Instantaneous Three-Channel Image Processing to Facilitate Instruction

Liping Di and *Donald C. Rundquist*
Center for Advanced Land Management Information Technologies, Conservation and Survey Division, Institute of Agriculture and Natural Resources, University of Nebraska-Lincoln, Lincoln, NE 68588-0517

ABSTRACT: Digital processing of remotely sensed data is typified by both large multispectral datasets and computationally demanding software algorithms. Such a situation leads not only to a major consumption of computing resources but also to time delays in processing. Because there are certain instances, such as during classroom training, when time delays in image processing are intolerable, we have developed a multispectral image-analysis procedure which allows execution of principal components, classification, and other standard analysis techniques on three channels of 864 by 1024 pixels in near-real time using a VAXstation II/GPX. There is little or no difference between the results produced by our instantaneous approach and those produced by conventional computational algorithms.

INTRODUCTION

DIGITAL IMAGE PROCESSING TECHNIQUES have been widely used to analyze remotely sensed data since the early 1970s (e.g., Berstein, 1973; Hoffer, 1973; Bernstein and Ferneyhough, 1975; Goldsbrough, 1977). Several authors have described software and/or procedures for the teaching of image processing techniques (e.g., Kiefer and Gunther, 1983; Jensen, 1983; Eyton, 1983; Williams *et al.*, 1983; Harrington *et al.*, 1986; Merchant, 1989).

One well-known advantage of computer-assisted analysis of image data is the greater signal-level discrimination by the computer than is possible by means of visual interpretation. The majority of all digital image analyses focus on the tone/color of objects in a scene, not only because of the computer's capability for signal-level discrimination but also because it is more difficult to develop software algorithms to address spatial information such as size, shape, texture, pattern, shadow, etc., than it is to write software to manipulate the gray levels of a digital image.

Typical digital procedures developed to exploit multispectral image tone/color include histogram manipulation, multispectral data transformations such as fast-Fourier (FFT), principal components (PCA), intensity-hue-saturation (IHS), image ratios, and multispectral data classifications (both supervised and unsupervised). Most of these techniques are computationally intensive (i.e., time consumptive), and when the number of input bands, the sophistication of the calculation, and/or the number of output classes are increased, the CPU-time requirements increase exponentially. Thus, digital processing of remotely sensed data is typified by both large multispectral datasets and intensive, computationally demanding algorithms.

PROBLEM

In a training environment, whether for formal classroom instruction or for short-course/workshop, the delays typically associated with the processing of image data often become intolerable. Indeed, the learning process is hindered when students are forced to wait several minutes, or even longer, for the processing of large datasets to be completed. One "solution" to the problem is to reduce either the size of the array of pixels or the number of channels being processed, but both the evaluation of results and the appreciation of certain concepts/algorithms are diminished in effectiveness when such an approach is taken. Therefore, it is necessary to develop a mechanism to increase significantly the processing speed of large, multi-channel datasets.

PURPOSE

In previous papers (Di and Rundquist, 1988; Di and Rundquist, 1989), we described a technique for displaying three-channel color-composites on an eight-bit graphics workstation. The purpose of the current paper is to introduce a new multispectral image-analysis procedure to accomplish fast processing on a similar display device while retaining a higher level of accuracy in the result than was possible with our earlier compositing method. Although the new technique introduces minor error in precision, we feel that the tremendous speed of processing more than compensates for this disadvantage; especially when viewed from the perspective of classroom training. Even though we believe the errors to be small, we do not propose our method as a research tool.

HARDWARE

We implemented our approach on specific hardware, the Digital Equipment Corporation VAXStation-II/GPX.* The GPX is a relatively inexpensive engineering-oriented graphics workstation. The CPU time to complete a color-composite display of an image of 864 by 1024 pixels is about one minute (Di and Rundquist, 1989). The GPX has one eight-bit graphics plane with a hardware 24-bit (eight bits for each color) color look-up table of length 256.

GENERAL THEORETICAL BACKGROUND

In order to explain our technique for fast multispectral image processing, we begin with a hypothetical multispectral image containing b bands where every pixel has an eight-bit gray level (0 to 255). Thus, every pixel in the multispectral dataset can be represented as a point in b-dimensional spectral space; in our case, b equals 3 and we call this 3-d space "color space." Therefore, the total number of possible points is $256^{**}b$. Each pixel in our hypothetical space can be summarized by a b-dimensional vector which, for our purposes, we term a "spectral vector." For three channels, we use the term "color vector."

In the case of standard multispectral image-processing algorithms, computations proceed on a pixel-by-pixel basis. For a one-channel image array which is n by n in size, an analysis algorithm must, of course , be executed n by n times. For a digital array n by n in size with b bands, a given calculation

*Any use of trade names and/or trademarks in this publication is for descriptive purposes only and does not constitute endorsement by the University of Nebraska-Lincoln. VAXstation-II/GPX is a trademark of Digital Equipment Corporation.

PHOTOGRAMMETRIC ENGINEERING & REMOTE SENSING,
Vol. 57, No. 3, March 1991, pp. 305–311.

0099-1112/91/5703–305$03.00/0

must still be executed n by n times, but calculation is more difficult because the array is b-dimensional. In such a dataset, the distinct spectral vectors far outnumber the actual physical objects or terrain phenomena (e.g., crop types) found on an image, but are far fewer than the number of pixels in a scene.

Common (remote sensing) logic tells us that the spectral response for the same or similar ground objects/phenomena should also be the same or similar. Pixels corresponding to the same or a similar group of ground objects in a certain part of spectral space can be described by a "mean spectral vector." Because the number of distinct ground objects/phenomena in an image usually is far fewer than the number of distinct spectral vectors, the number of mean spectral vectors is also far fewer.

The spectral vectors for a given land-cover group are either the same or very similar. Therefore, when a processing function is applied to that group, the result as it applies to pixels within the group will also be the same or similar. But, suppose that we apply a processing function to just the mean spectral vector and use that result to represent the result for every pixel within a group. Such an approach will, of course, lead to error. However, because the groups are defined, in the first place, by clustering similar pixels, the error is actually minor (as explained below in the section on "Error Analysis"). In other words, a substantial time savings will result if we apply a given image-processing function to only mean spectral vectors instead of every pixel, as is the case with traditional image-processing methods.

PROCEDURE

According to the contention presented above, we propose a three-step fast multispectral image-processing approach.

Step 1: Creation of "Membership Image"

We begin our procedure by clustering all pixels comprising the hypothetical image of interest into L groups (classes); i.e., we conduct an unsupervised classification. A "membership image" is created by assigning every pixel to groups ranging from 1 to L, and each group can be summarized by a mean spectral vector. The membership image which we have prepared is merely a record of the group number for each pixel. Thus, the input b-band image becomes a one-band membership image, which is used later as an index to a look-up table. The bit length for each pixel comprising this image depends on the number of groups (e.g., eight bits for up to 256 groups). The mean spectral vector for a group is merely an arithmetic average of the spectral vectors comprising the group (Di and Rundquist, 1989).

Suppose we have a b-band image $\vec{\mathbf{I}}$ which is n by n pixels in size. Then, each pixel can be represented by a vector

$$\vec{\mathbf{I}}(i,j) = \{I_1(i,j), I_2(i,j), \ldots, I_b(i,j)\}^{\top}$$
$$i,j = 1, 2, \ldots, n$$

Let there be L data groups denoted where W_k is the group name

$$W_k \quad , \quad k = 1, 2, \ldots, L$$

into which image pixels are to be grouped.

The grouping procedure reduces the original image to L groups, and a membership image is created by

$$M(i,j) = k \quad \text{if } \vec{\mathbf{I}}(i,j) \in W_k$$
$$i,j = 1, 2, \ldots, n$$
$$k = 1, 2, \ldots, L$$

with a mean vector for a specific group k:

$$\overrightarrow{\mathbf{MV}}(k) = \frac{1}{H(k)} \sum_{\vec{\mathbf{I}}(i,j) \in W_k} \vec{\mathbf{I}}(i,j)$$
$$k = 1, 2, \ldots, L$$

where $\overrightarrow{\mathbf{MV}}(k)$ is a b-dimensional vector and $H(k)$ is the frequency of the kth group in the membership image. So, H is the histogram of the membership image. If required, we can calculate the covariance matrix for the kth group:

$$\mathbf{CV}(k) = \frac{1}{H(k)} \sum_{\vec{\mathbf{I}}(i,j) \in W_k} (\vec{\mathbf{I}}(i,j) - \overrightarrow{\mathbf{MV}}(k)) (\vec{\mathbf{I}}(i,j) - \overrightarrow{\mathbf{MV}}(k))^{\top}$$
$$k = 1, 2, \ldots, L$$

where $\mathbf{CV}(k)$ is a b by b covariance matrix. The detailed method to create a membership image and mean spectral vectors can be found in Di and Rundquist (1989).

From the mean vectors $\overrightarrow{\mathbf{MV}}$ and membership image, we can derive the estimated description of the original image:

$$\vec{\mathbf{I}}(i,j) = \overrightarrow{\mathbf{MV}}(M(i,j))$$
$$i,j = 1, 2, \ldots, n$$

The difference between the original image and the estimated image is

$$\vec{\mathbf{E}}(i,j) = \vec{\mathbf{I}}(i,j) - \vec{\mathbf{I}}(i,j)$$
$$i,j = 1, 2, \ldots, n$$

where, $\vec{\mathbf{E}}$ is an error image. At this point, we have derived the membership image M, the mean vector $\overrightarrow{\mathbf{MV}}$ and the histogram H of the membership image. This first step is the most time-consuming of the procedure because the calculations are based upon the entire original image.

Step 2: Application of Functions

In the second step of our procedure, we apply an image processing function to the L mean spectral vectors instead of every pixel in the original image, and L outputs are obtained. If L is 256 and the image size is 512 by 512 pixels, the computational time is 1024 times (512•512/256) faster than would be the case for a standard pixel-by-pixel computation.

Because the length of $\overrightarrow{\mathbf{MV}}$ is L, any selected operation must be executed L times on $\overrightarrow{\mathbf{MV}}$. We denote the selected operation (function) as F. Some functions may use histogram information; some may use the group covariance matrices. Then, the processing result (vector table), $\vec{\mathbf{R}}$, is the function of $\overrightarrow{\mathbf{MV}}$, H, and/or $\mathbf{CV}$, which can be written as

$$\vec{\mathbf{R}}(k) = F(\overrightarrow{\mathbf{MV}}(k), H(k), \mathbf{CV}(k))$$
$$k = 1, 2, \ldots, L$$

The dimension of $\vec{\mathbf{R}}(k)$ varies from 1 to b depending on the choice of analysis operation. For example, the dimension of $\vec{\mathbf{R}}(k)$ is 1 when the analysis is a classification alogrithm. If PCA is the function selected, however, the dimension of $\vec{\mathbf{R}}(k)$ is less than or equal to b depending on the number of component images desired.

Step 3: Assignment of a Result to Every Pixel

Once $\vec{\mathbf{R}}$ is obtained, the resulting image is formed by using the formula

$$\vec{\mathbf{G}}(i,j) = \vec{\mathbf{R}}(M(i,j))$$
$$i,j = 1, 2, \ldots, n$$

where $\vec{G}$ is a pixel vector of the resulting image at location (i,j). The dimension of $\vec{G}$ depends on the dimension of $\vec{R}$. Here, $\vec{R}$ actually is a lookup table.

The membership image is used as an index to the L outputs and, thus, every pixel has an output. This final step is completed by use of the look-up table and requires little time if realized by software (and can be ignored if realized by hardware).

It is clear that once the membership image, the mean vector, and the histogram of membership are formed, any multispectral image-analysis operation can be applied to the mean vector and histogram instead of the whole original image. Different results can be obtained by merely changing the algorithm as applied to $\vec{MV}$ and H. The membership image does not have to be recalculated in any later processing. Such a procedure allows for very fast processing, due largely to the fact that the membership image remains in refresh memory and our procedure involves minimal computation and I/O. If the table-looking operation is not considered, the speed gain of processing for an n by n image array can be calculated using

$$g_s = \frac{n*n}{L}$$

where g is speed gain and L is the length of look-up table. We also do not include the time for creating the membership image and mean spectral vectors in this calculation because those computations occur as part of the ordinary image display when using our hardware configuration (Di and Rundquist, 1989).

ERROR ANALYSIS

As was demonstrated in the previous section, we can approximate the original image by using the group number of a pixel in the membership image as an index to the mean spectral vectors. Differences between the approximated image and the original image are errors caused by the first step in our approach. Therefore, the original image may be expressed as

$$\vec{I}(i,j) = \hat{\vec{I}}(i,j) + \vec{E}(i,j)$$
$$i,j = 1, 2, \ldots, n$$

where $\vec{I}(i,j)$ is a b-dimensional spectral vector at pixel (i,j), $\hat{\vec{I}}(i,j)$ is $\vec{I}(i,j)$'s estimation and is one of the mean spectral vectors, and $\vec{E}(i,j)$ is the error vector between $\vec{I}(i,j)$ and $\hat{\vec{I}}(i,j)$. Suppose we want to apply an image processing function F to the original image. Thus,

$$F\{\vec{I}(i,j)\} = F\{\hat{\vec{I}}(i,j) + \vec{E}(i,j)\}$$
$$i,j = 1, 2, \ldots, n$$

If the function is derivable, the above formula can be written approximately as

$$F\{\vec{I}(i,j)\} \approx F\{\hat{\vec{I}}(i,j)\} + F'\{\hat{\vec{I}}(i,j)\} * \vec{E}(i,j)$$
$$i,j = 1, 2, \ldots, n$$

where $F\{\vec{I}(i,j)\}$ is the exact result of the original image, $F\{\hat{\vec{I}}(i,j)\}$ is the result of our approach, and $F'\{\hat{\vec{I}}(i,j)\}$ is the derivative value of an image-processing function at vector $\hat{\vec{I}}(i,j)$. Then, the average error brought by our approach is

$$\bar{\xi} = \frac{1}{n*n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|F\{\vec{I}(i,j)\} - F\{\hat{\vec{I}}(i,j)\}\right|$$

$$= \frac{1}{n*n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|F'\{\hat{\vec{I}}(i,j)\} * \vec{E}(i,j)\right|$$

$$= \frac{1}{n*n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|\left\{\frac{\partial F}{\partial \hat{I}_1(i,j)}, \frac{\partial F}{\partial \hat{I}_2(i,j)}, \ldots, \frac{\partial F}{\partial \hat{I}_b(i,j)}\right\} * \{e_1(i,j), e_2(i,j), \ldots, e_b(i,j)\}^T\right|$$

$$\le \frac{1}{n*n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\left|\frac{\partial F}{\partial \hat{I}_1(i,j)}e_1(i,j)\right| + \left|\frac{\partial F}{\partial \hat{I}_2(i,j)}e_1(i,j)\right| + \ldots + \left|\frac{\partial F}{\partial \hat{I}_b(i,j)}e_b(i,j)\right|\right)$$

Many multispectral image-processing functions are linear, such as FFT, PCA, IHS, etc. For these functions, the derivative values are constant for the different $\hat{I}_i(i,j)$. Therefore, the upper limit of the average error brought by our method for linear transform function is

$$\bar{\xi} \le \left|\frac{\partial F}{\partial \hat{I}_1(i,j)}\right| * \bar{e}_1 + \left|\frac{\partial F}{\partial \hat{I}_2(i,j)}\right| * \bar{e}_2 + \ldots + \left|\frac{\partial F}{\partial \hat{I}_b(i,j)}\right| * \bar{e}_b$$

Because the average error brought by the first step of our approach is a mere one grey level (Di and Rundquist, 1989), e_1, e_2, ... , and e_b are equal to 1. Then,

$$\bar{\xi} \le \left|\frac{\partial F}{\partial \hat{I}_1(i,j)}\right| + \left|\frac{\partial F}{\partial \hat{I}_2(i,j)}\right| + \ldots + \left|\frac{\partial F}{\partial \hat{I}_b(i,j)}\right|$$

Because the result of a linear-transform function always retains the same gray scale as the original, the summation of the coefficients is always near 1 and the upper limit of the average error will be near one grey level. For example, the first component of a PCA transform, described later as Figure 3, can be obtained by (Table 2):

$$\text{PCA}_1(i,j) = 0.0393*\hat{I}_1(i,j) + 0.0736*\hat{I}_2(i,j) - 0.9965*\hat{I}_3(i,j) + C$$

where C is a constant. If the average error brought by the first step of our approach is equal to one grey level, the upper limit of the average result error brought by our approach in the first component of PCA is

$$\bar{\xi} \le |0.0393| + |0.0736| + |-0.9965| = 1.1094$$

The error analysis described above is based on applying a derivable function to a multispectral image. However, image classifications are not always entirely derivable. For a given pixel, image classifications first compute similarity values between the pixel and classes and then assign the pixel to the class with the highest similarity value. Although the function used to calculate the similarity values is derivable, the actual assigning of a pixel to a class with the highest similarity value is not derivable (i.e., cannot be differentiated). Therefore, the above formulas can be used to analyze the error of similarity values brought by our approach, but the error of the final result cannot be predicted. The classification error brought by our approach is determined both by error inherent in the first step of our approach and by the separability of classes. If there are no errors caused by the first step, then the final result will be error free.

High separability between two classes means that there is minimal overlap between the distributions of classes. A highly separable class will have large differences between first- and second-highest similarity values, which means that the small change in similarity value caused by the error of first step of our approach will not result in the switch of a class number between the first- and the second-highest similarity values. Low

separability means that this switch may occur, and, therefore, results from our approach may differ from those of traditional methods. Our approach produces the most errors when it treats those pixels whose first- and second-highest similarity values are nearly equal (i.e., near the decision boundary between any two classes). Of course, we are considering our classification error in terms of results produced through the traditional approach, and it does not necessarily follow that results obtained from our approach are lower in classification accuracy than the traditional approach with respect to actual ground truth. While the highest occurrences of error with our approach are in the overlapped part of two classes, it is worth noting that classification algorithms always have a higher mis-classification rate in this part of spectral space.

As noted above, the first step of our approach actually is an unsupervised classification of the raw data into a maximum of L classes (in our case, hardware limits us to 256 data groups) (Di and Rundquist, 1989). For a real application, the number of classes is rarely over 30. The second step of our approach continuously groups L classes into a user-specified number of information classes if/when the user selects either a supervised or unsupervised classification function. A large number of data classes insures high classification accuracy in our approach.

According to above analysis, we find that any error in the result tends to be caused by the first step of our approach. To reduce the error, two things are critical: (1) the number of groups (L); and (2) the algorithm which creates both the membership image and mean spectral vectors. If the number of groups (L) equals the number of distinct spectral vectors in a given image, we merely use the distinct spectral vectors as mean spectral vectors and the result is error-free. But the number of the distinct spectral vectors in a given image is usually very large. Correspondingly, the processing time in the second step of our approach will increase. For a fixed L, an algorithm is required which groups a given multispectral image into L groups with minimum error, but this algorithm has no known fast solution (Gray *et al.*, 1980).

IMPLEMENTATION

To test our approach, we implemented the technique on a Digital Equipment Corporation VAXstation II/GPX, an engineering workstation which runs both ELAS and LAS software at the Center for Advanced Land Management Information Technologies (CALMIT), University of Nebraska-Lincoln (Junkin *et al.*, 1981; Goddard Space Flight Center, 1987; Di and Rundquist, 1988).

We developed a method which compresses a three-band (eight-bits/band) image to one eight-bit composite image and uses a look-up table to form the color-composite image on a GPX (Di and Rundquist, 1989). The hardware feature of the color look-up table makes the "looking operation" very fast; it only takes one refresh period (1/30 second) to convert an entire image (in refresh memory) to the desired corresponding image on screen.

As described above, when a three-band image is displayed on the GPX, the membership image M, the histogram of membership image H, and the mean vector $\vec{\mathbf{MV}}$ have all been calculated and stored. The length of the hardware lookup table is 256, so the number of groups in the membership image is also 256.

The flow chart for multispectral image processing using our procedure on a GPX is depicted as Figure 1. The first part of the illustration is the display procedure for color-compositing, while the second part is the procedure of instantaneous three-band image processing.

To examine the speed of our technique, consider a hypothetical situation where we wish to process an image of 864 by 1024 pixels on a GPX. Thus, g_s equals 3456(864*1024/256), which means the analysis of a three-band image sized 864 by 1024 which normally would consume 57.6 minutes of CPU time on a VAXstation-II will require only 1 second with our method.

Several functions were executed on a test image in order to compare the speed and quality of results of our method with traditional procedures. Because our processing proceeds so quickly, it is difficult to record the expenditure of time. Therefore, we cannot provide the reader with data on CPU usage in deriving the results shown in the example which follows. Suffice it to say that results appear on the display screen instantaneously when a keyboard command is given.

The test array (200 by 200 pixels) consists of three channels of airborne MSS (Daedalus AADS-1260) data acquired on 20 May 1984 over Clay County, Nebraska (Plate 1). First, PCA was run on the test dataset. With a PCA transformation, the covariance matrix and the mean of every band for the image must be calculated first. With the conventional method, the covariance matrix and the means can be obtained by using

$$\vec{\mathbf{M}} = \frac{1}{n*n} \sum_{i=1}^{n} \sum_{j=1}^{n} \vec{\mathbf{I}}(i,j)$$

and

$$\mathbf{CV} = \frac{1}{n*n} \sum_{i=1}^{n} \sum_{j=1}^{n} (\vec{\mathbf{I}}(i,j) - \vec{\mathbf{M}}) * (\vec{\mathbf{I}}(i,j) - \vec{\mathbf{M}})^{\top}$$

Notice that **CV** is the covariance matrix for the whole image; it is not the $\mathbf{CV}(i)$ which is the covariance matrix of the ith data group (discussed above). It is easy, using our technique, to obtain the mean and covariance matrix from the histogram for the membership image (H) and the mean vector ($\vec{\mathbf{MV}}$):

$$\vec{\mathbf{M}} = \frac{1}{n*n} \sum_{k=1}^{256} H(k) * \vec{\mathbf{MV}}(k)$$

and

$$\mathbf{CV} = \frac{1}{n*n} \sum_{k=1}^{256} H(k) * (\vec{\mathbf{MV}}(k) - \vec{\mathbf{M}}) * (\vec{\mathbf{MV}}(k) - \vec{\mathbf{M}})^{\top}$$

In our example, the number of bands is three; $\vec{\mathbf{M}}$ is a three-dimensional vector representing the means of three bands and the **CV** is a matrix 3 by 3 in size.

Once the **CV** has been obtained, the transformation matrix **T** can be formed by the eigenvectors solved from **CV**. **T** is a 3 by 3 matrix which is used to transform the three-band image into a three-component image. The lookup table can be calculated by

$$\vec{\mathbf{R}}(k) = \mathbf{T} * (\vec{\mathbf{MV}}(k) - \vec{\mathbf{M}})$$

$$k = 1, 2, \ldots, 256$$

where $\vec{\mathbf{R}}(i)$ is a three-dimensional vector which represents the three component values of pixels belonging to the ith group. Of course, values of $\vec{\mathbf{R}}(i)$ are real numbers which should be converted to eight-bit integers for display and other purposes.

Figure 2 is the first component image (of Plate 1) created by the traditional method, and Figure 3 is the first component image (also of Plate 1) created by out method. Visual inspection leads us to conclude that there are no differences between two images. Figure 4 is the difference image between Figure 2 and Figure 3 (multiplied by 100 to convert small real-number differences to displayable integers). Table 1 summarizes the mean vectors and the eigenvalues for both methods, while Table 2 contains the covariance and transformation matrices. The statistics seem very similar.

In order to further compare the traditional and our procedure, two superviseal classifications, minimum distance and Bayes

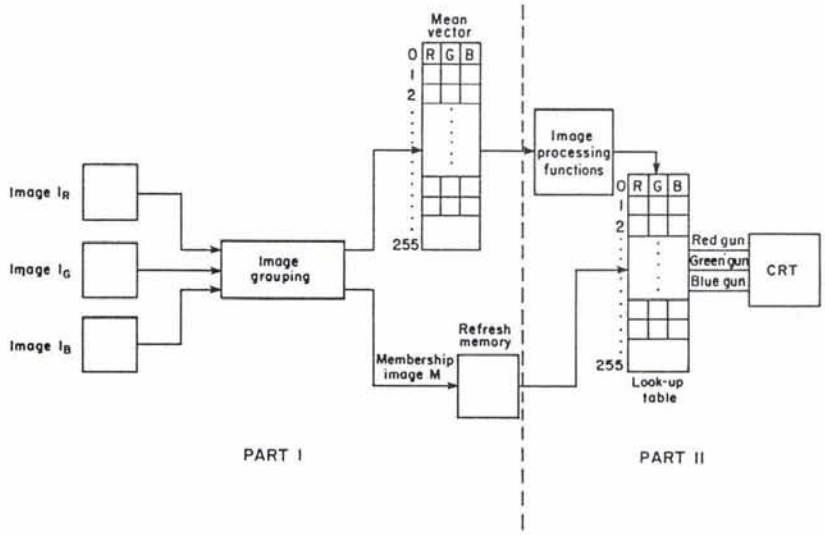

FIG 1. Schematic illustrating fast image processing using one eight-bit graphic plane.

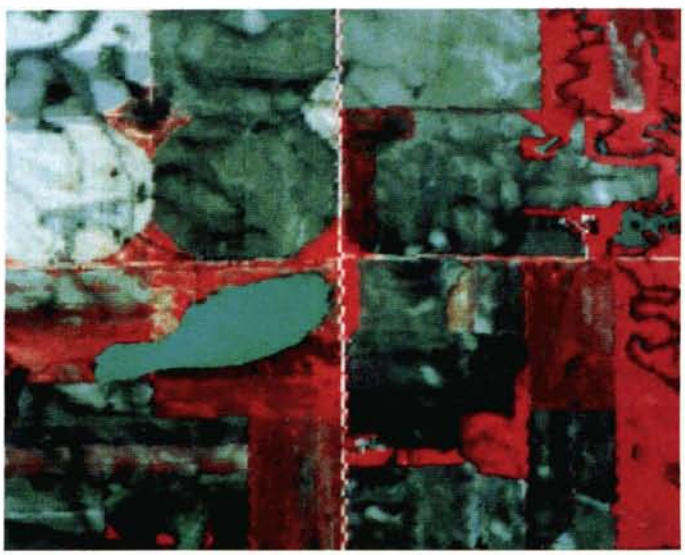

PLATE 1. Histogram-equalized color-composite image of test dataset.

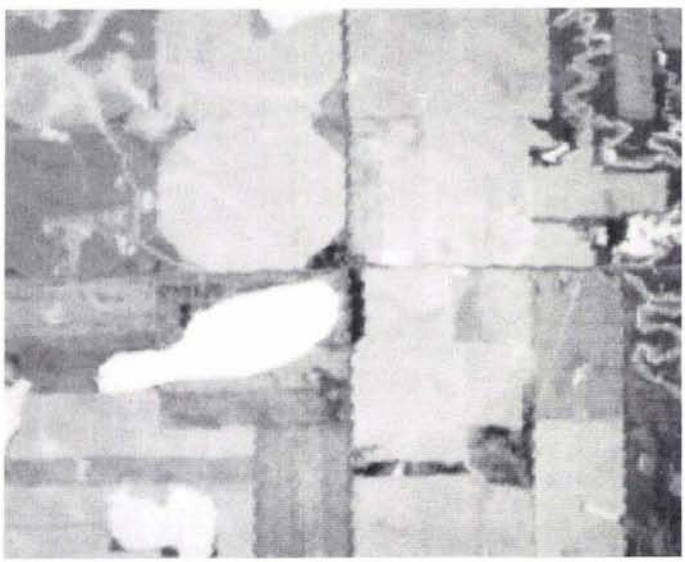

FIG 2. First-component image of Plate 1 created by the conventional method.

(maximum-likelihood) were conducted on the test image (Plate 1). Five classes of objects have been identified by visual inspection and knowledge of the site: (1) water, (2) vegetation, (3) non-vegetated farmland, (4) wet soil and (5) wetland vegetation. The same size and location of training areas were used for both the traditional and new procedures.

Figure 5 is the difference image between the two procedures for the minimum distance classification. The statistics (Table 3) show that the new and traditional methods classified most of the pixels (in fact, 98 percent) in the same class. Thus, it can be said there is no difference between the two methods for minimum-distance classification.

Figure 6 is the difference image between the two procedures for the maximum-likelihood classification. We can detect some differences on Figure 6, especially for the water class (compare to Plate 1). The statistical results (Table 4) again show general classification agreement (90 percent) between the traditional and new methods. This small discrepancy is no doubt the result of using the mean vector to represent all pixels in each of the groups, which, in turn, leads to slight differences in the convariance matrix. Bigger (or mulit-polygon) training areas would probably alleviate this difficulty because a given information class could consist of several data classes. Another solution to the prolem would be to calculate the covariance matrix within the data class while the original image is being grouped and keep the matrices for later processing. In our case, it would mean that we need to keep 256 matrices of 3 by 3. From the standpoint of memory, this would not be a big burden, but it would increase computing time slightly during the formation of data classes. The latter adjustment would solve the covariance matrix problem completely.

CONCLUSION

The method described in this paper increases processing speed from several hundred to several thousand times depending on the size of the image under consideration. In our implementation on a VAXstation II/GPX, almost all multispectral analyses involving three bands of 864 by 1024 pixels can be done within one second. As presented above, the processing precision achieved by this method is relatively high, with little or no

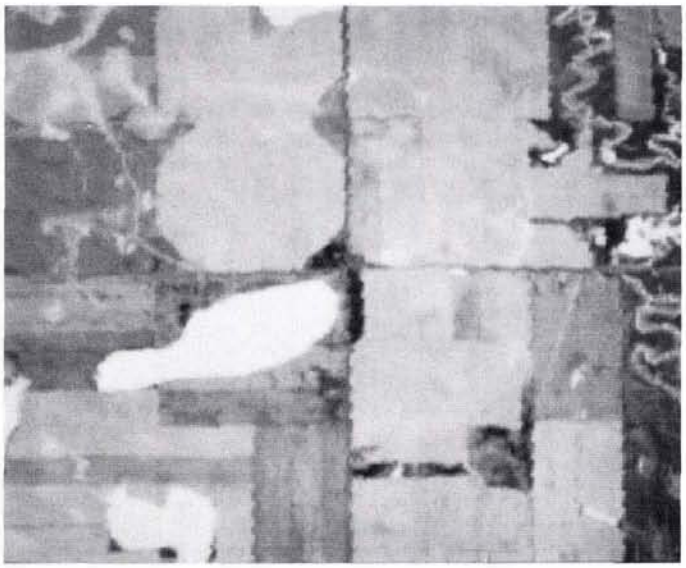

FIG 3. First-component image of Plate 1 created by the look-up table method.

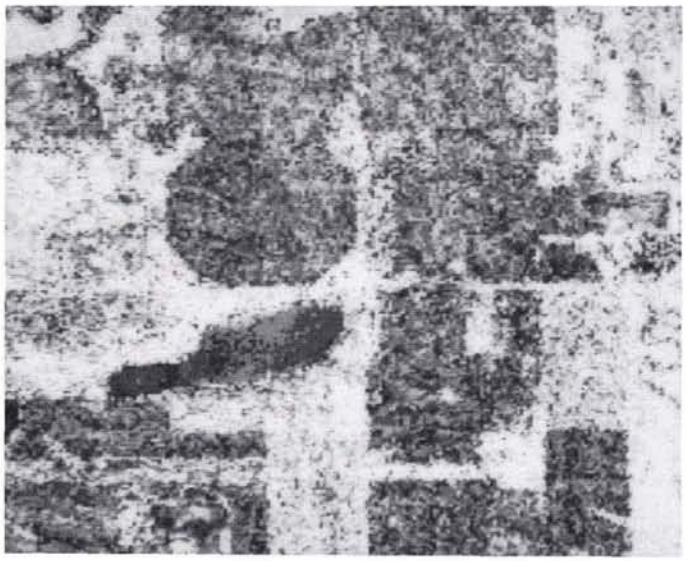

FIG 4. The difference image between Figure 2 and Figure 3 (multiplied by 100).

TABLE 1. MEANS AND EIGENVALUES: CONVENTIONAL VS. LOOK-UP TABLE METHOD

Method	Means			Eigenvalues		
	Ch1	Ch2	Ch3			
Conventional	80.0	99.9	64.6	366.8	197.7	0.781
Look-up table	80.0	99.9	64.6	383.3	195.8	0.130

TABLE 2. COVARIANCE AND TRANSFER MATRICES: CONVENTIONAL VS. LOOK-UP TABLE METHOD

	Covariance Matrix			Transfer Matrix		
Conventional	46.5	83.5	−6.4	0.0389	0.0728	−.9966
	83.5	153.2	−12.3	−.4788	−.8740	−.0825
	−6.4	−12.3	365.6	−.8771	0.4803	0.0009
Look-up table	45.6	83.0	−6.4	0.0393	0.0736	−.9965
	83.0	151.4	−12.4	−.4793	−.8737	−.0834
	−6.4	−12.4	362.1	−.8768	0.4809	0.0009

FIG 5. The difference image between the conventional method and the look-up table method for a minimum distance classification.

TABLE 3. PIXEL-BY-PIXEL CLASSIFICATION CROSSTABULATION: CONVENTIONAL VS. LOOK-UP TABLE METHOD (MINIMUM DISTANCE)

	Look-up Table				
Cls	1	2	3	4	5
1	2225			6	
2		4808	13	23	10
3		2	3263	148	
4	81	12	31	22736	145
5	7			504	5986

difference between the results produced by traditional and new methods.

In the experiment, our method is realized on a particular hardware display device, but the configuration of the GPX is similar to many popular microcomputers and workstations. However, our technique can be implemented totaly by software. But, if done with software, both the membership image and look-up table must be stored in inner memory instead of in refresh memory. Another difference is that the looking operation must be done by software instead of hardware; but, even so, such an operation will require little computation time. In fact, there are some benefits when the new method is realized completely by software; there are no limitations on the image size, the number of bands, and the length of lookup table (a hardware limitation).

As we mentioned before, the processing precision depends both on the performance of the grouping algorithm and the

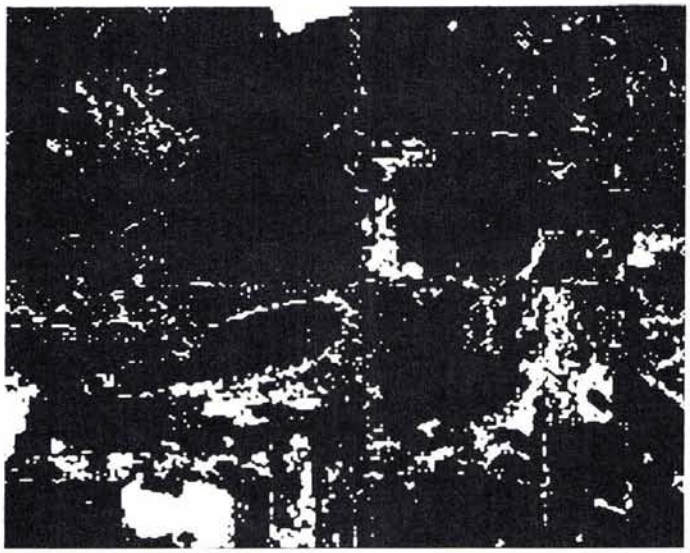

FIG 6. The difference image between the conventional method and the look-up table method for a maximum-likelihood classification.

TABLE 4. PIXEL-BY-PIXEL CLASSIFICATION CROSSTABULATION: CONVENTIONAL VS. LOOK-UP TABLE METHOD (MAXIMUM LIKELIHOOD)

	Look-up Table				
Cls	1	2	3	4	5
1	1187				
2	905	10563	171	447	1624
3		93	3870	119	
4		556	183	19609	50
5		1			621

length of the look-up table. However, an increase in the latter will not only slow the processing but also will increase the requirement of the memory for the membership image. When the length of the look-up table is fixed, the performance of the grouping algorithm will dictate the processing precision of our method. Further research is required to find a fast, optimal grouping algorithm.

We feel that the software package outlined in this paper gives us the speed and sufficient precision we need for teaching techniques of digital image processing. Students receive instantaneous response after keyboard input, thus enriching the training experience significantly.

REFERENCES

Bernstein, R., 1973. Scene correction (precision processing) of ERTS sensor data using digital image processing techniques. *Third ERTS Symposium*, Vol. 1, Section A, NASA SP-351, pp. 1909–1928.

Bernstein, R., and D. G. Ferneyhough, Jr., 1975. Digital image processing. *Photogrammetric Engineering & Remote Sensing*, 41:12, pp. 1465–1476.

Di, L., and D. C. Rundquist, 1988. Color-composite image generation on an eight-bit graphics workstation. *Photogrammetric Engineering & Remote Sensing*, 54:12, pp. 1745–1748.

———, 1989. A 'Classification-Based Method' for color-composite image generation on an eight-bit graphics workstation. in review, *Geocarto International*.

Eyton, J. R., 1983. A hybrid image classification instructional package. *Photogrammetric Engineering & Remote Sensing*, 49:8, pp. 1175–1181.

Goddard Space Flight Center, 1987. *LAS (Land Analysis System) User's Manual (Version 4.0)*. National Aeronautics and Space Administration, Greenbelt, Maryland.

Goldsbrough, P. F., 1977. Digital processing of analog thermal infrared scanner data. *Photogrammetric Engineering & Remote Sensing*, 43:2, pp. 145–153.

Gray, R. M., J. C. Kieffer, and Y. Linde, 1980. Locally optimal block quantizer design, *Information and Control* 45, pp. 178–198.

Harrington, J. A., Jr., K. F. Cartin, and R. Lougeay, 1986. The Digital Image Analysis System (DIAS): microcomputer software for remote sensing education. *Photogrammetric Engineering & Remote Sensing*, 52:4, pp. 545–550.

Hoffer, R. M., 1973. Techniques for computer-aided analysis of ERTS-1 data, useful in geologic, forest and water resource surveys. *Third ERTS Symposium*, Vol. 1, Section A, NASA SP-351, pp. 1687–1708.

Jensen, J. R., 1983. Educational image processing: an overview. *Photogrammetric Engineering & Remote Sensing*, 49:8, pp. 1151–1157.

Junkin, B. G., *et al.*, 1981. *ELAS (Earth Resources Laboratory Applications software)*. National Aeronautics and Space Administration, National Space Technology Laboratories, Earth Resources Laboratory, Report 183.

Kiefer, R. W., and F. J. Gunther, 1983. Digital image processing using the Apple II microcomputer. *Photogrammetric Engineering & Remote Sensing*, 49:8, pp. 1167–1174.

Merchant, J. W., 1989. Teaching digital data analysis strategies. *Current Trends in Remote Sensing Education* (Nellis, D., R. Lougeay, and K. Lulla, eds), Hong Kong: Geocarto International Centre, pp. 65–80.

Williams, T. H. L., C. Gunn, and J. Siebert, 1983. Instructional use of a mainframe interactive image analysis system. *Photogrammetric Engineering & Remote Sensing*, 49:8, pp. 1159–1165.

(Received 11 January 1990; accepted 12 April 1990; revised 11 June 1990)