2.4 Training Samples and Estimation Precision

A disadvantage of this class/subclass scheme is that more spectral classes must be trained, and this requires more training samples. It is characteristic of the remote sensing situation that the number of training samples available is always less than might be desirable. The process of labeling samples within a data set to be analyzed so they can be used for training can often be a complex task. It is usually situation-specific and so each case must be dealt with in a unique fashion.

Thus, on the one hand, one would like to spend as little time on the training sample labeling process as possible. On the other hand, the precision with which the classes are defined, and therefore the accuracy of the resulting classification, is strongly dependent upon the number of samples available by which to estimate the parameters of the class description. So far, we have only described one classifier algorithm, the maximum likelihood Gaussian one. There are many others, some of which we will examine later. This strong dependence on the number of training samples is true regardless of the algorithm used. One cannot expect a good quality output from the classifier

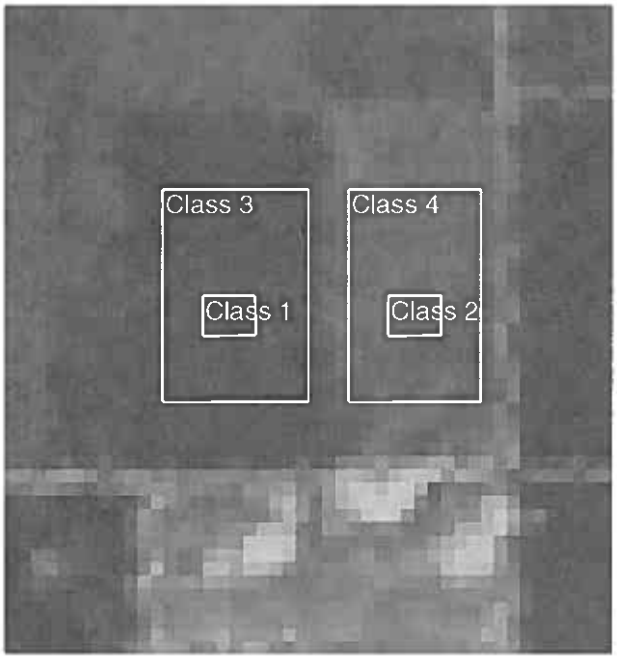

Figure 10-2 A small portion of a Landsat TM scene with test areas for two agricultural fields.

unless there is a good quality input to it, in terms of an accurate and precise quantitative description of the classes desired.

A simple example may help to make the situation clearer. Figure 10-2 shows a small portion of a Landsat TM frame over an agricultural area with test areas for two agricultural fields marked. If the areas marked Class 1 and Class 2, each containing 12 pixels, were to be used as training areas for those classes, and it was proposed to use Thematic Mapper bands 1 and 3 for the classification, the estimated mean values and covariance matrices would be

$$\mu_1 = \begin{bmatrix} 83.4 \\ 25.7 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 85.2 \\ 29.3 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 1.17 & 0.06 \\ 0.06 & 0.24 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1.66 & 0.73 \\ 0.73 & 2.97 \end{bmatrix}$$

$$\rho_{13_1} = 0.11 \quad \rho_{13_2} = 0.33$$

Figure 10-3 shows a plot of the 12 data points, showing band 1 (abscissa) vs. band 3 (ordinate). In addition, the *area of concentration* is shown for each class for a Gaussian density with the same mean vector and covariance matrix as the training data.

The elements of μ_i are the mean or average values of the 12 training pixels in band 1 (upper element of μ_i) and band 3 (lower element of μ_i). The elements of Σ_i, on the major diagonal of the matrix, are the variances, σ_j^2 of the data in the individual bands. Thus, for band 1 of class 1, $\sigma_1^2 = 1.17$ quantifies how much the data varies about its mean value of 83.4, while $\sigma_3^2 = 0.24$ quantifies the variation in band 3 about its mean value of 25.7. The off-diagonal element of Σ_i is the covariance value between band 1 and band 3, σ_{13}. It relates to how the data in bands 1 and 3 vary with respect to one another.

A normalized form of this covariance element is computed as

$$\rho_{13} = \frac{\sigma_{13}}{\sqrt{\sigma_1^2 \sigma_3^2}}$$

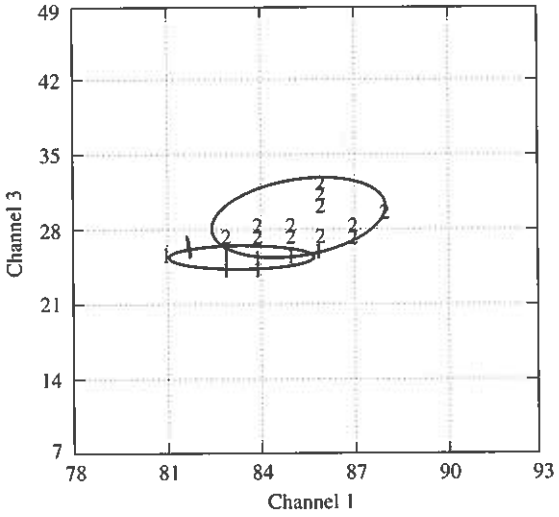

Figure 10-3 A scatter plot of the training samples for Classes 1 and 2. The ovals show the area of concentration for a Gaussian density with the same mean vector and covariance matrix.

This quantity is the correlation coefficient of the data between the two bands. A correlation coefficient can vary over a range $-1 \leq \rho_{jk} \leq +1$, indicating the degree to which data in the two bands tend to vary in the same direction (positive correlation) or in opposite directions (negative correlation). This turns out to be very useful information for a classifier. Two bands that have positive correlation tend to be distributed along a line slanted at 45 degrees upward to the right, assuming equal variances. The higher the correlation, the more closely the data approaches the line. The distribution is similar for negative correlation, but along a line upward to the left. Correlation values near zero imply distributions that tend to be circularly distributed, not having any favored direction.

Correlation between bands may in some instances seem undesirable. For example, it can suggest redundancy. However, in the case of classification, another more positive interpretation is appropriate. The mean value of a class defines where the class distribution is located in the feature space. The covariance matrix provides information about the shape of the distribution. Here, it is seen that the higher the correlation between features, the more concentrated the distribution is about a 45-degree line. Zero correlation means it is circularly distributed, thus occupying a greater area (volume) in the feature space. In this case, there may be a greater likelihood that the distribution will overlap with a neighboring one.

With this in mind, let us return to the consideration of the effect of the size of the training set. Consider defining the same user classes, but with a larger number of training samples. If, instead of the areas marked Class 1 and Class 2 in Fig. 10-2, the areas marked Class 3 and Class 4, each containing 200 points, are used as training samples, the corresponding results would be (Fig. 10-4)

$$\mu_3 = \begin{bmatrix} 83.5 \\ 26.2 \end{bmatrix} \quad \mu_4 = \begin{bmatrix} 86.9 \\ 31.2 \end{bmatrix}$$

$$\Sigma_3 = \begin{bmatrix} 1.86 & 0.13 \\ 0.13 & 1.00 \end{bmatrix} \quad \Sigma_4 = \begin{bmatrix} 3.31 & 2.42 \\ 2.42 & 4.43 \end{bmatrix}$$

$$\rho_{13_3} = 0.09 \quad \rho_{13_4} = 0.63$$

Comparing μ_1 with μ_3 and μ_2 with μ_4 shows that there is relatively little change, indicating that the locations of the two distributions were reasonably well determined by

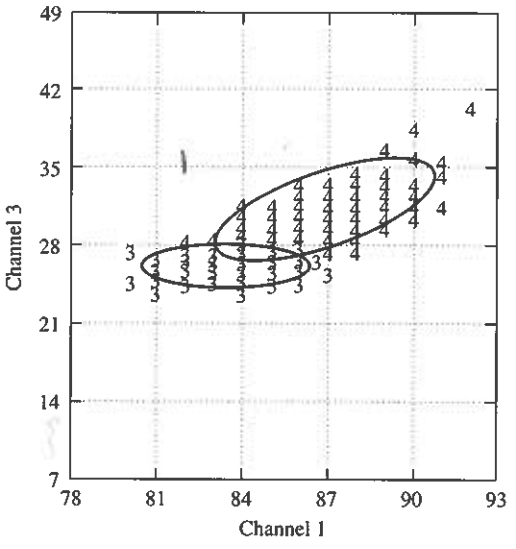

Figure 10-4 A scatter plot of the training samples for Classes 3 and 4 of the areas in Figure 10-2. The ovals show the area of concentration for a Gaussian density with the same mean vector and covariance matrix as the data.

the smaller training sets. However, the change in the corresponding covariance matrices is greater. The implication of this change is that the shape of the distributions was not as well determined by the smaller training sets.

This is a well-known result. The mean vector is known as a *first-order statistic*, because it involves only one variable. The covariance matrix is called a *second-order statistic*, because it involves the relationship between two variables; the correlation shows how two variables relate to one another. Higher-order statistics involve the relationships between more variables. To perfectly describe an arbitrary class density function would require knowing the value of statistics of all orders. However, this would require an infinite number of samples by which to estimate the statistics of all orders.

It is also the case that as the order of the statistic grows, the estimation process using a finite number of samples becomes more problematic. This is why one would in general expect that the mean vector would be reasonably well estimated with a smaller number of samples than would the covariance matrix, the circumstance we observed in the above example.

10.2.5 Training Samples and the Number of Bands

Another factor in the training process that relates to the size of the training set has to do with the number of bands or spectral features that are to be used. Some years ago, Hughes (1968) derived a very general but very useful theoretical result that bears on the problem at hand. The result is shown in Fig. 10-5. This graph, which was derived relative to pattern recognition problems in general rather than specifically to remote sensing data, shows the relationship between expected classification accuracy (averaged over the ensemble of all classifiers) and measurement complexity. Measurement complexity relates to the number of discrete locations in the feature space. In the case of multispectral data, measurement complexity thus relates to the number of spectral bands and the number of discrete values in each spectral band. If the data has N spectral bands and there are R discrete values in each band, then the total number of discrete locations in the feature space is R^N. The parameter m in Fig. 10-5 is the number of training samples used. The graph assumes a two-class problem with the two classes equally likely.

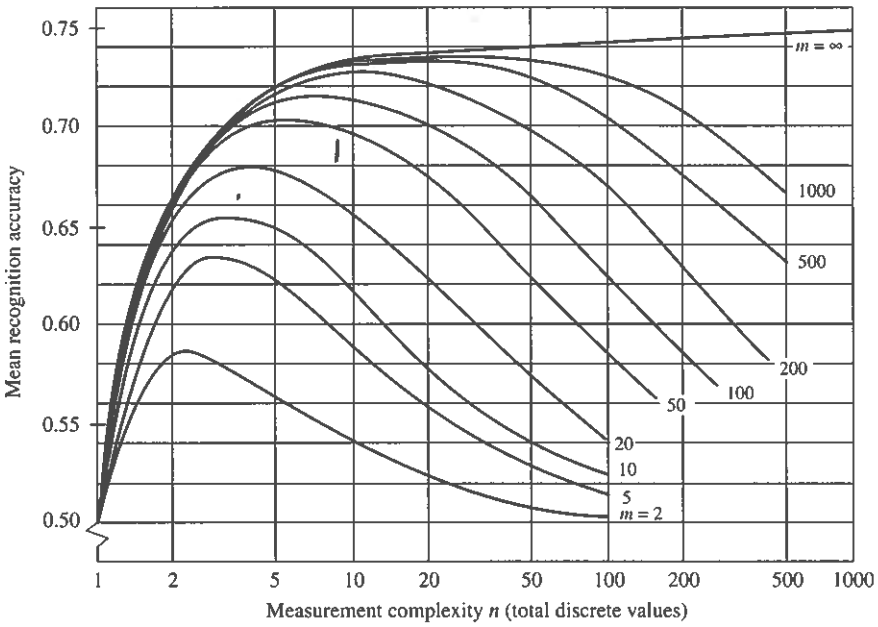

Figure 10-5 Mean recognition accuracy vs. measurement complexity for the finite training case.

If $m \longrightarrow \infty$, implying perfectly precise definition of the class statistics, the expected accuracy increases continuously with increasing measurement complexity, rapidly at first, but then more slowly. However, in a practical circumstance with a finite number of training samples, the curve has a maximum, indicating that there is an optimum measurement complexity. Using too many spectral bands would result in less than optimal performance. Notice also that the peak of accuracy moves upward and to the right as m is increased. This indicates that greater accuracy can be expected in general by increasing the number of spectral bands, but to achieve it, greater numbers of training samples would be required.

This graph is a theoretical result, and since it is quite general, analysis results may not conform to it exactly in any specific case. However, practical analyses do tend to show this general behavior. That is, if one were to analyze a given data set with a fixed (finite) number of training samples, varying the number of spectral bands, the accuracy of the result would tend to increase with increasing number of spectral bands to a point, then decrease. This reinforces the importance of choosing the right number of spectral features to use in any given analysis, using as many training samples as possible. It also makes clear that the number of bands to be used, the number of training samples, and the expected accuracy are variables that are all interrelated. We shall return to this point shortly.

10.2.6 Other Classification Algorithms

In Section 10.2.2, the Gaussian probability density function was described as a suitable discriminant function for many circumstances. The advantage of this model for multivariate data is that it utilizes both first-order and second-order variations of the data in feature space. It thus results in decision boundaries in feature space that are, in general, segments of second-order surfaces. One of the forms in which the Gaussian density was expressed as a discriminate function is

$$g_i(x) = \ln p(\omega_i) - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x - \mu_i)^T\Sigma_i^{-1}(x - \mu_i) \quad (10\text{-}7)$$

However, as has just been seen, the size of the training set to be used in estimating the parameters of a class is limited. The imprecise parameter estimates that result from this limitation in the training set size causes the error rate to be higher than necessary. We also noted that the estimation precision tends to affect higher-order statistics most. In this case, that means there may be more problems with the estimated covariance matrix, Σ_i, than with the mean vector, μ_i. As a result, a simpler classifier may outperform this more complex one.

The classifier known as the *Fisher linear discriminant* is one simplification of the Gaussian distribution classifier. In this case, it is assumed that all classes have a common class covariance and thus the training samples from all classes may be used to estimate the one common covariance matrix. Even though the decision boundary in feature space is now restricted to be segments of linear surfaces instead of second-order surfaces, higher accuracy can result from the greater estimation precision. The discriminate function in this case becomes

$$g_i(x) = \ln p(\omega_i) - \frac{1}{2}(x - \mu_i)^T\Sigma^{-1}(x - \mu_i) \quad (10\text{-}8)$$

Note the differences between Eq. 10-8 and Eq. 10-7. The Σ now has no subscript, since there is only one, and the second term on the right of Eq. 10.7 can be dropped, as it would be the same for all classes.

A further simplification results in the *Minimum distance to means* classifier. In this case, the covariance term is dropped completely, again resulting in linear decision boundaries, but without influence in their orientation based upon the shape of the class distributions in feature space. In this case, the discriminate function becomes

$$g_i(x) = \ln p(\omega_i) - \frac{1}{2}(x - \mu_i)^T(x - \mu_i) \quad (10\text{-}9)$$

An additional advantage in this case is that eliminating the covariance matrix from the discriminant functions means that significantly less computation is needed in the classification process. There are even simpler classifiers than that defined by Eq. 10-9, which require even less computation. However, one quickly reaches a point of diminishing returns, depending on the complexity of the task, with performance of the classifier falling below acceptable standards.

There are many classifiers that are more complex, or utilize different approaches. Some are based upon different schemes for modeling the class distributions, and some on how the locations of the decision boundaries are located and how the training process is implemented. Popular classifiers of the former type are *K-nearest neighbor* schemes, those based upon the theory of *fuzzy sets*, and *Parzen density estimators*. In the latter category, *neural network* implementations are an example.

Neural networks and Parzen density estimators are said to be *non-parametric* or *distribution-free* schemes, in that the class probability density functions have no initially prescribed forms and thus may be seen as perfectly general. However, in fact, all classifiers necessarily have parameters, and the more general they are, the more parameters they must have to describe that generality quantitatively. The Hughes phenomenon of Fig. 10-5 makes clear that there is a price to be paid for that generality. The greater the complexity of the class description in terms of the number of parameters used, the greater the size of the training set required to adequately quantify the required amount of detail.

10.2.7 Clustering: Unsupervised Classification

As was pointed out earlier, achieving a quantitative definition of the classes to be used is perhaps the most critical step in the classification process. A maximally effective set of classes must be (1) of informational value, (2) separable, and (3) exhaustive. The use of training samples to define the classes is referred to as *supervised classification,* because, via the training samples, the human analyst is supervising the definition of the classifier and thereby ensuring that the resulting classes will be of informational value. However, simply listing training samples for the desired set of classes does not necessarily lead to a set of classes that are separable or exhaustive.

Another frequently useful type of classification is called *clustering* or *unsupervised classification.* Though it is seldom useful for directly achieving a final classification, it can be very helpful in establishing a set of training data that meets the three required conditions stated above. The basic concept is to group the pixels into an appropriate number of clusters in feature space, in a manner that satisfies an appropriate optimality criterion. We will illustrate the concept with a (perhaps oversimplified) example using two-dimensional real data. Figure 10-6 shows in image form a small area from an agricultural region. The dotted box designates an area that appears to be dominated by two informational classes. Without any information about what is contained in the two agricultural fields, suppose we wish to divide the pixels in the dotted box into two groups based upon their spectral similarity in all the bands contained in the data set.

The data from the dotted box area of Fig. 10-6 in two spectral bands is plotted in Fig. 10-7. The desired classification of the pixels into two groups can be accomplished by using a clustering algorithm. There are many different clustering algorithms in the literature. In general, three basic capabilities are needed in a working clustering algorithm:

1. a measure of similarity or distance between points
2. a measure of similarity or distance between point sets
3. a clustering criterion

In essence, one uses the measure of distance between points to decide which points are close to one another in N-dimensional space. Points that are close together are

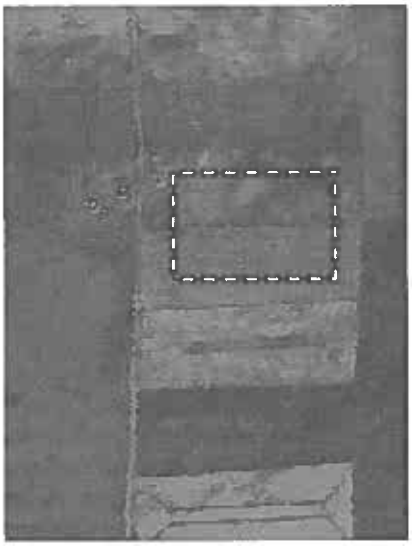

Figure 10-6 An image of a small area in an agricultural region with a test area including parts of two fields marked.

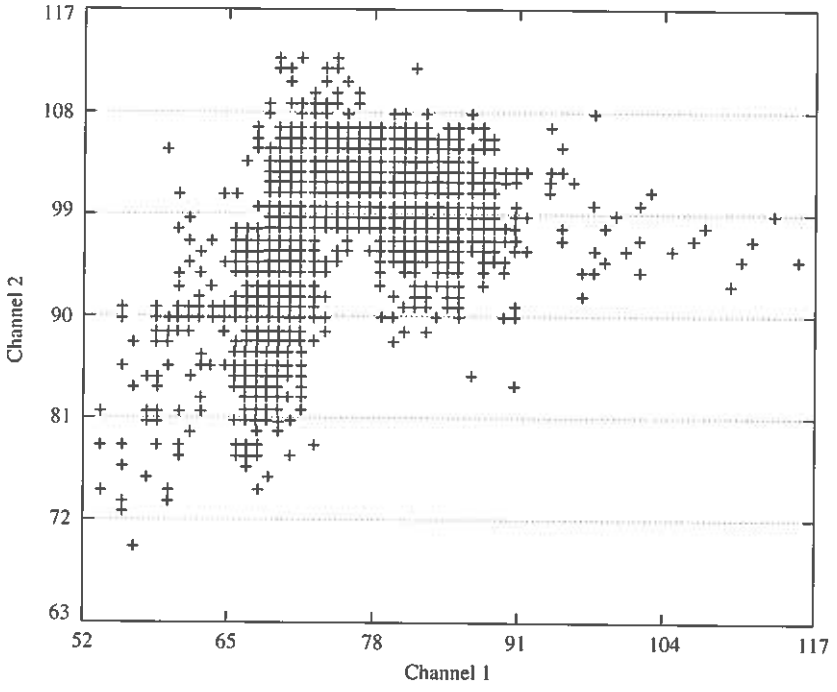

Figure 10-7 A scatter plot of the data in two bands from the region marked in Figure 10-6.

grouped to form point sets, or clusters. The measure of distance between point sets is used to determine whether the clusters are sufficiently distinct from neighboring clusters. The clustering criterion is then used to determine if each cluster is sufficiently compact and also adequately separable from the other clusters. The process is usually an iterative one in which points are compared with tentative cluster centers and then the clusters are tested to see if they meet the cluster criterion.

To begin the process, one must have a means for establishing initial cluster centers as a starting point. A typical sequence of steps for a clustering algorithm is as follows.

1. Estimate or specify the number of clusters needed, and select (often arbitrarily) an initial cluster center for each.
2. Assign each point to the nearest cluster center, using the measure of distance between points.
3. Compute the mean value of the points assigned to each cluster, and compare this with the previous center. If the mean value is not at the cluster center, assign the mean value as the cluster center and return to Step 2 using the new cluster centers.
4. Determine if the clusters have the characteristics required
   a. Are they sufficiently compact? This might be done, for example, by using the squared sum of the distances between the points and their cluster center. If they are not sufficiently compact, subdivide any that are too distributed, choose centers for the new clusters, and go back to Step 2.
   b. Are they sufficiently separated from other clusters, using the measure of distance between point clusters? If not, combine those that are too close together, assume a new cluster center and go back to Step 2. If so, clustering is complete.

Notice that such a procedure has the ability to end up with either more or fewer cluster centers than it started with.

It should be noted that clustering is usually carried out on higher-dimensional data, for which it is not possible to have points plotted to visualize what clusters might be appropriate or even where would be logical points for the initial cluster centers. This is a major reason an algorithm is needed, rather than simply doing the clustering manually. We use this two-dimensional situation here only to show the concept.

Continuing the example of Figs. 10-6 and 10-7, we must select the number of clusters desired and the initial location of the cluster centers. We will seek two clusters in this case. One way to select initial cluster centers is to space them equally along the major diagonal of the data to be clustered. The two points indicated by the round dots in Fig. 10-8 are the result of doing so in this case.

Then, using the algorithm steps listed with simple Euclidean distance as the measure of difference between points and the sum of the squared distances as the clustering criterion, the cluster centers move in the directions shown in Fig. 10-8 in nine iterations to the final points indicated by the diamonds in Fig. 10-8. The final cluster labels of the points are shown in Fig. 10-9.

A clustering algorithm can be very useful in helping to establish training sets for desired classes, because it gives an initial indication of what data might be separable from what. When presented with a seven-band data set for an area containing perhaps ten or so classes, for example, one might have no idea how to set up classes and subclasses initially. By first clustering the data, one could begin to tell which areas of the data set are similar to one another spectrally and which are not. Note, however, that clustering or unsupervised classification cannot normally be expected to indicate informational classes which are separable for you. For example, using the cluster results on the example would give the classification result shown in Fig. 10-10. Clearly, as a

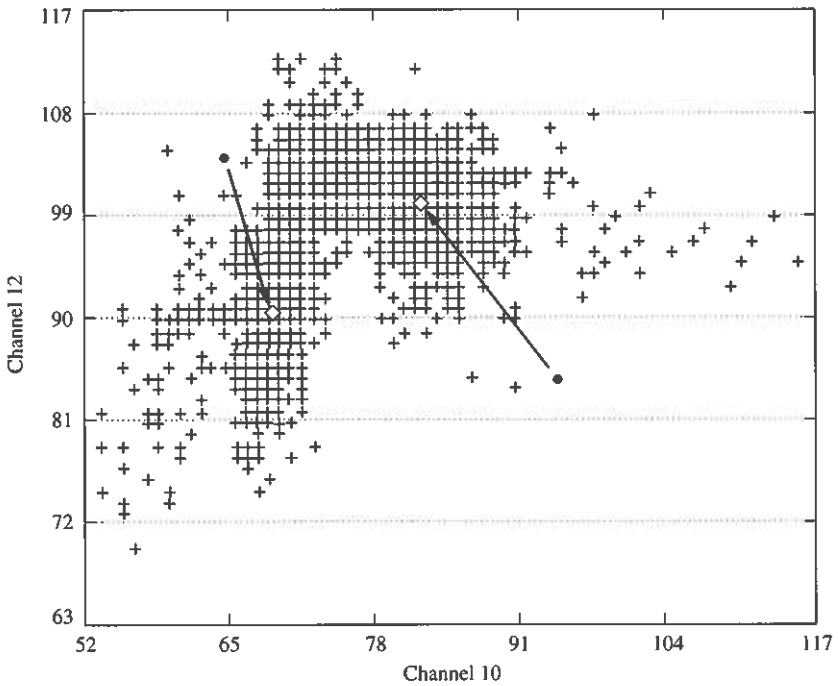

Figure 10-8 The scatter plot of Figure 10-7 showing the migration of the cluster centers during the iterations of the cluster process.

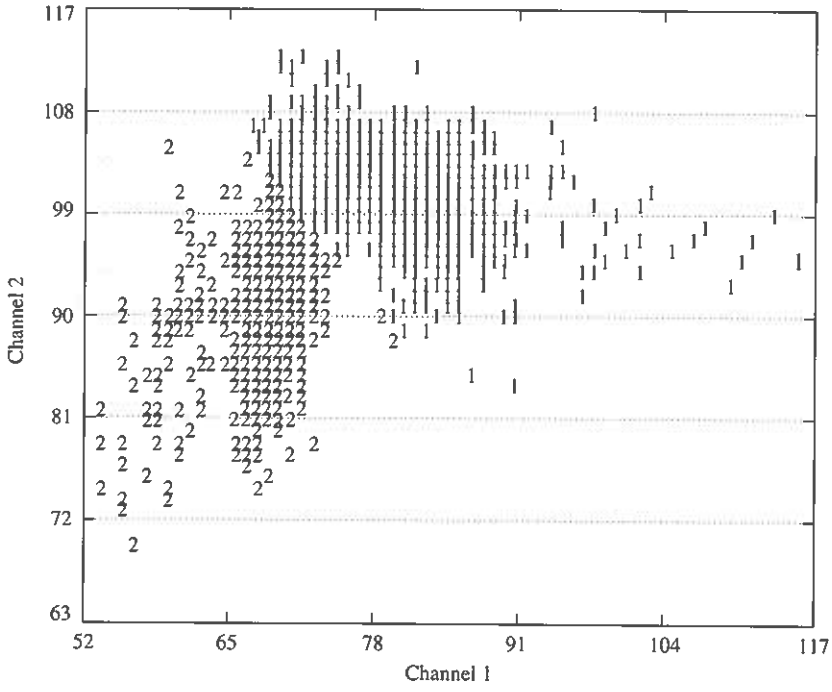

Figure 10-9 The scatter plot of Fig. 10-7 showing the final cluster assignments of the pixels resulting from the clustering.

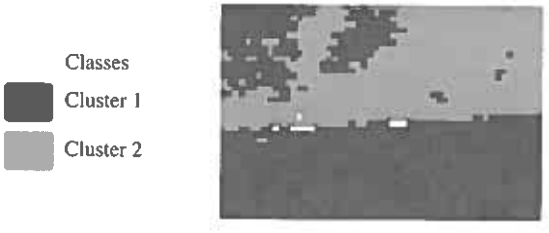

Figure 10-10 A thematic presentation showing how the pixels of Figure 10-6 were assigned to the two classes as a result of the clustering.

final classification, the error rate is higher than desirable and, by itself, poorer than what is possible, even using only two bands.

10.3 FEATURE REDUCTION AND SPECTRAL TRANSFORMATIONS

As was shown in Section 10.2.5, the use of too many features could lead to suboptimal results because of parameter estimation imprecision due to the limited size of training sets. It is not a matter of having sensors with smaller numbers of bands, because a different set of spectral features is optimal for every different problem. Thus, generally, it is desirable to have data gathered in a large number of spectral bands, but to then be able to determine the best subset of spectral features to use for classification in a problem-specific way. A number of feature reduction methods have appeared in the literature. A representative sampling of them is given in this section.

10.3.1 Subset Selection

One of the most straightforward ways to reduce the dimensionality of the data set is to simply select a subset of the available bands. A convenient means for finding an optimal spectral band subset is to use a separability measure to find the best (most separable) M bands of the N available, $M < N$. Given the training samples for a desired set of classes, for example, one might seek to find the best 4 bands out of the 7 available. This method rests on the ability to project classification accuracy from the training data and each possible subset without actually doing the classification for the entire data set.

There are a number of methods for projecting the relative classification accuracy from training statistics. One common and very useful method is to use a statistical distance measure, which indicates the relative separation between two distributions in N-dimensional space. A particularly effective measure for this purpose is the *Bhattacharyya distance*. In terms of the mean vectors μ_i and the covariance matrices Σ_i of two classes, the Bhattacharyya distance is given by

$$B = \frac{1}{8}(\mu_1 - \mu_2)^T\left[\frac{\Sigma_1 + \Sigma_2}{2}\right]^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\ln\left[\frac{\left|\frac{1}{2}(\Sigma_1 + \Sigma_2)\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}\right] \tag{10-10}$$

This quantity is known to have a nearly linear relationship with classification accuracy. Notice that the first term on the right measures the separability due to the difference in mean values between the two classes, while the second term on the right measures the separability due to the covariance matrices.

One can use such a metric to examine the various subsets of size M that exist in the N dimensions. To see how this might work, consider the following example. Assume that there are 9 classes defined for a 12-band data set and it is desired to find the best 4 of the 12 bands to use for classification. Then the Bhattacharyya distance between each pair of the 9 classes would be computed for each of the possible subsets of 4 bands out of the 12. Use of the binomial coefficient

$$\binom{12}{4} = \frac{12!}{4!(12-4)!} = 495 \tag{10-11}$$

shows that there are 495 4-tuples in 12-dimensional data that must be examined. There are 28 possible class pairs in 9 classes. Thus, one would compute the Bhattacharyya distance between each of the 28 class pairs for each of the 495 4-tuples. Then one might rank order the results based on the average Bhattacharyya distance for each 4-tuple. The result might be as shown in Table 10-1.

Table 10-1 suggests that, based upon the average of the Bhattacharyya distances between class pairs, bands 1, 6, 9, and 12 would be the best 4 bands of the 12 to use, because the average of the interclass distances is 11.5, which is greater than that for any other 4-tuple row.

Another possibility is to choose the 4-tuple with the largest minimum interclass distance. For example, in Table 10-1, the pair of classes 1 and 5 appears to be more difficult to separate. Looking down the 15 column, one sees that bands 1, 6, 9, and 10 have a Bhattacharyya distance of 1.84, compared with only 1.69 for 1, 6, 9, and 12. Thus, even though the average for 1, 6, 9, and 10 is slightly smaller than that for 1, 6, 9, and 12, the 4-tuple of 1, 6, 9, and 10 may do a better job on the difficult problem of separating classes 1 and 5.

This method of finding an optimum subset is very effective in many cases. It works especially well when the number of bands available is not too large (< 10 or so). However, it begins to present problems as the dimensionality increases. For example, to find

Table 10-1 Bhattarcharyya Interclass Distances

		Class pairs	1 2	1 3	1 4	1 5	1 6	...	1 9	2 3	...
Rank	Bands	Average									
1	1 6 9 12	11.50	20.6	3.75	5.16	1.69	12.5	...	30.3	16.5	...
2	1 6 9 11	11.23	20.2	3.37	4.81	1.65	12.2		30.0	16.3	
3	1 6 9 10	10.87	19.8	4.73	4.26	1.84	11.7		29.8	16.4	
4	2 6 9 12	10.65	19.2	3.40	5.44	1.22	12.1		30.3	16.4	
5	1 6 8 9	10.59	19.8	1.61	4.74	1.53	14.2	...	30.5	16.6	...
6	1 7 9 12	10.53	18.5	3.82	4.51	1.34	12.6		23.7	14.4	
7	1 6 8 12	10.43	15.9	3.90	5.34	1.63	14.2		23.7	11.4	
8	1 6 10 12	10.30	15.1	6.12	4.41	1.58	10.5		23.7	11.6	
9	1 8 9 12	10.26	16.8	3.87	4.44	1.22	14.2		21.0	13.8	
10	6 9 10 12	10.23	19.3	5.96	4.82	1.25	10.1	...	27.6	17.1	...
	.		.				.			.	
	.		.				.			.	
	(495 rows)		.				.			.	

the best 10 of 50 bands would require a table as in Table 10-1 with 10,272,278,170 rows. For the best 10 of 100, the number of 10-tuples becomes 1.73×10^{13}.

Another limitation of this approach, especially in high-dimensional cases, stems from its exclusive nature. For example, in a case of 100 bands available, picking only 10 completely excludes any separability characteristics of the other 90. Though the contribution of the rejected bands to separability may be somewhat smaller than the 10 picked, it may be significant in aggregate. Is there some way this problem can be reduced, and still achieve the desired dimensionality reduction? Fortunately, there is. It comes by allowing linear combinations of the original bands to form new features.

10.3.2 Principal Components Transformation

Suppose one has data distributed over a two-dimensional region as shown in Fig. 10-11. Since the region is diagonally oriented, data from the two axes are correlated. Define a new set of axes such that the data are horizontally or vertically oriented, so that in this new space, the data will not be correlated (Fig. 10-12).

A transformation to accomplish this is known as a *principal components transformation*. The name derives from the fact that, after deriving the new coordinate values, in

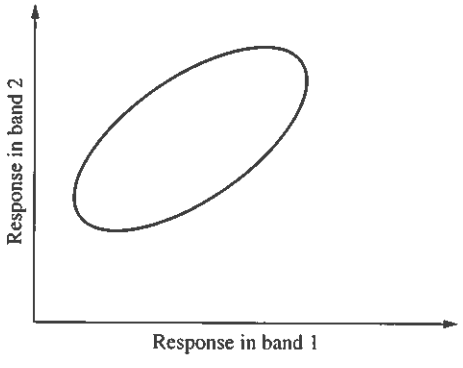

Figure 10-11 A hypothetical data distribution in two-dimensional feature space, for purposes of studying a transformation concept.

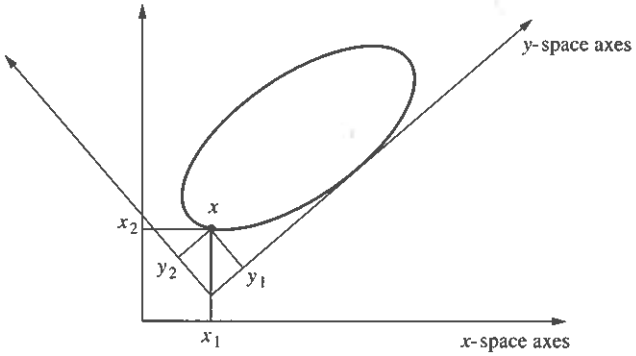

Figure 10-12 Principal component transformation of the data shown in Fig. 10-11. The x-axes are the original components and the y-axes are the principal components.

the form of $y_1 = a_{11}x_1 + a_{12}x_2$ and $y_2 = a_{21}x_1 + a_{22}x_2$, the one with the largest variance is chosen as the first one, and the second one, orthogonal to the first, is the one with the next highest variance, and so on. Data in the new coordinate system will be uncorrelated from feature to feature. Since data in the new coordinate system are now composed of a combination of bands, the term *feature* will be used to refer to them instead of *band*.

Following is a brief example illustrating the effect of a principal component transformation. Figure 10-13 shows images of the first four Thematic Mapper bands for a representative agricultural area. There is good contrast in all four bands, indicating a moderate dynamic range and thus a normal variance for the data in each band. The actual variances for the four bands are 50.41, 24.01, 75.69, and 408.04, respectively.

The result of a principal component calculation carried out on this data is shown in Table 10-2. The eigenvalues for the new features are the variances of the data in the new coordinates. It is seen that the first principal component now has a variance very much larger than the others, and only the first two have a variance of significant size. The coefficients (the a_{ij}'s) for forming the new features are given in Table 10-3. Images made with the new components are shown in Fig. 10-14. The first two have significant contrast, but the dynamic range of the last two is so small as to show mostly noise.

Principal components analysis has some substantial advantages. It is common and widely known, and conceptually, it is easy to understand. Since it is not case-specific and the individual class definitions are not used, the entire data set can be used to determine the required transformation. This means that the parameter estimation problem is minimized. It performs well in lower-dimensional situations, such as for MSS or Thematic Mapper data.

However, it, too, has some significant shortcomings, especially in higher-dimensional cases. Because the transformation does not use the individual class information, it cannot be optimal with respect to class discrimination. It really optimizes the *representation* of the whole data set, rather than the *discrimination between classes.*

In high-dimensional cases, a principal components transformation can actually be detrimental. Suppose, for example, in a 100-band data set that one of the classes to be identified has a spectral feature that is completely and clearly diagnostic of the class that occurs in only one of the 100 bands. Such narrow diagnostic features occur, for example, in geologic mapping problems, where a specific mineral may have a molecular absorption feature in a very narrow region. Since the variation in only one of the 100 bands would be a very small portion of the total variation, this narrow-band feature would be manifest in only the high-numbered principal components, and would therefore probably not be present in the components selected for use. In short, principal component analysis will often de-emphasize any narrow-band feature.

Another possible problem with this transformation results from the fact that the data are represented digitally and thus with a finite range. A well-designed remote sensor data system

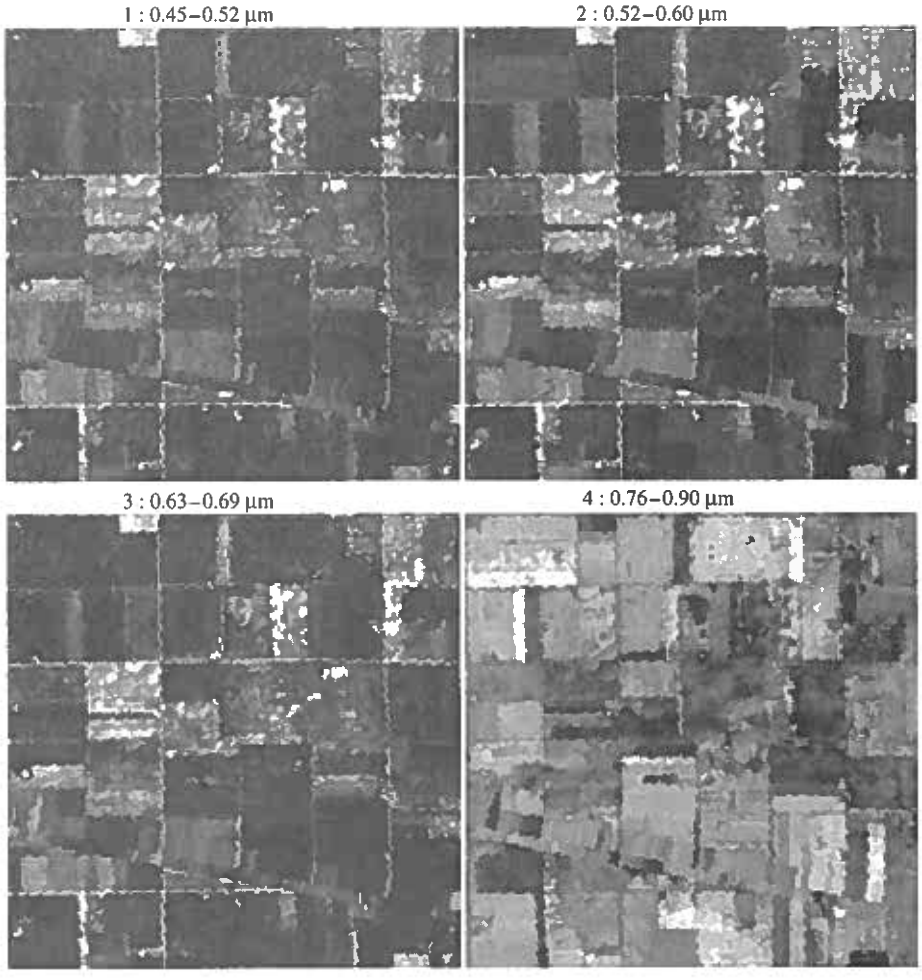

Figure 10-13 Thematic Mapper data of the first four bands expressed in image form.

Table 10-2 Principal Component Transformation of Data from Fig. 10-13

Component	Eigenvalue	Percent	Cumulative percent
1	460.4417	82.3284	82.3284
2	93.7505	16.7629	99.0913
3	3.1477	0.5628	99.6541
4	1.9346	0.3459	100.0000

Table 10-3 Coefficients for Principal Component Transformation of Data from Fig. 10-13

Component	Band 1	Band 2	Band 3	Band 4
1	0.21241	0.12417	0.28065	–0.92774
2	0.54514	0.40520	0.63343	0.37067
3	–0.81072	0.28293	0.51247	0.00727
4	0.02079	–0.86043	0.50732	0.04307

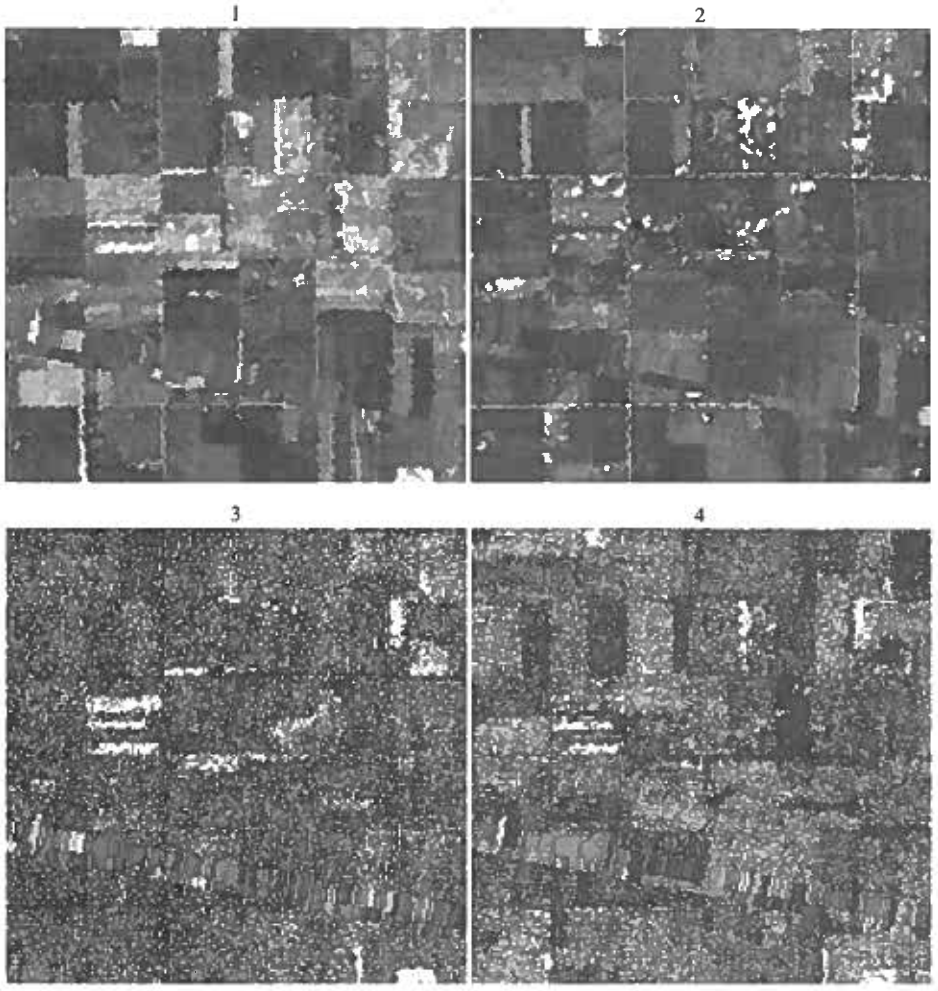

Figure 10-14 Data from Fig. 10-13 after a principal component transformation.

utilizes a significant portion of the available dynamic range over the ensemble of data to be collected. For example, if a sensor system has a 10-bit data system, this means that the digital data would have $2^{10} = 1024$ possible digital values. Then, based upon the expected brightness of the targets to be sensed and the sensitivity of the sensors, the amplifier gains preceding the conversion to digital form would be set so that a good portion of these 1024 values would be used. A principal component transformation significantly alters the data ranges of the various features, making some very much larger and some very much smaller. The features with smaller ranges would be dropped to accomplish the desired dimensionality reduction. On the other hand, it is possible and indeed quite likely that the larger dynamic ranges would substantially exceed the 1024 dynamic range, and some corrective action would need to be taken, reducing its effectiveness. However and worse yet, if this saturation effect went unnoticed, there would be significant distortion in the transformed data.

10.3.3 Discriminant Analysis

The principal components transformation is based upon the global covariance of the entire data set and is thus not explicitly sensitive to inter-class structure. It often works as a

feature reduction tool because, in remote sensing data, classes are frequently distributed in the direction of maximum data scatter. Discriminant analysis is a method that is optimized based on class separability. Consider the hypothetical situation depicted in Fig. 10-15. In Fig. 10-15*a*, the two classes are not separable with one feature axis in the original space, nor with regard to the principal component axis. In Fig. 10-15*b*, the classes would be separable with one feature if the dotted axis could be found. The problem is to find this axis.

Inspection of Fig.10-15*b* reveals that the primary axis of this new transformation should be oriented such that the classes have the maximum separation between their means on this axis, while at the same time they should appear as small as possible in their individual spreads. If the former is characterized by σ_B in the figure and the latter by σ_{W1} and σ_{W2}, then it is desired to find the new axis such that

$$\frac{\sigma_B^2}{\sigma_W^2} = \frac{\text{between-class variance}}{\text{average within-classes variance}}$$

is maximized, where σ_W^2 is the average of σ_{W1}^2 and σ_{W2}^2. In matrix form, the within-class scatter matrix Σ_W and the between-class scatter matrix Σ_B may be defined (Fukunaga, 1990) as

$$\Sigma_W = \sum_i p(\omega_i)\Sigma_i$$

$$\Sigma_B = \sum_i p(\omega_i)(\mu_i - \mu_o)(\mu_i - \mu_o)^T$$

$$\mu_o = \sum_i p(\omega_i)\mu_i$$

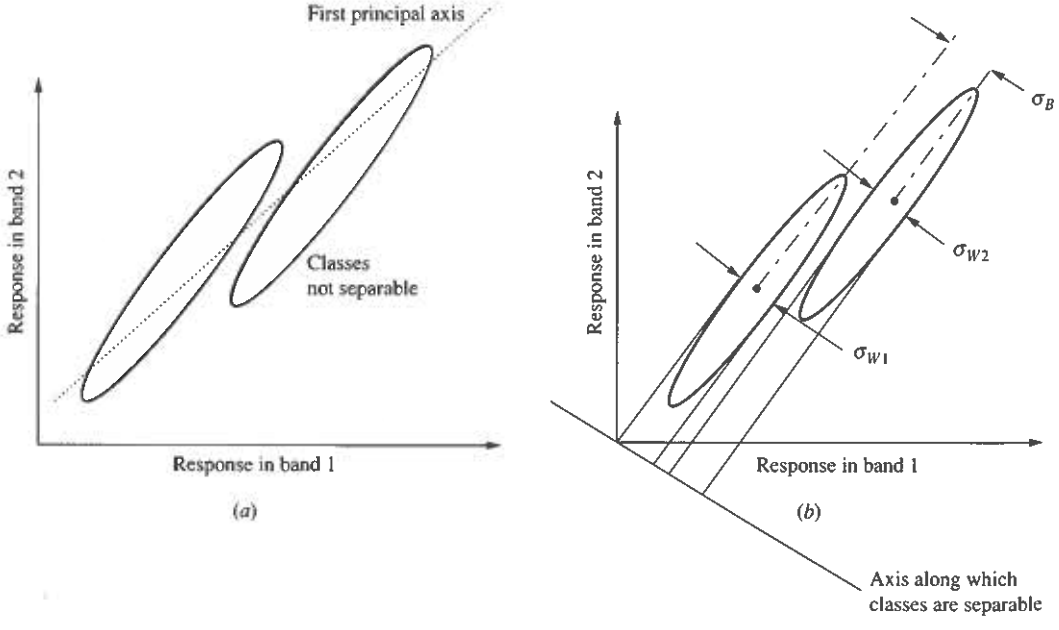

Figure 10-15 The concept for discriminant analysis feature extraction.

where μ_i, Σ_i, and $p(\omega_i)$ are the mean vector, the covariance matrix, and the prior probability of class ω_i, respectively. The criterion for optimization is defined as

$$J_1 = \text{tr}(\Sigma_W^{-1}\Sigma_B) \tag{10-12}$$

New feature vectors are selected to maximize this criterion. This method, referred to as *discriminant analysis feature extraction* (DAFE), like principal components analysis, results in new features that are linear combinations of the original bands. However, DAFE maximizes class separability rather than overall data variation. It is usually a very effective feature extraction method, being a rather short computation.

However, this method also has some shortcomings. Since discriminant analysis mainly utilizes class mean differences, the feature vectors defined by discriminant analysis are not reliable if mean vectors are near to one another, a circumstance that is not uncommon, especially in high-dimensional cases. Also, by using the lumped covariance in the criterion, discriminant analysis may lose information contained in class covariance differences. Another problem with the criteria functions using scatter matrices is that the criteria generally do not have a direct relationship to the error probability. Further, the features defined are only reliable up to one less than the number of classes. Nevertheless, it is an effective and practical means for deriving effective features in many circumstances.

10.3.4 Decision Boundary Feature Extraction

Another approach to feature extraction has been devised that does not have the limitations of the preceding ones. It is based directly upon the decision boundary in feature space and the training samples that define it (Lee and Landgrebe, 1993). Discriminately informative features have a component that is normal to the decision boundary at least at one point, while discriminately redundant features are orthogonal to a vector normal to the decision boundary at every point on the boundary. Based upon this distinction, a decision boundary feature matrix (DBFM) may be defined to extract discriminately informative and discriminately redundant features from the decision boundary. The rank of the DBFM is the smallest dimension where the same classification accuracy can be obtained as from the original feature space, and the eigenfunctions of the DBFM corresponding to nonzero eigenvalues are the features necessary to achieve the same accuracy as in the original feature space. The calculation process uses the training samples themselves, rather than statistics from them, to determine the location of the decision boundary, and then from that, the DBFM. The details are contained in the referenced work.

This method of feature extraction does not have any of the limitations imposed by the previous methods. It works well whether the classes have similar means or not, and it is not limited in any way by the number of classes. However, it is a much longer calculation than either of the previous two and it does not perform well when the training sample sets are small.

Thus there are a variety of methods available by which to reduce the dimensionality of the analysis process, each with its strengths and limitations. It is thus important to understand these strengths and weaknesses in terms of the details of the analysis situation being dealt with.

10.3.5 Ad Hoc Transforms

There have been many other *ad hoc* transforms introduced over the years. They are primarily of value for image enhancement purposes when the data are to be analyzed

by image interpretation techniques (Richardson and Wiegand, 1977). One of the more common ones occurring in the literature is the normalized difference vegetative index (NDVI). Depending on the particular sensor, it is calculated as

$$\text{NDVI} = \frac{\text{IR} - \text{VIS}}{\text{IR} + \text{VIS}} \tag{10-13}$$

where IR is the digital count of an IR band for a pixel, and VIS is the digital count of a visible band. The idea is that the greater the value of the IR return over the visible, the larger the vegetative index. The sum in the denominator is used to normalize the quantity, thus the name. Note that this transformation is sensor-dependent, since the value of NDVI would depend not only upon the net sensor system gain in the channels used, but also upon the specific bandwidth and location as well. It thus provides a relative indication of the vegetation, rather than an absolute measure, and even in this application, it is dependent upon the degree to which the relative size of the IR band response to that in the visible implies vegetation and only vegetation.

10.4 A PROCEDURE FOR ANALYZING MULTISPECTRAL DATA

It should be apparent from the preceeding discussion that the key to a successful analysis of a multispectral data set of any dimensionality is in the definition of the desired classes. As previously noted, for a completely successful analysis, the classes must be exhaustive, separable, and of informational value. Each class must also be defined with adequate precision. Thus the primary characteristic of any procedure for analyzing multispectral data should be to define a set of classes satisfying these conditions as precisely as possible.

The analysis of a multispectral or hyperspectral image data set may follow any of a number of approaches and processing steps. However, a typical generic list of steps might be to proceed in an interactive mode through the following steps, after appropriate offline preprocessing.

1. *Data review.* To gain general familiarity with the data set, its quality, and its general characteristics, a qualitative review is first carried out. This is usually done, at least in part, by viewing the data in multiband B/W and color IR image form. Thus some type of image display is needed.
2. *Class definition.* Quantitative definition of the set of classes meeting the above three conditions is usually accomplished by the analyst labeling, within the data itself, a set of pixels large enough to be adequately representative of each class.
3. *Feature determination.* The specific features to be used in the analysis must be identified or calculated. If the dimensionality is low (about 10 or less), this may simply be a process of selecting an optimal subset of the available spectral bands. For higher-dimensional situations, feature extraction procedures such as those described in Section 10.3 should be used in order to obtain optimal performance without the need for an excessive number of training samples per class.
4. *Analysis.* The specific analysis algorithm is applied to the data set to carry out the desired identification or discrimination.
5. *Evaluation of results.* Both quantitative and qualitative means are used to determine the quality and characteristics of the results obtained. Based on this assessment, it may be desirable to return to one of the previous steps and repeat a portion of the process.

Given that the labeling of the training samples is so key to the success of the analysis, the analyst must decide how many training samples are needed and what information to incorporate to do the labeling.

The optimum number of training samples cannot be precisely determined. If a maximum likelihood classifier involving both first-order and second-order statistics is to be used, there must be at least one sample more than the dimensionality of the data to be classified, in order that the covariance matrix not be singular. However, usually this is not nearly enough. A number of samples many times the number of features is required to achieve good performance. This is why feature selection or feature extraction procedures become so useful in the analysis process.

The question of where the information comes from to label the training samples for each class is also difficult to state precisely. The difficulty arises because of the wide variety of situations possible in the collection of the data, the classes in the ground scene, and the level of prior knowledge the analyst may have about the scene. Though it may at first seem desirable for the process to be automatic, this rarely turns out to be the case. The Earth's surface is a very dynamic and variable place, with the spectral responses of classes of interest in many cases changing on a week-to-week and even a minute-to-minute basis and also on a kilometer-to-kilometer basis. Thus, except for very simple classes, it is not likely that one can successfully apply prior measurements or measurements made at another location to a data set without operator involvement in the process. Further, for most classes of ground cover that are likely to be of interest, the limits of the definition of what is desired for each class can only be established by the analyst during the labeling process.

For example, in an urban area, the specification of a class called "roof" must define the types of roof that are to be included, including what materials are to be included, whether the size or height of buildings will be restricted, whether a parking garage that has cars parked on the "roof" will be included, etc. For an agricultural situation, the definition of a class called "corn" must specfiy how complete the canopy of a corn field must be before the label for the field changes from "bare soil" to "corn" and how much weed content is to be permitted in a pixel. If an area has become "lodged" (laid over) due to storm effects or animal activities, it must still be included in some class definition. All of these kinds of considerations ultimately must be specified in the definition of a class by labeling pixels that are to be considered descriptive and representative of the class the user desires. In all but very special cases, the labeling of training samples will be an interactive process and desirably so. It is also clear that a class cannot be adequately defined by a single spectrum, as is often implied by the term *spectral signature*.

So where does the information come from that the analyst uses to label training samples? Here are some examples.

- *Observations from the ground.* In some circumstances, it may be possible for an observer to be present on the ground at or near the time that the image data is being collected. Observations could be recorded on a perhaps crude map of a small portion of the area, for example.
- *Observations of the ground.* It may not be necessary to actually be on the ground at the time of data collection. Rather, one may be able to identify examples of desired classes from images either of the data itself or of higher-resolution images from another sensor. For example, one might have photographs taken from a low-altitude aircraft (MacDonald and Hall, 1980; Swain and Davis, 1978). The example from Washington, DC, mall data cited below is another illustration where this means was used.

- *Deterministic features from the pixel spectra.* In some cases, it may be possible to label individual pixels based upon features that can be seen in a spectral plot of a pixel. An example where this may be possible is in the use of hyperspectral data for the mapping of minerals based upon specific narrow-band absorption features shown by the minerals. The name *imaging spectroscopy* is sometimes used for this, because of the similarity with that which the chemical spectroscopist does in the laboratory. However, in the case at hand (Hoffbeck and Landgrebe, 1996), instead of attempting to classify the pixels directly, a more robust and accurate performance is obtained by labeling a significant number of pixels by manually examining the spectrum of each, then using the sets of labeled pixels as training for a classifier.

There are certainly other possibilities than these few examples. Clearly, the method by which the needed class definition information is obtained is very case-dependent.

REFERENCES

Cooper, G. R., and McGillem, C. D. 1999. *Probability Methods of Signal and System Analysis.* 3rd edition. Oxford: Oxford University Press.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition.* San Diego, CA: Academic Press.

Hoffbeck, J. P., and Landgrebe, D. A. 1996. Classification of remote sensing images having high spectral resolution. *Remote Sensing of Environment* 57(3):119–126.

Hughes, G. F. 1968. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory* 14(1):55–63.

Kettig, R. L., and Landgrebe D. A. 1976. Computer classification of remotely sensed multispectral image data by extraction and classification of homogeneous objects. *IEEE Transactions on Geoscience Electronics* 14(1):19–26.

Landgrebe, D. A. 1980. The development of a spectral-spatial classifier for earth observational data. *Pattern Recognition* 12(3):165–175.

Landgrebe D. 1997. The evolution of landsat data analysis. Special Issue commemorating the 25th anniversary of the launch of Landsat 1, July 1972. *Photogrammetric Engineering and Remote Sensing* 63(7):859–867.

Landgrebe, D. 1999. Information extraction principles and methods for multispectral and hyperspectral image data. Chapter 1 in *Information Processing for Remote Sensing.* Chen, C. H., ed. River Edge, NJ: World Scientific Publishing.

Lee, C., and Landgrebe, D. A. 1993. Feature extraction based on decision boundaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(4): 388–400.

MacDonald, R. B., and Hall, F. G. 1980. Global crop forecasting. *Science* 208:670–679.

Richards, J. A., and Jia, X. 1999. *Remote Sensing Digital Image Analysis: An Introduction.* 3rd edition, Berlin: Springer-Verlag.

Richardson, A. J., and Wiegand, C. L. 1977. Distinguishing vegetation from soil background information. *Photogrammetric Engineering and Remote Sensing* 43(12):1541–1552.

Swain, P. H., and Davis, S. M., eds. 1978. *Remote Sensing: The Quantitative Approach.* McGraw-Hill.

Swain and Davis, eds. 1978. Large Area Land-Use Inventory. In *Remote Sensing: The Quantitative Approach.* McGraw-Hill pp. 309–314.