The quantity $\Sigma(BV_{ik} \times BV_{il})$ is called the *uncorrected sum of products*. The relationship of $SP_{kl}$ to the *sum of squares* (SS) can be seen if we take $k$ and $l$ as being the same; for example,

$$SP_{kk} = \frac{\sum_{i=1}^{n} BV_{ik}^2 - \left(\sum_{i=1}^{n} BV_{ik}\right)^2}{n} = SS_k \tag{4-3}$$

Just as simple variance is calculated by dividing the sums of squares, SS, by $(n - 1)$, covariance is calculated by dividing SP by $(n - 1)$. Therefore, the covariance between brightness values in bands $k$ and $l$, $Cov_{kl}$, is equal to (Davis, 1973)

$$Cov_{kl} = \frac{SP_{kl}}{n-1} \tag{4-4}$$

$$= \frac{n\sum_{i=1}^{n}(BV_{ik} \times BV_{il}) - \sum_{i=1}^{n} BV_{ik} \sum_{i=1}^{n} BV_{il}}{n(n-1)} \tag{4-5}$$

The sums of products (SP) and sums of squares (SS) can be computed for all possible combinations of the four spectral variables in Table 4–1. These data may then be arranged in a $4 \times 4$ variance-covariance matrix as shown in Table 4–3. All elements in the matrix not on the diagonal have one duplicate (e.g., $Cov_{1,2} = Cov_{2,1}$, so that $Cov_{kl} = Cov_{lk}$).

**TABLE 4–3** Format of a Variance-Covariance Matrix

| | Band 1 | Band 2 | Band 3 | Band 4 |
|---|---|---|---|---|
| Band 1 | $SS_1$ | $Cov_{1,2}$ | $Cov_{1,3}$ | $Cov_{1,4}$ |
| Band 2 | $Cov_{2,1}$ | $SS_2$ | $Cov_{2,3}$ | $Cov_{2,4}$ |
| Band 3 | $Cov_{3,1}$ | $Cov_{3,2}$ | $SS_3$ | $Cov_{3,4}$ |
| Band 4 | $Cov_{4,1}$ | $Cov_{4,2}$ | $Cov_{4,3}$ | $SS_4$ |

The computation of variance for the diagonal elements of the matrix and covariance for the off-diagonal elements of the data shown in Table 4–1 are provided in Table 4–4. The manual computation of the covariance between band 1 and band 2 is shown in Table 4–5.

To estimate the degree of interrelation between variables in a manner not influenced by measurement units, the *correlation coefficient*, $r$, is used (Cooley and Lohnes, 1971). Correlation is the ratio of the covariance of two variables to the product of their standard deviations, thus:

$$r_{kl} = \frac{Cov_{kl}}{s_k s_l} \tag{4-6}$$

Pictometry EX2002 (Part 4 of 13)
Roofr v. Pictometry
IPR2023-00436



OFEK06751

**TABLE 4–4** Variance-Covariance Matrix of the Sample Data

| | Band 1 | Band 2 | Band 3 | Band 4 |
|---|---|---|---|---|
| Band 1 | 562.5 | | | |
| Band 2 | 135 | 264.8 | | |
| Band 3 | 718.75 | 275.25 | 1007.5 | |
| Band 4 | 537.5 | 64 | 663.75 | 570 |

Because the correlation coefficient is a ratio, it is a unitless number. Covariance may equal but cannot exceed the product of the standard deviation of its variables, so correlation ranges from +1 to −1. In our example, a correlation coefficient of +1 indicates a positive, perfect relationship between the brightness values in two of the bands (i.e., as one band's pixels increase in value, the other band's values also increase in a perfectly systematic fashion). Conversely, a correlation coefficient of −1 would indicate that one band was inversely related to the other (i.e., as brightness values in one band increased in value, corresponding pixels in the other band would systematically decrease in value). A continuum of less-than-perfect relationships exists between correlation coefficients of −1 and +1. A correlation coefficient of zero suggests that there is no linear relationship between the two bands of remote sensor data. The between-band correlations are usually stored in a correlation matrix such as the one shown in Table 4–6, which contains the between-band correlations of our sample data. Usually, only the coefficients below the diagonal are displayed because the diagonal terms are 1.0 and the terms above the diagonal are redundant.

Sample brightness values of band 1 are highly correlated with those of bands 3 and 4. A high correlation suggests that there is a substantial amount of redundancy in the information content among these bands. Perhaps one or more bands could be deleted from the analysis to reduce subsequent computation. Conversely, the low

**TABLE 4–5** Computation of Covariance Between Bands 1 and 2 of the Sample Data

| Band $1^2$ | Band 1 | Band 1 × Band 2 | Band 2 | Band $2^2$ |
|---|---|---|---|---|
| 16,900 | 130 | 7,410 | 57 | 3,249 |
| 27,225 | 165 | 5,775 | 35 | 1,225 |
| 10,000 | 100 | 2,500 | 25 | 625 |
| 18,225 | 135 | 6,750 | 50 | 2,500 |
| 21,025 | 145 | 9,425 | 65 | 4,225 |
| 93,375 | 675 | 31,860 | 232 | 11,824 |

Where $SP_{12} = (31{,}860) - \frac{(675)(232)}{5} = 540$

$$Cov_{12} = \frac{540}{4} = 135$$



OFEK06752

**TABLE 4–6** Correlation Matrix of the Sample Data

| | Band 1 | Band 2 | Band 3 | Band 4 |
|---|---|---|---|---|
| Band 1 | — | | | |
| Band 2 | 0.35 | — | | |
| Band 3 | 0.95 | 0.53 | — | |
| Band 4 | 0.94 | 0.16 | 0.87 | — |

correlation between band 2 and all other bands suggests that this band provides some type of information not found in the other bands. More sophisticated methods of selecting the most useful bands will be described in later sections.

The STATISTICS program listed in Appendix A can compute both univariate and multivariate statistics for up to eight bands of remotely sensed data at one time, then display the data in tables similar to those shown. The results of running this program on the Charleston, S.C., TM scene are summarized in Table 4–7. The blue band (1) exhibits the greatest range of brightness values (i.e., from 51 to 242). This

**TABLE 4–7** Statistics for the Charleston, S.C., Thematic Mapper Scene Composed of Seven Bands Each 240 × 256 Pixels

| Band number: | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| μm: | 0.45–0.52 | 0.52–0.60 | 0.63–0.69 | 0.76–0.90 | 1.55–1.75 | 2.08–2.35 | 10.4–12.5 |
| | | | *Univariate statistics* | | | | |
| Mean | 64.8 | 25.6 | 23.7 | 27.3 | 32.4 | 15.0 | 110.6 |
| Std. dev. | 10.05 | 5.84 | 8.30 | 15.76 | 23.85 | 12.45 | 4.21 |
| Variance | 100.9 | 34.1 | 68.8 | 248.4 | 568.8 | 154.9 | 17.78 |
| Minimum | 51 | 17 | 14 | 4 | 0 | 0 | 90 |
| Maximum | 242 | 115 | 131 | 105 | 193 | 128 | 130 |
| | | | *Variance-covariance matrix* | | | | |
| 1 | 100.93 | | | | | | |
| 2 | 56.60 | 34.14 | | | | | |
| 3 | 79.43 | 46.71 | 68.83 | | | | |
| 4 | 61.49 | 40.68 | 69.59 | 248.40 | | | |
| 5 | 134.27 | 85.22 | 141.04 | 330.71 | 568.84 | | |
| 6 | 90.13 | 55.14 | 86.91 | 148.50 | 280.97 | 154.92 | |
| 7 | 23.72 | 14.33 | 22.92 | 43.62 | 78.91 | 42.65 | 17.78 |
| | | | *Correlation matrix* | | | | |
| 1 | 1.00 | | | | | | |
| 2 | 0.96 | 1.00 | | | | | |
| 3 | 0.95 | 0.96 | 1.00 | | | | |
| 4 | 0.39 | 0.44 | 0.53 | 1.00 | | | |
| 5 | 0.56 | 0.61 | 0.71 | 0.88 | 1.00 | | |
| 6 | 0.72 | 0.76 | 0.84 | 0.76 | 0.95 | 1.00 | |
| 7 | 0.56 | 0.58 | 0.66 | 0.66 | 0.78 | 0.81 | 1.00 |


OFEK06753

range is not surprising because atmospheric scattering of blue wavelength energy is expected. The reflective and middle-infrared bands (4, 5, and 6) all have minimums near or at zero. These values are low because much of the Charleston scene is composed of open water which absorbs most of the incident near- and middle-infrared radiant flux, thereby causing low reflectance in these bands. Bands 1, 2, and 3 are all highly correlated with one another ($r > 0.95$), indicating that there is substantial redundant spectral information in these channels. Although not to the same degree, there is also considerable redundancy between the reflective and middle-infrared bands (4, 5, and 6), as they exhibit correlations ranging from 0.76 to 0.95. Not surprisingly, the lowest correlations occur when a visible band is compared with an infrared band, especially bands 1 and 4 ($r = 0.39$). In fact, band 4 is the least redundant infrared band when compared with all three visible bands (1, 2, and 3). For this reason, band 4 (0.76 to 0.90 $\mu$m) will be used as an example throughout much of the text. The thermal-infrared band 7 (10.4 to 12.5 $\mu$m) is not highly correlated with any of the other bands, as it records emitted rather than reflected spectral radiation.

## THE HISTOGRAM AND ITS SIGNIFICANCE TO DIGITAL IMAGE PROCESSING OF REMOTE SENSOR DATA

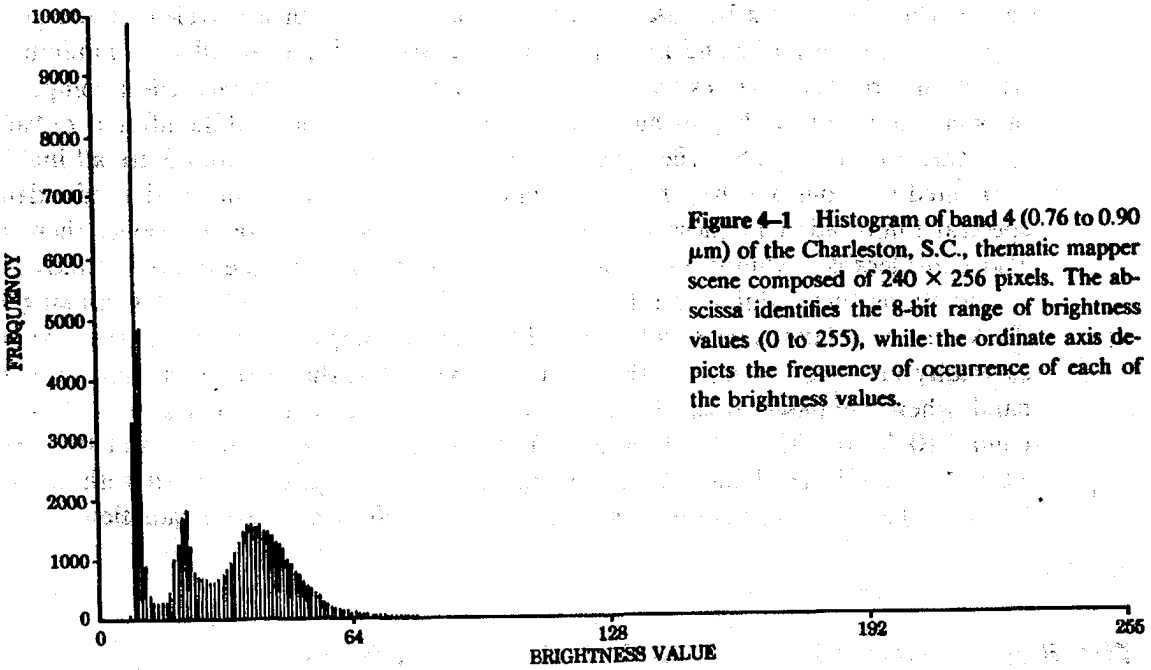
A single band of remotely sensed imagery, $k$, was previously defined as being composed of $i$ rows and $j$ columns, with a brightness value $BV_{ijk}$ at each pixel location. Normally, brightness values will range $BV_r$ from $2^6$ to $2^8$ ($2^6$ = values from 0 to 63, $2^7$ = values from 0 to 127, and $2^8$ = values from 0 to 255). Tabulating frequencies of occurrence of each of the brightness values within the image provides statistical information which may be displayed graphically in a *histogram* (Castleman, 1978). The range of quantized values, $BV_r$, is provided on the abscissa, while the frequency of occurrence of each of these values, $f(BV_i)$ is displayed on the ordinate. For example, there might be 175 pixels with a brightness value of 50 in the scene.

A histogram of band 4 of the Charleston S.C., thematic mapper scene is shown in Fig. 4–1. It was produced using the HISTOGRAM program found in Appendix A. Note how the data are compressed into the lower one-fourth of the range 0 to 255, suggesting that it is relatively low in contrast. The various peaks and valleys in the multimodal histogram correspond to dominant types of land cover in the image. The peak with the highest frequency is associated with water pixels in the scene.

A histogram of predawn thermal infrared (8.5 to 13.5 $\mu$m) imagery obtained over a portion of the Savannah River is shown in Fig. 4–2. The modes in this histogram are associated with the relatively cool temperature of the Savannah River swamp, the slightly warmer temperature (12° C) of the Savannah River, and the higher brightness values associated with a warm thermal plume. The plume enters the Savannah River via a creek that carries hot water used to cool industrial activities.

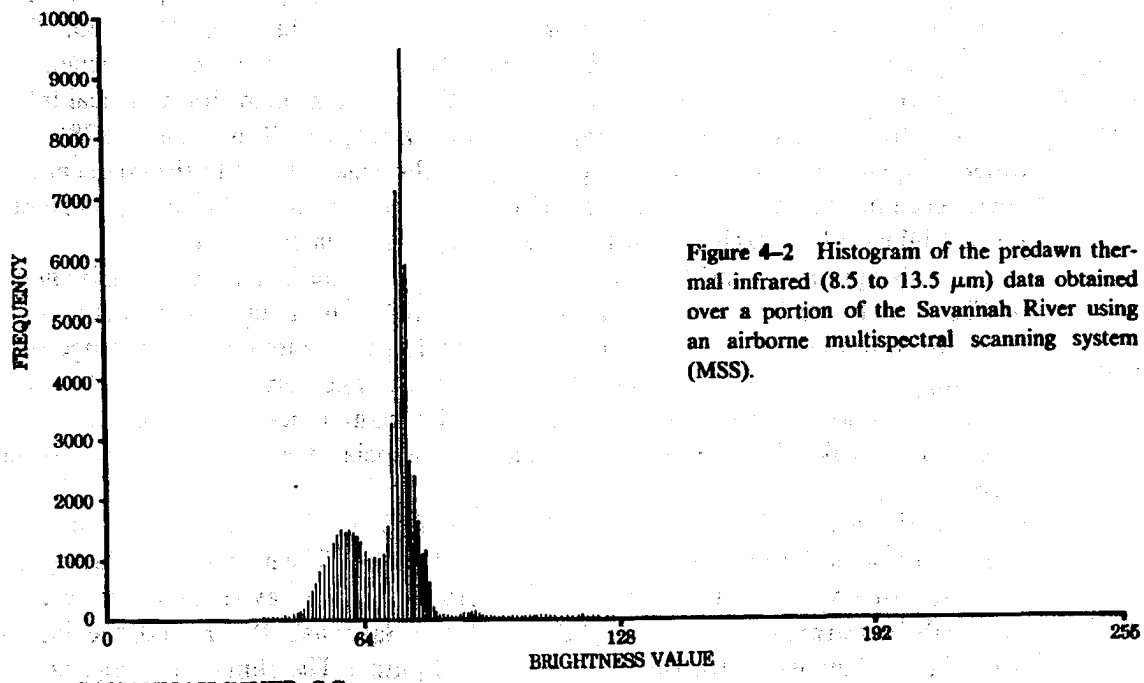
The most fundamental histogram is a plot of the number of pixels occurring


OFEK06754




CHARLESTON, S.C.
THEMATIC MAPPER
BAND 4 (ORIGINAL DATA)

**Figure 4–1** Histogram of band 4 (0.76 to 0.90 μm) of the Charleston, S.C., thematic mapper scene composed of 240 × 256 pixels. The abscissa identifies the 8-bit range of brightness values (0 to 255), while the ordinate axis depicts the frequency of occurrence of each of the brightness values.




SAVANNAH RIVER, S.C.
AIRCRAFT MULTISPECTRAL SCANNER
THERMAL INFRARED (ORIGINAL DATA)

**Figure 4–2** Histogram of the predawn thermal infrared (8.5 to 13.5 μm) data obtained over a portion of the Savannah River using an airborne multispectral scanning system (MSS).



OFEK06755

for each of the brightness values. Often, however, the raw count of certain brightness values becomes prohibitively difficult to display, especially when an unusually large number of pixels have the same brightness value. When this occurs, it is useful to scale the frequency of occurrence according to the relative percentage of pixels within the image at each brightness level. If there are $n$ pixels in the subimage, then the percentage of pixels, $p$, at each brightness value, $BV_i$, may be computed using the equation

$$p(BV_i) = \frac{f(BV_i)}{n} \times 100 \tag{4-7}$$

where $f$ is the frequency of occurrence of the pixels having brightness value $BV_i$. Once this is computed, it is necessary to allocate a certain number of pixels to be used in the histogram. For example, if 1000 pixels were allocated for the vertical bars in the entire histogram display, a given brightness value with a frequency of 100 would be allocated 10% of the pixels and would be represented by a vertical bar 100 pixels in height. In this manner, it is possible to highlight the information content found in the histogram which may be masked when certain brightness values have an inordinate frequency of occurrence. Examples of this type of histogram display are provided in Fig. 4–3, where the ordinate axis ranges from 0 to 10%.

The histogram is a useful graphic representation of the information content of

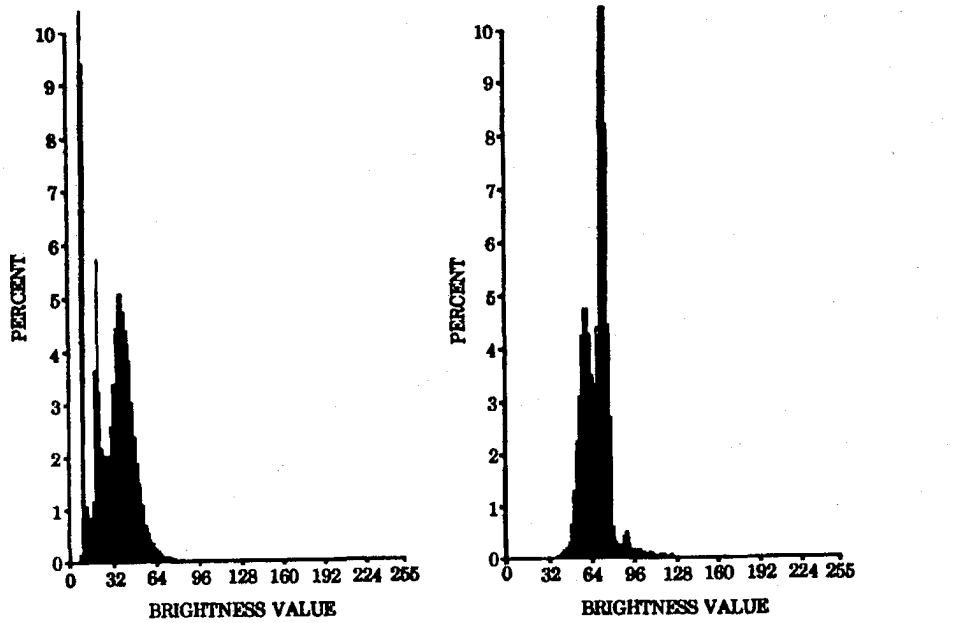



Figure 4–3 Histograms of the Charleston and Savannah River images displayed using percentages instead of frequencies on the ordinate axes.



OFEK06756

a remotely sensed image. Histograms for each of the bands of imagery are often displayed in many studies because they provide the reader with an appreciation of the quality of the original data (e.g., whether it is low in contrast, high in contrast, or multimodal in nature). In fact, many analysts routinely provide both before (original) and after histograms of the imagery to document the effects of applying an image enhancement technique. Examples will be provided in Chapter 7.

## REFERENCES

CASTLEMAN, K. E. 1978. "The Gray Level Histogram," Chapter 5 in *Digital Image Processing*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., pp. 68–84.

COOLEY, W. W., and P. R. LOHNES. 1971. *Multivariate Data Analysis*. New York: John Wiley & Sons, Inc., pp. 49–97.

DAVIES, H. G., and G. A. HICKS. 1975. *Mathematics for Scientific and Technical Students*. New York: Longman, Inc., pp. 312–331.

DAVIS, J. C. 1973. *Statistics and Data Analysis in Geology*. New York: John Wiley & Sons, Inc., pp. 68–79.

HAMMOND, R., and P. S. MCCULLAGH. 1980. *Quantitative Techniques in Geography*. Oxford, England: Clarendon Press.

SHORT, N. M. 1982. *The Landsat Tutorial Workbook: Basics of Satellite Remote Sensing*. Washington, D.C.: NASA Scientific and Technical Information Branch, pp. 174–175.



OFEK06757

# 5

# Initial Display Alternatives

## INTRODUCTION

Black-and-white aerial photography exhibits a continuous range of gray values from black through gray to white. Visual interpretation of such imagery is possible because human beings process continuous-tone images throughout their daily lives. But what about the interpretation of digital remotely sensed data available from sensor systems such as the Landsat multispectral scanner or thematic mapper? How do interpreters convert the brightness values (BVs) stored on a computer-compatible tape (CCT) into an image that begins to approximate the continuous-tone photographs they are used to interpreting? The answer is the creation of a brightness map, also commonly referred to as a gray-scale map.

A *brightness map* is a computer graphic display of the brightness values ($BV_{ijk}$) found in digital remote sensor data (Fig. 2–1). Ideally, there is a one-to-one relationship between input brightness values and the resultant intensities of the output brightness values on the display [i.e., a BV of 0 would result in a very dark (black) intensity on the map, while a BV of 255 would produce a bright (white) intensity]. All brightness values between 0 and 255 would be displayed as a continuum of grays from black to white. Unfortunately, it is not always easy to maintain this ideal relationship.

Most digital image processing systems designed for remote sensing education or research purposes have the capability to produce brightness maps (Jensen, 1983). The quality and interpretability of a remote sensing brightness map is dependent on (1) the type of output device and (2) the type of algorithm(s) or logic used in its creation.

This chapter describes the creation of remote sensing brightness maps using



0960 OFEK06758

two fundamentally different output devices: hard-copy and video displays. *Hard-copy displays* are based on the use of line printers, line plotters, or film writers to produce tangible hard copies of the imagery for visual examination. Conversely, *video displays* are usually based on the use of black-and-white or color cathode ray tubes (CRTs) that display a temporary image for visual examination (Green, 1983). The temporary image may be discarded or subsequently routed to a hard-copy display if a hard copy is desired. This chapter summarizes the logic and function of displaying the remote sensor data using these display alternatives.

## BLACK-AND-WHITE HARD-COPY IMAGE DISPLAY

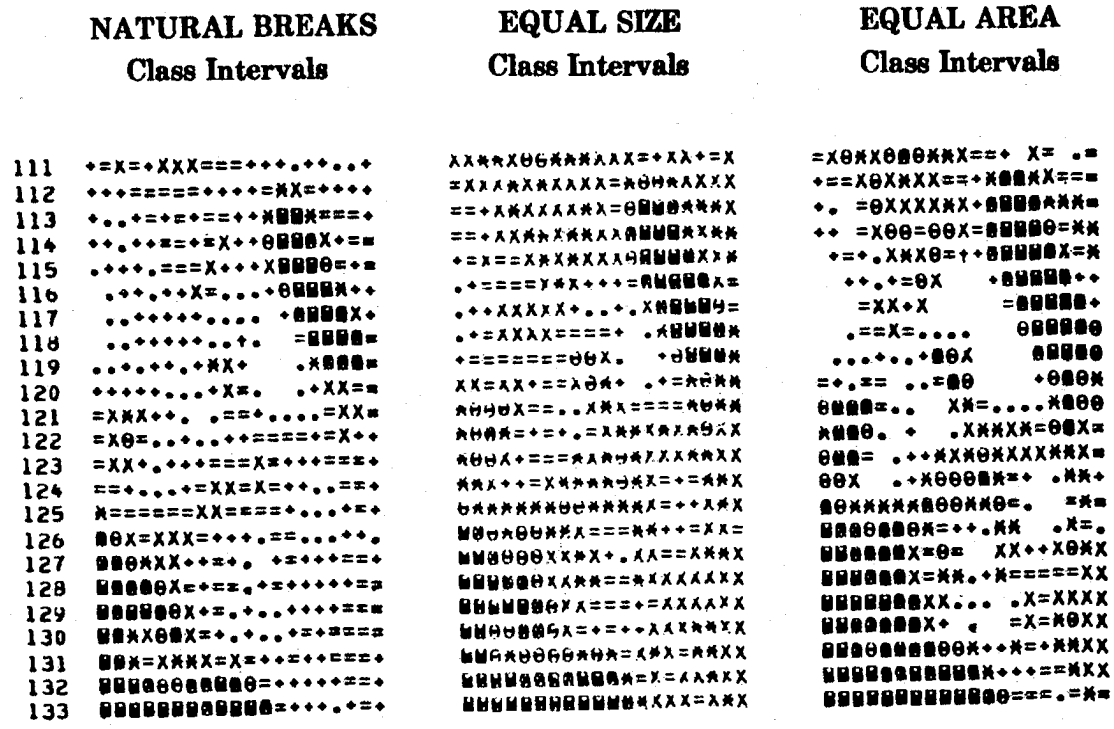
### Line Printer Brightness Maps

The most prevalent method of producing a brightness map is to output a "density-sliced" map to a standard line printer. *Density slicing* refers to the conversion of the continuous tone of an image into a series of discrete class intervals or slices where each interval corresponds to a specific brightness value range (Haralick, 1973). Such slices are analogous to class intervals used in cartography, where upper and lower class limts are selected for choropleth, isarithmic, and dasymetric mapping (Robinson et al., 1978).

To produce density-sliced maps on the line printer, it is necessary to select (1) an appropriate number of class intervals, (2) the size or dimension of each class interval, and (3) the symbolization to be assigned to each class interval. Crucial to this task is an awareness of the statistical nature of the scene. Analysis of the histogram provides valuable information when selecting the number and dimension of class intervals. One can expect to successfully display about eight class intervals on a conventional line printer map (i.e., a 3-bit range, $2^3 = 8$). Thus this severely degrades the potential of displaying a truly continuous-tone rendition of a Landsat thematic mapper image whose brightness values range from 0 to 255 (8 bits).

There are numerous methods of selecting class interval boundaries, including natural breaks, equal size, and equal areas (MacDougall, 1976). Remote sensing specialists often select the natural break method, whereby they analyze the histogram and then select $m$ mutually exclusive class intervals (i.e., density slices) to represent the image. A nine-class-interval density-sliced map using natural break criteria and produced on a line printer is shown in Fig. 5–1a. This is a very small area (just 23 rows by 20 columns) of the Landsat thematic mapper band 4 image of Charleston, S.C. A panchromatic aerial photograph of approximately the same geographic area is shown in Fig. 5–2. The equal-size and equal-area class interval schemes applied to the same area are shown in Fig. 5–1b and c, respectively. In an equal-size class interval scheme, the size of the class intervals is held as constant as possible. If the data are not uniformly distributed throughout the range of possible brightness values, this usually results in a relatively low contrast density-sliced map. Conversely, in an equal-area scheme, the class intervals are adjusted so that approximately equal


OFEK06759




| # | SYMBOL | NATURAL BREAKS RANGE | EQUAL SIZE RANGE | EQUAL AREA RANGE |
|---|---|---|---|---|
| 1 | ■ | 0 — 10 | 7 — 14 | 7 — 14 |
| 2 | ■ | 11 — 16 | 15 — 21 | 15 — 27 |
| 3 | ● | 17 — 20 | 22 — 28 | 28 — 32 |
| 4 | ✱ | 21 — 24 | 29 — 35 | 33 — 35 |
| 5 | X | 25 — 29 | 36 — 42 | 36 — 38 |
| 6 | = | 30 — 36 | 43 — 49 | 39 — 41 |
| 7 | + | 37 — 46 | 50 — 56 | 42 — 45 |
| 8 | • | 47 — 59 | 57 — 63 | 46 — 49 |
| 9 | | 60 — 255 | 64 — 74 | 50 — 74 |

**Figure 5–1** Three remote sensing brightness maps of a small portion of the Charleston, South Carolina, band 4 thematic mapper image produced on an eight-line-per-inch line printer. The different displays are the result of using (a) natural breaks, (b) equal size, or (c) an equal-area class interval scheme. The symbolization is held constant.

numbers of pixels fall within each of the $m$ class intervals. This scheme guarantees that every symbol selected will be used in the display. The natural breaks and equal-area class interval schemes are often used for initial (quick-look) visual evaluations of remotely sensed data.

The few symbols on a line printer available for creating the impression of continuous tone are a serious constraint. However, because the line printer is so ubiquitous,



OFEK06760

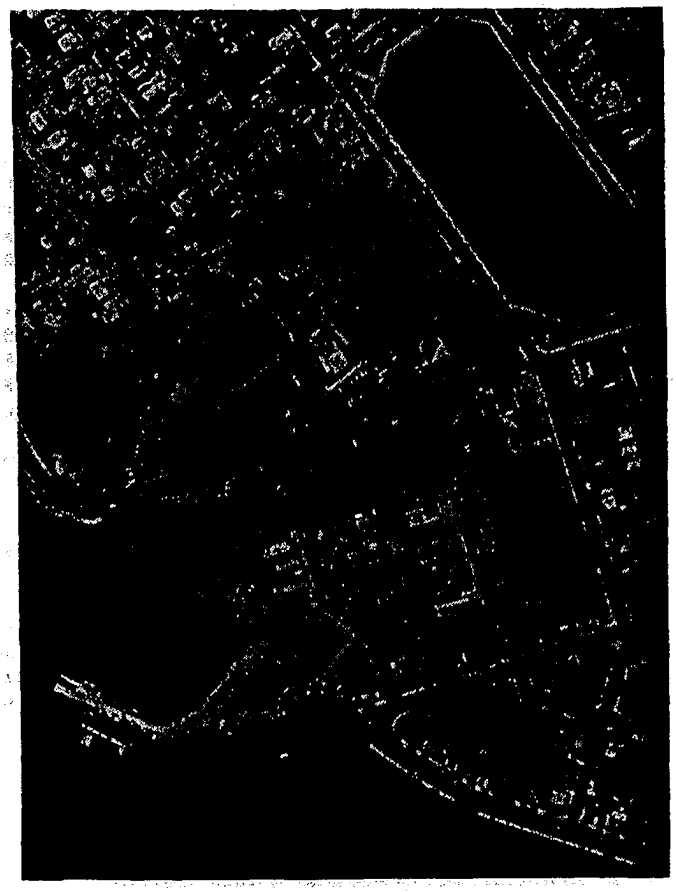


**Figure 5–2** Panchromatic aerial photography of approximately the same geographic area of Charleston, South Carolina, as shown in Fig. 5–1. The original scale was 1:12,000.

much research has gone into identifying optimum single-print and overprint symbolization to approximate the continuous-tone gray scale. For example, Smith (1980) evaluated 47 standard line printer symbols often used in computer cartography thematic mapping (Table 5–1). Using a transmission densitometer, the "percent area inked" of *areas* of symbols was measured, which included the empty space between symbols. The Williams adjustment (see Robinson et al., 1978) was then applied to the densitometer values to produce "perceived grayness values." The lightest average gray attainable was 23.2% (measured value, not converted) and the darkest was 65.1%, giving a range of 41%. Converted to perceptual equivalence using the Williams curve (Fig. 5–3), the extreme values became 34.5 and 64.1 (Table 5–1), representing a range of only 29.6%. Thus the 10-character-per-inch and 8-line-per-inch format of the typical line printer caused even the most dense overprint to yield only a 65% black area as perceived by the viewer because of the empty space between characters. Such conditions reduce the effectiveness of line printer density-sliced maps when used to approximate a continuous-tone remotely sensed image.

When selecting the line printer symbol for each class interval, one should try to develop a graded series that provide adequate contrast between gray-level symbols. Some manufacturers do provide optional gray-scale character sets that may be used



OFEK06761

**TABLE 5–1** Symbolization and Perceived Grayness Based on Transmission Densitometer Measurements[a]

| Symbol | Perceived grayness (%) | Symbol | Perceived grayness (%) |
|---|---|---|---|
| . | [34.5][b] | M | 46.4 |
| – | 25.6 | L. | 46.4 |
| ' | 36.4 | I– | 48.0 |
| + | [38.8] | X– | [48.5] |
| / | 39.8 | /= | 48.6 |
| = | [40.5] | X= | 50.1 |
| 1 | 40.6 | X+ | 50.5 |
| 7 | 41.5 | Z= | 50.5 |
| L | 41.8 | I= | 51.6 |
| * | 42.0 | VT | 52.0 |
| I | 42.0 | O– | [52.9] |
| 5 | 43.5 | VA | 53.9 |
| 4 | 43.5 | O= | 54.4 |
| H | 44.0 | O/ | 55.3 |
| 3 | 44.1 | O+ | 55.3 |
| 2 | 44.5 | MW | 55.9 |
| S | 44.8 | O* | 56.5 |
| X | [44.8] | TVA | 57.6 |
| N | 44.9 | HHH/ | 58.0 |
| 6 | 45.4 | OX | [58.1] |
| 0 | 45.4 | OH= | 60.3 |
| 9 | 45.5 | MEW | [62.0] |
| /– | 45.6 | OXAV | 62.5 |
| 8 | 46.3 | HIXO | 64.1 |

[a] Symbols evaluated in Smith (1980).

[b] The bracketed symbol sets were used in the creation of Figure 5–1. They were chosen for diagrammatic purposes only and should not be considered ideal symbolization for the creation of line printer brightness maps.

in this regard (Hamill, 1977). Also, some line printers may be adjusted to eradicate the scale distortion problem (Ballew and Lyon, 1977). But for the majority of users who only have access to standard 8- to 6-line-per-inch printers, the creation of remote sensing brightness maps continues to be a major problem.

### *Line Plotter Brightness Maps*

The creation of brightness maps using line plotters is another feasible alternative. Most educational and research institutions obtain line plotters, electrostatic printer-plotters, and/or dot matrix printer-plotters with dot addressable graphics options.



OFEK06762

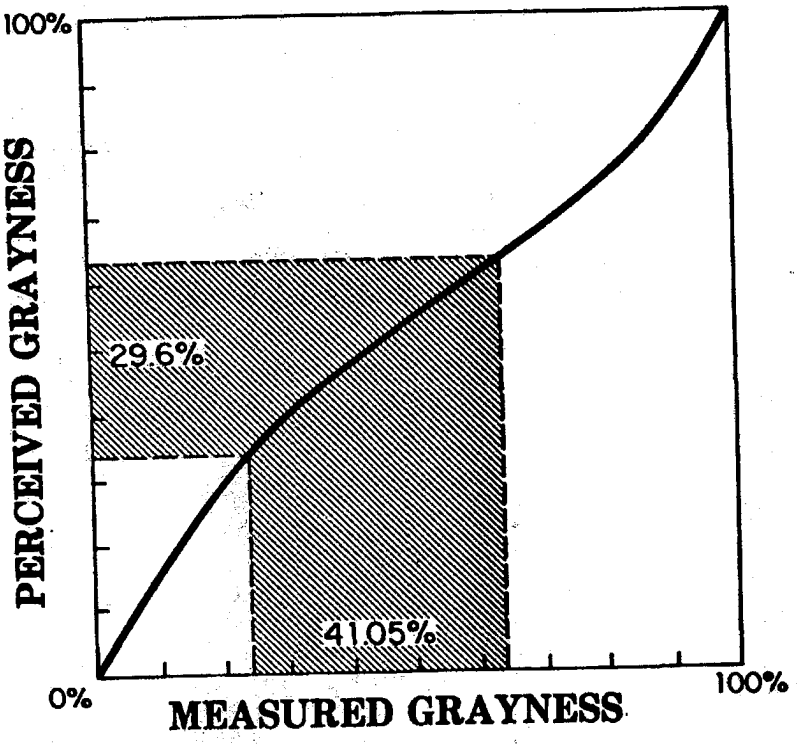



Figure 5–3 The range of gray tones produced by overprinting symbols on a line printer is reduced from 41% to approximately 30% after adjustment according to the Williams curve (after Smith, 1980). The 10-character-per-inch and eight-line-per-inch format typical of line printers causes even the most dense symbolization to yield only 65% black. The blank areas between symbols contributes a great deal to this condition. This graph is based on the assumption that the lightest class symbol will not be blank (completely white) because in cartography this is not considered good practice, as the map information may be visually confused with the border area surrounding it. However, most remote sensing practitioners often use the blank symbol for the highest (brightest) class interval. (From Jensen and Hodgson, 1983.)

Brightness maps produced by plotters provide (1) greater scaling capability and (2) the opportunity to create a unique sequence of symbols designed specifically for remote sensing brightness mapping.

The ability to scale a brightness map so that it accurately overlays a planimetric map base or other image is essential for most serious remote sensing endeavors. Scaling is accomplished by calling "scale" subroutines available in the software supplied with most plotters. Any previously defined alphanumeric character or symbol (e.g., a "*") in the system software may be called. These are often more useful than the standard line printer symbols. However, this type of symbolization is still inadequate and even overprinting of symbols only approximates the results produced in the line printer brightness maps.

Other researchers have programmed plotters to turn on specific dots within a specific pixel area (Craig et al., 1978). For example, dot matrix printer-plotter output is shown in Fig. 5–4. Such dot matrices may be used to display as many as 16 gray



OFEK06763

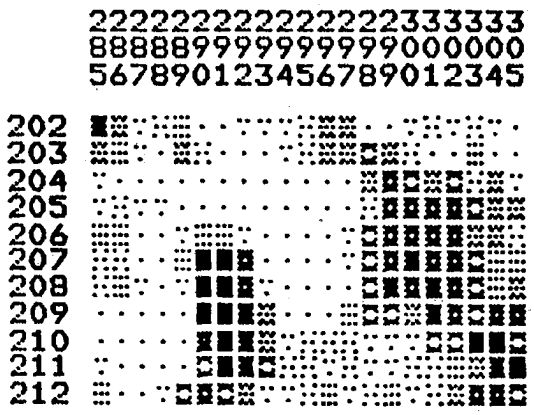



Figure 5–4 Portion of a brightness map produced on a matrix printer-plotter by turning on specific dots within a pixel. Each of the 10 patterns in this example were individually programmed. (From Craig et al., 1978.)

levels. However, this procedure still necessitates compressing the data down to a 4-bit resolution versus the 7- or 8-bit resolution of most digital remote sensor data. What is needed are algorithms to produce any shade of gray desired (from black to white; 0 to 255, respectively) for each pixel.

### *Crossed-Line Shading*

The impression of continuous tone may be produced by using parallel lines, crossed lines, or circular dots located adjacent to one another. This discussion focuses on the use of crossed-line shading (Jensen and Hodgson, 1983). As the name implies, crossed-line shading is produced by intertwining two perpendicular sets of equally spaced parallel lines (Fig. 5–5). The lightness or darkness of the shading is controlled by reducing or increasing the separation between neighboring lines. The width or

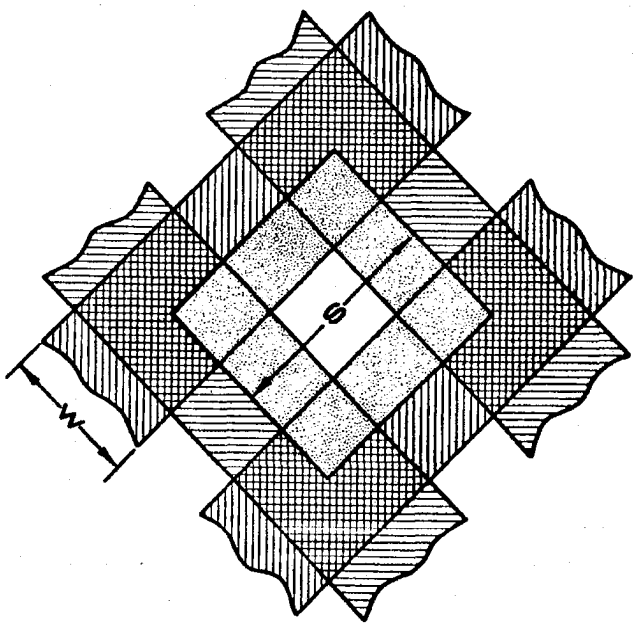



Figure 5–5 Geometry of cross-line shading (after Monmonier, 1980). The line ($w$) is generally held constant while line separation ($s$) is varied using equation (5-5) to produce varying shades of gray within a pixel. (From Jensen and Hodgson, 1983.)



OFEK06764

line thickness ($w$) is generally held constant while line separation ($s$) is varied to produce different shades of gray (Monmonier, 1980).

To understand how the proportion of the area inked is computed, consider a typical cell in the shading pattern (Fig. 5–5) as an uninked square bounded by a half-line-thick frame (Monmonier, 1979). The inked, or "black" proportion, $p$, is the area of this frame, with thickness $w/2$, divided by the total area of the cell formed by lines with separation $s$, or

$$p = \frac{2s(w/2) + 2(s - 2(w/2))(w/2)}{s^2} \tag{5-1}$$

which reduces to

$$p = \frac{2sw - w^2}{s^2} \tag{5-2}$$

Solving this quadratic equation for the plotting separation, $s$, between neighboring lines yields

$$s = \frac{w + w(1-p)^{0.5}}{p} \tag{5-3}$$

In the original work by Tobler (1973) on crossed-line continuous-tone mapping, the proportion, $p$, was rescaled according to the psychophysical power function

$$R = S^{1/a} \tag{5-4}$$

which relates response of the map reader, $R$, to map stimulus, $S$. In the rescaled formula,

$$s = \frac{w}{p^a} \quad [1 + (1 - p^a)^{0.5}] \tag{5-5}$$

where $p$ now represents the intended proportion rather than the apparent proportion of the area inked (Monmonier, 1979).

Based on work by Stoessel (1972), Tobler derived a correction for continuous-tone, crossed-line shading with the exponential function: area inked = value$^{1.4}$, which represents the perceptual function: values = (area inked)$^{0.7}$ (Monmonier, 1980). This transformation is labeled "exponential-Stoessel" (Tobler, 1973) in Fig. 5–6. Thus, when the exponent $a$ is set to 1.4, the nonlinear response of the human perceptual system is taken into consideration. Other exponents have been proposed, such as: value = (area inked)$^{0.647}$, which, obviously, produces similar results (Peterson, 1979).

To use equation (5-5), the remotely sensed brightness value, $BV_{ijk}$, must be transformed into value $p$, which lies in the range from 0 to 1. Each pixel value $BV_{ijk}$ is recomputed according to the formula

$$p = \frac{BV_{ijk} - R_{min}}{R_{max} - R_{min}} \tag{5-6}$$



OFEK06765

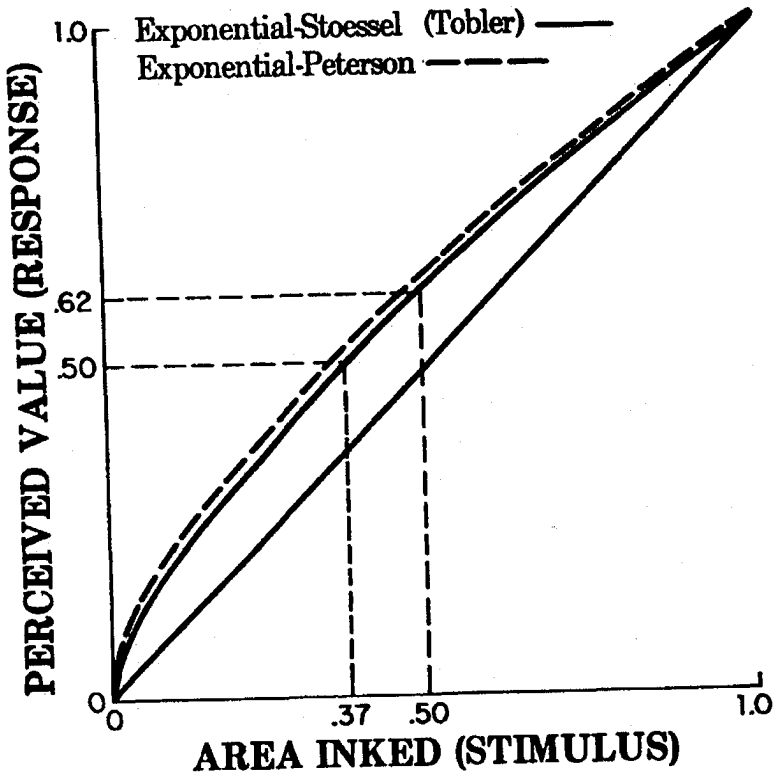



**Figure 5–6** Two models that compensate for the nonlinear response of the human eye to percent-area-inked stimuli. (From Jensen and Hodgson, 1983.)

where $R_{min}$ and $R_{max}$ represent the extreme points in the brightness value range that $BV_{ijk}$ might take upon itself (e.g., if $BV_r = 255$, then $R_{min} = 0$ and $R_{max} = 255$).

Program BRIGHTNESS MAP listed in Appendix A uses this logic to create brightness maps on an electrostatic plotter. A small portion of the Charleston, S.C., band 4 TM scene is displayed using the BRIGHTNESS MAP program in Fig. 5–7. Note the variety of gray shades present compared with the illustrations in Fig. 5–1. This same logic is applied to a 240 × 256 submatrix of the TM scene in Fig. 5–8. The original of this illustration was approximately 30 by 36 in.; therefore, it has undergone substantial photographic reduction.

This type of brightness map display produces results that are similar in appearance to continuous-tone output. The method is superior to the creation of line printer gray maps because (1) it uses a graded series of the same symbol (the crosshatch) rather than totally different symbols that introduce texture into the display, and (2) all cells are scaled to user specifications, alleviating the problem of the ⅘ or ⅗ scale problem associated with typical 8- or 6-line-per-inch line printers. The most effective display of the remote sensor data, however, is based on the use of video displays, which have improved brightness map display capability.



OFEK06766

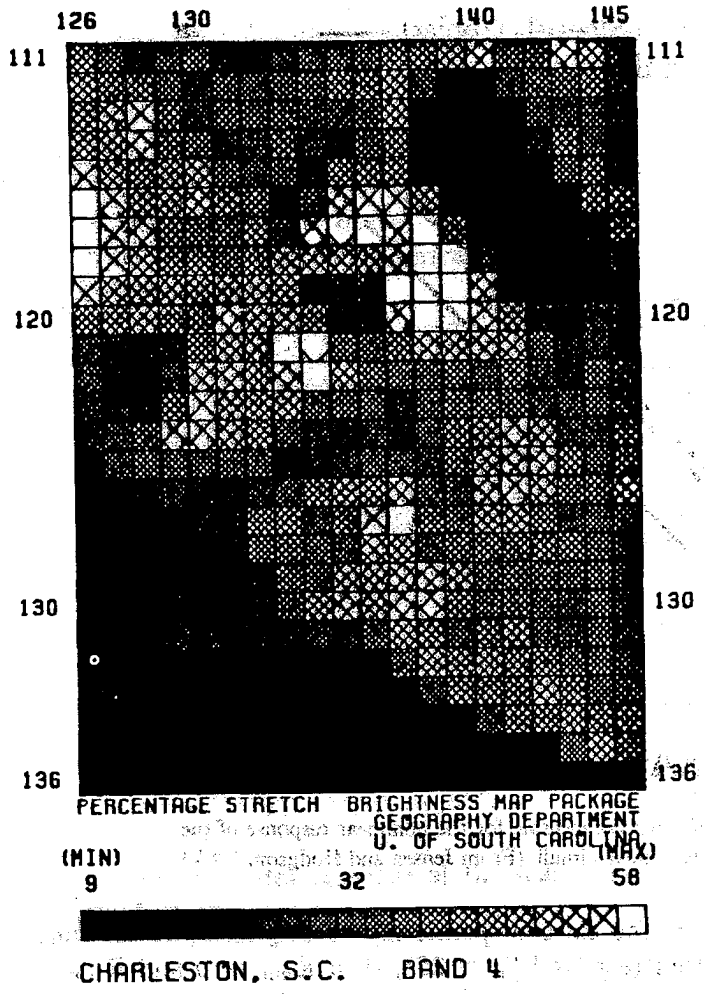



**Figure 5–7** Remote sensing brightness map of the Charleston, S.C., thematic mapper band 4 image produced on a 200-dot-per-inch electrostatic plotter using the **BRIGHTNESS MAP** program and equation (5-5). The original pixel size was 0.2 in. square with a line width ($w$) of 0.02 in. Because the data was so low in contrast, it was enhanced using a percentage linear contrast stretch discussed in Chapter 7.

## VIDEO IMAGE DISPLAY

### Black-and-White and Color Brightness Maps

A *video image display* simply means that the image is provided in a video format on a black-and-white or color monitor for visual examination and can easily be modified or discarded. The CPU generally reads the digital remote sensor data from a mass storage device (either disk or tape) and transfers these data to the image processor. The *image processor* is a collection of display memory composed of $i$ lines by $j$ columns and $b$ bits, which may be accessed sequentially, line by line (Fig. 5–9). Each line of digital values stored in the image processor display memory is continually scanned by a "read mechanism." The contents of the image processor memory are normally "read" every $\frac{1}{30}$ of a second, referred to as the "refresh rate" of the system. The brightness values encountered during this scanning process are passed onto a digital-to-analog converter (DAC) that prepares an analog video signal suitable for



OFEK06767



**Figure 5–8** A remote sensing brightness map based on the same principles as Fig. 5–7 except that an area 240 × 256 pixels is displayed.

display on the CRT screen. Thus the analyst viewing a CRT screen is actually looking at the video expression of the digital values stored in the image processor memory. The $\frac{1}{30}$-s refresh rate is so fast that the analyst normally does not see any significant amount of flicker on the CRT screen.

But how does the entire range of brightness values in a digital data set get displayed properly? This is a function of the number of bits per pixel and the "video lookup tables" associated with the image processor (Greenberg et al., 1982; Green, 1983). Many image processing systems have 8 bits of color resolution per pixel. As discussed earlier, this means that each pixel may assume any one of 256 values, from 0 to 255. However, the exact nature of the display is more precisely controlled by the size and characteristics of a separate block of memory called a "lookup table,"



OFEK06768

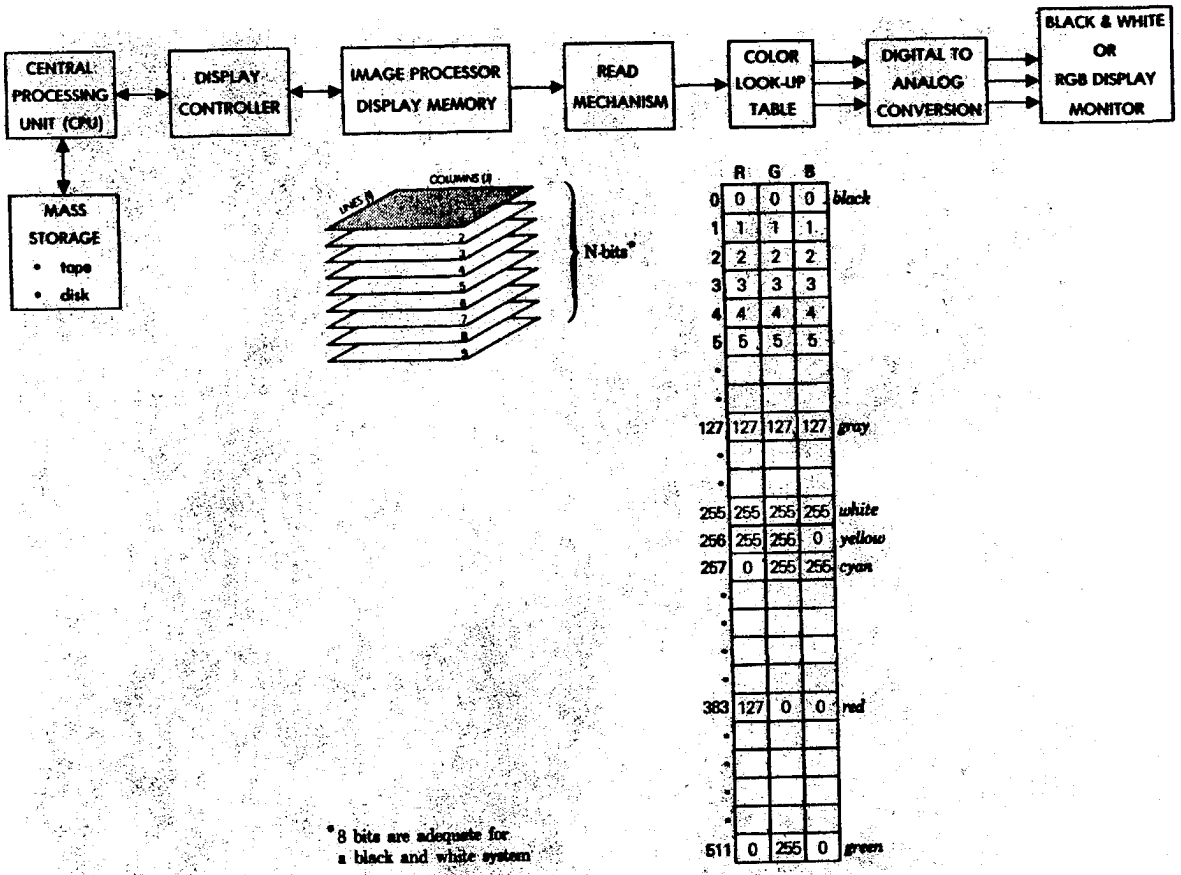



**Figure 5–9** Components of a 9-bit digital image processing system. The image processor display memory is filled with the 8-bit remote sensing brightness values from a single band of imagery. These values are then manipulated for display by specifying the contents of the 512-element color lookup table. An 8-bit digital-to-analog converter (DAC) then converts the digital lookup table values into analog signals that are used to modulate the intensity of the red, green, and blue (RGB) "guns" that create the image on the color CRT screen.

which contains the exact disposition of each combination of red, green, and blue (RGB) values associated with each 8-bit pixel. Evaluating the nature of a 9-bit image processor and associated color lookup tables will provide some insight into the way the brightness values and color lookup table must interact.

In Fig. 5–9 one sees that the first 256 elements of the lookup table coincide with progressively greater values of the red, green, and blue (RGB) components, ranging from 0, 0, 0, which would be black on the screen, to 127, 127, 127, which would be gray, and of course 255, 255, 255, for a very bright white. Thus if a pixel in a single band of imagery had a brightness value of 127, the RGB value located at table entry 127 (with RGB values of 127, 127, 127) would be passed on to the 8-bit digital-to-analog converter and a midgray pixel would be displayed. This type of logic was applied to create the black-and-white thematic mapper band 4 image of Charleston, S.C., shown in Fig. 5–10. This is often referred to as a true 8-bit



OFEK06769

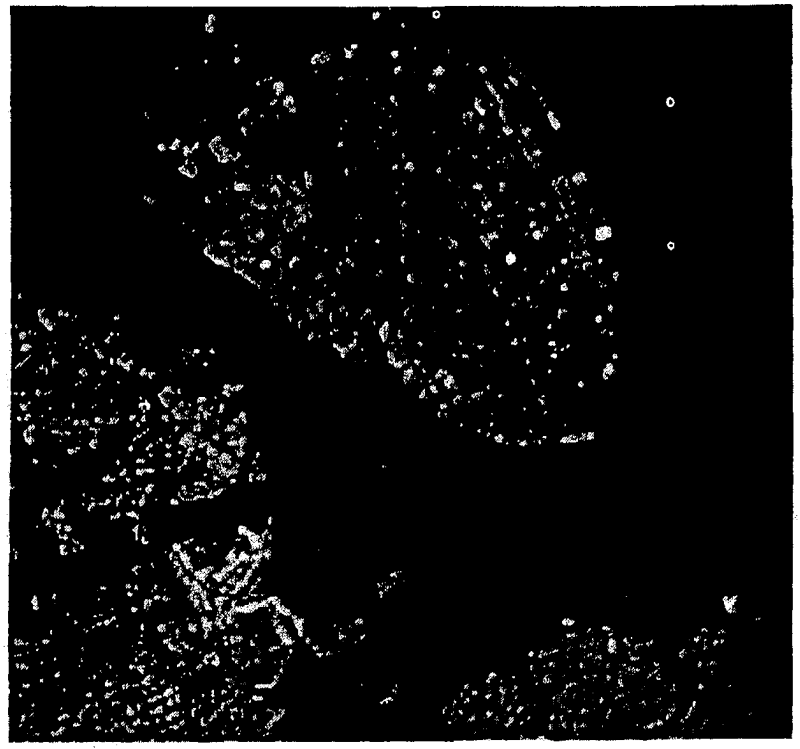


**Figure 5–10** Black-and-white display of the band 4 thematic mapper image of Charleston, S.C. The data were collected on November 11, 1982, at path 16, row 37 of the World Reference System (WRS). Each of the 240 × 256 pixels displayed represent an area 30 × 30 meters on the ground.

black-and-white display because there is no generalization of the original remote sensor data. This means that there is a one-to-one relationship between the 8 bits of remote sensor input data and the 8-bit color lookup table. This is the ideal mechanism for displaying remote sensor data. This same logic was applied to produce the 8-bit display of the predawn thermal infrared image of the Savannah River in Fig. 5–11 (Jensen et al., 1983).

If desired, the lookup table may be filled with much different values. For instance, the first entry in the table might be given an RGB value of 255, 0, 0, which would mean that any pixel with a value of 0 would be displayed as bright red. In this way, it is possible to create a special-purpose lookup table with those colors in it which are of greatest value or use to the analyst. This is precisely the mechanism by which a single band of remote sensor data is color "density sliced." For example, Table 5–2 lists the upper and lower brightness values for the line printer density-sliced map shown in Fig. 5–1a. If the analyst takes these same class intervals (e.g., interval 1 is from 0 to 10) and assigns all the values in the color lookup table the desired value (let's say, dark blue; 0, 0, 120), then all red, green, and blue (RGB) values in the lookup table from 0 to 10 will have 0, 0, and 120 in them, respectively. If class interval 2 is to contain all brightness values from 11 to 16 and be bright


OFEK06770



**Figure 5–11** Black-and-white display of a thermal plume in the Savannah River in South Carolina. Each of the 240 × 256 pixels in the thermal infrared image (8.5 to 13.5 μm) is approximately 2.8 × 2.8 meters on the ground. The data were collected on March 28, 1981, at 4:28 A.M. at 1220 meters above ground level (AGL).

**TABLE 5–2** Class Intervals and Color-Lookup-Table Values for Color Density Slicing the Band 4 Charleston, S.C., Thematic Mapper Scene Shown in Fig. 5–12 (Color Section)

| Class interval | Visual color | Color-lookup-table value | | | Brightness value | |
|---|---|---|---|---|---|---|
| | | Red | Green | Blue | Low | High |
| 1 | Dark blue | 0 | 0 | 120 | 0 | 10 |
| 2 | Bright blue | 0 | 0 | 255 | 11 | 16 |
| 3 | Dark green | 0 | 80 | 0 | 17 | 20 |
| 4 | Green | 0 | 140 | 0 | 21 | 24 |
| 5 | Bright green | 0 | 255 | 0 | 25 | 29 |
| 6 | Yellow | 255 | 255 | 0 | 30 | 36 |
| 7 | Dark red | 120 | 60 | 0 | 37 | 46 |
| 8 | Bright red | 255 | 0 | 0 | 47 | 59 |
| 9 | White | 255 | 255 | 255 | 60 | 255 |



OFEK06771

blue, the color lookup table from 11 to 16 would be allocated values 0, 0, 255 (RGB). Table 5–2 summarizes all the values of the color lookup table used to prepare the density-sliced image shown in Fig. 5–12 (color section). Similar logic was applied to density slice the predawn Savannah River thermal infrared image [Fig. 5–13 (color section)]. The temperature class intervals and associated lookup table values for the Savannah River color density-sliced image are found in Table 5–3.

Because there are 9 bits available in this particular image processor memory (Fig. 5–9), it is possible to allocate an additional 256 colors to the lookup table, bringing the total to 512 (i.e., $2^9 = 512$). In this case element number 256 in the table might be yellow, composed of equal proportions of red and green, and no blue (e.g., 255, 255, 0) or element 257 could be cyan (0, 255, 255). Obviously, when the 9 bits of refresh memory are exhausted, no additional colors or shades of gray may be displayed at one time on the CRT screen.

Much more flexibility is provided when multiple 8-bit images can be stored and evaluated all at one time (Ferns, 1984). For example, Fig. 5–14 depicts the configuration of a 24-bit image processing system complete with three 8-bit banks of image processor memory and three 8-bit color lookup tables, one for each image plane. Thus three separate 8-bit images could be stored at full resolution in the image processor memory (one in the red, one in the green, and one in the blue). Three separate 8-bit digital-to-analog converters (DACs) in effect read the brightness value of a pixel in each of the red, green, and blue image planes and transform this digital value into an analog signal that may be used to modulate the intensity of the red, green, and blue (RGB) tricolor guns on the face of the CRT screen. For example, if pixel (1, 1) in the red image plane has a brightness value of 255, and pixel (1, 1) in both the green and blue image planes have brightness values of 0, then a bright red pixel (255, 0, 0) would be displayed at location (1, 1) on the CRT screen. More

**TABLE 5–3** Class Intervals and Color-Lookup-Table Values for Color Density Slicing the Savannah River Thermal Infrared Image Shown in Fig. 5–13 (Color section)

| Class interval | Visual Color | Color-lookup-table value | | | Apparent temperature (°C) | | Brightness value | |
|---|---|---|---|---|---|---|---|---|
| | | Red | Green | Blue | Low value | Upper value | Low | High |
| 1. Land | Gray | 127 | 127 | 127 | −3.0 | 11.6 | 0 | 73 |
| 2. River ambient | Dark blue | 0 | 0 | 120 | 11.8 | 12.2 | 74 | 76 |
| 3. +1°C | Light blue | 0 | 0 | 255 | 12.4 | 13.0 | 77 | 80 |
| 4. 1.2–2.8°C | Green | 0 | 255 | 0 | 13.2 | 14.8 | 81 | 89 |
| 5. 3.0–5.0°C | Yellow | 255 | 255 | 0 | 15.0 | 17.0 | 90 | 100 |
| 6. 5.2–10.0°C | Orange | 255 | 50 | 0 | 17.2 | 22.0 | 101 | 125 |
| 7. 10.2–20°C | Red | 255 | 0 | 0 | 22.2 | 32.0 | 126 | 176 |
| 8. >20°C | White | 255 | 255 | 255 | 32.2 | 48.0 | 176 | 255 |

*Source*: Jensen et al. (1983).



OFEK06772

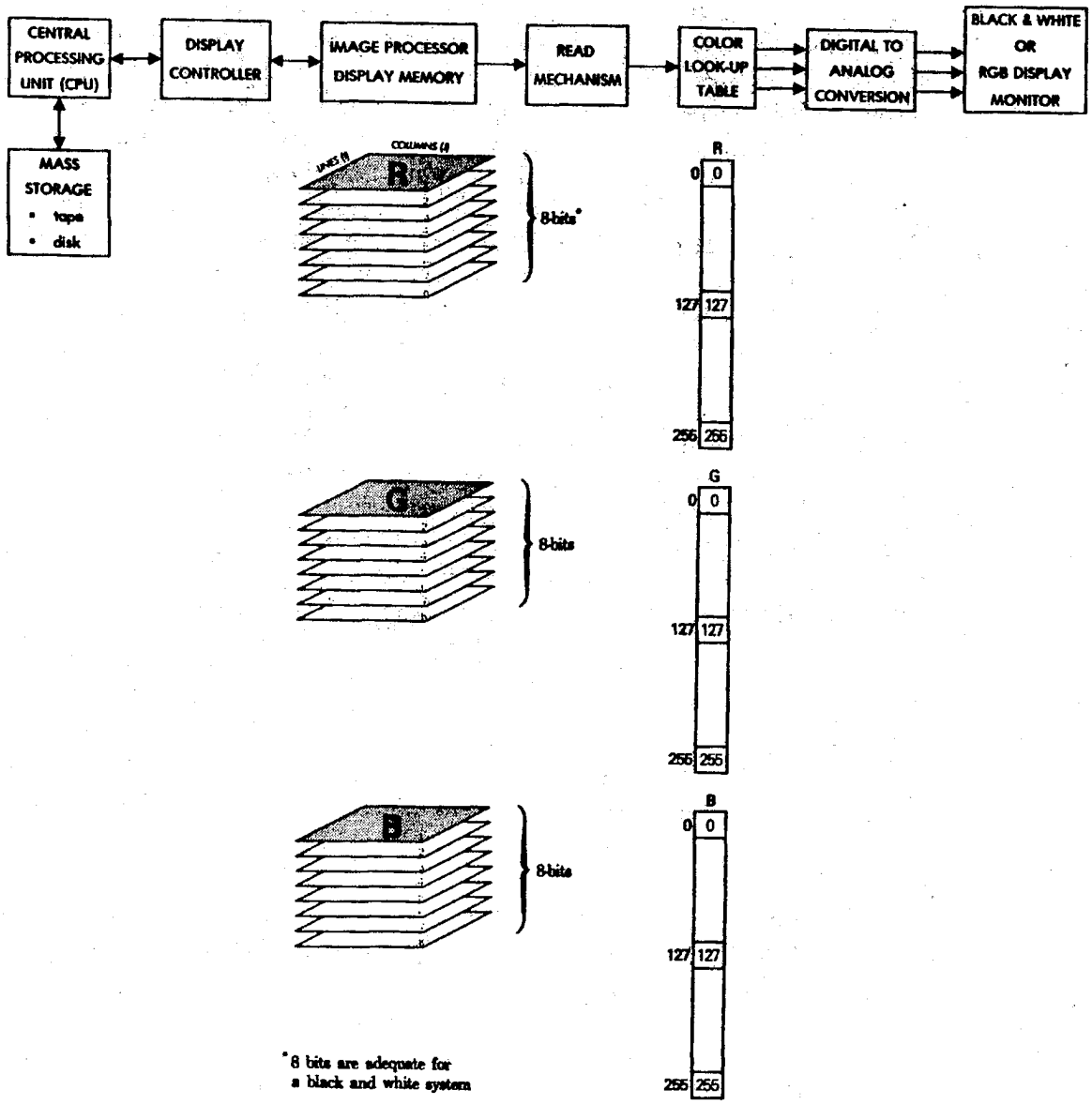



**Figure 5–14** Components of a 24-bit digital image processing system. The image processor can store and continuously evalute up to three 8-bit remotely sensed images. Three 8-bit digital-to-analog converters (DACs) scan the contents of the three 8-bit color lookup tables. The pixel color displayed on the CRT screen will be just one possible combination (e.g., red = 225, 0,0) out of a possible 16,777,216 colors that can be displayed at any time using the 24-bit image processing system.

than 16.7 million RGB color combinations may be produced using this configuration (i.e., $2^{24} = 16{,}777{,}216$). Obviously, this provides a much more complete palette of colors to choose from, all of which may be displayed on the CRT at one time. What has just been described is the fundamental basis behind the creation of color composites.



OFEK06773

If one wanted to display a black-and-white version of a single band of imagery (e.g. band 4) using this configuration, it would be necessary to fill all three image planes (RGB) with the same band 4 values in order to create a normal-looking black-and-white image on the CRT screen.

### *False Color Composites*

High color resolution becomes very important when producing color composites. For example, if a false-color reflective-infrared thematic mapper image is to be displayed accurately, each 8-bit input image (green, red, and reflective infrared) must be assigned 8 bits of blue, green, and red, respectively. In this case, the 24-bit system with three 8-bit color lookup tables just described would provide a true-color rendition of the exact color of each pixel on the screen, as discussed previously. This means that if a pixel had a red brightness value of 150, a green value of 150, and a blue value of 0, the resultant yellow pixel on the screen would truly be a product of these three brightness values. This is true additive color combining with no generalization taking place.

If on the other hand, the image processor had only 9 bits of color, each of the green, red, and reflective-infrared bands of remotely sensed data could be allocated only 3 bits of color [i.e., only eight shades of blue ($2^3 = 8$), eight shades of green, and eight shades of red]. This would result in only 512 possible combinations of colors displayable on the monitor at one time, when the data have the potential to yield over 16.7 million color combinations. Therefore, there must be generalization and loss of color visual detail. In fact, if the data are very low in contrast (i.e., the data are compressed into a very narrow range of brightness values), it will be difficult to produce a useful color composite. However, if the individual bands of imagery are contrast stretched, a reasonable color composite might be attained despite the obvious generalization.

A color infrared color composite of bands 2, 3, and 4 of the Charleston, S.C., thematic mapper scene is displayed in Fig. 5–15 (color section). It was produced using an image processing system with 15 bits of spectral resolution (i.e., 5 bits allocated to the blue, green, and red lookup tables, respectively). This means that the green band (2) brightness values were scaled down from 0 to 255 to values ranging from 0 to 31 as were the red (3) and reflective-infrared (4) bands. The 15-bit display enables $2^{15}$ possible color combinations (32,768) to be displayed on the color monitor at one time out of a possible 16.7 million colors ($2^{24}$). Image processing systems that can display $\geq 2^{15}$ colors on the monitor at one time generally provide a very effective color composite. Systems that display fewer colors (e.g., 9 or 12 bits) run the risk of artificial contouring in the color composites, due simply to the extreme generalization that the data must undergo.

In addition to color infrared color composites, it is possible to create other false-color composites. For example, natural color displays of the terrain may be created using exactly the same logic except that the image processor blue, green, and red memory planes are filled with blue, green, and red brightness values. In the case of the Landsat TM data, this would mean that brightness values found in



OFEK06774

bands 1, 2, and 3 would be placed in the image processor blue, green, and red refresh memory planes. But what about all the other three-band color composites that may be produced from the same TM data? Besides trial-and-error visual analysis, is there no way quantitatively to identify which three bands should produce the optimum color composite?

**TABLE 5–4** Optimum Index Factors (OIF) for Six of the Charleston, S.C., Thematic Mapper Bands

| Rank | Combination[a] | OIF |
|---|---|---|
| 1 | 1, 4, 5 | 27.137[b] |
| 2 | 2, 4, 5 | 23.549 |
| 3 | 3, 4, 5 | 22.599 |
| 4 | 1, 5, 6 | 20.785 |
| 5 | 1, 4, 6 | 20.460 |
| 6 | 4, 5, 6 | 20.100 |
| 7 | 1, 3, 5 | 19.009 |
| 8 | 1, 2, 5 | 18.657 |
| 9 | 1, 3, 4 | 18.241 |
| 10 | 2, 5, 6 | 18.164 |
| 11 | 2, 3, 5 | 17.920 |
| 12 | 3, 5, 6 | 17.840 |
| 13 | 1, 2, 4 | 17.682 |
| 14 | 2, 4, 6 | 17.372 |
| 15 | 3, 4, 6 | 17.141 |
| 16 | 2, 3, 4 | 15.492 |
| 17 | 1, 3, 6 | 12.271 |
| 18 | 1, 2, 6 | 11.615 |
| 19 | 2, 3, 6 | 10.387 |
| 20 | 1, 2, 3 | 8.428 |

[a] Six bands combined three at a time allows 20 combinations. The thermal infrared band 7 (10.4 to 12.6 $\mu$m) was not used.

[b] OIF is calculated using the equation

$$\text{OIF} = \frac{\sum_{k=1}^{3} s_k}{\sum_{j=1}^{3} \text{Abs}\,(r_j)}$$

where $s_k$ is the standard deviation for band $k$, and $r_j$ is the absolute value of the correlation coefficient between any two of the three bands being used (Chavez et al., 1982). For example,

$$\frac{10.05 + 15.76 + 23.85}{0.39 + 0.56 + 0.88} = 27.137$$



OFEK06775

Chavez et al. (1982, 1984) developed an optimum index factor (OIF) that ranks the 20 three-band combinations that may be made from six bands of TM data (not including the thermal infrared band). The technique, however, is applicable to any multispectral remote sensing data set. It is based on the amount of total variance and correlation within and between various band combinations. The algorithm used to compute the OIF for any subset of three bands is

$$\text{OIF} = \frac{\sum_{k=1}^{3} s_k}{\sum_{j=1}^{3} \text{Abs}\,(r_j)} \tag{5-7}$$

where $s_k$ is the standard deviation for band $k$, and $r_j$ is the absolute value of the correlation coefficient between any two of the three bands being evaluated. The three-band combination with the largest OIF generally will have the most information (as measured by variance) with the least amount of duplication (as measured by correlation). Combinations within two or three rankings of each other may produce similar results. Application of the OIF criteria to the Charleston, S.C., thematic mapper data set (excluding the thermal band) resulted in the 20 combinations summarized in Table 5–4. Table 4–7 is the source of the standard deviations and between band correlation coefficients. A three-band combination using bands 1, 4, and 5 should provide the optimum color composite with bands 2, 4, and 5, and 3, 4, and 5 just about as good. Generally, the best three-band combinations include one of the visible bands (TM 1, 2, or 3) and one of the longer-wavelength infrared bands (TM 5 or 6) together with TM band 4. TM band 4 was present in five of the first six rankings. Such information may be used to select the most useful bands for three-band color composites. The analyst must then decide what color to assign each band (red, green, or blue) in the color composite.

The only time a large color lookup table is absolutely necessary is when a color composite must be produced. In most other cases, 8 bits (256 colors) are generally sufficient for producing density-sliced brightness maps, land cover classification maps, and so on. This is not surprising since most cartographers who produce computer graphic maps get along quite well with a color palette of just 256 displayable colors ($2^8$).

## TRANSFORMING VIDEO DISPLAYS TO HARD-COPY DISPLAYS

There are numerous instruments that will sample the image being displayed on a video display and produce a permanent hard copy. Many of the "color camera"-type instruments literally have a black-and-white CRT display inside them which is used to expose black-and-white or color film. Most are based on a simple three-



OFEK06776

pass exposure system, where first the blue, then the green, and finally the red information stored in the refresh memory are sequentially exposed to the film.

Other more sophisticated hard-copy instruments are based on light-emitting diodes (LEDS) or laser scanning systems that very precisely "write" or expose every pixel onto photographic film by modulating the intensity of the light source in direct proportion to the brightness value encountered in the image processor memory (Bruno, 1984). These "film writer" devices are capable of maintaining positional accuracy in the hard copy down to a few micrometers. Such devices provide the most impressive hard copy from video displays but are also expensive. Some instruments can both "read" (digitize) and "write" (expose) photographic film (Fig. 2–4). Several hard-copy devices capable of reproducing remotely sensed data displayed on a CRT are summarized in Table 5–5.

**TABLE 5–5** Hard-Copy Output Devices Capable of Reproducing Remotely Sensed Data Displayed on a CRT Screen

| Hard-copy device | Quality | Cost |
|---|---|---|
| Color film writer/plotter (e.g., Optronics, Dicomed) | **** | ***** |
| Black-and-white film writer/plotter | ***** | **** |
| Color cameras (e.g., Matrix, Dunn, Modgraph) | *** | *** |
| Ink-jet printers | ** | *** |
| Color dot matrix printers | * | ** |
| Camera (35mm) screen photography | ** | * |

*Source:* Ferns (1984).

They are listed in descending order of quality and cost.

## REFERENCES


BALLEW, G. I., and R. J. P. LYON. 1977. "The Display of Landsat Data at Large Scales by Matrix Printer," *Photogrammetric Engineering and Remote Sensing*, Vol. 43, pp. 1147–1150.

BRUNO, M. 1984. "Electronic Imaging in the Graphic Arts," *Electronic Imaging*, Vol. 3, pp. 30–37.

CHAVEZ, P. S., G. L. BERLIN, and L. B. SOWERS. 1982. "Statistical Method for Selecting Landsat MSS Ratios," *Journal of Applied Photographic Engineering*, Vol. 8, pp. 23–30.

CHAVEZ, P. S., S. C. GUPTILL, and J. A. BOWELL. 1984. "Image Processing Techniques for Thematic Mapper Data," *Technical Papers*, 50th Annual Meeting of the American Society of Photogrammetry, Vol. 2, pp. 728–742.

CRAIG, M. E., R. SIGMAN, and M. CARDENAS. 1978. "Area Estimates by Landsat: Kansat 1976 Winter Wheat," *Report of the New Techniques Section*. Washington, D.C.: Economics, Statistics and Cooperative Service of the U.S. Department of Agriculture.

FERNS, D. C. 1984. "Microcomputer-Based Systems for Remote Sensing," *Electronic Imaging*, Vol. 3, pp. 38–42.




OFEK06777

GREEN, W. B. 1983. "Image Display," Chapter 5 in *Digital Image Processing: A Systems Approach*. New York: Van Nostrand Reinhold Company, Inc., pp. 113–134.

GREENBERG, D., A. MARCUS, A. H. SCHMIDT, and V. GORTER. 1982. *The Computer Image: Applications of Computer Graphics*. Reading, Mass.: Addison-Wesley Publishing Company, Inc.

HAMILL, P. 1977. "Line Printer Modification for Better Gray Level Pictures," *Computer Graphics and Image Processing*, Vol. 6, pp. 485–491.

HARALICK, R. M. 1973. "Glossary and Index to Remotely Sensed Image Pattern Recognition Concepts," *Pattern Recognition*, Vol. 5, pp. 391–403.

JENSEN, J. R. 1983. "Education Image Processing: An Overview," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 1151–1157.

JENSEN, J. R., and M. E. HODGSON. 1983. "Remote Sensing Brightness Maps," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 93–102.

JENSEN, J. R., P. J. PACE, and E. J. CHRISTENSEN. 1983. "Remote Sensing Temperature Mapping: The Thermal Plume Example," *The American Cartographer*, Vol. 10, pp. 111–127.

MACDOUGALL, E. B. 1976. *Computer Programming for Spatial Problems*, London: Edward Arnold Ltd.

MONMONIER, M. S. 1979. "Modeling the Effect of Reproduction Noise on Continuous-Tone Area Symbols," *The Cartographic Journal*, Vol. 16, pp. 86–96.

MONMONIER, M. S. 1980. "The Hopeless Pursuit of Purification in Cartographic Communication: A Comparison of Graphic Arts and Perceptual Distortions of Graytone Symbols," *Cartographica*, Vol. 17, pp. 24–39.

PETERSON, M. P. 1979. "An Evaluation of Unclassed Cross-Line Choropleth Mapping," *The American Cartographer*, Vol. 6, pp. 21–37.

ROBINSON, A. H., R. SALE and J. MORRISON. 1978. *Elements of Cartography*. New York: John Wiley & Sons, Inc.

SMITH, R. M. 1980. "Improved Areal Symbols for Computer Line-Printer Maps," *The American Cartographer*, Vol. 7, pp. 51–58.

STOESSEL, O. C. 1972. "Standard Color Screen Tint Systems for Department of Defense Mapping, Charting and Geodesy Services: Part II. Standard Printing Screen System," *Proceedings*, ACSM Fall Technical Meeting, pp. 111–149.

TOBLER, W. R. 1973. "Choropleth Maps without Class Intervals?" *Geographical Analysis*, Vol. 5, pp. 259–262.



OFEK06778

# 6

# Image Preprocessing

## INTRODUCTION

The ideal or perfect remote sensing system has yet to be developed. Also, the terrain is amazingly complex and does not lend itself well to being recorded by relatively simplistic remote sensing devices that have constraints such as spatial, spectral, temporal, and radiometric resolution. Consequently, error creeps into the data acquisition process and can degrade the quality of the remote sensor data collected. These errors may have an impact on the accuracy of subsequent human or machine-assisted image analysis. Therefore, it is usually necessary to *preprocess* the remotely sensed data prior to analyzing it to remove some of these errors.

*Image restoration* is concerned with the correction of distortion, degradations, and noise introduced during the imaging process (Moik, 1980). Image restoration produces a corrected image that is as close as possible, both geometrically and radiometrically, to the radiant energy characteristics of the original scene (Estes et al., 1983).

To correct the image data, internal and external errors must be determined. *Internal errors* are created by the sensor itself. They are generally systematic (predictable) and stationary (constant), and may be determined from prelaunch or in-flight calibration measurements. *External errors* are due to platform perturbations and the modulation of scene characteristics, which are variable in nature. Such unsystematic errors may be determined by relating points on the ground (i.e., ground control points) to sensor system measurements. Radiometric and geometric error are the most common types of error encountered in remotely sensed imagery.



0981 OFEK06779

## RADIOMETRIC CORRECTION OF REMOTE SENSOR DATA

Multispectral scanning systems (MSS) such as those described in Chapter 2 require calibration and data correction to provide the user with reliable sensor data. A most important operation is to determine the relationship between the individual detector output and the input scene radiance, and to correct the data for any detector malfunctions or distortions.

### Detector Response Error

Remote sensors use detectors that convert the sensed radiance into a voltage or brightness value. The Landsat MSS has six detectors operable during each scan measuring in four spectral regions, while the TM has 16 detectors functioning during each scan measuring near- and middle-infrared energy in six spectral regions. There are several detector anomalies that can arise, including line dropouts, striping, and/or line start problems.

If one of the six detectors in the Landsat MSS or one of the 16 detectors in the TM fails to function during a scan, this can result in a brightness value of zero for every pixel, $j$, in a particular line, $i$. This is often called a *line dropout* and may appear as a completely black line in the band, $k$, of imagery. Obviously, this is a very serious condition. There is no way to restore data that were never acquired; however, it is possible to improve the visual interpretability of the data by introducing estimated brightness values for each bad scan line. The first problem is to locate each bad line. A simple thresholding algorithm can flag any scan line having a mean brightness value at or near zero. Once identified, it is then possible to evaluate the output for a pixel in the preceding line ($BV_{i-1,j,k}$) and succeeding line ($BV_{i+1,j,k}$) and assign the output pixel ($BV_{i,j,k}$) in the "dropout" line the average of these two brightness values:

$$BV_{ijk} = \text{Int}\left(\frac{BV_{i-1,j,k} + BV_{i+1,j,k}}{2}\right) \tag{6-1}$$

This is performed for every pixel in a bad scan line (Guindon, 1984). The result is an image consisting of interpolated data every $n$th line that is much more visually interpretable than one with horizontal black lines running systematically throughout the entire image. The restoration method is also applicable to dropout lines that occur randomly.

Sometimes, a detector does not fail completely, but simply goes out of adjustment (e.g., provides readings perhaps twice as great as the other detectors for the same band). This is often referred to as $n$-line *striping* or *banding*. The data are valid, but should be corrected (or restored) to have the same general contrast as the other detectors per scan. First, the bad scan lines must be identified in the scene. This is usually accomplished by computing a histogram of the values for each of the $n$ detectors over a homogeneous area such as a body of water. If one detector's mean or median is significantly different from the others, it is probable that this detector is



OFEK06780

out of adjustment. It may require a bias (additive or subtractive) correction or a more severe gain (multiplicative) correction. Then the errors arising from the maladjusted detector are not as noticeable.

Occasionally, scanning systems fail to collect data at the beginning of a scan line. This is called a *line start* problem. Also, a detector may abruptly stop collecting data somewhere along a scan and produce results similar to the line dropout discussed previously. Ideally, when data are not collected the sensor system would be programmed to remember what was not collected and place any good data in their proper location within the scan. Unfortunately, this is not always the case. For example, the first pixel in band $k$, on line $i$ (i.e., $BV_{i,1,k}$) might be improperly located at column 50 (i.e., $BV_{i,50,k}$). If the line start problem is always associated with a horizontal bias of 50 columns, it is a simple horizontal adjustment. However, if the amount of the line start displacement is random, it is very difficult to restore the data without extensive human interaction on a line-by-line basis. A considerable number of MSS data collected by Landsats 2 and 3 have this problem. Data not recorded by the detectors can never be restored.

### Atmospheric Effects

There is really no such thing as atmospheric "error" in remotely sensed data. The energy emanating from the sky and recorded by the sensor is a true signal even when it completely destroys our ability to measure the spectral reflectance patterns

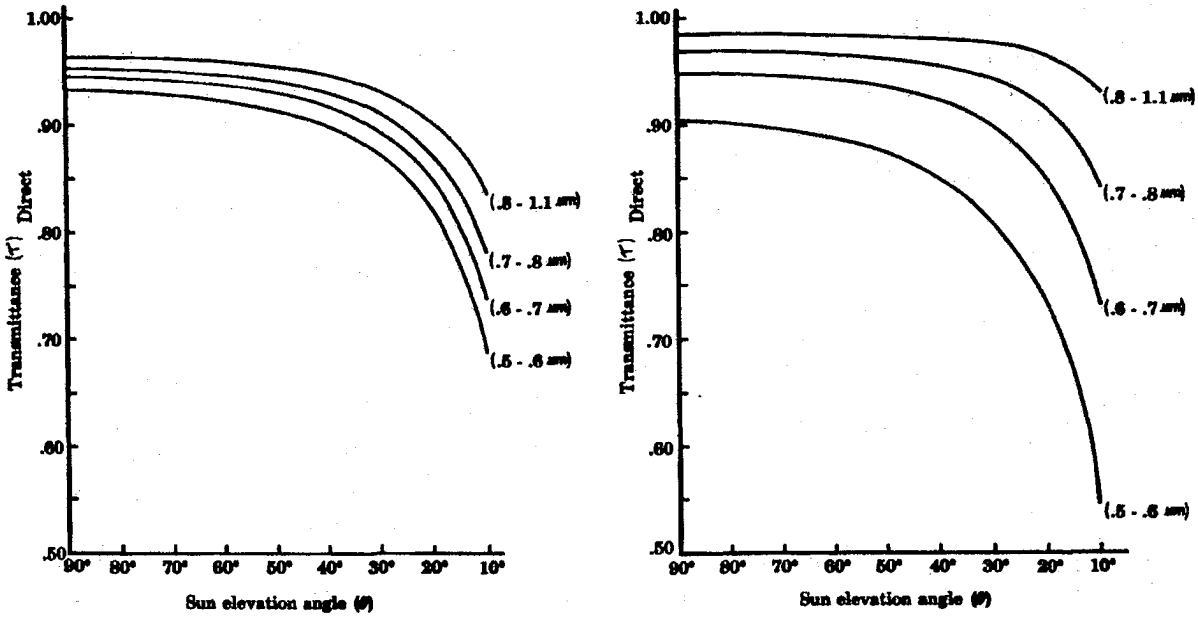



Figure 6-1 (a) Effects of molecular (Rayleigh) scattering and sun angle on incident radiation transmitted to the earth's surface; (b) effects of aerosol (Mie) scattering and sun angle on radiation transmitted to the earth's surface. (From Taranik, 1978.)



OFEK06781

of the terrain. Nevertheless, we often consider the deleterious effects of atmospheric scattering and absorption as being sources of error that can minimize our ability to extract useful terrain information from remotely sensed data.

Solar radiation is largely unaffected as it travels through the vacuum of space. However, when it interacts with the earth's atmosphere it is selectively scattered and absorbed. This introduces external (unsystematic) radiometric distortion into remotely sensed data. For Landsat MSS data, the atmospheric scattering of green (band 4) wavelength energy is four times greater than the scattering of infrared (band 6) radiation (Fig. 6–1). Atmospheric absorption by water vapor and other gases in the atmosphere mostly affects radiation of wavelengths longer than 0.8 μm, especially MSS band 7 (Fig. 6–2). Thus atmospheric scattering adds brightness, whereas atmospheric absorption subtracts brightness from landscape spectral measurements (Fig. 6–3). The effect of the atmosphere on spectral brightness creates problems when trying to compare in situ measurements of spectral response and MSS spectral measurements for the same geographic area. It can also effect classification accuracy. Fortunately, it is possible to apply atmospheric scattering adjustments to the MSS or TM data to minimize some of these effects. Two techniques will be discussed which are based primarily on the fact that infrared data (> 0.7 μm) are largely free of atmospheric scattering effects, whereas the visible region (0.4 to 0.7 μm) is strongly influenced by them (Taranik, 1978).

**Histogram adjustment.** This involves evaluating the histograms of the various bands of remotely sensed data of the desired scene. Normally, the data collected in the visible wavelengths (e.g., TM bands 1 to 3) have a higher minimum value because of the increased atmospheric scattering taking place in these wavelengths. For example, in the data shown in Fig. 6–4, the TM band 1 data have a minimum of 51 and a maximum of 242. Conversely, atmospheric absorption subtracts brightness from the data recorded in the longer wavelength intervals [e.g., TM bands 4, 5,

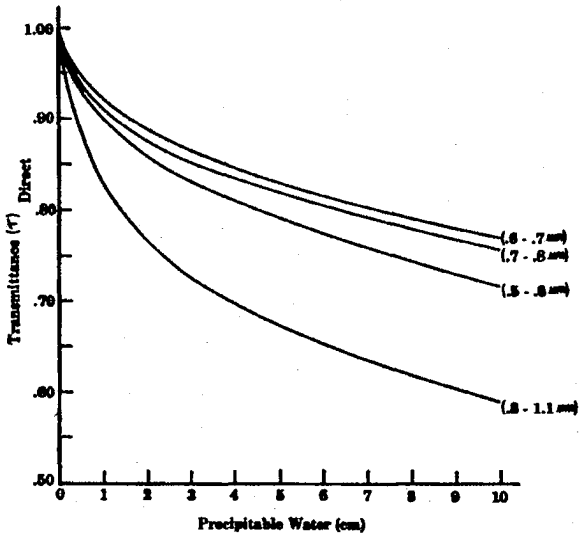



Figure 6–2 Effects of atmospheric water vapor absorption on radiation transmitted to the earth's surface. (From Taranik, 1978.)



OFEK06782

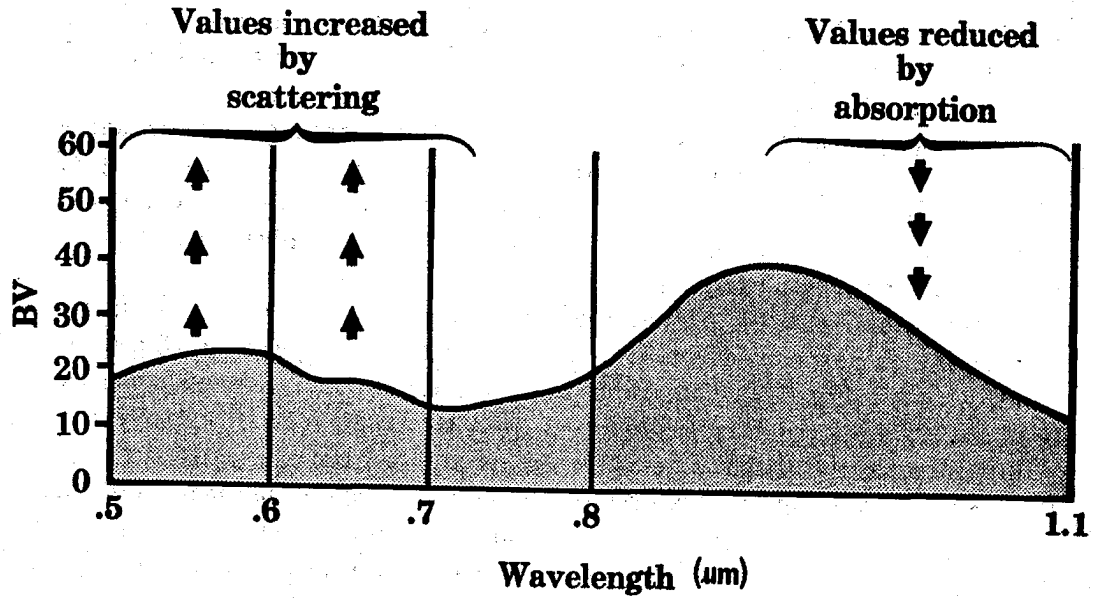



Figure 6–3 Combined effects of scattering and absorption on the brightness values (BV) eventually produced by the Landsat MSS sensor system. (From Taranik, 1978.)

and 6 (band 6 in our discussion is 2.08 to 2.35 $\mu$m)]. This effect commonly causes data from the infrared bands to have minimums close to zero, even when no objects in the scene truly have a reflectance of zero (Fig. 6–4).

If the histograms in Fig. 6–4 are shifted to the left so that zero values appear in the data, the effects of atmospheric scattering will be somewhat minimized. This simple algorithm models the first-order effects of atmospheric scattering, or "haze." It is based on a subtractive bias established for each spectral band. The bias is determined by evaluating the original histograms discussed previously. The atmospheric-effects correction algorithm is defined as

$$\text{output } BV_{i,j,k} = \text{input } BV_{i,j,k} - \text{bias} \qquad (6\text{-}2)$$

where input $BV_{i,j,k}$ = input pixel value at line $i$ and column $j$ of band $k$

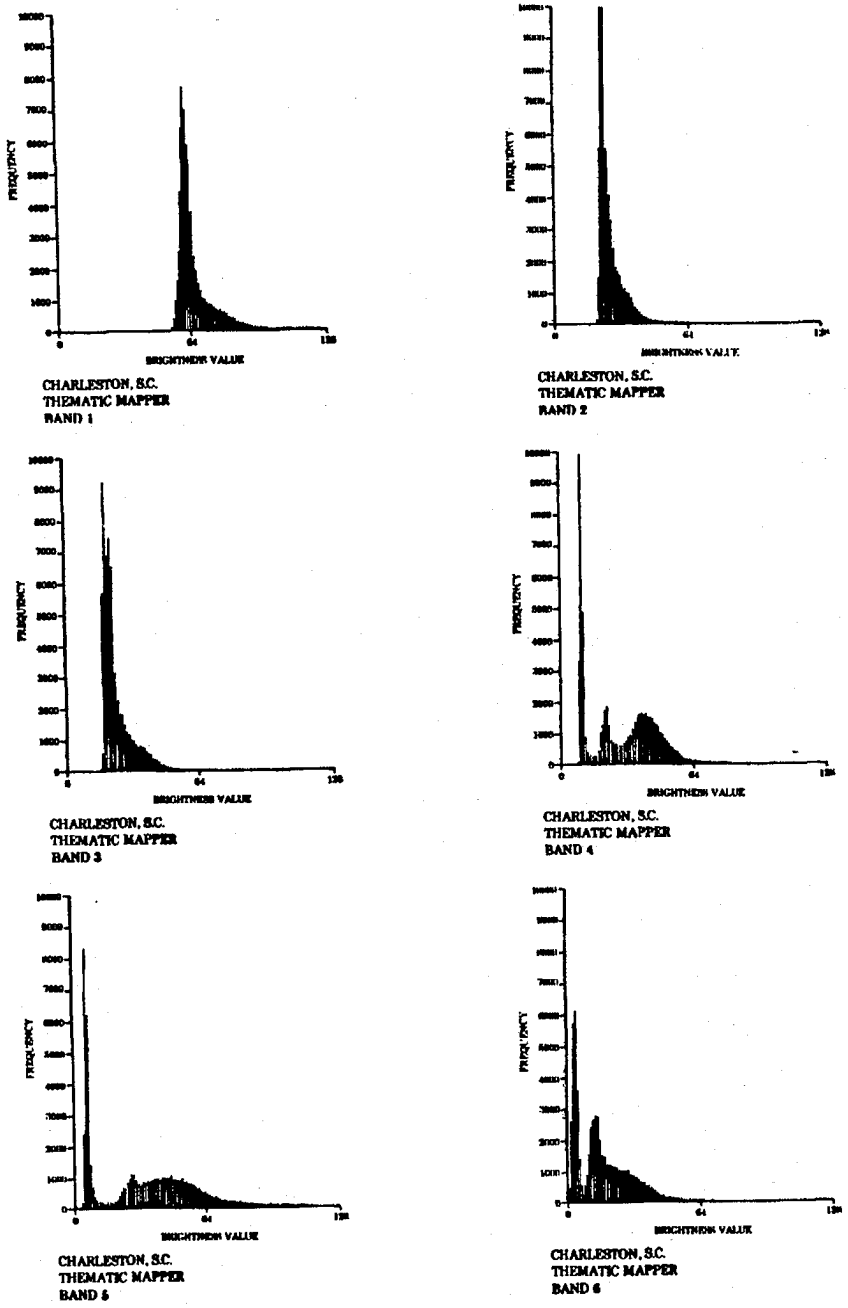
output $BV_{i,j,k}$ = corrected pixel value at the same location

In our example, the appropriate bias was determined for each of the histograms in Fig. 6–4 and subtracted from the data. The histograms of the corrected data are displayed in Fig. 6–5. It was not necessary to adjust bands 5 and 6 for first-order atmospheric effects because they originally had minimums of zero.

**Regression adjustment.** This atmospheric (haze) correction technique requires that the analyst identify an area in an image either in shadow or in homogeneous deep, nonturbid water. The pixel brightness values for this area are then extracted from each of the bands of digital data under investigation. Then, for each pixel in the area, the brightness values of a visible band (e.g., TM band 1) are plotted against


OFEK06783

# ORIGINAL



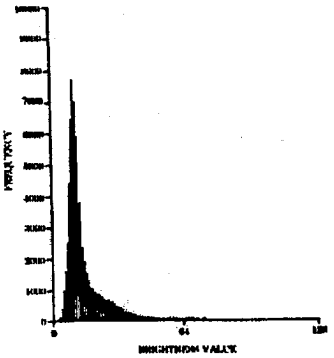

| BAND | MINIMUM | MAXIMUM |
|---|---|---|
| 1 | 51 | 242 |
| 2 | 17 | 115 |
| 3 | 14 | 131 |
| 4 | 4 | 105 |
| 5 | 0 | 193 |
| 6 | 0 | 128 |

**Figure 6–4** Original histograms of six bands of the Charleston, S.C., thematic mapper scene. Atmospheric scattering in the visible regions has increased the minimum brightness values in bands 1, 2, and 3. Generally, the shorter the wavelengths sensed by each band, the greater the offset from a brightness value of zero.

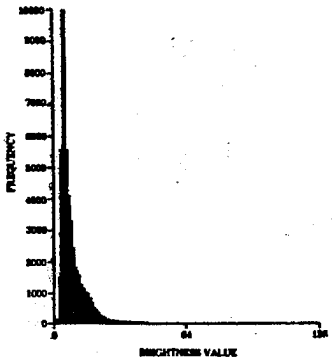


OFEK06784

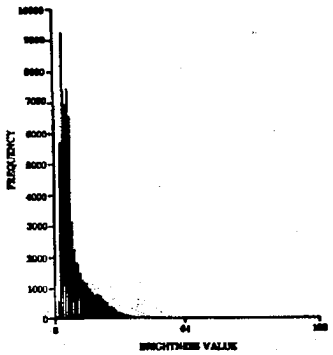
# ADJUSTED



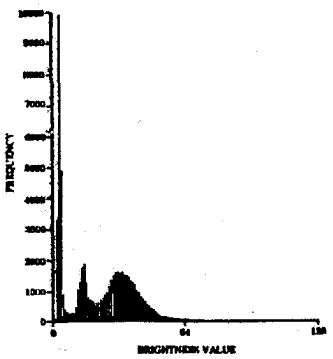
CHARLESTON, S.C.
THEMATIC MAPPER
BAND 1



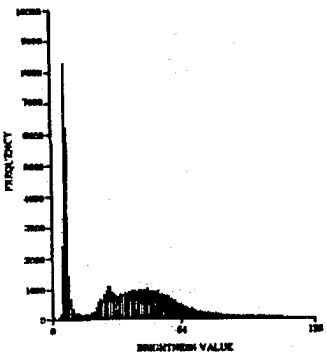
CHARLESTON, S.C.
THEMATIC MAPPER
BAND 2



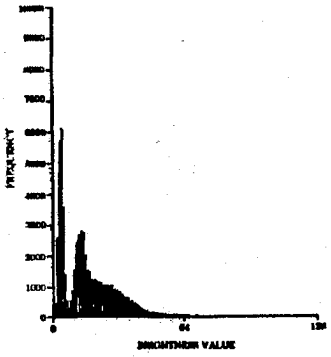
CHARLESTON, S.C.
THEMATIC MAPPER
BAND 3



CHARLESTON, S.C.
THEMATIC MAPPER
BAND 4



CHARLESTON, S.C.
THEMATIC MAPPER
BAND 5



CHARLESTON, S.C.
THEMATIC MAPPER
BAND 6

| BAND | MINIMUM | MAXIMUM |
|---|---|---|
| 1 | 0 | 191 |
| 2 | 0 | 98 |
| 3 | 0 | 117 |
| 4 | 0 | 91 |
| 5 | 0 | 193 |
| 6 | 0 | 128 |

**Figure 6–5** Result of applying a simple histogram adjustment atmospheric scattering correction to the data shown in Fig. 6–4. Only the first four thematic mapper bands required the adjustment. This method does not correct for atmospheric absorption.



OFEK06785

the corresponding values found at the same location in an infrared band (e.g., TM band 6). For example, 25 pixels of open water were extracted from TM band 1 and TM band 6. The data were plotted as a scattergram in Fig. 6–6. A straight line was then "fit" through the distribution using a least-squares technique. If no haze or atmospheric scattering exists, the line would be expected to pass through the origin of the distribution. However, this is rarely the case. Instead, the line usually intersects the $x$-axis (in this case the TM band 1 axis at a brightness value of 48). This $x$-intercept represents the amount of bias caused by atmospheric scattering. It is the amount that should be subtracted from all the original data in TM band 1 for this scene. Data from each of the other bands are then plotted against the band 6 data in a similar fashion to derive their offset values. This technique requires many more passes through the digital data than when using the histogram adjustment method discussed previously. There is no guarantee that it will provide superior results.

The subtraction of a bias or offset as described in these two techniques results in an image that is low in contrast. Therefore, these techniques are rarely employed without also using a gain (i.e., multiplicative) adjustment to the new brightness values. This amounts to first subtracting a bias from each BV and then multiplying the resulting BV by a constant (gain) to expand the values to fill the entire dynamic range of the output device. This is a form of linear contrast stretching (discussed in Chapter 7). It improves the visual interpretability of the atmospherically corrected remotely sensed data.

The atmospheric corrections discussed here deal only with simple haze corrections. Adjustments for atmospheric absorption are difficult to determine because atmospheric absorption is largely a function of water vapor in the atmosphere. The

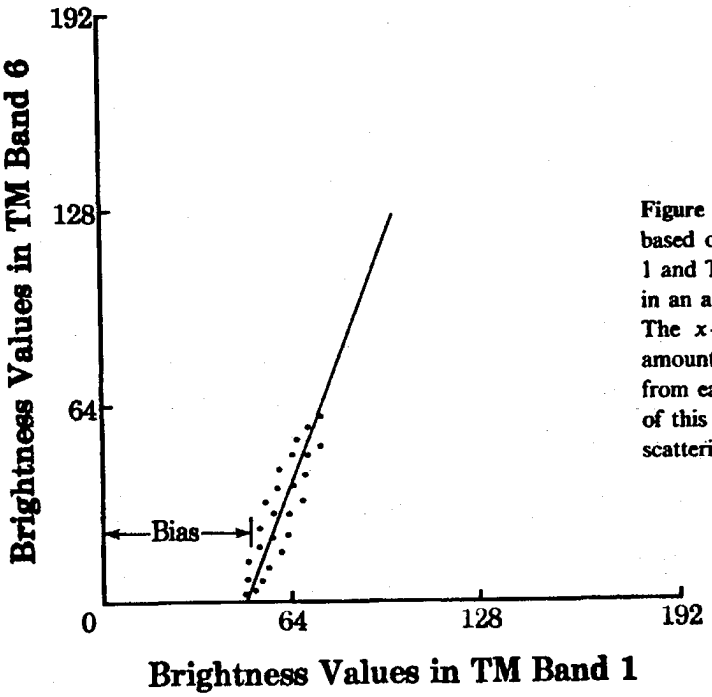



**Figure 6–6** Atmospheric haze correction based on a linear regression of TM band 1 and TM band 6 brightness values found in an area of deep, nonturbid open water. The $x$-intercept (a value of 48) is the amount of bias that should be subtracted from each brightness value in TM band 1 of this data set to adjust for atmospheric scattering.



OFEK06786

amount of water vapor may be variable from scene to scene, but can be modeled for one scene if sufficient weather station data are available on total precipitable water for the date and time that the MSS data are acquired. Even then, much more sophisticated models must be employed, such as those summarized in Turner (1978), Mekler and Kaufman (1980), and Jackson et al. (1983).

## GEOMETRIC CORRECTION OF REMOTE SENSOR DATA

Remotely sensed data usually contain both systematic and nonsystematic geometric errors. Some of the more important sources are summarized in Table 6–1 (Bernstein and Ferneyhough, 1975; Bernstein, 1983). These errors may be divided into two classes: those that can be corrected using data from platform ephemeris and knowledge of internal sensor distortion, and those that cannot be corrected with acceptable accuracy without a sufficient number of ground control points. A *ground control point* (GCP) is a point on the surface of the earth where both image coordinates

**TABLE 6–1** Sources of Image Geometry Errors in Landsat MSS and TM Data

| |
|---|
| *Systematic distortions* |
| *Scan skew:* Caused by the forward motion of the platform during the time required for each mirror sweep. The ground swath is not normal to the ground track but is slightly skewed, producing cross-scan geometric distortion. |
| *Mirror scan velocity:* The MSS mirror scanning rate is usually not constant across a given scan, producing along-scan geometric distortion. |
| *Panoramic distortion:* The ground area imaged is proportional to the tangent of the scan angle rather than to the angle itself. Because data are sampled at regular intervals, this produces along-scan distortion. |
| *Platform velocity:* If the speed of the platform changes, the ground track covered by successive mirror scans changes, producing along-track scale distortion. |
| *Earth rotation:* The earth rotates as the MSS scans the terrain. This results in a shift of the ground swath being scanned, causing along-scan distortion. |
| *Perspective:* For some applications it is desirable to have the MSS images represent the projection of points on the earth upon a plane tangent to the earth with all projection lines normal to the plane. This introduces along-scan distortion. |
| *Nonsystematic distortions* |
| *Altitude:* If the MSS platform departs from its normal altitude, this produces changes in scale. |
| *Attitude:* One sensor system axis is usually maintained normal to the earth's surface and the other parallel to the spacecraft's direction of travel. If the sensor surface departs from this attitude, geometric distortion results. |

*Source:* Bernstein and Ferneyhough (1975); Bernstein (1983).



OFEK06787

(measured in rows and columns) and map coordinates (measured in degrees of latitude and longitude, feet, or meters) can be identified. Those geometric distortions that can be corrected through analysis of sensor characteristics and ephemeris include scan skew, mirror-scan velocity nonlinearities, panoramic distortion, spacecraft velocity, and perspective geometry (including earth curvature). Those that can be corrected only through the use of GCPs are sensor system attitude (roll, pitch, and yaw) and/or altitude (Bernstein, 1983).

Most of the digital Landsat MSS data acquired through the EOSAT Company have already had systematic error removed by the Master Data Processor at the NASA Goddard Space Flight Center (NASA, 1979; Billingsley, 1983). Unless otherwise processed, however, the unsystematic error remains in the image, making it nonplanimetric. This section focuses on the two geometric corrections most often used by earth scientists to make the digital remote sensor data truly useful; geometric rectification and geometric registration.

*Rectification* is the process by which the geometry of an image area is made planimetric (Haralick, 1973). It may not, however, remove distortion caused by topographic relief displacement in images. The process almost always involves relating GCP pixel coordinates (row and column) with their map coordinate counterparts (e.g., meters in northing and easting in a transverse Mercator map projection). This is the most precise geometric correction since each pixel can be referenced not only by its row and column in a matrix after rectification is completed, but it is also rigorously referenced in degrees, feet, or meters in a standard map projection. Whenever accurate area, direction, and distance measurements are required, geometric rectification is required. This is often referred to as an image-to-map rectification.

Conversely, there may be times when such geometric precision is not necessary. For example, one might want simply to compare two images obtained at different dates to see if any change has taken place in them. Certainly, it is possible to rectify both of the images to a standard map projection and then evaluate them (and this is often done), but this may not be necessary. Instead, *image registration* might be applied, which is the translation and rotation alignment process by which two images of like geometries and of the same set of objects are positioned concident with respect to one another so that corresponding elements of the same ground area appear in the same place on the registered images (Haralick, 1973). This is often called image-to-image registration.

The same general image processing principles are used in both image rectification and image registration. The difference is that in image rectification the reference is a map in a standard map projection, whereas in image registration the reference is another image. It should be obvious that if an image is used as the reference base (rather than a map), any other image registered to it will inherit the geometric errors existing in the reference image. Because of this characteristic, most serious earth science remote sensing research is based on analysis of data that have been rectified to a map base. Therefore, this discussion will focus on image rectification.



OFEK06788

### Geometric Rectification

Two basic operations must be performed in order to geometrically rectify a remotely sensed image to a map coordinate system (Green, 1983):

1. The geometric relationship between the input pixel location (row and column) and the associated map coordinate of this same point $(x, y)$ must be identified. This will establish the nature of the geometric coordinate transformation that must be applied to rectify or relocate every pixel in the original input image $(x', y')$ to its proper position in the rectified output image $(x, y)$. This process is called *spatial interpolation.*
2. Pixel brightness values must be determined. Unfortunately, there is not a direct one-to-one relationship between the movement of input pixel values to output pixel locations. It will be shown that often a pixel in the rectified output image requires a value from the input pixel grid that does not fall neatly on a row-and-column coordinate. When this occurs there must be some mechanism for determining the brightness value (BV) to be assigned to the new rectified pixel. This process is called *intensity interpolation*.

**Spatial interpolation using coordinate transformations.** As discussed earlier, some distortions in MSS data may be removed or mitigated using techniques that model systematic orbital and sensor characteristics. Unfortunately, this does not remove error produced by changes in attitude (roll, pitch, and yaw) or altitude. Such errors are generally unsystematic and are best removed by identifying ground control points (GCPs) in the original imagery and on the reference map and then mathematically modeling the geometric distortion present. This procedure generally requires that polynominal equations be fit to the control point data using least-squares criteria to model the corrections directly in the image domain without explicitly identifying the source of the distortion. For moderate distortions in a relatively small area of an image (e.g., a quarter of a Landsat or TM scene), a six-parameter affine transformation is sufficient to rectify the imagery to a geographic frame of reference.

This type of transformation can model six kinds of distortion in the remote sensor data, including: translation in $x$ and $y$, scale changes in $x$ and $y$, skew, and rotation (Billingsley, 1983). When all six operations are combined into a single expression, it becomes

$$x' = a_0 + a_1 x + a_2 y$$
$$y' = b_0 + b_1 x + b_2 y \qquad (6\text{-}3)$$

where $x$ and $y$ are positions in the output-rectified image or map, and $x'$ and $y'$ represent corresponding positions in the original input image.

By the use of these six coordinate transform coefficients that model distortions in the original scene, it is then possible to transfer (relocate) pixel values from the


OFEK06789

original distorted image $x'$, $y'$ to the grid of the rectified output image, $x$, $y$. However, before applying the rectification to the entire set of data, it is important to determine how well the six coefficients derived from the least-squares regression of the initial GCPs accounts for the geometric distortion in the input image. The method used most often involves computation of the root mean square error ($RMS_{error}$) for each of the ground control points.

Let us consider for a moment the nature of the GCP data. We first identify a point in the image, such as a street intersection. Its original column and row coordinates in the original image we will call $x_{orig}$ and $y_{orig}$. The $x$, $y$ position of the same road intersection is then measured from the reference map in degrees, feet, or meters. These two sets of GCP coordinates and many others selected by the analyst are used to compute the six coefficients discussed in equation (6-3). Now, if we were to input the map $x$ and $y$ values for the first GCP back into equation (6-3) with all the coefficients in place, we would get computed $x'$ and $y'$ values that are supposed to be the location of this point in the image space. Ideally, $x'$ would equal $x_{orig}$ and $y'$ would equal $y_{orig}$. Unfortunately, this is rarely the case. Any discrepancy between the values represents image geometric distortion not corrected for by the six-coefficient coordinate transformation.

A simple way to measure such distortion is by computing the $RMS_{error}$ for each control point using the equation

$$RMS_{error} = \sqrt{(x' - x_{orig})^2 + (y' - y_{orig})^2} \tag{6-4}$$

where $x_{orig}$ and $y_{orig}$ are the original row and column coordinates of the GCP in the image, and $x'$ and $y'$ are the computed or estimated coordinates in the original image. The square root of the squared deviations represents a measure of the accuracy of this GCP in the image. By computing $RMS_{error}$ for all GCPs, it is possible (1) to see which GCPs exhibit the greatest error, and (2) to sum all the $RMS_{error}$.

Normally, the user specifies a certain amount (a threshold) of acceptable total RMS error. If an evaluation of the total $RMS_{error}$ reveals that a given set of control points exceeds this threshold, it is common practice to (1) delete from the analysis the GCP that has the greatest amount of individual error, (2) recompute the six coefficients, and (3) recompute the $RMS_{error}$ for all points. This process continues until one of the following occurs: the total $RMS_{error}$ is less than the threshold specified, or too few points remain to perform a least-squares regression to compute the coefficients. Once the acceptable $RMS_{error}$ is reached, one can proceed to the intensity interpolation phase of geometric rectification, which attempts to fill an output grid ($x$, $y$) with brightness values found within the original input grid ($x'$, $y'$).

**Intensity interpolation.** This process involves the extraction of a brightness value from an $x'$, $y'$ location in the original (distorted) input image and its relocation to the appropriate $x$, $y$ coordinate location in the rectified output image. This "pixel filling" logic may be used to produce the output image line by line, column by column (Castleman, 1978). Most of the time, the $x'$ and $y'$ coordinates to be sampled in the input image are real numbers (i.e., they are not integers). For example, in Fig.



OFEK06790

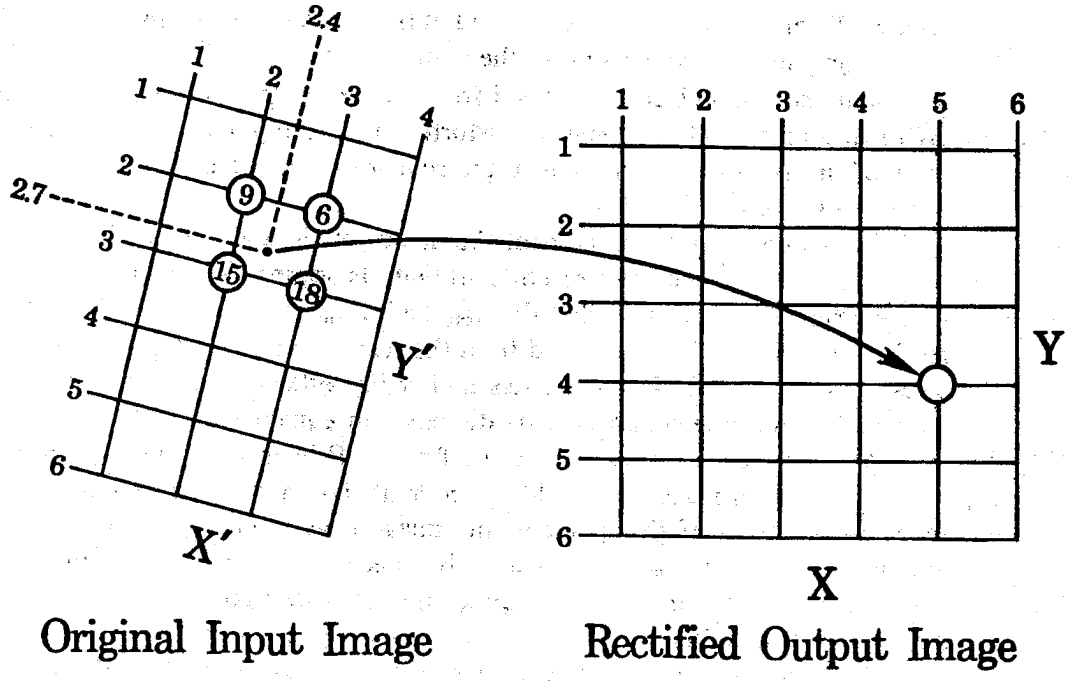



**Figure 6–7** This diagram illustrates how the rectified output grid ($X$, $Y$) is filled with brightness values from the geometrically distorted input grid ($X'$, $Y'$). In particular, we are trying to fill pixel 5, 4 in the output grid with a value from the appropriate location in the input grid. The appropriate location is computed using equation (6-3) with the six necessary coefficients and a value of 5 for the $X$ value and a value of 4 for the $Y$ value in the output image. The equation then computes the $X'$, $Y'$ coordinates of the location in the original image to obtain the appropriate brightness value. The location in this example is 2.4, 2.7. Because these coordinates are not integers, it is necessary to use an intensity interpolation technique to compute the exact brightness value to be placed in $X$, $Y$ location 5, 4 of the rectified image.

6–7 we see that pixel 5, 4 ($x$, $y$) in the output image is to be "filled" with the value from coordinates 2.4, 2.7 ($x'$, $y'$) in the input image. When this occurs, there are several methods of brightness value (BV) interpolation that can be applied, including nearest-neighbor, bilinear interpolation, and cubic convolution. The practice of brightness value interpolation is commonly referred to as *resampling*.

In zero-order, or *nearest-neighbor* interpolation, the brightness value of the pixel closest to the $x'$, $y'$ coordinate specified is assigned to the output $x$, $y$ coordinate. For example, in Fig. 6–7, the output pixel 5, 4 ($x$, $y$) requests the brightness value in the original input image at location 2.4, 2.7 ($x'$, $y'$). There is no value at this location; however, there are nearby values at the integer grid intersections. A nearest-neighbor rule would assign the output pixel ($x$, $y$) the value of 15, which is the value found at the nearest input pixel.

This is a computationally efficient procedure. It is especially liked by earth scientists because it does not alter the pixel brightness values during resampling. It is often the very subtle changes in brightness value that make all the difference when discriminating between one type of vegetation versus another, an edge associated



OFEK06791

with a geologic lineament, or different levels of turbidity or temperature in a lake. Other interpolation techniques to be discussed use averages to compute the output intensity value, often removing valuable spectral information.

First-order, or *bilinear interpolation* assigns output pixel values by interpolating brightness values in two orthogonal directions in the input image. It basically fits a plane to the four pixel values nearest the desired position $(x', y')$ in the input image and then computes a new brightness value based on the weighted distances to these points. For example, the distances from the requested $x'$, $y'$ position at 2.4, 2.7 in the input image in Fig. 6–7 to the closest four input pixel coordinates (2,2; 3,2; 2,3; 3,3) are computed in Table 6–2. The closer a pixel is to the desired $x'$, $y'$ location, the more weight it will have in the final computation of the average. The weighted average of the new brightness value ($BV_{wt}$) is computed according to the equation

$$BV_{wt} = \frac{\sum_{k=1}^{4} (Z_k / D_k^2)}{\sum_{k=1}^{4} (1/D_k^2)} \tag{6-5}$$

where $Z_k$ are the surrounding four data point values, and $D_k^2$ are the distances squared from the point in question $(x', y')$ to these data points. In our example, the weighted average of $BV_{wt}$ is 13.54 (truncated to 13), as shown in Table 6–2. The average without weighting is 12. In many respects this method acts as a spatial moving filter that subdues extremes in brightness value throughout the output image. The method is also more computationally demanding than the nearest-neighbor method.

*Cubic convolution* resampling assigns values to output pixels in much the same manner as bilinear interpolation, except that the weighted values of 16 input pixels surrounding the location of the desired $x'$, $y'$ pixel are used to determine the value of the output pixel.

**Implementation.** To appreciate digital image rectification, it is useful to demonstrate the logic by applying it to a real data set, such as the Charleston, S.C.,

**TABLE 6–2** Bilinear Interpolation of a Weighted Brightness Value ($BV_{wt}$) at Location $x'$, $y'$ Based on the Analysis of Four Sample Points in Fig. 6–7

| Sample point location column row | Value at sample point, $Z$ | Distance from $x'$, $y'$ to the sample point $D$ | $D_k^2$ | $Z/D_k^2$ | $1/D_k^2$ |
|---|---|---|---|---|---|
| 2, 2 | 9 | 0.806 | 0.65 | 13.85 | 1.530 |
| 3, 2 | 6 | 0.922 | 0.85 | 7.05 | 1.176 |
| 2, 3 | 15 | 0.500 | 0.25 | 60.00 | 4.000 |
| 3, 3 | 18 | 0.670 | 0.45 | 40.00 | 2.222 |
| | | | | Σ 120.90 | Σ 8.928 |
| | | | | $BV_{wt} = 120.90/8.928 = 13.54$ | |



OFEK06792

thematic mapper image. In doing so we are concerned with the rectification of the remotely sensed data to a map (i.e., an image-to-map rectification). Obviously, the type of map and its associated projection are important parameters and must be carefully selected.

There are several types of map projections to be considered in this context. On an equivalent (equal-area) map projection, a circle of diameter $n$ drawn at any location on the map will encompass exactly the same total geographic area. This characteristic is useful if the scientist is interested in comparing land-use area, density, and so on. Unfortunately, to maintain the equal-area attribute, the shapes, angles, and scale in parts of the map may be distorted.

Conversely, a conformal map projection maintains correct shape and distance around a given point on the map. Because angles at each point are correct on conformal maps, the scale in every direction around any point is constant. This allows the analyst to measure distance and direction between relatively near points with good precision. For our purposes, this means that for image areas covering a few contiguous 7.5-minute quadrangle sheets, accurate spatial measurement is possible if the data are rectified to a conformal map projection.

One of the most often used projections for rectifying remotely sensed data is the transverse Mercator projection. It is made from a normal Mercator projection by rotating the cylinder (the developable surface) so that it lies tangent along a meridian. The central meridians, the equator, and each line 90° from the central meridian are straight lines (Fig. 6–8). The central meridian normally has a constant scale. Any lines parallel to the central meridian are lines of constant scale. The universal transverse Mercator projection often used by the USGS in their topographic mapping program has a central scale factor of 0.9996 and is composed of 60 zones, each 6° of longitude wide, with a central meridian placed every sixth meridian beginning with 177° west.

In our example, the Charleston, S.C., 7.5-minute quadrangle (photorevised 1979) was selected as the appropriate base map with which to rectify the TM data (Fig. 6–9). It lies in UTM zone 17 and has a 1000-m UTM grid overlay. Twenty ground control points were located on the map (Fig. 6–9) and the UTM easting and northing of each point were identified (Table 6–3). The same 20 GCPs were then identified in the TM data according to their row and column coordinates (also in Table 6–3). The GCPs should normally be located throughout the region to be rectified and not congested into one small area simply because there are more easily identifiable points in that area.

The 20 GCPs were input to the least-squares regression procedure previously discussed to identify (1) the coefficients of the coordinate transformation, and (2) the individual and total $RMS_{error}$ associated with the GCPs. A threshold of 0.5 was not satisfied until 13 GCPs were deleted from the analysis. The order in which the 13 GCPs were deleted and the total $RMS_{error}$ found after each deletion is summarized in Table 6–3. The seven GCPs finally selected that produced an acceptable $RMS_{error}$ are shown in Table 6–4. GCP 11 would have been the next point deleted if the threshold had not been satisfied. The six coefficients derived from the seven suitable GCPs are found in Table 6–5.



OFEK06793

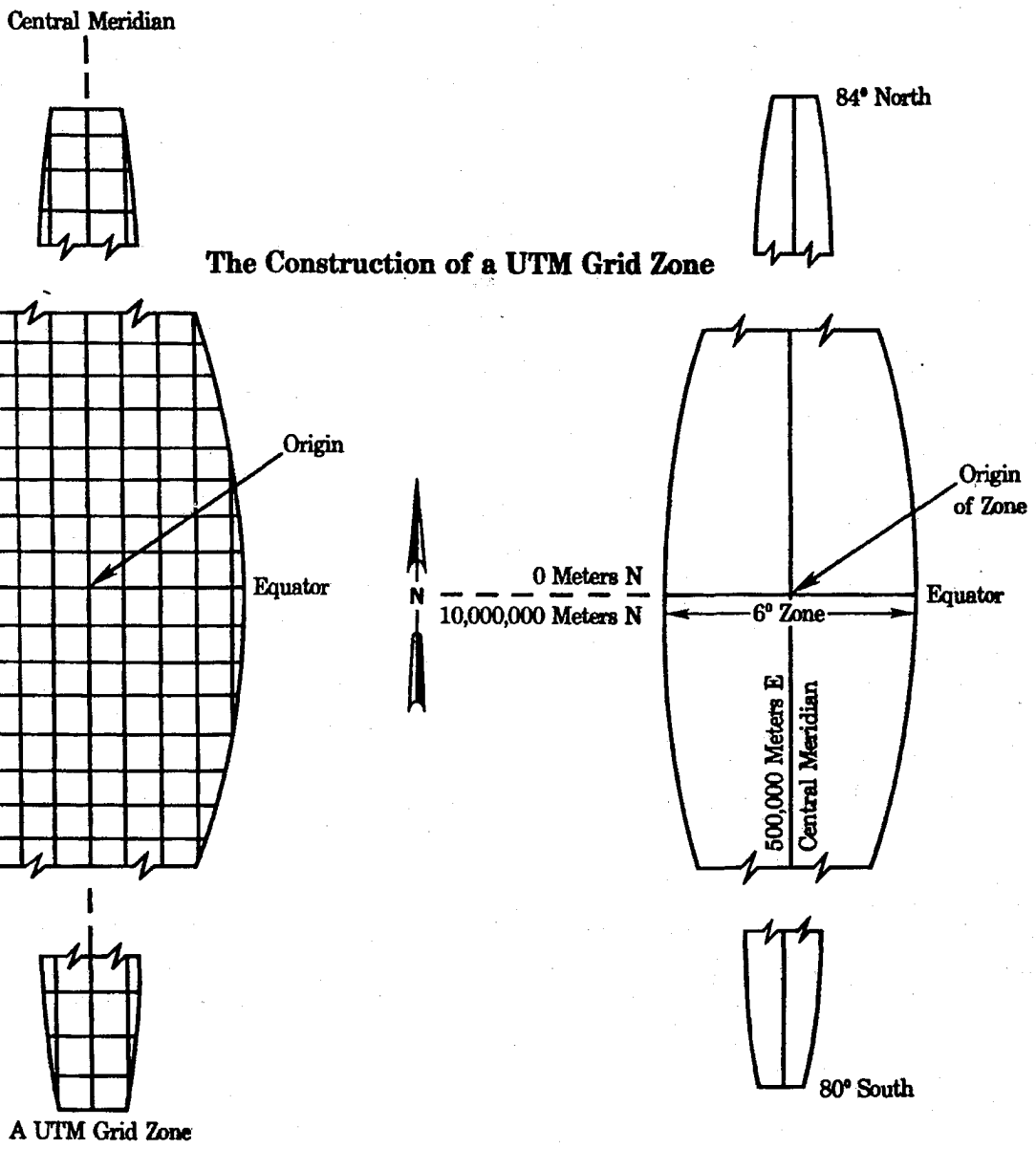



**Figure 6-8** Universal transverse Mercator (UTM) grid zone with associated parameters. This projection is often used when rectifying remote sensor data to a base map. It is found on U.S. Geological Survey 7.5-minute and 15-minute quadrangles.

It is instructive at this point to demonstrate how the six coefficients are computed. This is done using only the final seven GCPs since these represent the set that produced an acceptable $RMS_{error}$ of $< 0.50$. Remember, however, that this same operation was performed with 20 points, then 19, and so on, before arriving at just seven acceptable points.

A least-squares multiple regression approach may be used to compute the coefficients. Two equations are necessary. One equation computes the image $y'$ coordinate



OFEK06794

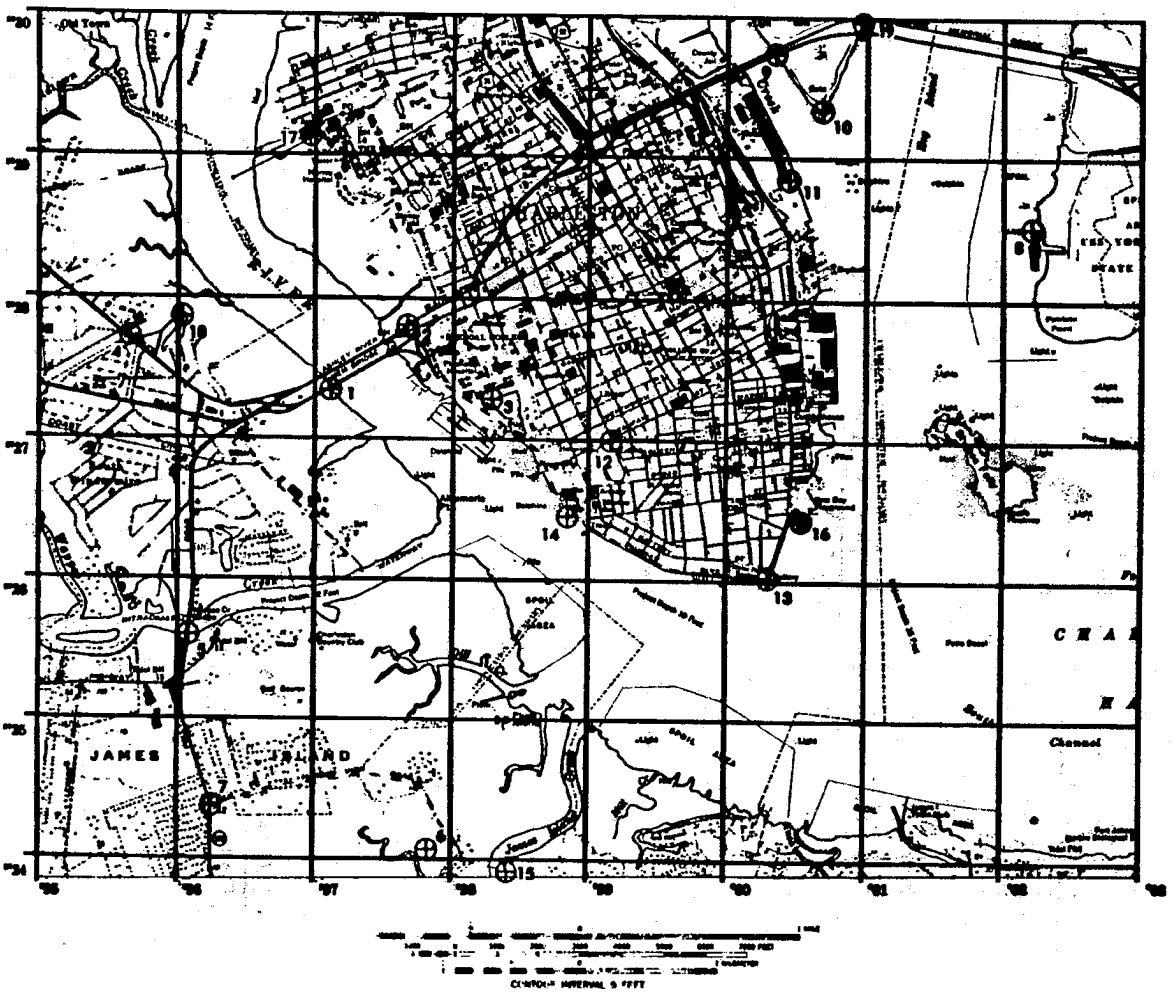



**Figure 6–9** Reduced version of the Charleston, S.C., 7.5-minute quadrangle. The UTM coordinates and 19 of the 20 ground control points (GCPs) used to rectify the Charleston thematic mapper scene are displayed.

(the dependent variable) as a function of the $x$ and $y$ map coordinates (representing two independent variables). The second computes the image $x'$ coordinates as a function of the same map $(x, y)$ coordinates. Three coefficients are determined using each algorithm. The mathematics for computing the column coordinates $(x')$ in the image will now be presented.

**Multiple regression coefficients computation.** First, we will let

$Y =$ either the $x'$ or $y'$ location in the image, depending on which is being evaluated; in this example it will represent the $x'$ values
$X_1 =$ easting coordinate $(x)$ of the map GCP
$X_2 =$ northing coordinate $(y)$ of the map GCP.


OFEK06795

**TABLE 6–3** Characteristics of Twenty Ground Control Points Used to Rectify the Charleston, S.C., Thematic Mapper Scene

| Point number | Order of points deleted[a] | Easting on map, $X_1$ | Northing on map, $X_2$ | $X'$ pixel | $Y'$ pixel | Total $RMS_{error}$ after this point deleted |
|---|---|---|---|---|---|---|
| 1 | 12 | 597,120 | 3,627,050 | 150 | 185 | 0.501 |
| 2 | 9 | 597,680 | 3,627,800 | 166 | 165 | 0.663 |
| 3 | Kept | 598,285 | 3,627,280 | 191 | 180 | — |
| 4 | Kept | 595,650 | 3,627,730 | 98 | 179 | — |
| 5 | 2 | 596,750 | 3,625,600 | 123 | 252 | 6.569 |
| 6 | 13 | 597,830 | 3,624,820 | 192 | 294 | 0.435 |
| 7 | Kept | 596,250 | 3,624,380 | 137 | 293 | — |
| 8 | Kept | 602,200 | 3,628,530 | 318 | 115 | — |
| 9 | Kept | 600,350 | 3,629,730 | 248 | 83 | — |
| 10 | 5 | 600,680 | 3,629,340 | 259 | 93 | 1.291 |
| 11 | Kept | 600,440 | 3,628,860 | 255 | 113 | — |
| 12 | 10 | 599,150 | 3,626,990 | 221 | 186 | 0.601 |
| 13 | 8 | 600,300 | 3,626,030 | 266 | 211 | 0.742 |
| 14 | 6 | 598,840 | 3,626,460 | 211 | 205 | 1.113 |
| 15 | 3 | 598,940 | 3,623,430 | 214 | 295 | 4.773 |
| 16 | Kept | 600,540 | 3,626,450 | 272 | 196 | — |
| 17 | 4 | 596,985 | 3,629,350 | 134 | 123 | 1.950 |
| 18 | 7 | 596,035 | 3,627,880 | 109 | 174 | 0.881 |
| 19 | 11 | 600,995 | 3,630,000 | 269 | 71 | 0.566 |
| 20 | 1 | 601,700 | 3,632,580 | 283 | 12 | 8.542 |
| | | | | Total $RMS_{error}$ with all 20 GCPs used: | | 11.016 |

[a] For example, GCP 20 was the first point deleted. After it was deleted the total $RMS_{error}$ dropped from 11.016 to 8.542. Point 5 was the second point deleted. After it was deleted the total $RMS_{error}$ dropped from 8.542 to 6.569.

**TABLE 6–4** Information concerning the Final Seven Ground Control Points Used to Rectify the Charleston, S.C., Thematic Mapper Scene

| Point number | Easting on map | Adjusted easting $X_1$[a] | Northing on map | Adjusted northing $X_2$[b] | $Y$[c] $X'$ pixel, | $Y'$ pixel |
|---|---|---|---|---|---|---|
| 3 | 598,285 | 2635 | 3,627,280 | 2900 | 191 | 180 |
| 4 | 595,650 | 0 | 3,627,730 | 3350 | 98 | 179 |
| 7 | 596,250 | 600 | 3,624,380 | 0 | 137 | 293 |
| 8 | 602,200 | 6550 | 3,628,530 | 4150 | 318 | 115 |
| 9 | 600,350 | 4700 | 3,629,730 | 5350 | 248 | 83 |
| 11 | 600,440 | 4790 | 3,628,860 | 4480 | 255 | 113 |
| 16 | 600,540 | 4890 | 3,626,450 | 2070 | 272 | 196 |
| | | 24165 | | 22300 | 1519 | 1159 |
| | Minimum = 595,650 | | Minimum = 3,624,380 | | | |

[a] Adjusted easting values ($X_1$) used in the least-squares computation of coefficients. This is an independent variable.
[b] Adjusted northing values ($X_2$) used in the least-squares computation of coefficients. This is an independent variable.
[c] The dependent variable ($Y$) discussed in the text. In this example it was used to predict the $X'$ pixel location.



0998

OFEK06796

**TABLE 6–5** Coefficients Used to Rectify the Charleston, S.C., Thematic Mapper Scene

$$x' = -382.0783 + 0.034187x + (-0.005481)y$$
$$y' = 130162 + (-0.005576)x + (-0.349150)y$$

where $x$ and $y$ are map coordinates, and $x'$ and $y'$ are predicted image coordinates

It is practical to use $X_1$ and $X_2$ instead of $x$ and $y$ to simplify mathematical notation.

The seven coordinates used in the computation of the coefficients are shown in Table 6–4. Notice that the independent variables ($X_1$ and $X_2$) have been adjusted (adjusted value = original value − minimum value) so that the sums of squares or sums of products do not become so large that they overwhelm the precision of the computation in the CPU being used. Note for example, that most of the original northing UTM measurements associated with the map GCPs are already in the range of 3 million meters. The minimums subtracted in Table 6–4 are added back into the analysis at the final stage of coefficient computation. Below are the mathematics necessary to compute the coefficients. It is technical but should be of value to those interested in the fundamental principles of how to compute the coefficients used in equation (6-3).

I. Find $(X^TX)$ and $(X^TY)$ in deviation form:

$n = 7$, the number of control points used

A. First compute:

$$\sum_{i=1}^{n} Y_i = 1519 \qquad \sum_{i=1}^{n} X_{1i} = 24{,}165 \qquad \sum_{i=1}^{n} X_{2i} = 22{,}300$$

$$\bar{Y} = 217 \qquad \bar{X}_1 = 3452.1428 \qquad \bar{X}_2 = 3185.7142$$

$$\sum_{i=1}^{n} Y_i^2 = 366{,}491 \qquad \sum_{i=1}^{n} X_{1i}^2 = 119{,}151{,}925 \qquad \sum_{i=1}^{n} X_{2i}^2 = 89{,}832{,}800$$

$$\sum_{i=1}^{n} X_{1i}Y_i = 6{,}385{,}515 \qquad \sum_{i=1}^{n} X_{2i}Y_i = 5{,}234{,}140 \qquad \sum_{i=1}^{n} X_{1i}X_{2i} = 91{,}550{,}500$$

B. Compute sums of squares

1. $$\sum_{i=1}^{n} X_{1i}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} X_{1i}\right)^2 = 119{,}151{,}925 - \frac{1}{7}(24{,}165)^2$$
$$= 35{,}730{,}896.4285$$

2. $$\sum_{i=1}^{n} X_{2i}^2 - \frac{1}{n}\left(\sum_{i=1}^{n} X_{2i}\right)^2 = 89{,}832{,}800 - \frac{1}{7}(22{,}300)^2$$
$$= 18{,}791{,}373.7142$$



OFEK06797

3. $$\sum_{i=1}^{n} X_{1i}X_{2i} - \frac{1}{n}\left(\sum_{i=1}^{n} X_{1i}\right)\left(\sum_{i=1}^{n} X_{2i}\right)$$

$$= 91{,}550{,}500 - \frac{1}{7}(24{,}165)(22{,}300) = 14{,}567{,}714.2857$$

where

$$(X^TX) = \begin{Bmatrix} 35{,}730{,}896.4285 & 14{,}567{,}714.2857 \\ 14{,}567{,}714.2857 & 18{,}791{,}373.7142 \end{Bmatrix}$$

4. Covariance between $Y$ and $X_1$:

$$\sum_{i=1}^{n} X_{1i}Y_i - \frac{1}{n}\left(\sum_{i=1}^{n} X_{1i}\right)\left(\sum_{i=1}^{n} Y_i\right) = 6{,}385{,}515 - \frac{1}{7}(24{,}165)(1519)$$
$$= 1{,}141{,}710$$

5. Covariance between $Y$ and $X_2$:

$$\sum_{i=1}^{n} X_{2i}Y_i - \frac{1}{n}\left(\sum_{i=1}^{n} X_{2i}\right)\left(\sum_{i=1}^{n} Y_i\right) = 5{,}234{,}140 - \frac{1}{7}(22{,}300)(1519)$$
$$= 395{,}040$$

where

$$(X^TY) = \begin{Bmatrix} 1{,}141{,}710 \\ 395{,}040 \end{Bmatrix}$$

II. Find the inverse of $(X^TX) = (X^TX)^{-1}$.

A. First, find the determinant of the $2 \times 2$ matrix:

$$|X^TX| = (35{,}730{,}896.4285)(18{,}791{,}373.7142) - (14{,}567{,}714.2857)^2$$
$$= 459{,}214{,}334{,}417{,}386.5$$

B. Determine adjoint matrix of $(X^TX)$, where adjoint equals the transpose of the cofactor matrix:

$$\text{Adjoint}^* = \begin{Bmatrix} 18{,}791{,}373.7142 & -14{,}567{,}714.2857 \\ -14{,}567{,}714.2857 & 35{,}730{,}896.4285 \end{Bmatrix}$$

* Note that if

$$A = \begin{Bmatrix} a & b \\ c & d \end{Bmatrix}, \quad \text{then } A^{-1} = \frac{1}{\det A}\begin{Bmatrix} d & -b \\ -c & a \end{Bmatrix}$$

C. Get $(X^TX)^{-1}$ by multiplying the adjoint of $(X^TX)$ by the det $(X^TX)$ under 1.

$$(X^TX)^{-1} = \left(\frac{1}{459{,}214{,}334{,}417{,}386.5}\right)$$
$$\begin{Bmatrix} 18{,}791{,}373.7142 & -14{,}567{,}714.2857 \\ -14{,}567{,}714.2857 & 35{,}730{,}896.4285 \end{Bmatrix}$$



OFEK06798

$$= \begin{Bmatrix} 0.000000041 & -0.000000032 \\ -0.000000032 & 0.000000078 \end{Bmatrix}$$

III. Now compute coefficients, using $a_i = (X^TX)^{-1}(X^TY)$.

A. $$\begin{Bmatrix} 0.000000041 & -0.000000032 \\ -0.000000032 & 0.000000078 \end{Bmatrix} \cdot \begin{Bmatrix} 1{,}141{,}710 \\ 395{,}040 \end{Bmatrix}$$

$$= \begin{matrix} a_1 = (\ 0.000000041)(1{,}141{,}710) + (-0.000000032)(395{,}040) \\ a_2 = (-0.000000032)(1{,}141{,}710) + (\ 0.000000078)(395{,}040) \end{matrix}$$

$$= \ 0.0341877$$
$$= -0.0054810$$

B. Now compute $a_0$, the intercept from $a_0 = \overline{Y} - \left(\sum_{i=1}^{2} a_i \overline{X_i}\right)$. [The minimums of $X_1$ and $X_2$ (595,650 and 3,624,380, respectively) must be accounted for here. See Table 6–4.]

$$a_0 = 217 - [(0.0341877)(3452.1428 + 595{,}650) + (-0.0054810)(3185.7142 + 3{,}624{,}380)] = -382.0783091$$

Therefore, the equation becomes

$$Y = -382.0783091 + 0.0341877X_1 + (-0.0054810)X_2$$

Because we actually evaluated the dependent variable $x'$, this becomes

$$x' = -382.0783091 + 0.0341877x + (-0.0054810)y$$

with $x$ and $y$ representing the map coordinates, and $x'$ being the predicted column coordinate in the input image.

Similar procedures are required to compute the other three coefficients for the row ($y'$) location in the input image. This would require inserting the seven $y'$ pixel values in the equation in Table 6–4 instead of the $x'$ pixel values just used. The $X_1$ and $X_2$ values associated with the $x$, $y$ coordinates of the map remain the same.

With the coefficients established, it was necessary to (1) identify the UTM coordinates of the area on the map to be rectified, (2) select the type of resampling to be performed (e.g., nearest-neighbor, bilinear, or cubic convolution), and (3) specify the desired output pixel size. The northwesternmost coordinate was N(3,630,000), E(595,000) and the southeasternmost coordinate was N(3,624,000), E(603,000). A nearest-neighbor resampling algorithm was selected with a desired output pixel dimension grid of 30 × 30 m. Obviously, the finer the dimension of the output grid, the greater the number of computations required to fill it. Normally, the size of the pixel is made square (e.g., 30 × 30; 50 × 50; 80 × 80) to facilitate scaling considerations when the rectified data are displayed on CRTs, film writers, plotters, and so on.

Because the data are rectified to the USGS Charleston, S.C., 7.5-minute quadrangle, it is posssible to route the remotely sensed data to a plotter or printer and



OFEK06799

have them precisely overlay the 1:24,000 scale map (assuming that proper scaling adjustments are made). The capability to overlay the rectified image data onto map data is very important in many image analysis applications because Landsat MSS images at 57 × 79 m spatial resolution and TM images at 30 × 30 m spatial resolution are still relatively coarse, making it very difficult to orient within a scene.

The geometric rectification process described here represents the most fundamental approach. More sophisticated methods do exist. For example, Friedman et al. (1983) used improved resampling techniques to rectify Landsat imagery. They suggest that their technique requires only four ground control points (GCPs) to rectify up to 10 Landsat scenes. Others have incorporated digital elevation models (DEMs) as an additional image file to improve both image rectification and classification (Seidel et al., 1983; Naraghi et al., 1983). Furthermore, multiple Landsat scenes can now be mosaicked together by simultaneously map projecting the remotely sensed data while adjusting the edges of the frame so that no radiometric or geometric seams are present (Zobrist et al., 1983).

## REFERENCES


BERNSTEIN, R. 1983. "Image Geometry and Rectification," Chapter 21 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: American Society of Photogrammetry and Remote Sensing, Vol. 1, pp. 875–881.

BERNSTEIN, R., and D. G. FERNEYHOUGH. 1975. "Digital Image Processing," *Photogrammetric Engineering and Remote Sensing*, Vol. 41, pp. 1465–1476.

BILLINGSLEY, F. C. 1983. "Data Processing and Reprocessing," Chapter 17 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: American Society of Photogrammetry and Remote Sensing, Vol. 1, pp. 758–763.

CASTLEMAN, K. E. 1978. "Geometric Operations," Chapter 8 in *Digital Image Processing*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., pp. 111–112.

ESTES, J. E., E. J. HAJIC, and L. R. TINNEY. 1983. "Fundamentals of Image Analysis: Analysis of Visible and Thermal Infrared Data," Chapter 24 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: American Society of Photogrammetry and Remote Sensing, Vol. 1, pp. 1039–1040.

FRIEDMAN, D. E., J. P. FRIEDEL, K. L. MAGNUSSEN, R. KWOK, and S. RICHARDSON. 1983. "Multiple Scene Precision Rectification of Spaceborne Imagery with Very Few Ground Control Points," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 1657–1667.

GREEN, W. B. 1983. "Quantitative Image Processing Techniques," Chapter 4 in *Digital Image Processing*. New York: Van Nostrand Reinhold Company, Inc., pp. 77–105.

GUINDON, G. 1984. "Algorithms for the Estimation of Failed Detector Data," *Proceedings*, Machine Processing of Remotely Sensed Data Symposium. West Lafayette, Ind.: Laboratory for the Applications of Remote Sensing, pp. 39–46.

HARALICK, R. M. 1973. "Glossary and Index to Remotely Sensed Image Pattern Recognition Concepts," *Pattern Recognition*, Vol. 5, pp. 391–403.

JACKSON, R. D., P. N. SLATER, and P. J. PINTER. 1983. "Adjusting the Tasseled-Cap Brightness and Greenness Factors for Atmospheric Path Radiance and Absorption on a Pixel by Pixel Basis," *International Journal of Remote Sensing*, Vol. 4, No. 2, pp. 313–323.




OFEK06800

KAUFMAN, Y. J., and R. S. FRASER. 1984. "Atmospheric Effect on Classification of Finite Fields," *Remote Sensing of Environment*, Vol. 15, pp. 95–118.

MEKLER, Y., and Y. J. KAUFMAN. 1980. "The Effect of Earth's Atmosphere on Contrast Reduction for a Nonuniform Surface Albedo and 'Two Halves' Fields," *Journal of Geophysical Research*, Vol. 85, pp. 4067–4083.

MOIK, J. G. 1980. *Digital Processing of Remotely Sensed Images*. Washington, D.C.: NASA Scientific and Technical Information Branch, NASA SP-431, pp. 77–126.

NARAGHI, M., W. STROMBERG, and M. DAILY. 1983. "Geometric Rectification of Radar Imagery Using Digital Elevation Models," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 195–199.

NASA. 1979. *Synthetic Aperture Radar/Landsat MSS Image Registration*. Washington, D.C.: NASA Scientific and Technical Information Branch, NASA RP-1039.

SEIDEL, K., F. ADE, and J. LICHTENEGGER. 1983. "Augmenting LANDSAT MSS Data with Topographic Information for Enhanced Registration and Classification," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. GE-21, pp. 252–258.

TARANIK, J. V. 1978. *Principles of Computer Processing of Landsat Data for Geologic Applications*. U.S. Geological Survey Open File Report 78–187. Washington, D.C.: U.S. Department of Interior.

TURNER, R. E. 1978. "Elimination of Atmospheric Effects from Remote Sensor Data," *Proceedings*, 12th International Symposium on Remote Sensing of Environment. Ann Arbor, Mich.: Environmental Research Institute of Michigan.

ZOBRIST, A. L., N. A. BRYANT, and R. G. MCLEOD. 1983. "Technology for Large Digital Mosaics of Landsat Data," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 1325–1335.



OFEK06801

# 7

# Image Enhancement

## INTRODUCTION

Image enhancement algorithms are applied to remotely sensed data to improve the appearance of an image for human visual analysis, or occasionally for subsequent machine analysis. There is no such thing as the ideal or best image enhancement because the results are ultimately evaluated by human beings who make subjective judgments as to whether a given image enhancement is useful. This chapter identifies a variety of point and local image enhancement techniques which have proven of value for visual analysis of remote sensor data and/or subsequent machine analysis. *Point operations* modify the brightness values of each pixel in an image data set independently. *Local operations* modify the value of each pixel in the context of the brightness values surrounding it.

## IMAGE REDUCTION AND MAGNIFICATION

### Image Reduction

In the early stages of remote sensing research or educational projects it is often necessary to view the entire image in order to locate the row and column coordinates of a subimage (or multiple subimages) which encompass the study area. Most image processing systems can display only $\leq 512 \times 512$ pixels at one time; therefore, it is desirable to have some relatively simple procedure for reducing the original image data set down to one or a few manageable images. To *reduce* a digital image to





OFEK06802

just $1/m^2$ of the original data, every $m$th row and $m$th column of the imagery are systematically selected and displayed. For example, consider a Landsat MSS image composed of 2340 rows by 3240 columns. If only every other row and every other column (i.e., $m = 2$) were selected for a single band, the entire scene could be displayed as a sampled image consisting of just 1170 rows by 1620 columns. This reduced data set would contain only one-fourth (25%) of the pixels found in the original scene. The logic associated with such a reduction is shown diagrammatically in Fig. 7–1. It is possible to sample the rows and columns with unequal integer values (e.g., sample very second row and every third column). This produces skewed displays that are often difficult to interpret. Therefore, to maintain geometric relationships it is usual practice to sample the rows and columns with the same integer value, $m$.

Most image processing systems cannot display an image that is 1170 × 1620 pixels in dimension as described in the previous example. Therefore, it is often necessary to sample the remotely sensed data intensively. For example, consider the original Charleston, S.C., thematic mapper scene that consists of 5940 rows by 6960 columns per band. The Charleston TM band 4 image displayed in Fig. 7–2 was produced by sampling every thirtieth row and column (i.e., $m = 30$) of the original imagery. Thus this image represents only 1/900 of the original data and is composed of 198 rows and 232 columns. A color composite produced using sampled data from bands 2, 3, and 4 is shown in Fig. 7–3 (color section).

If we compare the original data with the reduced data, there is an obvious loss of detail because so many of the pixels are not present. Therefore, we rarely

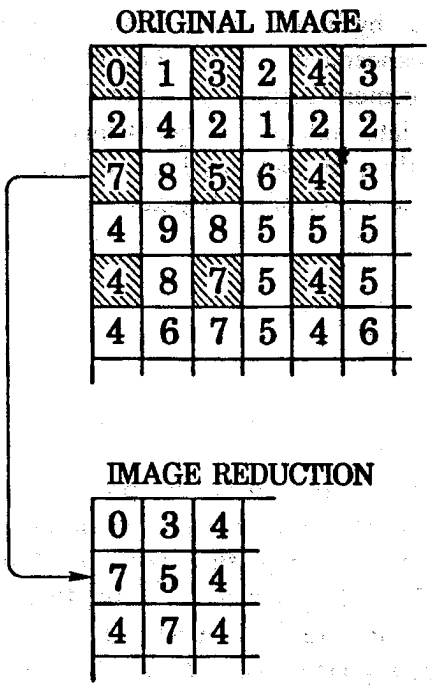



**Figure 7–1** Image reduction achieved by sampling every other row and column of the original data. This operation results in a new image consisting of only ¼ (25%) of the original data.



OEEK06803

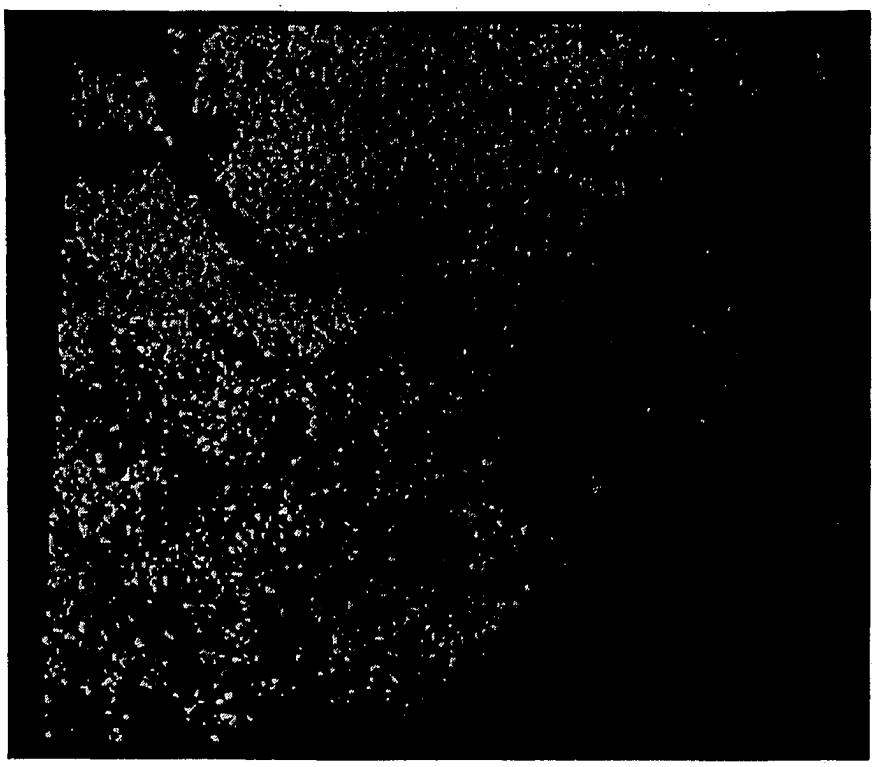


**Figure 7–2** This 198-row by 232-column image represents only 1/900 of the data found in the 5940-row by 6960-column original thematic mapper image. It was created by sampling every thirtieth row and thirtieth column of the band 4 image.

interpret image reductions. Instead, they are used for orienting within a scene and locating the row and column coordinates of specific study areas which can then be extracted at full resolution for analysis.

## *Image Magnification*

Digital image *magnification* (often referred to as *zooming*) is usually performed to improve the scale of a display for visual interpretation purposes, or occasionally, to match the scale of another image. Just as row and column deletion is the simplest form of image reduction, row and column replication represents the simplest form of digital image magnification. To magnify a digital image by a factor $m^2$, each pixel in the original image is usually replaced by an $m \times m$ block of pixels, all with the same brightness value as the original input pixel. An example of the logic of a four-times magnification is shown in Fig. 7–4. This form of magnification is characterized by visibly square tiles of pixels in the output display. Image magnifications of two, three, four, eight, and sixteen times the original Charleston TM data are shown in Fig. 7–5. Note how the building shadows become progressively more apparent as the magnification is increased. Magnifications of two, three, and four



OFEK06804

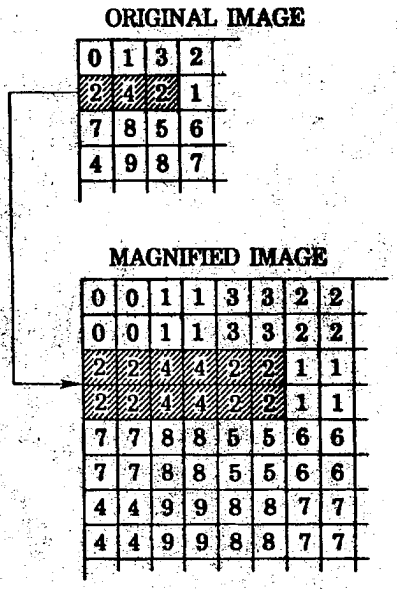



**Figure 7–4** Image magnification achieved by replicating every row and column in the original data. This operation results in a new image consisting of four times as many pixels as the original scene.

times the Savannah River thermal-infrared data are shown in Fig. 7–6. The scan lines with line start problems become readily apparent as the magnification is increased.

Magnification of the original imagery is useful when the analyst is trying to obtain detailed information about the spectral reflectance or emittance characteristics of a relatively small geographic area of interest. During the "training" phase of a supervised classification it is especially useful to be able to zoom in to the raw sensor data to isolate a particular field, body of water, and so on. For example, some of the sites shown in the color composite in Fig. 7–7 (color section) at various magnifications are used in the supervised classification discussed in Chapter 8.

In addition to magnification, many digital image processing systems provide a mechanism whereby the analyst can *pan* or *roam* about a much larger geographic area (e.g., 1024 × 1024) while only viewing a portion (e.g., 512 × 512) of this area at any one time. This allows the analyst to view parts of the data base much more rapidly. It also requires a more sophisticated image processor.

## TRANSECTS

The ability to extract brightness value information along a user-specified *transect* from point A to point B in an image is important in many remote sensing applications. The data are often simply extracted from the remote sensing data and listed in tabular format. Ideally, however, the brightness values encountered along the transect are allowed to keep their spatial relationship by being plotted in a histogram format. For example, consider transect A shown in the density-sliced band 4 image of Charleston in Fig. 5–12 (color section) and displayed as a color-coded histogram in Fig. 7–8 (color section). The colors associated with the density-sliced class intervals (listed in Table 5–2) are retained along this 5940-m transect (198 pixels at 30 m/pixel =



OFEK06805

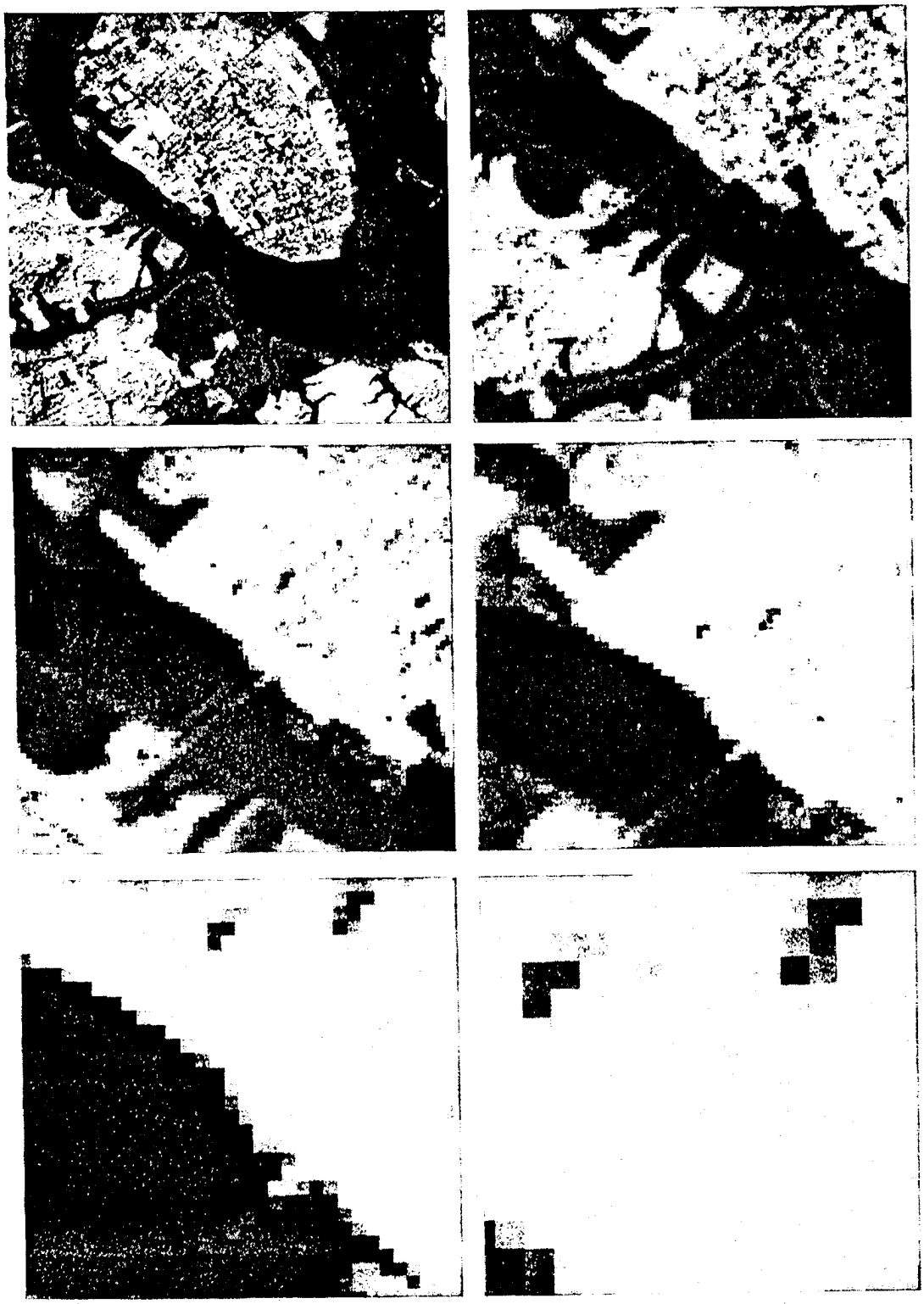


**Figure 7–5** The band 4 thematic mapper data of Charleston, S.C., has been magnified 0, 2, 3, 4, 8, and 16 times in these illustrations. Note the two large buildings and their associated shadows as the magnification increases.





OFEK06806

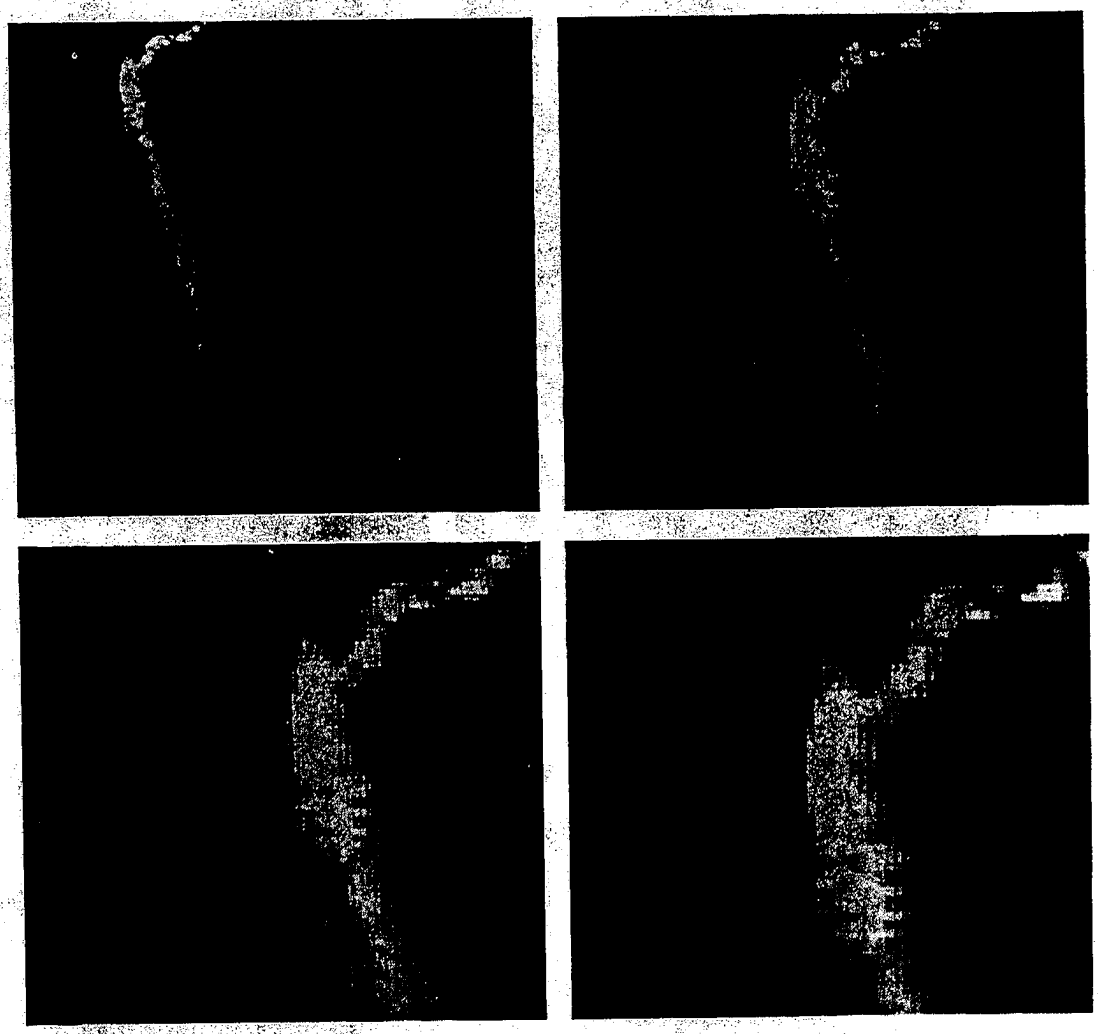


**Figure 7–6** The thermal infrared MSS data of the Savannah River is magnified 0, 2, 3, and 4 times in these illustrations. Note the line start problem associated with a few of the scan lines which becomes progressively more apparent with magnification.

5940 m). The farther inland, the greater the brightness value of the pixels encountered in the transect. This condition is expected since this is an infrared image (TM band 4) and water absorbs most of the incident radiant flux.

Transects passed through the predawn Savannah River thermal infrared data serve to demonstrate the quantitative power of the transect enhancement technique [Fig. 5–13 (color section)]. A color-coded histogram of transect B is shown in Fig. 7–9 (color section). By counting the number of pixels in specific temperature class intervals within the plume and counting the total number of pixels in river width, it is possible to determine the proportion of the plume falling within the temperature class intervals (Jensen et al., 1983). The results of evaluating transects A through G are summarized in Table 7–1. The values reported in this table were obtained



OFEK06807

**TABLE 7-1** Savannah River Thermal Plume Apparent Temperature Transects (Jensen et al., 1983)

| Transect[a] | Average width of river[b] | D. blue | L. blue | Green | Yellow | Orange | Red | White |
|---|---|---|---|---|---|---|---|---|
| | | Ambient | +1°C | 1.2–2.8 | 3.0–5.0 | 5.2–10. | 10.2–20 | >20 |
| A | 32 pixels = 89.6 m | 15/42[c] | 17/47.6 | — | — | — | — | — |
| B | 38 pixels = 106.4 m | 25/70 | 1/2.8 | 1/2.8 | 2/5.6 | 1/2.8 | 5/14. | 3/8.4 |
| C | 34 pixels = 95.2 m | 19/53.2 | 2/5.6 | 2/5.6 | 2/5.6 | 6/16.8 | 3/8.4 | — |
| D | 34 pixels = 95.2 m | 16/44.8 | 2/5.6 | 3/8.4 | 5/14. | 8/22.4 | — | — |
| E | 32 pixels = 89.6 m | 18/50.4 | 3/8.4 | 2/5.6 | 6/16.8 | 3/8.5 | — | — |
| F | 36 pixels = 100.8 m | 8/22.4 | 18/50.4 | 2/5.6 | 8/22.4 | — | — | — |
| G | 34 pixels = 95.2 m | 3/8.4 | 24/67.2 | 6/16.8 | 1/2.8 | — | — | |

[a] Each transect was approximately 285 m in length (66 pixels at 2.8 m/pixel). Transect measurements were made only after the image was rotated so that the beginning and ending pixels of the transect fell on the same scan line.

[b] Includes one mixed pixel of land and water on each side of the river.

[c] Notation represents pixels and meters; for example, 15 pixels represent 42 m.



1010

OFEK06808

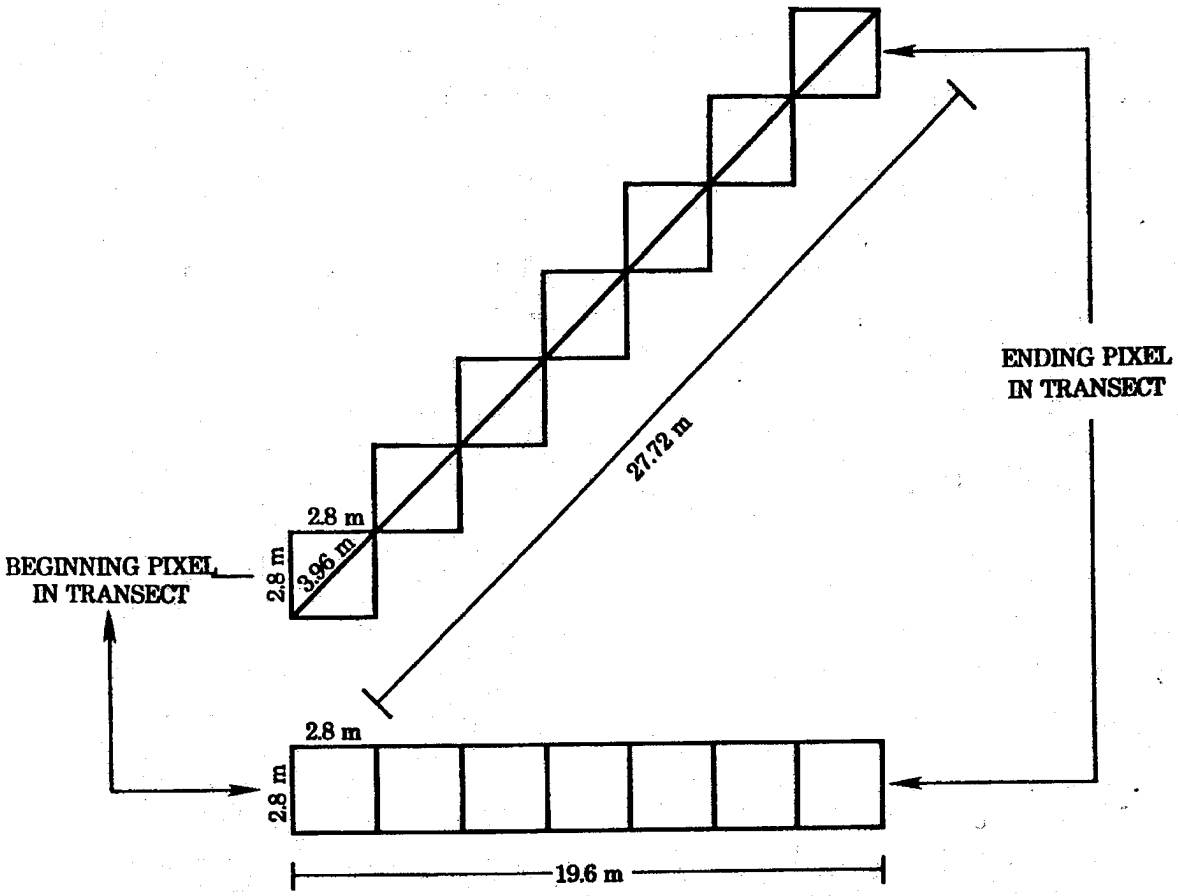



**Figure 7–10** Linear measurements along a transect are easily computed when the beginning and ending pixels of the transect fall on the same line. If this is not the case, the hypotenuse of "stair-stepped" pixels must be considered. In this example the pixels are 2.8 × 2.8 m in dimension.

only after the image had been geometrically rotated so that the endpoints of each transect fell on the same scan line. Thus seven individual geometric rotations were required (not shown). This ensured that the number of meters in each temperature class along each transect were accurately measured. If the analyst extracts transects where the endpoints do not fall on the same scan line, the hypotenuse of "stair-stepped" pixels must be considered instead of the simple horizontal pixel distance. This relationship is demonstrated in Fig. 7–10.

## CONTRAST ENHANCEMENT

Remote sensors record reflected and emitted radiant flux from earth surface materials. Ideally, one material would reflect a tremendous amount of energy in certain wavelengths, while another material would reflect much less energy in the same wavelengths. This would result in *contrast* between the two types of materials when recorded by



OFEK06809

a remote sensing system. Unfortunately, different materials often reflect similar amounts of radiant flux throughout the visible and near-infrared portion of the electromagnetic spectrum, resulting in a relatively *low contrast* image. In addition, besides this obvious low-contrast characteristic of biophysical materials, there are cultural factors at work. For example, people in developing countries often use natural building materials (e.g., wood, soil, etc.) in the construction of urban areas. This results in much-lower-contrast remotely sensed imagery for such areas than for urban areas in developed countries, where concrete, asphalt, and fertilized green vegetation may be more prevalent. Thus biophysical materials themselves are an important factor and man may further confuse the issue by bringing the materials together in diverse ways.

An additional factor in the creation of low-contrast remotely sensed imagery is the sensitivity of the detectors. For example, the detectors on most sensing systems are designed to record a relatively wide range of scene brightness values (e.g., 0 to 127 or 0 to 255) without becoming "saturated." Saturation occurs if the radiometric sensitivity of a detector is insufficient to record the full range of intensities of reflected or emitted energy emanating from the scene. The Landsat MSS detectors, for example, must be sensitive to reflectance from diverse biophysical materials such as basalt outcrops or snow (possibly represented as BVs of 0 to 127, respectively). However, very few scenes are composed of brightness values that utilize the full sensitivity range of the Landsat detectors. Therefore, this results in relatively low-contrast imagery, with brightness values that usually range from 10 to 50.

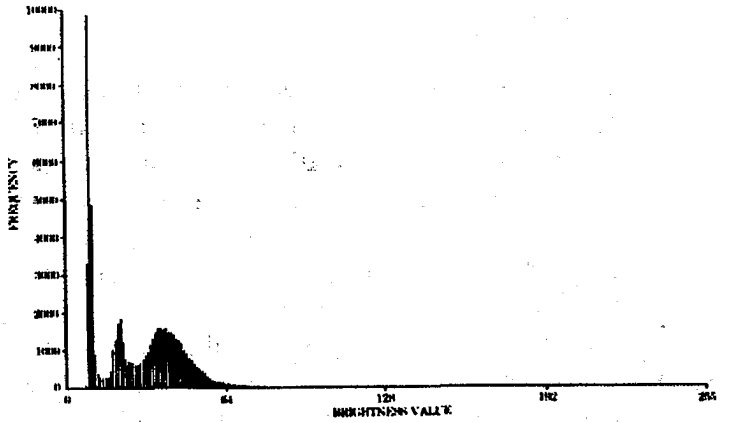
To improve the contrast of digital remotely sensed data, it is both desirable and possible to utilize the entire brightness range of the display medium, which is generally a video CRT display or hard-copy photographic film (discussed in Chapter 5). Digital methods may be more satisfactory than photographic techniques for contrast enhancement because of the precision and wide variety of processes that can be applied to the imagery. There are linear and nonlinear digital contrast enhancement techniques.

## Linear Contrast Enhancement

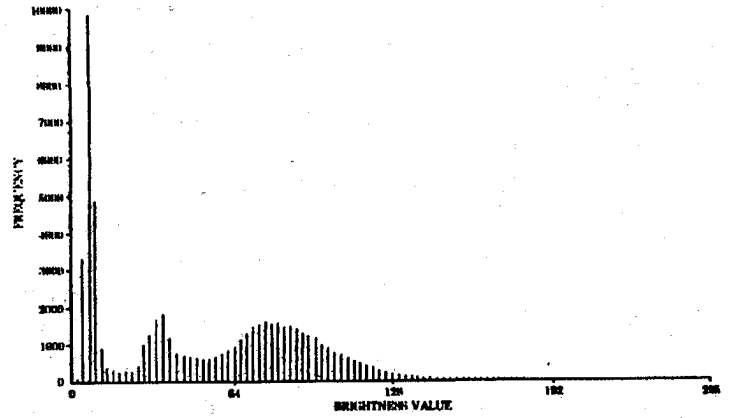
A contrast enhancement (often referred to as a *contrast stretch*) expands the original input brightness values to make use of the total range or sensitivity of the output device. To illustrate the linear contrast stretch process, consider the Charleston, S.C., TM band 4 image produced by a sensor system whose image output levels can vary from 0 to 255 (Fig. 5–10). A histogram of this image is provided in Fig. 7–11a. We will assume that our output device (a high-resolution black-and-white CRT) can display 256 shades of gray (0 to 255, respectively). The histogram and associated statistics of this subimage reveals that the scene is composed of brightness values ranging from a minimum of 4 to a maximum value of 104, with a mean of 27.3 and a standard deviation of 15.76 (Table 4–7). When these data are displayed on the CRT without any contrast enhancement, we use less than one-half of the full range of brightness values that could be displayed (i.e., brightness values between 0 and 3 and between 105 and 255 are not used). The image is rather dark, low in



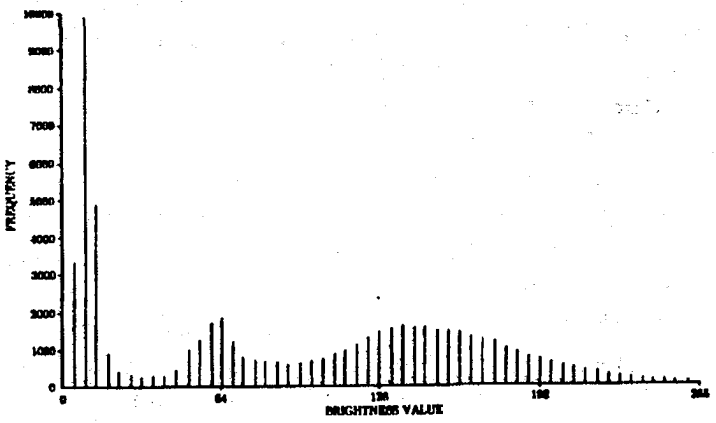
OFEK06810




CHARLESTON, S.C.
THEMATIC MAPPER
BAND 4 (ORIGINAL DATA)




CHARLESTON, S.C.
THEMATIC MAPPER
BAND 4 (MIN-MAX STRETCH)




CHARLESTON, S.C.
THEMATIC MAPPER
BAND 4 (PERCENTAGE STRETCH)

**Figure 7–11** (a) The histogram of the original band 4 Charleston, S.C., thematic mapper data; (b) the histogram after a minimum-maximum contrast stretch has been applied to the data; (c) the histogram after a 5% linear contrast stretch.





OFEK06811

contrast, with no distinctive bright areas (Fig. 5–10). It would be difficult to visually interpret such an image. A more useful display can be produced if we expand the range of original brightness values to use the full dynamic range of the CRT display.

Linear contrast enhancement is best applied to remotely sensed images with Gaussian or near-Gaussian histograms, that is, where all the brightness values fall generally within a single, relatively narrow range of the histogram and only one mode is apparent. Unfortunately, this is rarely the case, especially for scenes with extensive land and water bodies. To perform a linear contrast enhancement, the analyst examines the image histogram and determines the minimum and maximum brightness values in the image, $BV_{min}$ and $BV_{max}$, respectively. The output brightness value $BV_{out}$, is computed according to the equation:

$$BV_{out} = \frac{BV_{in} - BV_{min}}{BV_{max} - BV_{min}} \times BV_r \tag{7-1}$$

where $BV_{in}$ is the original input brightness value and $BV_r$ is the range of the brightness values that can be displayed on the CRT (e.g., 256). In this example, any pixel with a $BV_{in}$ of 4 would now have a $BV_{out}$ of 0, while any pixel with a $BV_{in}$ of 104 would now have a $BV_{out}$ of 255. All original brightness values between 5 and 103 would be linearly distributed between 1 and 254, respectively. The application of this enhancement to the Charleston band 4 TM data is shown in Fig. 7–12. This is commonly referred to as a *min-max contrast stretch*. The resultant histogram is shown in Fig. 7–11b. The logic of such a linear stretch is shown diagrammatically in Fig. 7–13a.

Image analysts often specify $BV_{min}$ and $BV_{max}$ that lie a certain percentage of pixels "into" the tails of the histogram. This is called a *percentage linear contrast*

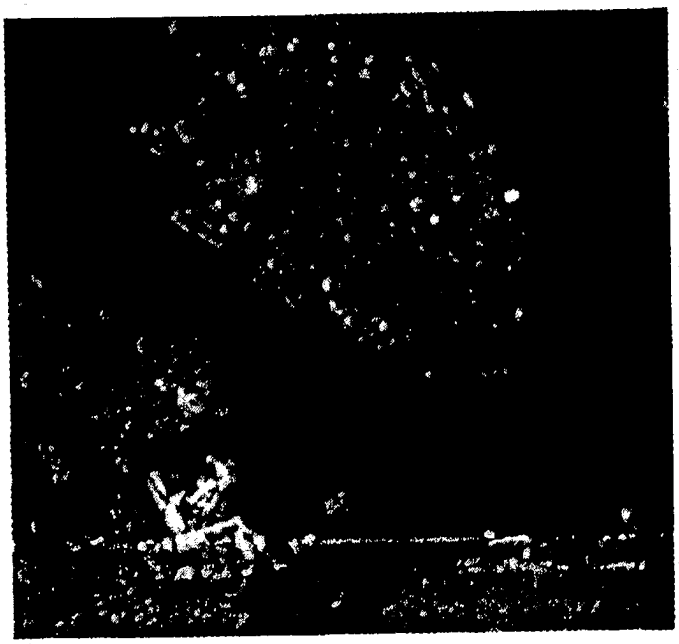


**Figure 7–12** Minimum-maximum linear contrast stretch applied to the band 4 Charleston, S.C., thematic mapper scene. The minimum and maximum were 4 and 104, respectively.



OFEK06812

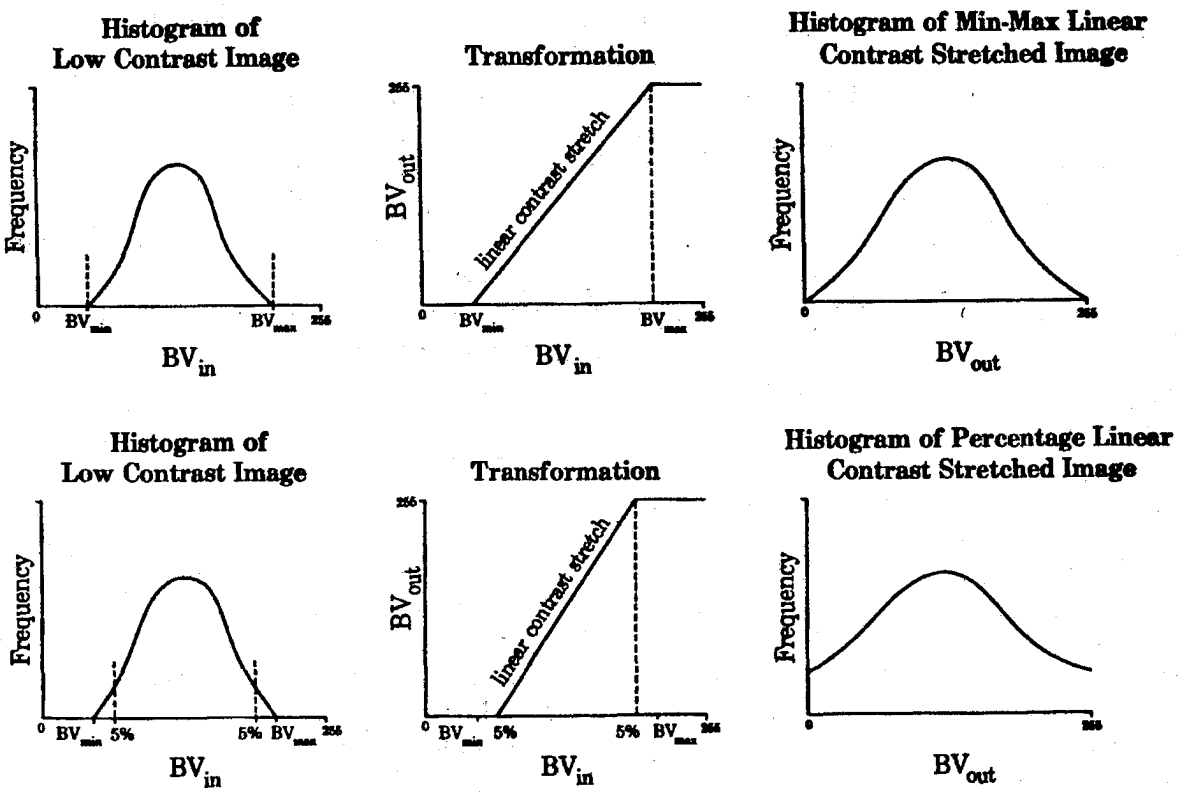



**Figure 7–13** Result of applying a minimum-maximum (a) or percentage linear contrast stretch (b) to normally distributed remotely sensed data. The histograms before and after the transformation are shown. Note that bringing the minimum and maximum values in by 5% greatly increases the slope of the linear contrast stretch transformation.

*stretch*. For example, consider coming in 5% from the minimum and 5% from the maximum. The minimum then becomes 15 and the maximum 86. All values between 15 and 86 are linearly contrast stretched to lie within the range 0 to 255. All values between 0 and 14 are now 0 and those between 87 and 104 are set equal to 255. This results in more pure black-and-white pixels in the Charleston scene, dramatically increasing the contrast of the image (Fig. 7–14). The histogram associated with this percentage contrast stretch is shown in Fig. 7–11c. The information content of the pixels that "saturated" at 0 and 255 are lost. Note that the slope of the percentage linear contrast stretch is much greater than for the simple min-max stretch (Fig. 7–13b).

It is not necessary that the same percentage be applied to each tail of the distribution. For example, what linear stretch would be appropriate if we were interested only in the water and wetland environments that have brightness values ranging from 4 to approximately 29 in band 4? Coming in 0% from the minimum and coming in 63% from the maximum value would result in only the values between 4 and 29 being linearly contrast stretched to lie between 0 and 255. All land pixels having brightness values $\geq$ 30 would be bright white with a value of 255. Similar logic



OFEK06813



Figure 7–14 Result of applying a 5% linear contrast stretch to the band 4 thematic mapper data. After coming in 5%, the minimum and maximum values were 15 and 86, respectively.

could be applied to enhance just the upland area at the expense of the water and wetland.

A percentage linear contrast stretch of the Savannah River thermal infrared data was performed that enhanced the plume and river while subduing the contrast of the swamp (Fig. 7–15). It involved coming in 30% from the minimum and 0% from the maximum to highlight the modes of the histogram associated with the plume and the river (see the original histogram in Fig. 4–2).

When the histogram of an image is not Gaussian in nature (i.e., it is bimodal, trimodal, etc.), it is possible to perform a piecewise linear contrast stretch to the imagery of the type shown in Fig. 7–16. Here the analyst identifies a number of linear enhancement steps that expand the brightness ranges in just the "modes" of the histogram. In effect, this corresponds to setting up a series of $BV_{min}$ and $BV_{max}$ segments and using equation (7-1) within user-selected regions of the histogram.

### *Nonlinear Contrast Enhancement*

Nonlinear contrast enhancements can also be applied to low-contrast imagery. One of the most useful is *histogram equalization*. First, the histogram of the image is derived. The user then specifies the number of output gray-scale classes (e.g., 32) into which the data are to be redistributed. The algorithm then passes through the data set and assigns approximately equal numbers of pixels to each of the output gray-scale classes. Thus there should be an approximately equal number of pixels found in each of the 32 brightness value classes. Such a contrast enhancement is shown in Fig. 7–17. Histogram equalization applies the greatest contrast enhancement to the most populated range of brightness values in the image. It automatically reduces



OFEK06814

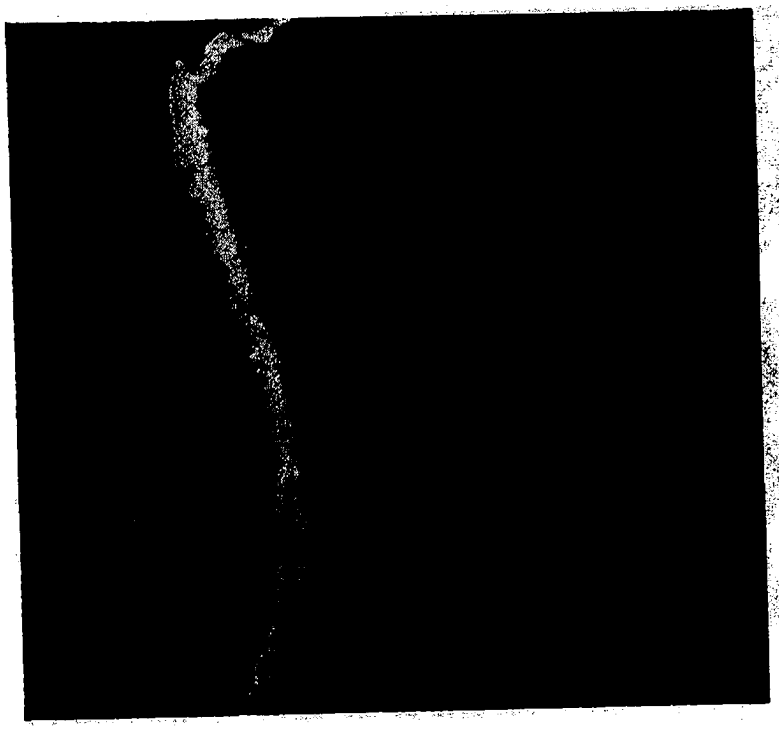


**Figure 7–15** Result of applying a percentage linear contrast stretch to the Savannah River thermal infrared data. In this case the minimum was moved in 30% while the maximum was moved in 0%. This enhanced the modes of the histogram associated with just the thermal plume and the river.

the contrast in the very light or dark parts of the image associated with the tails of a normally distributed histogram.

Histogram equalization contrast modification is found on many image processing systems because to implement it requires very little information from the analyst. Because of its wide availability, it is instructive to review how the equalization takes place using a hypothetical data set (modified from Gonzalez and Wintz, 1977). For example, consider an image that is composed of just 64 rows and 64 columns (4096 pixels) with the range of brightness values that each pixel can assume limited to just eight values (i.e., $BV_r = 8$). A histogram of this hypothetical image is shown in Fig. 7–18a and the frequency of occurrence of the individual brightness values, $f(BV_i)$ summarized in Table 7–2. For example, there are 790 pixels in the scene with a brightness value of 0 [i.e., $f(BV_0) = 790$] and 1023 pixels with a brightness value of 1 [i.e., $f(BV_1) = 1023$]. We can compute the probability of the $i$th brightness value, $p_i$, by dividing each of the frequencies $f(BV_i)$ by the total number of pixels in the scene (i.e., $n = 4096$). Thus the probability of encountering a pixel with a brightness value of 0 in the scene is approximately 19% [i.e., $p_0 = f(BV_i)/n = (790/4096) = 0.19$]. A plot of the probability of occurrence of each of the eight brightness values for the hypothetical scene is shown in Fig. 7–18b. This particular histogram has an exceptionally large number of pixels with low brightness values (0 and 1), making it a relatively low contrast scene.

The next step is to compute a transformation function, $s_i$, for each of the individual brightness values. This is done by computing the cumulative frequency curve of



OEEK06815

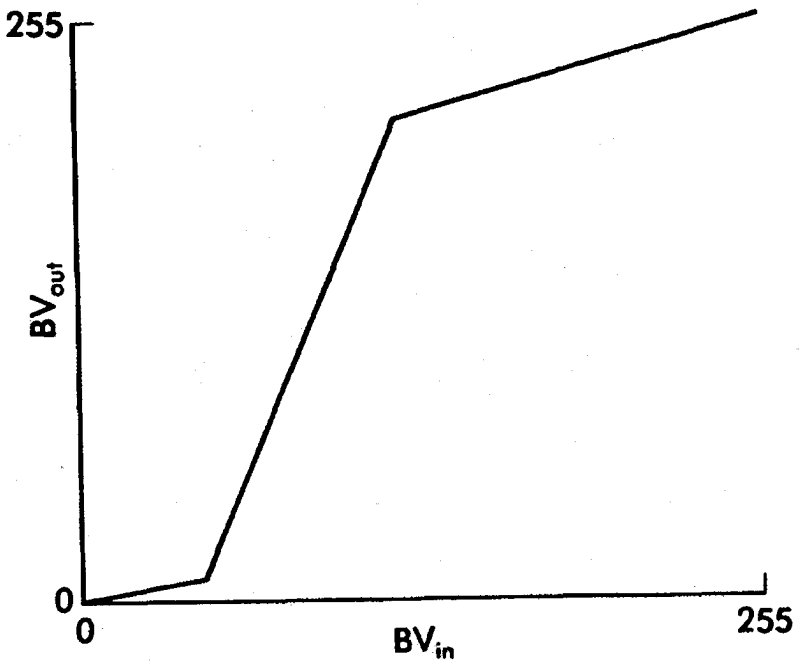



**Figure 7–16** Logic of a piecewise linear contrast stretch. This means that selective pieces of the histogram are linearly contrast stretched. Notice that the slope of the linear contrast enhancement changes.



**Figure 7–17** Result of applying a histogram equalization to the band 4 thematic mapper data of Charleston, S.C.



OFEK06816

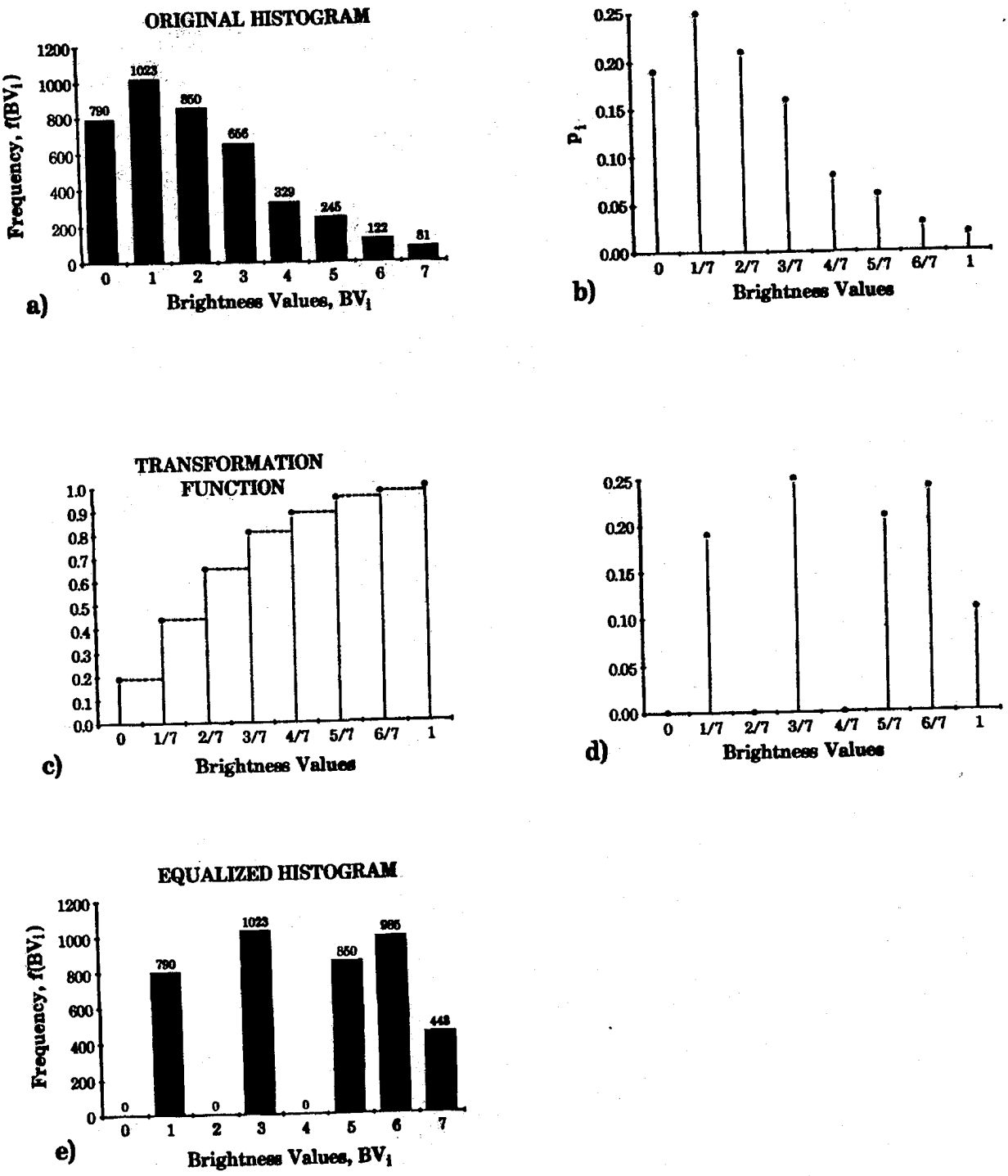



**Figure 7–18** Histogram equalization process applied to hypothetical data: (a) original histogram showing the frequency of pixels in each brightness value; (b) the original histogram expressed in probabilities; (c) the transformation function; (d) the equalized histogram expressed as probabilities; (e) the equalized histogram showing the frequency of pixels in each brightness value. (Modified from Gonzalez and Wintz, 1977.)



OFEK06817

**TABLE 7–2** Statistics for a 64 × 64 Hypothetical Image with Brightness Values from 0 to 7

| Brightness value, $BV_i$ | Frequency, $f(BV_i)$ | Probability, $p_i[f(BV_i)/n]$ |
|---|---|---|
| $BV_0 = 0/7 = 0.00$ | 790 | 0.19 |
| $BV_1 = 1/7 = 0.14$ | 1023 | 0.25 |
| $BV_2 = 2/7 = 0.28$ | 850 | 0.21 |
| $BV_3 = 3/7 = 0.42$ | 656 | 0.16 |
| $BV_4 = 4/7 = 0.57$ | 329 | 0.08 |
| $BV_5 = 5/7 = 0.71$ | 245 | 0.06 |
| $BV_6 = 6/7 = 0.85$ | 122 | 0.03 |
| $BV_7 = 7/7 = 1.00$ | 81 | 0.02 |

$n$ = 4096 pixels.

*Source*: Modified from Gonzalez and Wintz (1977).

the original histogram (Fig. 7–18c) using the following algorithm (Gonzalez and Wintz, 1977):

$$s_i = \sum_{i=0}^{i} p_i \tag{7-2}$$

For example, the transformation function for brightness value $BV_0$ is

$$s_0 = \sum_{i=0}^{0} 0.19 = 0.19$$

Similarly,

$$\begin{aligned} s_1 &= \sum_{i=0}^{1} p_i \\ &= p_0 + p_1 \\ &= 0.19 + 0.25 \\ &= 0.44 \end{aligned} \tag{7-3}$$

and

$$\begin{aligned} s_2 &= 0.65 \qquad s_3 = 0.81 \qquad s_4 = 0.89 \\ s_5 &= 0.95 \qquad s_6 = 0.98 \qquad s_7 = 1.00 \end{aligned}$$

In this example we want to try to equalize the number of pixels found in eight brightness values. Therefore, we now compare each of the transformation functions $s_i$ with each of the original brightness values $BV_i$, and assign each of the transformation functions to the brightness value closest to it in value. For example, consider the first transformation, $s_0$, which has a value of 0.19 (i.e., $s_0 = 0.19$). This value is not close to $BV_0 = 0/7 = 0$, not close to $BV_2 = 2/7 = 0.285$, but is close to $BV_1 = 1/7 = 0.14$. This same logic can be applied to each of the transformed values:



OFEK06818

$(s_0 = 0.19) \approx (1/7 = 0.14)$ $(s_1 = 0.44) \approx (3/7 = 0.42)$
$(s_2 = 0.65) \approx (5/7 = 0.71)$ $(s_3 = 0.81) \approx (6/7 = 0.85)$
$(s_4 = 0.89) \approx (6/7 = 0.85)$ $(s_5 = 0.95) \approx 1$
$(s_6 = 0.98) \approx 1$ $(s_7 = 1.0) \approx 1$

Now there are really only five distinct histogram-equalized brightness values. The notation may be redefined to take this condition into account:

$$s_0 = 1/7 \qquad s_1 = 3/7 \qquad s_2 = 5/7 \qquad s_3 = 6/7 \qquad S_4 = 1$$

This means that $BV_0$ is mapped to $s_0 = 1/7$, which now has all 790 pixels assigned to it. Similarly, there are now 1023 pixels with value $s_1 = 3/7$ and 850 pixels with value $s_2 = 5/7$. However, because both levels $BV_3$ and $BV_4$ were mapped to $s_3 = 6/7$, there are now 656 + 329 = 985 pixels with this new brightness value. Similarly, there are 245 + 122 + 81 = 448 pixels with value $s_4 = 1$ (Gonzalez and Wintz, 1977). The probability of encountering each of the five distinct brightness values is shown in Fig. 7–18d. It was produced by dividing the new frequencies by 4096: for example, the brightness value at $5/7 = 850/4096 = 0.19$. A histogram of the number of pixels in each of these new brightness values is shown in Fig. 7–18e. The histogram equalization is an approximation to a uniformly distributed probability density function.

Frei (1977) discusses histogram hyperbolization enhancement, which is another

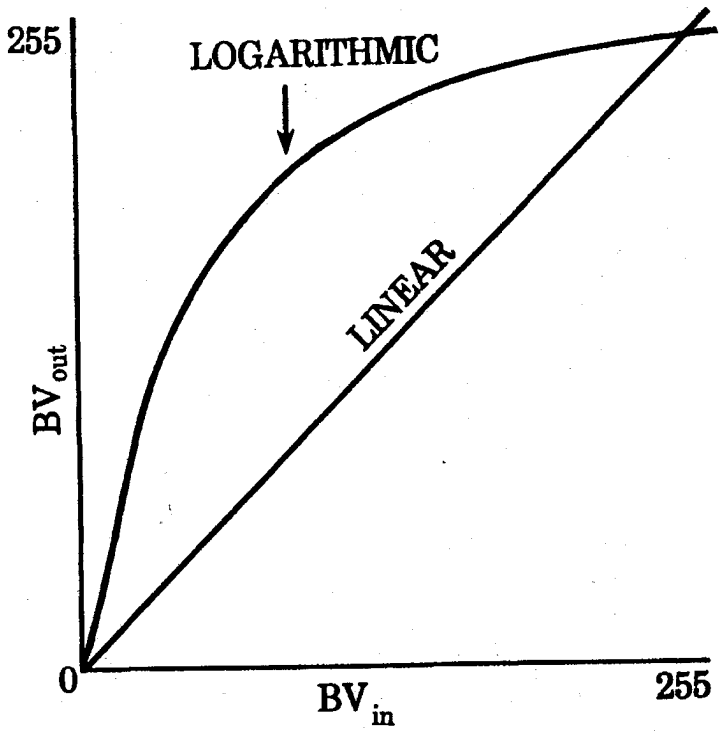



**Figure 7–19** Logic of a nonlinear, logarithmic contrast stretch.



OFEK06819

histogram modification technique, but one that takes into account human brightness perception characteristics. Still another type of nonlinear contrast stretch involves scaling the input data *logarithmically*, as suggested diagrammatically in Fig. 7–19. This enhancement has the greatest impact on the brightness values found in the darker part of the histogram. It could be reversed to enhance values in the brighter part of the histogram, if desired.

The selection of a contrast enhancement algorithm depends on the nature of the original histogram and the elements of the scene that are of greatest interest to the user. An experienced image analyst can usually identify an appropriate contrast enhancement algorithm by examining the image histogram and then experimenting until satisfactory results are obtained. Most contrast enhancements cause some useful information to be lost. Hopefully, that which remains is of value.

Contrast enhancement is applied primarily to improve visual image analysis. It is *not* good practice to contrast stretch the original imagery and then use the enhanced imagery for computer-assisted classification, change detection, and so on. Contrast stretching almost always changes the original pixel values, often in a nonlinear fashion.

## RATIOING

Sometimes the differences in brightness values from similar surface materials are caused by topographic conditions, shadows, or seasonal changes in sunlight illumination angle and intensity. These conditions may hamper the ability of an interpreter or classification algorithm to correctly identify surface materials or land use in a remotely sensed image. Fortunately, *ratio* transformations of the remotely sensed data can, in certain instances, be applied to reduce the effects of such environmental conditions (Friedman, 1978). In addition to minimizing the effects of environmental factors, ratios may also provide unique information not available in any single band that is useful for discriminating between soils and vegetation (Satterwhite, 1984).

The mathematical expression of the ratio function is:

$$BV_{i,j,r} = \frac{BV_{i,j,k}}{BV_{i,j,l}} \tag{7-4}$$

where $BV_{i,j,r}$ is the output ratio value for the pixel at row $i$, column $j$; $BV_{i,j,k}$ is the brightness value at the same location in band $k$; and $BV_{i,j,l}$ is the brightness value in band $L$. Unfortunately, the computation is not always simple since $BV_{i,j} = 0$ is possible. However, there are alternatives. For example, the mathematical domain of the function is 1/255 to 255 (i.e., the range of the ratio function includes all values beginning at 1/255, passing through 0 and ending at 255). The way to overcome this problem is simply to give any $BV_{i,j}$ with a value of 0, the value of 1. Alternatively, some like to add a small value (e.g., 0.1) to the denominator if it equals zero.

To represent the range of the function in a linear fashion and to encode the



OFEK06820

ratio values in a standard 8-bit format (values from 0 to 255), normalizing functions are applied. Using this normalizing function, the ratio value 1 is assigned the brightness value 128. Ratio values within the range 1/255 to 1 are reassigned values between 1 and 128 by the function:

$$BV_{i,j,n} = \text{Int}\,[(BV_{i,j,r} \times 127) + 1] \tag{7-5}$$

Ratio values from 1 to 255 are assigned values within the range 128 to 255 by the function:

$$BV_{i,j,n} = \text{Int}\left(128 + \frac{BV_{i,j,r}}{2}\right) \tag{7-6}$$

Ratios of remotely sensed data can also be computed using the following algorithm which does not require so much data evaluation (Hord, 1982):

$$BV_{i,j,n} = K\left(\arctan \frac{BV_{i,j,k}}{BV_{i,j,l}}\right) \tag{7-7}$$

If the constant $K$ is set equal to 162.34, the ratio $BV_{i,j,n}$ can take upon itself any value from 0 to 255.

Deciding which two bands to ratio is not always a simple task. Often, the analyst simply displays various ratios and then selects the most visually appealing. A more objective procedure was developed by Chavez et al. (1982, 1984) based on the optimum index factor (OIF), discussed in Chapter 5. It can be used to rank the utility of various combinations of bands for band ratioing. Also, Robinove (1982) suggests that instead of using raw brightness values in the numerator and denominator



**Figure 7–20** Image produced by taking the ratio of thematic mapper bands 3 and 4. The street pattern in the urban area is much more visually interpretable in this image than in any single band 4 image.



OEEK06821

of a band ratio, these data should first be transformed into physical values. He provides simple equations and the necessary parameters to transform raw Landsat 1, 2, and 3 MSS data into either reflectance or radiance [expressed in milliwatts per square centimeter per steradian (mW $cm^{-2}$ $sr^{-1}$)].

The ratio of Charleston, S.C., thematic mapper bands 3 and 4 (red and infrared) is displayed in Fig. 7–20. This type of ratio provides vegetation information, as will be discussed in the section on vegetation indices in this chapter. Generally, the darker the pixel ratio, the more vegetation present within the field of view of the picture element. Several other band ratio images for a portion of the Charleston scene are shown in Fig. 7–21. Sometimes, the inverse of a band ratio is of interest. For example, the brighter the pixel ratio, the greater the amount of vegetative matter present in Fig. 7–21b. The ratio of TM bands 2 and 5 is useful for identifying all water bodies

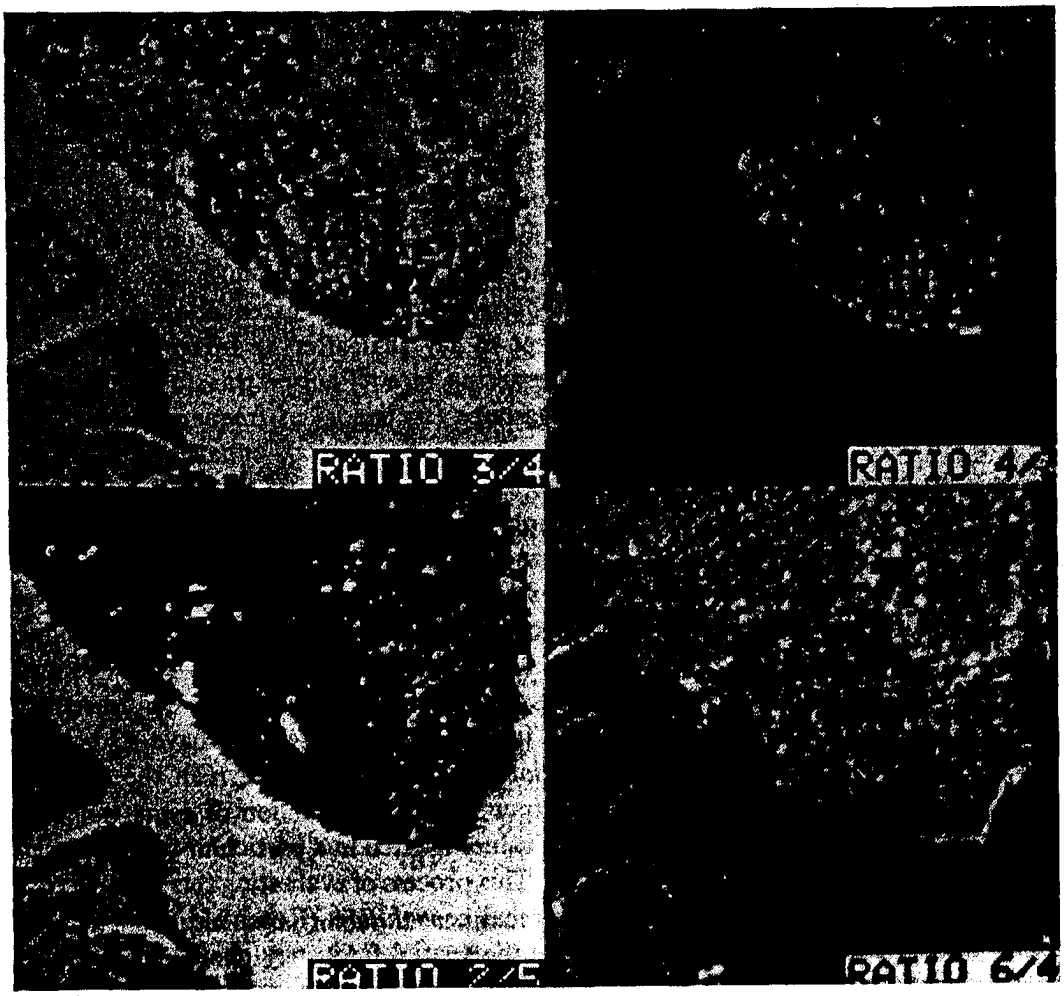



**Figure 7–21** Four band ratioed images produced for a portion of the Charleston, S.C., thematic mapper scene.



OFEK06822

and provides some subtle wetland information (Fig. 7–21c). The ratio of band 6 to band 4 is not very useful since both are infrared bands. Notice that the noise in the remotely sensed data is enhanced, particularly over the water (Fig. 7–21d). Band ratioing depicts in spatial terms the amount of between-band correlation. Generally, the lower the correlation between the bands, the greater the information content of the band-ratioed image.

## SPATIAL FILTERING

A characteristic of remotely sensed images is a parameter called *spatial frequency*, defined as the number of changes in brightness value per unit distance for any particular part of an image. If there are very few changes in brightness value over a given area, this is referred to as being a "low-frequency" area. Conversely, if the brightness values change dramatically over very short distances, this is an area of "high-frequency" detail.

Because spatial frequency by its very nature describes the brightness values over a spatial *region*, it is necessary to adopt a spatial approach to extracting quantitative spatial information. This is done by looking at the local or neighboring pixel brightness values rather than just an independent pixel value. This perspective allows the analyst to extract useful spatial frequency information from the imagery.

The mathematical technique for separating an image into its various spatial frequency components is called *Fourier analysis*. An extended discussion of this technique is not presented here except to say that after an image is separated into its component spatial frequencies (resulting in a "Fourier transform" of the image), it is possible to emphasize certain groups (or bands) of frequencies relative to others and recombine the spatial frequencies to produce an enhanced image. Algorithms to perform such enhancement are called *filters* because they suppress (deemphasize) certain frequencies and pass (emphasize) others. Filters that pass high frequencies, and hence emphasize fine detail and edges, are referred to as high-frequency filters. Conversely, low-frequency filters suppress the high-frequency content of the imagery while emphasizing gradual change. Hord (1982) provides a listing of a Fourier transform FORTRAN program.

Although the operation of spatial filters is conveniently described through the concept of Fourier analysis, the implementation does not have to be through the Fourier transform. A linear *spatial filter* is a filter for which the brightness value ($BV_{i,j}$) at location $i, j$ in the output image is a function of some weighted average (linear combination) of brightness values located in a particular spatial pattern around the $i, j$ location in the input image. This process of evaluating the weighted neighboring pixel values is called two-dimensional *convolution* (Rosenfeld and Kak, 1976). The procedure is often used to change the spatial frequency characteristics of an image. For example, a linear spatial filter that emphasizes high spatial frequencies may sharpen



OEEK06823

the edges in an image. A linear spatial filter that emphasizes the low spatial frequencies may be used to reduce noise in an image.

### Low Frequency Filtering in the Spatial Domain

Image enhancements that deemphasize or block the high spatial frequency detail are *low-frequency* or low-pass filters. The simplest low-frequency filter (LFF) evaluates a particular input pixel b_ightness value, $BV_{in}$, and the pixels surrounding the input pixel, and outputs a new brightness value $BV_{out}$ that is the mean of this convolution. The size of the convolution *mask* or kernel ($n$) is usually $3 \times 3$, $5 \times 5$, $7 \times 7$, $9 \times 9$, and so on. We will constrain this discussion primarily to $3 \times 3$ convolution masks with nine coefficients, $c_i$, defined at the following locations as:

$$\text{mask template} = \begin{matrix} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{matrix} \tag{7-8}$$

For example, the coefficients in a low-frequency mask might all be set equal to 1:

$$\text{mask A} = \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \tag{7-9}$$

The coefficients $c_i$, in the mask are multiplied by the following individual brightness values ($BV_i$) in the input image:

$$\begin{matrix} (c_1 \times BV_1) & (c_2 \times BV_2) & (c_3 \times BV_3) \\ (c_4 \times BV_4) & (c_5 \times BV_5) & (c_6 \times BV_6) \\ (c_7 \times BV_7) & (c_8 \times BV_8) & (c_9 \times BV_9) \end{matrix} \tag{7-10}$$

where

$$\begin{aligned} BV_1 &= BV_{i-1,j-1} \\ BV_2 &= BV_{i-1,j} \\ BV_3 &= BV_{i-1,j+1} \\ BV_4 &= BV_{i,j-1} \\ BV_5 &= BV_{i,j} \\ BV_6 &= BV_{i,j+1} \\ BV_7 &= BV_{i+1,j-1} \\ BV_8 &= BV_{i+1,j} \\ BV_9 &= BV_{i+1,j+1} \end{aligned} \tag{7-11}$$

The primary input pixel under investigation at any one time is $BV_5 = BV_{i,j}$. The convolution of mask A (with all coefficients equal to 1) and the original data will result in a low-frequency filtered image (Sullivan et al., 1984), where



OEEK06824

$$LFF_{5,\text{out}} = \text{Int}\left(\frac{\sum_{i=1}^{n=9} c_i \times BV_i}{n}\right) \tag{7-12}$$

$$= \text{Int}\left(\frac{BV_1 + BV_2 + BV_3 + BV_4 + BV_5 + BV_6 + BV_7 + BV_8 + BV_9}{9}\right)$$

The spatial moving average then shifts to the next pixel, where the average of all nine brightness values is computed. This operation is repeated for every pixel in the input image (Fig. 7–22). Such *image smoothing* is useful for removing periodic "salt and pepper" noise often encountered in remote sensor data acquired by electronic devices.

This simple smoothing operation will, however, blur the image, especially at the edges of objects. Blurring becomes more severe as the size of the kernel increases. To reduce blurring, unequal-weighted smoothing masks have been introduced, including (Wang et al., 1983):

$$\text{mask B} = \begin{matrix} 0.25 & 0.50 & 0.25 \\ 0.50 & 1.00 & 0.50 \\ 0.25 & 0.50 & 0.25 \end{matrix} \tag{7-13}$$

$$\text{mask C} = \begin{matrix} 1.00 & 1.00 & 1.00 \\ 1.00 & 2.00 & 1.00 \\ 1.00 & 1.00 & 1.00 \end{matrix} \tag{7-14}$$

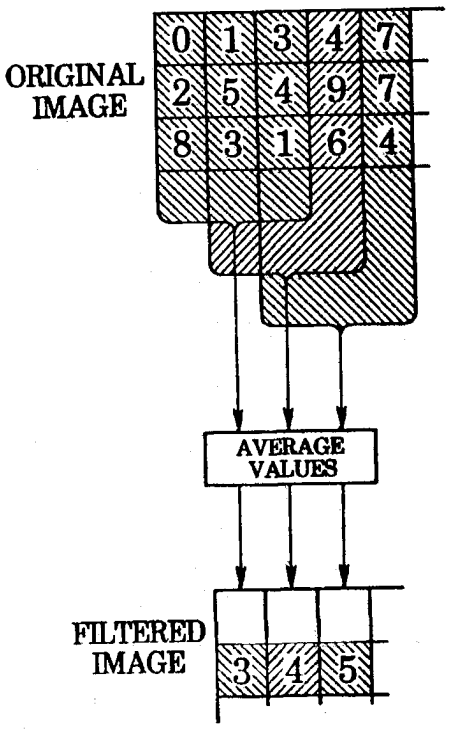



**Figure 7–22** Logic of a 3 × 3 spatial moving average to enhance low-frequency detail in an image. The nine coefficients of the 3 × 3 mask are all equal to 1 in this example.



OEEK06825

Chin and Yeh (1983) evaluate other smoothing algorithms that can preserve edges while still removing noise.

Using a 3 × 3 kernel can result in the low-pass image being two lines and two columns smaller than the original image. Techniques that can be applied to deal with this problem include (1) artificially extending the original image beyond its border by repeating the original border pixel brightness values, or (2) replicating the average brightness values near the borders, based on the image behavior within a few pixels of the border. This maintains the row and column dimension of the imagery but introduces some spurious information that should not be interpreted.

Images produced by applying the mask A spatial filter to the Charleston TM data and Savannah River thermal infrared data are shown in Figs. 7–23 and 7–24, respectively. The Charleston scene becomes blurred, suppressing the high-frequency detail. Only the general trends (similar to a trend-surface analysis) are allowed to pass through the filter. The smooth thermal plume is visually appealing, especially since the bad scan lines are subdued. This suggests that low-frequency filtering is most useful when there are significant low-frequency components in the scene as in the thermal plume example. In a heterogeneous, high-frequency urban environment, a high-frequency filter should provide superior results.

### *High-Frequency Filtering in the Spatial Domain*

High-pass filtering is applied to imagery to remove the slowly varying components and retain only the high-frequency local variations. One high-frequency filter ($HFF_{5,out}$) is computed by subtracting the output from the low-frequency filter ($LFF_{5,out}$) from twice the value of the original central pixel value, $BV_5$, that is,



**Figure 7–23** Result of applying a 3 × 3 low-frequency filter to the Charleston, S.C., band 4 thematic mapper image. All coefficients in the 3 × 3 mask were equal to 1.



OEEK06826

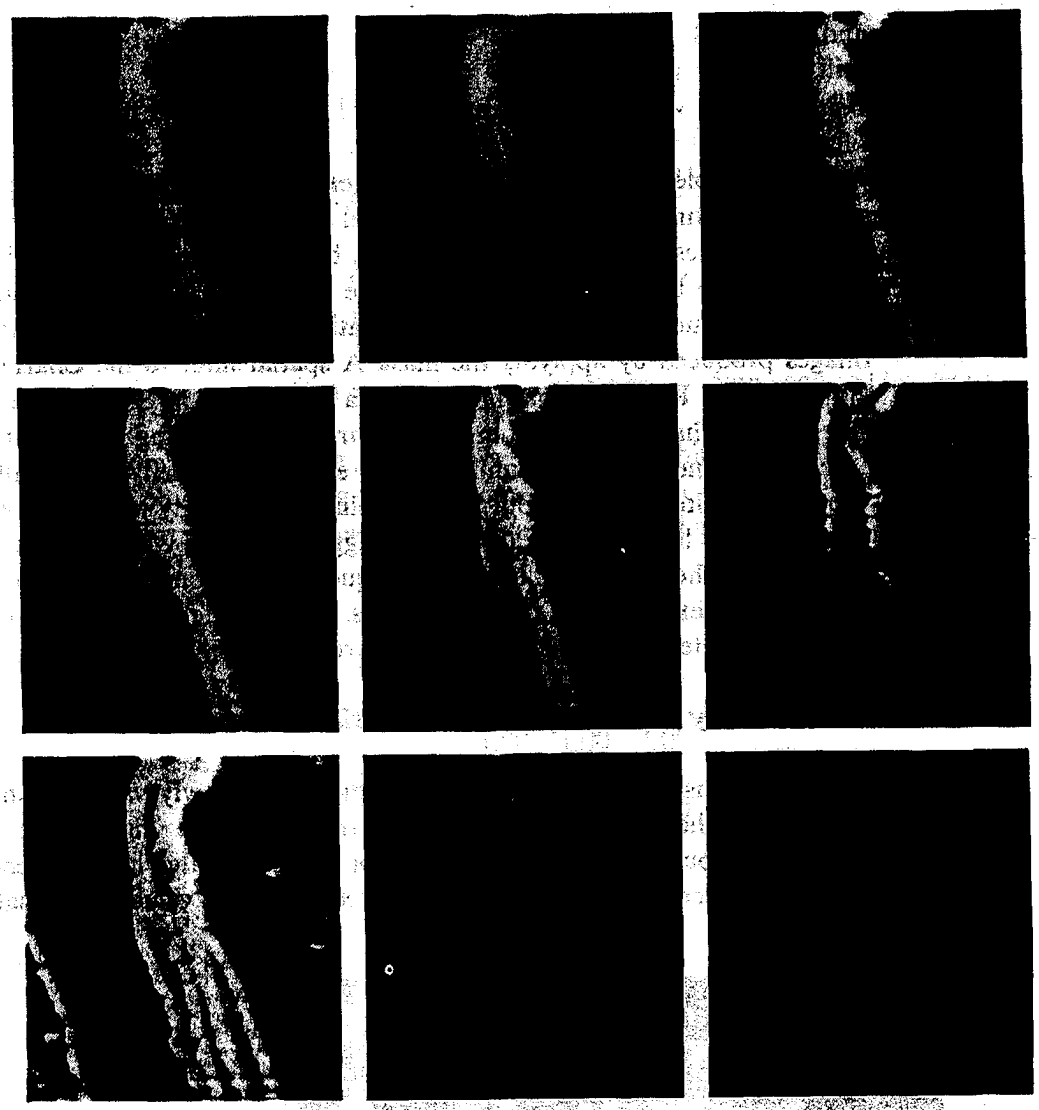


**Figure 7-24** Eight image enhancements applied to the Savannah River thermal-infrared data: (a) the original data; (b) a low-frequency filtered image; (c) a high-frequency filtered image; (d) a horizontal compass gradient mask edge enhancement; (e) a horizontal compass gradient mask edge enhancement applied to the high-frequency filtered image; (f) a Sobel edge enhancement; (g) a piecewise linear contrast stretch of the Sobel edge enhancement; (h) a Sobel edge enhancement of the high-frequency filtered image; (i) a Roberts edge enhancement.

$$HFF_{5,\text{out}} = (2 \times BV_5) - LFF_{5,\text{out}} \tag{7-15}$$

Brightness values tend to be highly correlated in a nine-element window. Thus the high-frequency filtered image will have a relatively narrow intensity histogram. This suggests that most high-frequency filtered images must be contrast stretched prior to visual analysis.



OEEK06827

High-pass filters that accentuate or sharpen edges can be produced using the following convolution masks (Pratt, 1978):

$$\text{mask E} = \begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix} \tag{7-16}$$

$$\text{mask F} = \begin{matrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{matrix} \tag{7-17}$$

The application of mask E to the Charleston scene is shown in Fig. 7–25. It is a very noisy image (as one would expect when enhancing high-frequency detail) and is not very interpretable. A high-frequency filtered image of the Savannah River thermal infrared data produced using equation (7-15) is shown in Fig. 7–24c. This enhancement accentuates the bad scan lines. However, it also begins to highlight the information content in the heart or core of the plume and is therefore a useful enhancement.

Ford et al. (1983) had good results using a 5 × 5 convolution mask with the following coefficients for the enhancement of high-frequency detail of urban areas in Austin, Texas:

$$\begin{matrix} -0.3 & -0.3 & -0.3 & -0.3 & -0.3 \\ -0.3 & -0.3 & -0.3 & -0.3 & -0.3 \\ -0.3 & -0.3 & 9.7 & -0.3 & -0.3 \\ -0.3 & -0.3 & -0.3 & -0.3 & -0.3 \\ -0.3 & -0.3 & -0.3 & -0.3 & -0.3 \end{matrix} \tag{7-18}$$



**Figure 7–25** Result of applying a 3 × 3 high-frequency filter (mask E) to the Charleston, S.C., band 4 thematic mapper data.



OEEK06828

## EDGE ENHANCEMENT

For many remote sensing earth science applications the most valuable information that may be derived from an image is contained in the edges surrounding various objects or features of interest. The edge enhancement operation delineates these edges and thereby makes the shapes and details comprising the image more conspicuous and perhaps easier to analyze. Generally, what the eyes see as pictorial edges are simply sharp changes in brightness value between two adjacent pixels. The edges may be enhanced using either *linear* or *nonlinear* edge enhancement techniques. A survey of edge detection techniques is found in Davis (1975), Shaw (1979), and Peli and Malah (1982).

### Linear Edge Enhancement

One simple method of extracting edges in remotely sensed imagery is the application of the *running difference* operation. The original image is shifted by one picture element, and then tested for a difference between corresponding pixels in the original and shifted images. A midgray brightness value (e.g., BV = 127) is substituted for the pixel if there is no difference, a darker value (e.g., BV = 30) is substituted for the pixel if there is a negative difference, and a lighter value is substituted (e.g., BV = 200) if there is a positive difference. Hence transitions from light to dark produce white lines; from dark to light, black lines. The result is an image with gray background and black-and-white lines surrounding the edges of all objects (Spatial Data, 1975). The algorithms for enhancing horizontal, vertical, and diagonal edges are, respectively:

$$BV_{i,j} = BV_{i,j} - BV_{i-1,j} \tag{7-19}$$

and if $BV_{i,j} = 0$, then $BV_{i,j} = 127$
if $BV_{i,j} < 0$, then $BV_{i,j} = 30$
if $BV_{i,j} > 0$, then $BV_{i,j} = 200$

$$BV_{i,j} = BV_{i,j} - BV_{i,j+1} \tag{7-20}$$

and if $BV_{i,j} = 0$, then $BV_{i,j} = 127$
if $BV_{i,j} < 0$, then $BV_{i,j} = 30$
if $BV_{i,j} > 0$, then $BV_{i,j} = 200$

$$BV_{i,j} = BV_{i,j} - BV_{i+1,j+1} \tag{7-21}$$

and if $BV_{i,j} = 0$, then $BV_{i,j} = 127$
if $BV_{i,j} < 0$, then $BV_{i,j} = 30$
if $BV_{i,j} > 0$, then $BV_{i,j} = 200$

By selectively applying these enhancements it is possible to enhance only vertical, horizontal, or diagonal edges in the imagery.

It is also possible to perform edge enhancement by convolving the original



OFEK06829

**TABLE 7–3** Scale of Delta versus Kernel Size

| Delta, Δ | Smoothness/roughness | Kernel size |
|---|---|---|
| ≤±3 | Very smooth | 9 × 9 |
| ±4 | Smooth | |
| ±5 | Semismooth | 7 × 7 |
| ±6 | Smooth/rough | |
| ±7 | Rough/smooth | 5 × 5 |
| ±8 | Semirough | |
| ±9 | Rough | 3 × 3 |
| ≥±10 | Very rough | 1 × 1 |

*Source:* After Chavez and Bauer (1982).

data with a weighted mask or kernel, as discussed previously. Chavez and Bauer (1982) point out that the optimum kernel size (e.g., 3 × 3, 5 × 5, 7 × 7, or 9 × 9) typically used in edge enhancement is a function of the surface roughness and sun angle characteristics at the time the data were collected. Taking these factors into consideration, they developed a procedure based on the "first difference in the horizontal direction" of the input image using the equation:

$$\text{Diff } BV_{i,j} = BV_{i+1,j} - BV_{i,j} + 126 \tag{7-22}$$

This operator is applied to every row, $i$, in the image or every $m$th row. The value 126 is added so that the output data are in the range 0 to 255. This results in a first difference or derivative image. A histogram of the new image reveals generally how many edges are contained in the image. The standard deviation of the first difference image is computed and multiplied by 2.3 yielding a delta value, Δ, closely associated with surface roughness (Table 7–3). The following algorithm and the information presented in Table 7–3 are then used to select the appropriate kernel size:

$$\text{kernel}_{\text{size}} = 12 - \Delta \tag{7-23}$$

The method is intended to be used by earth scientists to select the optimum kernel size for enhancing the very fine image detail without having to try several different versions before selecting the appropriate kernel size for their area of interest.

Once the size of the mask is selected, various coefficients may be placed in it. For example, consider the following 3 × 3 *compass gradient masks* that can be used to perform two-dimensional discrete differentiation edge enhancement (Robinson, 1977):

$$\text{mask F} = \text{north} = \begin{matrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{matrix} \tag{7-24}$$

$$\text{mask G} = \text{northeast} = \begin{matrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{matrix} \tag{7-25}$$



OEEK06830

$$\text{mask H} = \text{east} = \begin{matrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{matrix} \tag{7-26}$$

$$\text{mask I} = \text{southeast} = \begin{matrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{matrix} \tag{7-27}$$

$$\text{mask J} = \text{south} = \begin{matrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{matrix} \tag{7-28}$$

$$\text{mask K} = \text{southwest} = \begin{matrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{matrix} \tag{7-29}$$

$$\text{mask L} = \text{west} = \begin{matrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{matrix} \tag{7-30}$$

$$\text{mask M} = \text{northwest} = \begin{matrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{matrix} \tag{7-31}$$

The compass names suggest the slope direction of maximum response. For example, the west gradient mask produces a maximum output for horizontal brightness value changes from right to left. The gradient masks have zero weighting (i.e., the sum of the mask coefficients is zero) (Pratt, 1978). This results in no output response over regions with constant brightness values (i.e., no edges are present).

North and east compass gradient masks were applied to the Charleston band 4 TM scene (Fig. 7–26). The north mask enhanced east-west trending edges while the east mask enhanced north-south trending edges. Although it is systematically surveyed, the street pattern of Charleston is not in a north-south orientation. In fact, it changes orientation just north of Colonial Lake. Also, as mentioned previously, there is a high degree of spectral heterogeneity in this urban area. These two factors combine to produce edge-enhanced images that do not provide very much information of value, although the land–water interface (edge) is highlighted quite well using the north compass gradient mask. Edge enhancements performed in areas of high-frequency detail often do not produce useful results.

A horizontal, west compass gradient mask was applied to the Savannah River thermal infrared data (Fig. 7–24d). This caused the western side of the river to exhibit a distinct edge. Also, the heart of the plume begins to be represented by a thin, dark line. It is also possible to combine enhancements. For example, Fig. 7–24e is the result of applying the same horizontal compass gradient mask to the high-frequency image in Fig. 7–24c. The edges of the plume and both sides of the river are very distinct in this example.

Edge enhancement without regard to edge direction may be obtained by applying



OFEK06831

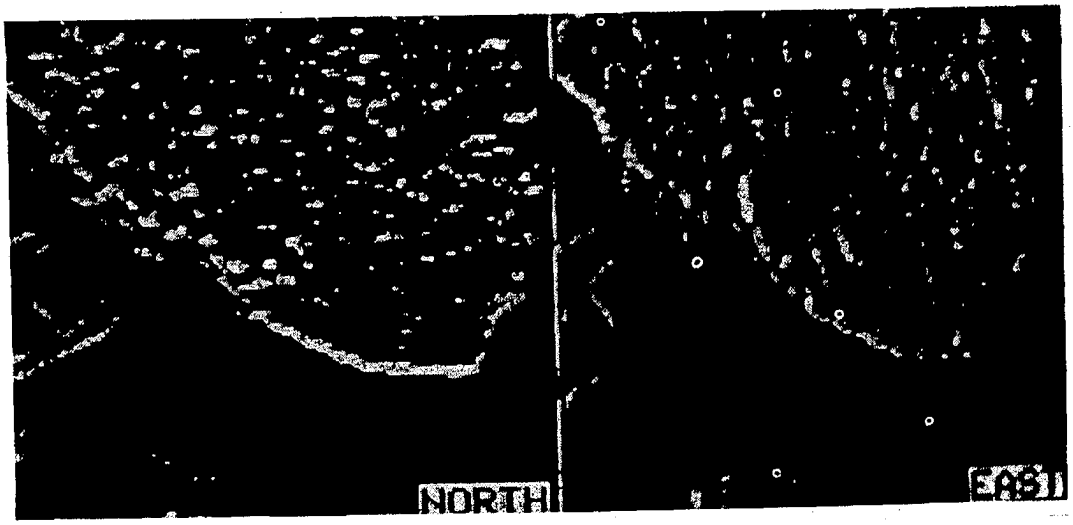



**Figure 7–26** Two compass gradient masks applied to the band 4 thematic mapper data of a portion of the Charleston, S.C., scene. The north mask enhances edges trending east-west, while the east mask enhances edges running north-south.

a *Laplacian* convolution mask to the imagery. The compass gradient is generally preferred for finding lines, whereas the Laplacian is more useful in edge and boundary detection. The Laplacian is a second derivative (as opposed to the gradient, which is a first derivative) and is consequently more troubled by image noise (Hord, 1982). Several 3 × 3 Laplacian filters (and weights) are shown below (Robinson, 1977; Pratt, 1978; Sullivan et al., 1984):

$$\text{mask N} = \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \tag{7-32}$$

$$\text{mask O} = \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix} \tag{7-33}$$

$$\text{mask P} = \begin{matrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{matrix} \tag{7-34}$$

The Laplacian edge enhancement emphasizes maximum values, or peaks, within the image. Human vision physiological research suggests that we see objects in much the same way. Hence the use of this operation produces a more natural look than that of many of the other edge-enhanced images. The computation of Laplacian mask N for a given 3 × 3 region in the image would be:

$$\text{Lap}_{5,\text{out}} = [-1(BV_2)] + [-1(BV_4)] + [-1(BV_6)] + [-1(BV_8)] + [4(BV_5)] \tag{7-35}$$

All three of these Laplacian masks were applied to the Charleston image (Fig. 7–27). Again, no enhancement provided particularly useful results in the urban area.



OEEK06832

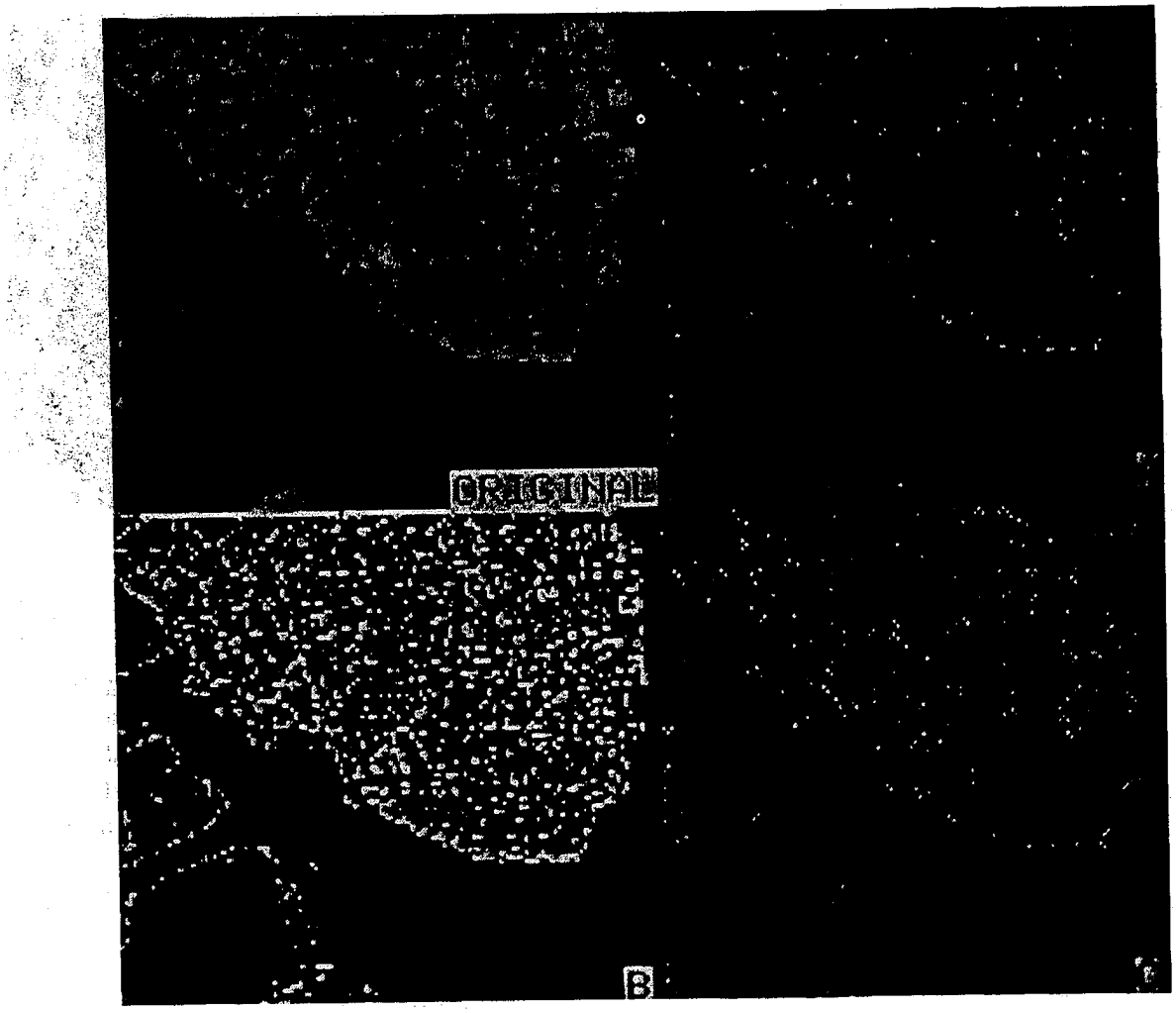



**Figure 7–27** Result of applying several Laplacian masks (N, O, and P) to the original band 4 thematic mapper data of Charleston, S.C.

However, mask O did accentuate the edge of the wetland–water interface in the southwest part of the image.

Often, a Laplacian edge enhancement is added back to the original image using the following mask (Bernstein, 1983):

$$\text{mask } Q = \begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix} \tag{7-36}$$

The result of applying this enhancement to the Charleston scene is shown in Fig. 7–28. This is perhaps the best enhancement of high-frequency detail of the urban area obtained thus far. Also, note the increased detail in the wetland.

There are numerous coefficients that can be placed in a 3 × 3 convolution



OEEK06833

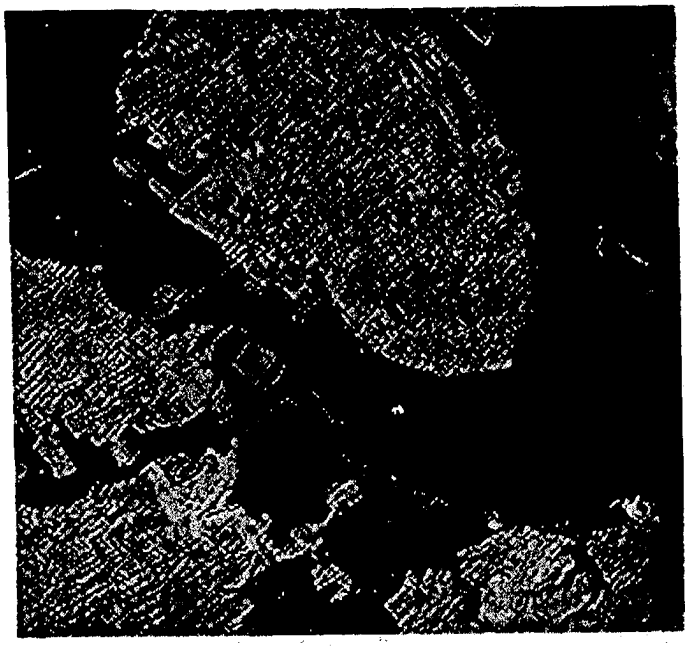


**Figure 7–28** Result of applying Laplacian mask Q to the remotely sensed data. This mask computes the Laplacian edge enhancement and adds it back onto the original image. Note the increased detail both in the urban area and especially in the wetland.

kernel or mask. Usually, the analyst works interactively with the remotely sensed data, trying new coefficients and selecting those that produce the most effective edge-enhancement results. It is also possible to use combinations of operators for edge detection. For example, Eberlein and Weszka (1975) found that a combination of a gradient and Laplacian edge operators was superior to using either edge enhancement alone.

### *Nonlinear Edge Enhancement*

Nonlinear edge enhancements are performed using nonlinear combinations of pixels. Many algorithms are applied using either 2 × 2 or 3 × 3 kernels. The *Sobel edge detector* is based on the notation of the 3 × 3 window described previously and is computed according to the relationship (Duda and Hart, 1973; Gonzalez and Wintz, 1977):

$$\text{Sobel}_{5,\text{out}} = \sqrt{X^2 + Y^2} \tag{7-37}$$

where

$$X = [BV_3 + (2(BV_6)) + BV_9] - [BV_1 + (2(BV_4)) + BV_7] \tag{7-38}$$

and

$$Y = [BV_1 + (2(BV_2)) + BV_3] - [BV_7 + (2(BV_8)) + BV_9] \tag{7-39}$$

A Sobel edge enhancement of the Savannah River thermal plume is shown in Fig. 7–24f. It is a very effective edge enhancement with the heart and sides of the plume surrounded by bright white lines. A piecewise linear contrast stretch of the



OFEK06834

Sobel edge enhancement is shown in Fig. 7–24g. This saturates some detail in the plume while picking up some edge information in the swamp. Finally, a Sobel edge enhancement is applied to the low-frequency filtered image (Fig. 7–24h). This is perhaps the most striking edge enhancement, as the hot core of the plume turns into a dark band that extends not only into the river but up into the tributary.

The *Robert's edge detector* is based on the following $2 \times 2$ pixel window:

$$\begin{matrix} BV_1 & BV_2 \\ BV_3 & BV_4 \end{matrix} \tag{7-40}$$

The new pixel value at pixel location $BV_1$ is computed according to the equation (Gonzalez and Wintz, 1977; Peli and Malah, 1982):

$$BV_{1,\text{out}} = \text{Abs}\,(BV_1 - BV_4) + \text{Abs}\,(BV_2 - BV_3) \tag{7-41}$$

where Abs refers to absolute values. A Robert's edge enhancement of the thermal plume is shown in Fig. 7–24i. It produced results very similar to Sobel edge enhancement.

The *Kirsch* nonlinear edge enhancement calculates the gradient at pixel location $BV_{i,j}$. To apply this operator, however, it is first necessary to designate a different $3 \times 3$ window numbering scheme than we have used in previous discussions:

$$\text{window numbering for Kirsch} = \begin{matrix} BV_0 & BV_1 & BV_2 \\ BV_7 & BV_{i,j} & BV_3 \\ BV_6 & BV_5 & BV_4 \end{matrix} \tag{7-42}$$

The algorithm applied is (Robinson, 1977; Gil et al., 1983):

$$BV_{i,j} = \max\left\{1, \max_{i=0}^{7} [\text{Abs}\,(5S_i - 3T_i)]\right\} \tag{7-43}$$

where

$$S_i = BV_i + BV_{i+1} + BV_{i+2} \tag{7-44}$$

and

$$T_i = BV_{i+3} + BV_{i+4} + BV_{i+5} + BV_{i+6} + BV_{i+7} \tag{7-45}$$

The subscripts of BV are evaluated modulo 8, meaning that the computation moves around the perimeter of the mask in eight steps. The edge enhancement computes the maximal compass gradient magnitude about input image point $BV_{i,j}$. The value of $S_i$ equals the sum of three adjacent pixels, while $T_i$ equals the sum of the remaining four adjacent pixels. The input pixel value at $BV_{i,j}$ is never used in the computation.

For readers interested in the detection of lines as well as edges, refer to the work by Gonzalez and Wintz (1977), Vanderbrug (1976), and Chittineni (1983). Several masks for detecting lines are

$$\text{mask R} = \begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix} \tag{7-46}$$



OFEK06835

$$\text{mask S} = \begin{matrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{matrix} \quad (7\text{-}47)$$

$$\text{mask T} = \begin{matrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{matrix} \quad (7\text{-}48)$$

$$\text{mask U} = \begin{matrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{matrix} \quad (7\text{-}49)$$

Mask R will enhance lines in the imagery that are oriented horizontally. Lines oriented at 45° will be enhanced using mask S. Vertical lines and lines oriented −45° will be enhanced using masks T and U, respectively.

## SPECIAL TRANSFORMATIONS

### Principal Components Analysis

*Principal Components Analysis* (also referred to as PCA, factor or Karhunen–Loeve analysis) has proven to be of significant value in the analysis of remotely sensed digital data. The transformation of the raw remote sensor data using PCA can result in new principal component images that are often more interpretable than the original data (Kaneko, 1978; Byrne et al., 1980). Also, PCA analysis may be used to compress the information content of a number of bands of imagery (e.g., seven thematic mapper bands) into just two or three transformed principal component images. This ability to reduce the *dimensionality* (i.e., the number of bands in the data set that must be analyzed to produce usable results) from seven to two or three bands is an important economic consideration, especially if the potential information recoverable from the transformed data is just as good as that from the original remote sensor data (Wheeler and Misra, 1976; Anuta, 1977; and Haralick and Fu, 1983).

To perform principal component analysis we apply a *transformation* to a correlated set of multispectral data. For example, the Charleston, S.C., TM scene is a likely candidate since bands 1, 2, and 3 are highly correlated, as are bands 5 and 6 (Table 7–4). The application of the transformation to the correlated remote sensor data will result in an *uncorrelated* multispectral data set that has certain ordered variance properties. This transformation is conceptualized best by considering the two-dimensional distribution of pixel values obtained in two TM bands which we will label simply $X_1$ and $X_2$. A scatterplot of all the brightness values associated with each pixel in each band is shown in Fig. 7–29a together with the location of the respective means $\mu_1$ and $\mu_2$. The spread or variance of the distribution of points is an indication of the correlation and quality of information associated with both bands. If all the data points clustered in an extremely tight zone in the two-dimensional space, these data would probably provide very little information of value.

The initial measurement coordinate axes ($X_1$ and $X_2$) may not necessarily be



OEEK06836

**TABLE 7–4** Charleston, S.C., Thematic Mapper Scene Statistics Used in the Principal Components Analysis

| Band Number: | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| μm: | 0.45–0.52 | 0.52–0.60 | 0.63–0.69 | 0.76–0.90 | 1.55–1.75 | 2.08–2.35 | 10.4–12.5 |
| | | | *Univariate statistics* | | | | |
| Mean | 64.8 | 25.6 | 23.7 | 27.3 | 32.4 | 15.0 | 110.6 |
| Std. dev. | 10.05 | 5.84 | 8.30 | 15.76 | 23.85 | 12.45 | 4.21 |
| Variance | 100.9 | 34.1 | 68.8 | 248.4 | 568.8 | 154.9 | 17.78 |
| Minimum | 51 | 17 | 14 | 4 | 0 | 0 | 90 |
| Maximum | 242 | 115 | 131 | 105 | 193 | 128 | 130 |
| | | | *Variance-covariance matrix* | | | | |
| 1 | 100.93 | | | | | | |
| 2 | 56.60 | 34.14 | | | | | |
| 3 | 79.43 | 46.71 | 68.83 | | | | |
| 4 | 61.49 | 40.68 | 69.59 | 248.40 | | | |
| 5 | 134.27 | 85.22 | 141.04 | 330.71 | 568.84 | | |
| 6 | 90.13 | 55.14 | 86.91 | 148.50 | 280.97 | 154.92 | |
| 7 | 23.72 | 14.33 | 22.92 | 43.62 | 78.91 | 42.65 | 17.78 |
| | | | *Correlation matrix* | | | | |
| 1 | 1.00 | | | | | | |
| 2 | 0.96 | 1.00 | | | | | |
| 3 | 0.95 | 0.96 | 1.00 | | | | |
| 4 | 0.39 | 0.44 | 0.53 | 1.00 | | | |
| 5 | 0.56 | 0.61 | 0.71 | 0.88 | 1.00 | | |
| 6 | 0.72 | 0.76 | 0.84 | 0.76 | 0.95 | 1.00 | |
| 7 | 0.56 | 0.58 | 0.66 | 0.66 | 0.78 | 0.81 | 1.00 |

the best arrangement in multispectral feature space to analyze the remote sensor data associated with these two bands. The goal is to use principal components analysis to *translate* and/or *rotate* the original axes so that the original brightness values on axes $X_1$ and $X_2$ are redistributed (actually reprojected) onto a new set of axes or dimensions, $X_1'$ and $X_2'$. For example, the best *translation* for the original data points from $X_1$ to $X_1'$ and from $X_2$ to $X_2'$ coordinate systems might be the simple relationship $X_1' = X_1 - \mu_1$ and $X_2' = X_2 - \mu_2$. Thus, the origin of the new coordinate system ($X_1'$ and $X_2'$) now lies at the location of both means in the original scatter of points (Fig. 7–29b).

The $X'$ coordinate system might then be *rotated* about its new origin ($\mu_1$, $\mu_2$) in the new coordinate system some $\phi$ degrees so that the first axis $X_1'$ is associated with the maximum amount of variance in the scatter of points (Fig. 7–29c). This new axis is called the first *principal component* ($PC_1 = \lambda_1$). The second principal component ($PC_2 = \lambda_2$) is perpendicular (orthogonal) to $PC_1$. Thus, the major and minor axes of the ellipsoid of points in bands $X_1$ and $X_2$ are called the principal components. The third, fourth, fifth, and so on, components contain monotonically decreasing amounts of the variance found in the data set.



OFEK06837

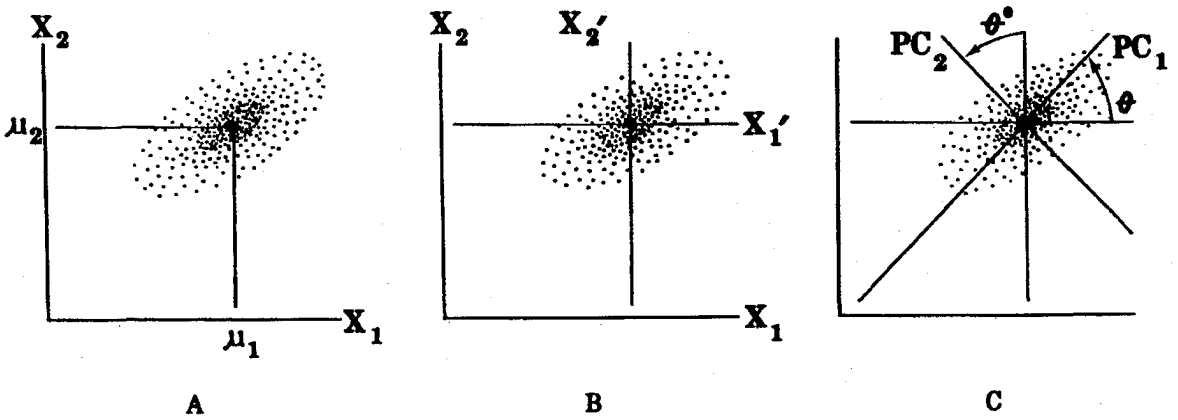



**Figure 7–29** Diagrammatic representation of the spatial relationship between the first two principal components. (a) A scatterplot of data points collected from two remotely bands labeled $X_1$ and $X_2$ with the means of the distribution labeled $\mu_1$ and $\mu_2$. (b) A new coordinate system is created by shifting the axes to an $X'$ system. The values for the new data points are found by the relationship $X_1' = X_1 - \mu_1$ and $X_2' = X_2 - \mu_2$. (c) The $X'$ axis system is then rotated about its origin ($\mu_1$, $\mu_2$) so that $PC_1$ is projected through the semimajor axis of the distribution of points and that the variance of $PC_2$ is a maximum. $PC_2$ must be perpendicular to $PC_1$. The PC axes are the principal components of this two-dimensional data space. Component 1 usually accounts for approximately 90% of the variance, with component 2 accounting for approximately 5%. (After Short, 1982.)

To transform (reproject) the original data on the $X_1$ and $X_2$ axes onto the $PC_1$ and $PC_2$ axes, we must obtain certain transformation coefficients that we can apply in a linear fashion to the original pixel values. The linear transformation required is derived from the covariance matrix of the original data set. Thus, this is a data-dependent process, with each new data set yielding different transformation coefficients.

The transformation is computed from the original spectral statistics as follows (Anuta, 1977; Short, 1982):

1. The $n \times n$ *covariance* matrix, Cov, of the $n$-dimensional remote sensing data set to be transformed is computed (Table 7–4).
2. The eigenvalues, $E = [\lambda_{1,1}, \lambda_{2,2}, \lambda_{3,3}, \ldots \lambda_{n,n}]$, and eigenvectors, $EV = [a_{kp}$ . . . for $k = 1$ to $n$ bands, and $p = 1$ to $n$ components] of the covariance matrix are computed such that:

$$\underset{[n \times n]}{EV}\ \underset{[n \times n]}{Cov}\ \underset{[n \times n]}{EV^T} = \begin{bmatrix} \lambda_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_{3,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{4,4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_{5,5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_{6,6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{n,n} \end{bmatrix} \quad E \quad (7\text{-}50)$$



OEEK06838

where $EV^T$ is the transpose of the eigenvector matrix, EV, and $E$ is a diagonal covariance matrix whose elements $\lambda_{ii}$, called *eigenvalues*, are the variances of the $p$th *principal components*, where $p = 1$ to $n$ components. The nondiagonal eigenvalues, $\lambda_{ij}$, are equal to zero and therefore can be ignored. The number of nonzero eigenvalues in an $n \times n$ covariance matrix always equals $n$, the number of bands examined. The eigenvalues are often called *components* (i.e., eigenvalue 1 may be referred to as principal component 1).

Eigenvalues and eigenvectors were computed for the Charleston, S.C., TM scene (Tables 7–5 and 7–6). Such computations can be performed using most statistical analysis packages such as SAS or SPSS. Also, Cooley and Lohnes (1971) provide a FORTRAN program that performs principal components analysis.

The eigenvalues contain important information. For example, it is possible to determine the percent of total variance explained by each of the principal components, $\%_p$, using the equation:

$$\%_p = \frac{\text{eigenvalue } \lambda_p \times 100}{\sum_{p=1}^{n} \text{eigenvalue } \lambda_p} \tag{7-51}$$

where $\lambda_p$ is the $p$th eigenvalue out of the possible $n$ eigenvalues. The first principal component (eigenvalue $\lambda_1$) of the Charleston TM scene accounts for 84.68% of the variance in the entire multispectral data set (Table 7–5). Component 2 accounts for

**TABLE 7–5** Eigenvalues Computed for the Covariance Matrix Found in Table 7–4.

| | Component$_p$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Eigenvalues, $\lambda_p$ | 1010.92 | 131.20 | 37.60 | 6.73 | 3.95 | 2.17 | 1.24 |
| Difference | 879.72 | 93.59 | 30.88 | 2.77 | 1.77 | 0.93 | — |

Total variance = 1193.81

Percent of total variance in the data explained by each component:

Computed as:

$$\%_p = \frac{\text{eigenvalue } \lambda_p \times 100}{\sum_{p=1}^{7} \text{eigenvalue } \lambda_p}$$

For example,

$$\sum_{p=1}^{7} \lambda_p = 1010.92 + 131.20 + 37.60 + 6.73 + 3.95 + 2.17 + 1.24 = 1193.81$$

$$\text{Percentage of variance explained by first component} = \frac{1010.92 \times 100}{1193.81} = 84.68$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Percentage: | 84.68 | 10.99 | 3.15 | 0.56 | 0.33 | 0.18 | 0.10 |
| Cumulative: | 84.68 | 95.67 | 98.82 | 99.38 | 99.71 | 99.89 | 99.99 |



OFEK06839

**TABLE 7–6** Eigenvectors ($a_{kp}$) Computed for the Covariance Matrix Found in Table 7–4.

| | | Component$_p$ | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Band$_k$ | 1 | 0.205 | 0.637 | 0.327 | −0.054 | 0.249 | −0.611 | −0.079 |
| | 2 | 0.127 | 0.342 | 0.169 | −0.077 | 0.012 | 0.396 | 0.821 |
| | 3 | 0.204 | 0.428 | 0.159 | −0.076 | −0.075 | 0.649 | −0.562 |
| | 4 | 0.443 | −0.471 | 0.739 | 0.107 | −0.153 | −0.019 | −0.004 |
| | 5 | 0.742 | −0.177 | −0.437 | −0.300 | 0.370 | 0.007 | 0.011 |
| | 6 | 0.376 | 0.197 | −0.309 | −0.312 | −0.769 | −0.181 | 0.051 |
| | 7 | 0.106 | 0.033 | −0.080 | 0.887 | 0.424 | 0.122 | 0.005 |

10.99% of the remaining variance. Cumulatively, these first two principal components account for 95.67% of the variance. The third component accounts for another 3.15%, bringing the total to 98.82% of the variance explained by the first three components (Table 7–5). Thus, the seven-band TM data set of Charleston might be compressed into just three new principal component images (or bands) that explain 98.82% of the variance.

But what do these new components represent? For example, what does component 1 stand for? By computing the correlation of each band $k$ with each component $p$, it is possible to determine how each band "loads" or is associated with each principal component. The equation is:

$$R_{kp} = \frac{a_{kp} \times \sqrt{\lambda_p}}{\sqrt{\text{Var}_k}} \tag{7-52}$$

where $a_{kp}$ = eigenvector for band $k$ and component $p$
$\lambda_p$ = $p$th eigenvector (component)
$\text{Var}_k$ = variance of band $k$ in the covariance matrix

This computation results in a new $n \times n$ matrix (Table 7–7) filled with *factor loadings*. For example, the highest correlations (i.e., factor loadings) for principal component 1 were for bands 4, 5, and 6 (0.894, 0.989, and 0.961, respectively, Table 7–7). This suggests that this component is a near- and middle-infrared reflectance band. Conversely, principal component 2 has high loadings only in the visible bands 1, 2, and 3 (0.726, 0.670, and 0.592). This is a visible spectrum component. Component 3 provides very little information of value with no loading above 0.287. Component 4 accounts for very little of the variance but is easy to label since it loads heavily (0.545) on the thermal infrared band 7. Components 5, 6, and 7 provide no useful information and should probably not be analyzed further.

Now that we understand what "information" each component contributes, it is useful to see what the images of these components look like. To do this it is necessary first to identify the brightness values ($BV_{i,j,k}$) associated with a given


OFEK06840

**TABLE 7-7** Degree of Correlation, $R_{kp}$, between Each Band$_k$ and Each Principal Component$_p$

Computed as:

$$R_{kp} = \frac{a_{kp} \times \sqrt{\lambda_p}}{\sqrt{\text{Var}_k}}$$

For example,

$$R_{1,1} = \frac{0.205 \times \sqrt{1010.92}}{\sqrt{100.93}} = \frac{0.205 \times 31.795}{10.046} = 0.649$$

$$R_{5,1} = \frac{0.742 \times \sqrt{1010.92}}{\sqrt{568.84}} = \frac{0.742 \times 31.795}{23.85} = 0.989$$

$$R_{2,2} = \frac{0.342 \times \sqrt{131.20}}{\sqrt{34.14}} = \frac{0.342 \times 11.45}{5.842} = 0.670$$

| | Component$_p$ | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | 0.649 | 0.726 | 0.199 | −0.014 | 0.049 | −0.089 | −0.008 |
| | 0.694 | 0.670 | 0.178 | −0.034 | 0.004 | 0.099 | 0.157 |
| | 0.785 | 0.592 | 0.118 | −0.023 | −0.018 | 0.115 | −0.075 |
| Band$_k$ | 0.894 | −0.342 | 0.287 | 0.017 | −0.019 | −0.002 | −0.000 |
| | 0.989 | −0.084 | −0.112 | −0.032 | 0.030 | 0.000 | 0.000 |
| | 0.961 | 0.181 | −0.152 | 0.065 | −0.122 | −0.021 | 0.004 |
| | 0.799 | 0.089 | −0.116 | 0.545 | 0.200 | 0.042 | 0.001 |

pixel. In this case we evaluate the first pixel in a hypothetical image at row 1, column 1 for each of seven bands. We will represent this as the vector $X$, such that:

$$X = \begin{bmatrix} BV_{1,1,1} = 20 \\ BV_{1,1,2} = 30 \\ BV_{1,1,3} = 22 \\ BV_{1,1,4} = 60 \\ BV_{1,1,5} = 70 \\ BV_{1,1,6} = 62 \\ BV_{1,1,7} = 50 \end{bmatrix} \tag{7-53}$$

We will now apply the appropriate transformation to these data such that they are projected onto the first principal component's axes. In this way we will find out what the new brightness value (new$BV_{i,j,p}$) will be for this component, $p$. It is computed according to the formula:

$$\text{new}BV_{i,j,p} = \sum_{k=1}^{n} a_{kp} BV_{i,j,k} \tag{7-54}$$



OFEK06841

where $a_{kp}$ = factor loading (eivenvector)
$BV_{ijk}$ = brightness value in band $k$ for the pixel at row $i$ column $j$
$n$ = number of bands

In our hypothetical example this would yield:

$$\begin{aligned}\text{new}BV_{1,1,1} &= a_{1,1}(BV_{1,1,1}) + a_{2,1}(BV_{1,1,2}) + a_{3,1}(BV_{1,1,3}) + a_{4,1}(BV_{1,1,4}) \\ &\quad + a_{5,1}(BV_{1,1,5}) + a_{6,1}(BV_{1,1,6}) + a_{7,1}(BV_{1,1,7}) \\ &= 0.205\,(20) + 0.127\,(30) + 0.204\,(22) + 0.443\,(60) + 0.742\,(70) \\ &\quad + 0.376\,(62) + 0.106\,(50) \\ &= 119.53\end{aligned}$$

This is a "pseudo-measurement" that is a linear combination of original brightness values and factor loadings (eigenvectors). The new brightness value for row 1, column 1 in principal component 1 after truncation to an integer is $\text{new}BV_{1,1,1} = 119$.

This procedure takes place for every pixel in the original data to produce the principal component 1 data set. Then $p$ is incremented by 1 and principal component 2 is created pixel by pixel. This is the method used to produce the principal component images (sometimes referred to as eigenimages) shown in Fig. 7–30. If desired, any two or three of the principal components can be additively filtered with blue, green, and/or red light to create a principal component color composite. These displays often depict more subtle differences in color shading and distribution than the traditional color–infrared color composite images.

If components 1, 2, and 4 are of value, perhaps the original seven bands of TM data can be set aside and the remainder of the image enhancement or classification can be performed using just these three principal component images. This greatly reduces the amount of data to be analyzed and completely bypasses the expensive and time-consuming process of "feature selection" so often necessary when classifying remotely sensed data (discussed in Chapter 8).

Gunther (1982) developed a two-mode PCA that computes the first and second principal components for three sets of MSS data: set one is MSS 4 and 5 data, set two is MSS 5 and 6, and set three is MSS 6 and 7 data. This produces six PCA axes and images that can be color composited to enhance the terrain.

## *Vegetation Indices*[1]

The collection of accurate, timely information on the world's food and fiber crops continues to be important. The terrible famine and starvation in Ethiopia in 1984–1985 is a sufficient example. The collection of such information using in situ techniques is often expensive, time consuming, and occasionally impossible. An alternative is the measurement of vegetative amount and condition based on an analysis of remote sensing spectral measurements.

[1] This section contains some information reprinted with permission from C. R. Perry, and L. F. Lautenschlager, 1984, "Functional Equivalence of Spectral Vegetation Indices," *Remote Sensing of Environment*, Vol. 14, pp. 169–182. © 1984 by Elsevier Science Publishing Co.


OFEK06842

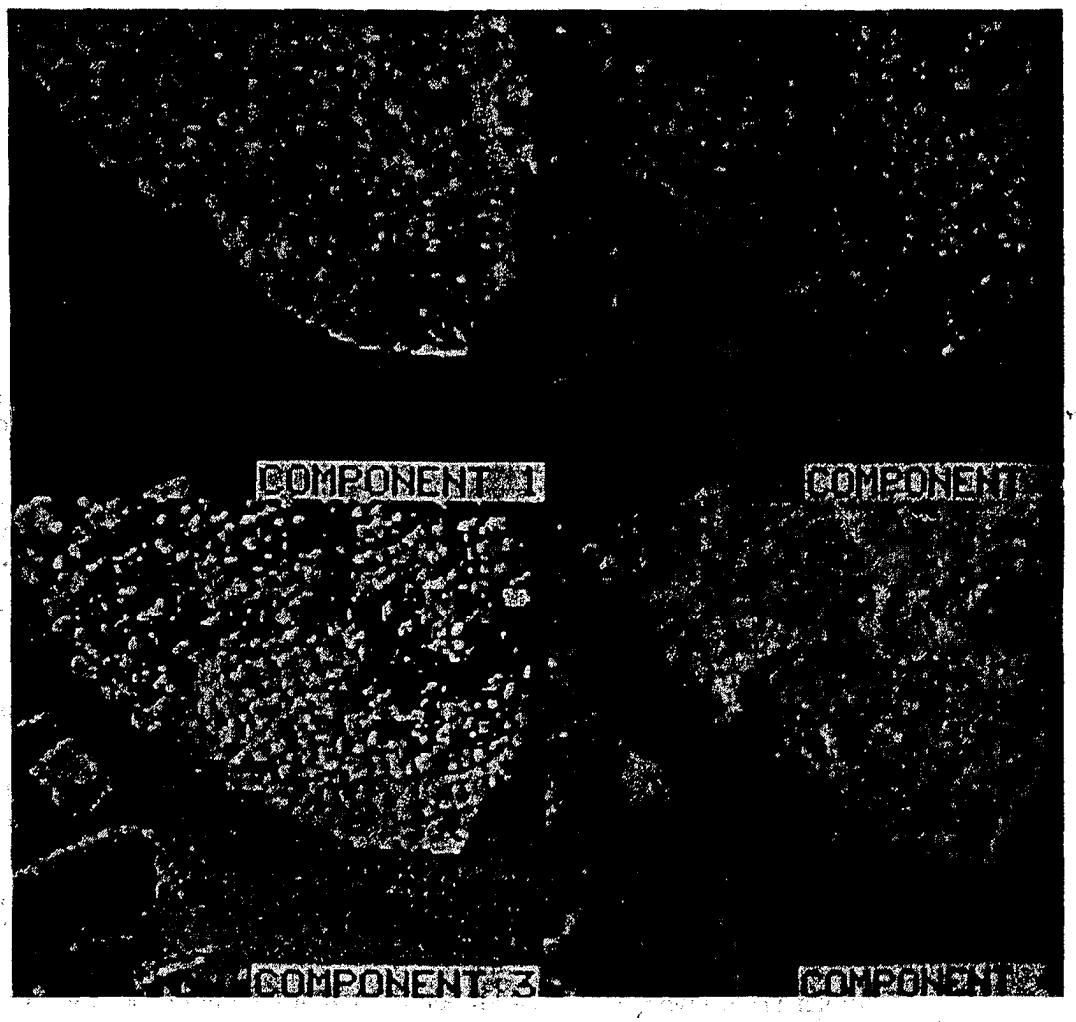



**Figure 7–30** First four principal component images of the Charleston, S.C., thematic mapper data computed using all seven bands. Component 1 consists of both near- and middle-infrared information (bands 4, 5, and 6). Component 2 contains primarily visible light information (bands 1, 2, and 3). Component 3 is difficult to interpret. Component 4 consists of the thermal infrared information contributed by band 7. Thus the seven-band TM data can be reduced in dimension to just three principal components (1, 2, and 4) that account for approximately 98% of the variance.

Much of the research in this area has involved the analysis of Landsat multispectral scanner (MSS) and thematic mapper (TM) data using digital image processing techniques. The goal has often been to reduce the four to seven bands of MSS or TM data down to a single number per pixel that predicts or assesses such canopy characteristics as biomass, productivity, leaf area, and/or percent vegetative ground cover. This section identifies several algorithms used to extract such information from remotely sensed data, which are referred to collectively as *vegetation indices* (VIs). It will be shown that many of the indices are redundant in information content and should be used judiciously.


OFEK06843

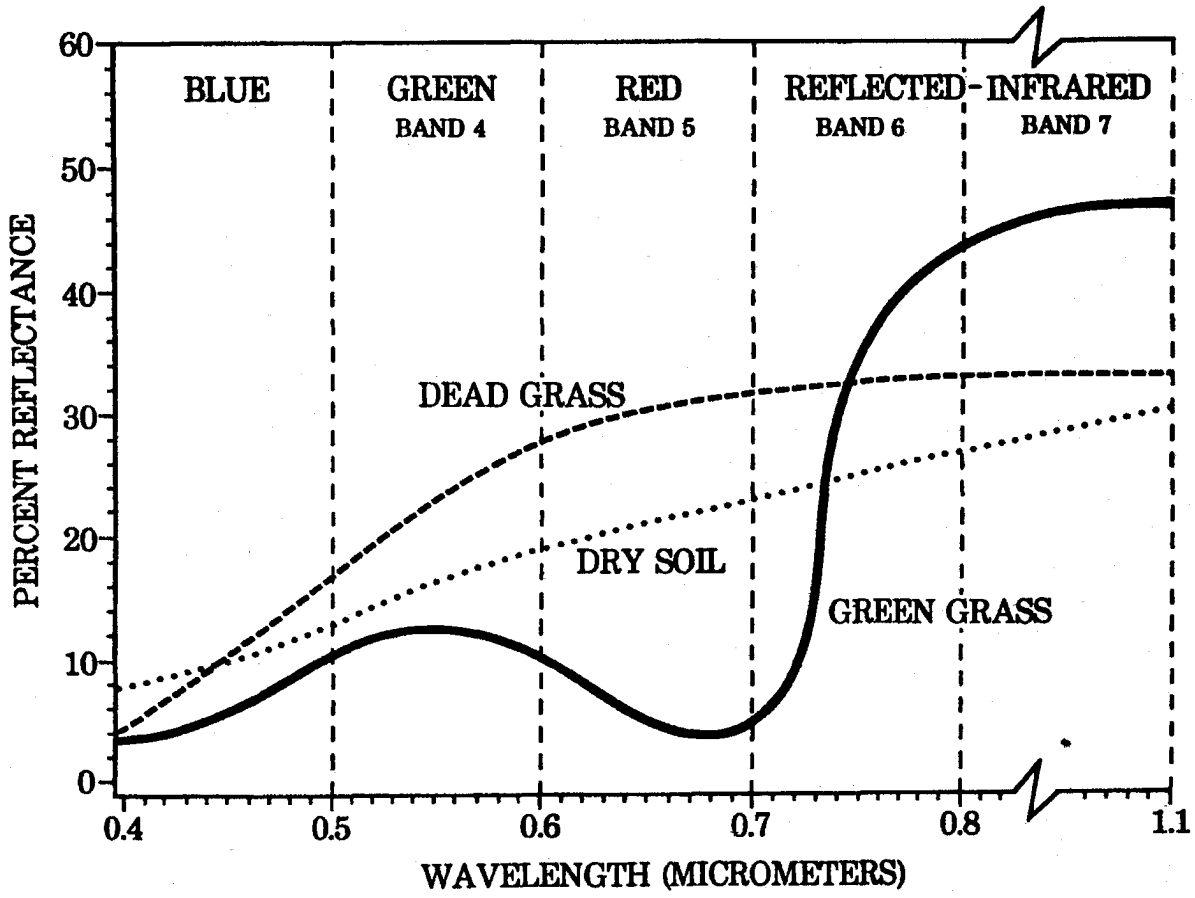



**Figure 7–31** Typical spectral reflectance characteristics for healthy green grass, dead or senescing grass, and bare dry soil for the wavelength interval from 0.4 to 1.1 μm. [After Deering et al., (1975) and Perry and Lautenschlager (1984)]. © 1984 Elsevier Science Publishing Co. Inc. Used with permission.

Typical spectral reflectance characteristics for healthy green vegetation, dead or sensecent vegetation, and dry soil are shown in Fig. 7–31. Healthy green vegetation generally reflects 40 to 50% of the incident near-infrared energy (0.7 to 1.1 μm), with the chlorophyll in the plants absorbing approximately 80 to 90% of the incident energy in the visible (0.4 to 0.7 μm) part of the spectrum (Jensen, 1983). Dead or senescent vegetation reflects a greater amount of energy than healthy green vegetation throughout the visible spectrum (0.4 to 0.7 μm). Conversely, it reflects less than green vegetation in the reflective infrared region. Dry soil generally has higher reflectance than green vegetation and lower reflectance than dead vegetation in the visible region, whereas in the near-infrared, dry soil generally has lower reflectance than green or senescent vegetation. Most vegetation indices are based on the fact that there are significant differences in the shape of these three curves. Remote sensing can provide no useful crop condition information if all three curves are situated on top of one another.


OFEK06844

The brightness values (BVs) from an individual MSS band (e.g., Landsat MSS4, MSS5, MSS6, MSS7) have been used as vegetation indexes to estimate percent ground cover and vegetative biomass (Wiegand et al., 1974; Tucker, 1979). Correlation coefficients from 0.30 for MSS7 with crop cover to 0.88 for MSS6 with leaf area index have been reported. This is certainly the most computationally simple type of vegetative index.

Between-band ratioing of MSS and TM brightness values have been used to estimate and monitor green biomass. The discussion on band ratioing earlier in this chapter describes how the ratios are computed. For example, a simple red/infrared ratio of TM bands 3 and 4 of the Charleston image is shown in Fig. 7–20. The darker the ratio pixel, the greater the amount of vegetative matter.

One of the first successful vegetation indices based on band ratioing was developed by Rouse et al. (1973). They computed the normalized difference of brightness values from MSS7 and MSS5 for monitoring vegetation. They called it the *normalized difference index*, hereafter referred to as ND7. Deering et al. (1975) added 0.5 to ND7 and took the square root, producing the *transformed vegetation index*, or TVI7. Similar algorithms using MSS6 and MSS5 were also used:

$$\text{ND6} = \frac{\text{MSS6} - \text{MSS5}}{\text{MSS6} + \text{MSS5}} \tag{7-55}$$

$$\text{ND7} = \frac{\text{MSS7} - \text{MSS5}}{\text{MSS7} + \text{MSS5}} \tag{7-56}$$

$$\text{TVI6} = \sqrt{\text{ND6} + 0.5} \tag{7-57}$$

$$\text{TVI7} = \sqrt{\text{ND7} + 0.5} \tag{7-58}$$

Perry and Lautenschlager (1984) found that the addition of 0.5 did not eliminate negative values altogether. Therefore, they proposed the following algorithms:

$$\text{TVI6} = \frac{\text{ND6} + 0.5}{\text{Abs}(\text{ND6} + 0.5)} \times \sqrt{\text{Abs}(\text{ND6} + 0.5)} \tag{7-59}$$

$$\text{TVI7} = \frac{\text{ND7} + 0.5}{\text{Abs}(\text{ND7} + 0.5)} \times \sqrt{\text{Abs}(\text{ND7} + 0.5)} \tag{7-60}$$

where Abs is the absolute value and 0/0 is equal to 1.

Transformed vegetation indices such as those discussed have been used extensively to measure vegetation amount (Curran, 1980; Lulla, 1983). For example, NOAA now provides a standard vegetation index map compiled weekly for both hemispheres from NOAA AVHRR (advanced very high resolution radiometer) meterological satellite data (Tarpley, 1983; Fig. 7–32). Also, a simple red/reflective infrared ratio was the basis of the FILE sensor system flown on board the Space Shuttle. It consisted of two imaging cameras used to perform on-board mapping of earth phenomena into one of four categories: water, vegetation, bare land, and clouds–snow–ice. Each camera contained charge-coupled detectors (100 × 100 elements) which measured



OFEK06845

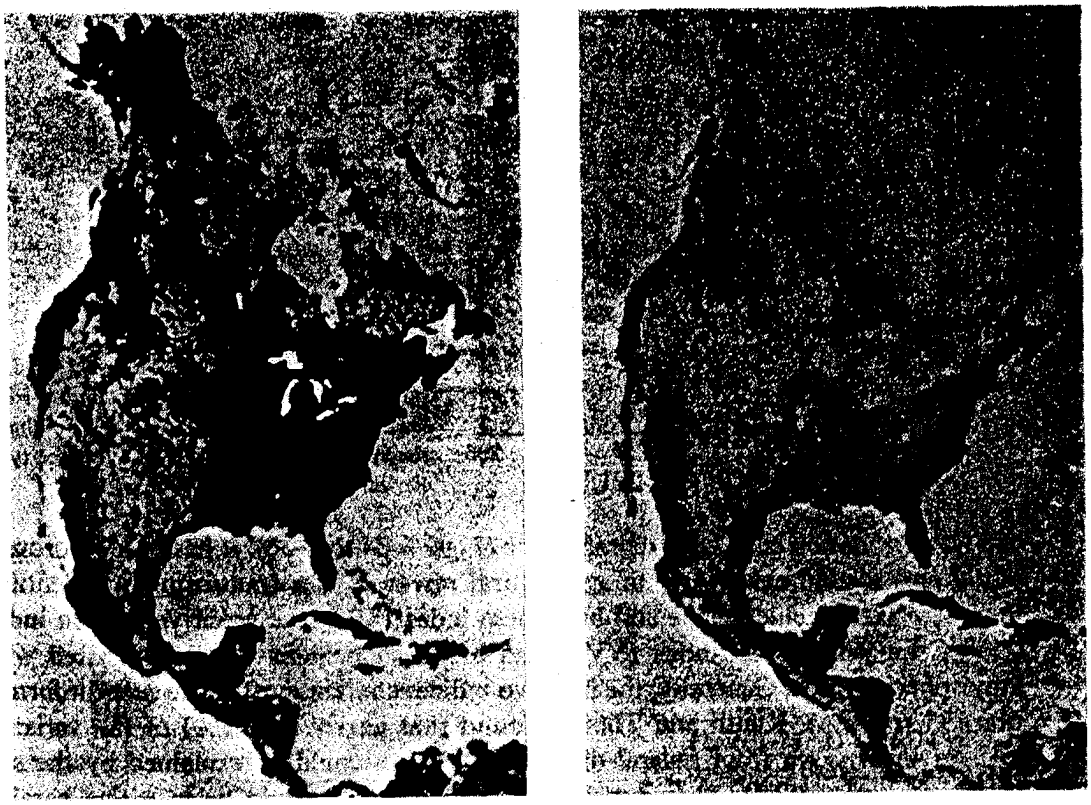


**Figure 7–32** Vegetation index maps produced by applying transformed vegetation index algorithms to NOAA advanced very high resolution radiometer (AVHRR) meteorological satellite data. (From Tarpley, 1983.)

surface reflectivity at wavelengths centered at 0.65 and 0.85 $\mu$m with bandwidths of 0.02 $\mu$m. Pattern recognition algorithms based on ratios of the red and reflected infrared reflectance values (Fig. 7–33) were used to make real-time classification decisions (Sivertson et al., 1982). This is one of the first real-time, thematic mapping sensors that could transmit the information to the earth or use it internally to control the operation of other sensors and avoid the collection of unwanted or unusable data (such as cloud-covered areas).

Two simple vegetation indices based on *image differencing* have found use. They include the DVI, developed by Richardson and Weigand (1977), and the AVI, proposed by Ashburn (1978):

$$\text{DVI} = 2.4\ \text{MSS7} - \text{MSS5} \tag{7-61}$$

$$\text{AVI} = 2.0\ \text{MSS7} - \text{MSS5} \tag{7-62}$$

One of the most important vegetation indices is the *tassled cap transformation* developed by Kauth and Thomas (1976). It is based on Gram-Schmidt sequential orthogonalization techniques that produce an orthogonal transformation of the original four-channel MSS data space to a new four-dimensional space. It is called the tassled



OFEK06846

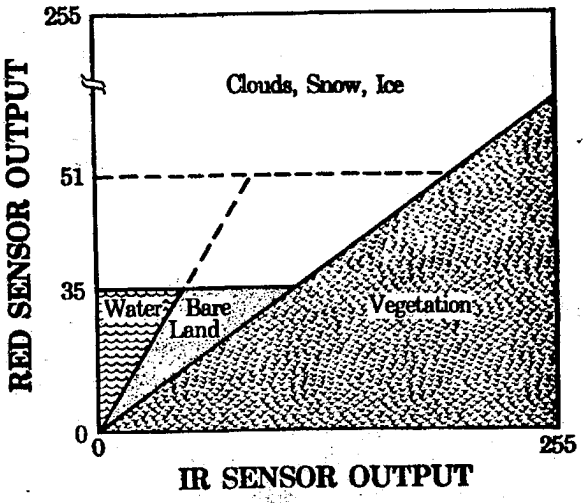



**Figure 7–33** Red and near-infrared band ratio data obtained by the Space Shuttle FILE sensor was used to partition the two-dimensional feature space into the land-cover types shown. (After Sivertson et al., 1982.)

cap transformation because of its cap shape (Figs. 7–34 to 7–36). It has been rigorously tested and is used extensively in agricultural research. The transformation identifies four new axes, including the soil brightness index (SBI), the green vegetation index (GVI), the yellow stuff index (YVI), and a non-such index (NSI) associated with atmospheric effects. Generally, the first two indices contain most of the scene information (95 to 98%). Kauth and Thomas found that nearly all (98%) of the variance in bare soil spectra from several different soil types could be explained by the soil brightness index (SBI). They concluded that bare soils would lie in a line parallel to the brightness axis and that a globally valid soil line exists that could be applied to Landsat MSS agricultural scenes. Greenness is an orthogonal deviation from the mean soil line and is used as a measure of the green vegetation present. The farther the distance perpendicular to the soil line, the greater the amount of vegetation present within the field of view of a pixel.

The coefficients in the following equations applied to Landsat 2 MSS data are from Kauth et al. (1979) and Thompson and Wehmanen (1980):

$$SBI = 0.332MSS4 + 0.603MSS5 + 0.675MSS6 + 0.262MSS7 \quad (7\text{-}63)$$

$$GVI = -0.283MSS4 - 0.660MSS5 + 0.577MSS6 + 0.388MSS7 \quad (7\text{-}64)$$

$$YVI = -0.899MSS4 + 0.428MSS5 + 0.076MSS6 - 0.041MSS7 \quad (7\text{-}65)$$

$$NSI = -0.016MSS4 + 0.131MSS5 - 0.452MSS6 + 0.882MSS7 \quad (7\text{-}66)$$

A greenness vegetation index (GVI) image is created by multiplying each of the four original Landsat MSS brightness values for each pixel by the corresponding tassled cap coefficient. Jackson (1983) describes the procedures for calculating the coefficients of $n$-space indices. Huete et al. (1984) and Ezra et al. (1984) provide additional insight into the importance of calculating the coefficients on a site-specific basis, if possible, rather than using global coefficients. Ezra et al. (1984) also review the environmental factors that should be considered whenever a vegetation index is applied. Crist and Kauth (1986) de-mystified the tassled cap transformation.



OFEK06847

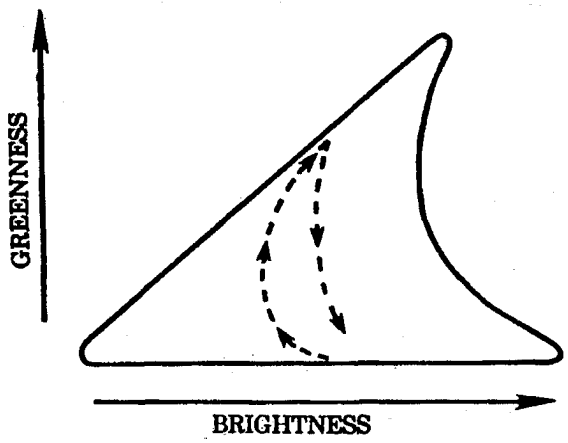



**Figure 7–34** Crop development in the Kauth–Thomas tassled cap brightness–greenness plane of vegetation. (From Crist, 1983.) © 1983 Purdue Research Foundation. Used with permission.

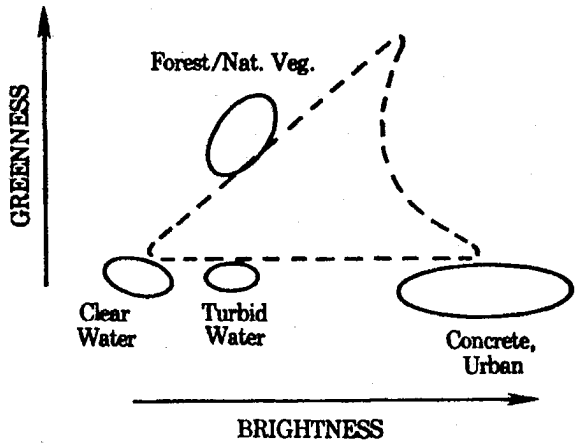



**Figure 7–35** Location of various types of land cover when plotted in the brightness–greenness spectral space. Brightness is highly correlated with bare soil while greenness is highly correlated with leaf area index, percent canopy closure, and/or biomass. (From Crist, 1983.) © 1983 Purdue Research Foundation. Used with permission.

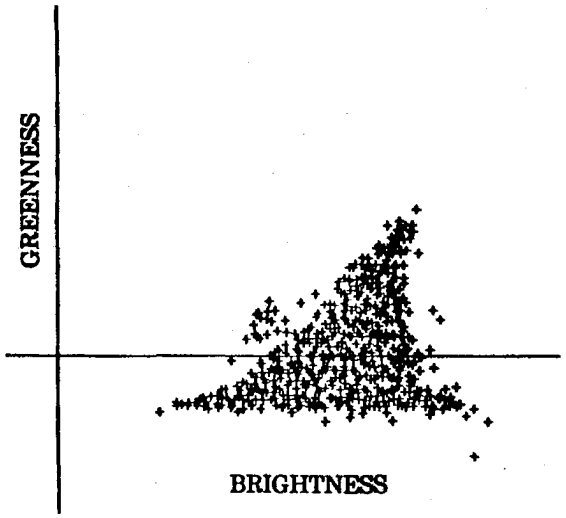



**Figure 7–36** Actual plot of brightness and greenness values for an agricultural area. Note that the shape of the distribution looks like a cap. (From Crist, 1983.) © 1983 Purdue Research Foundation. Used with permission.



OFEK06848

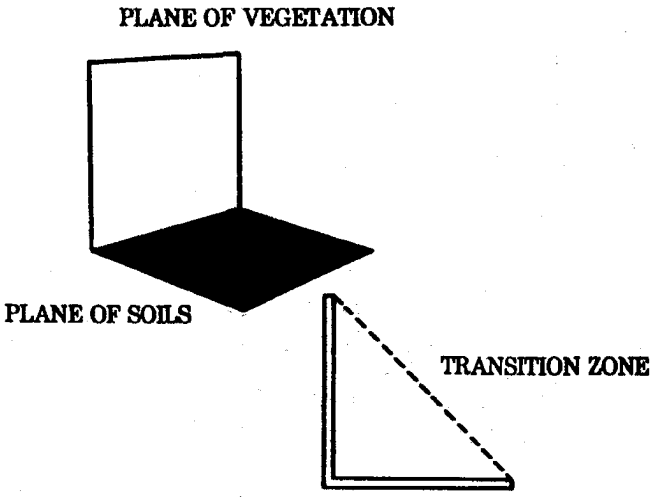



**Figure 7–37** Dispersion of six-band thematic mapper data based on the use of the Kauth–Thomas transformation. (From Crist 1983.) © 1983 Purdue Research Foundation. Used with permission.

Recent research has evaluated the Landsat 4 and 5 thematic mapper (TM) data to determine if it provides as useful vegetation index information as the MSS data. Crist and Cicone (1984) found that "the coefficients of the first two components are, for the TM bands which sample the same spectral regions as the MSS bands (TM bands 2, 3, and 4), comparable to those which define greenness and brightness." They found that the six-band TM data are dispersed into a three-dimensional space, and more precisely, define two perpendicular planes and a transition zone between

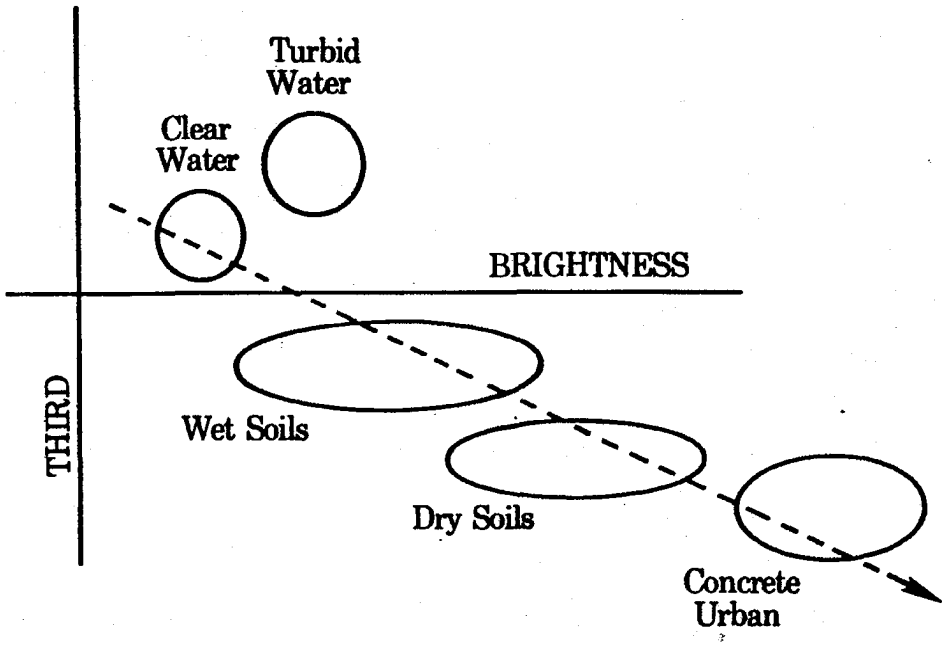



**Figure 7–38** Approximate direction of moisture variation in the plane of soils. The arrow points in the direction of less moisture. (From Crist, 1983.) © 1983 Purdue Research Foundation. Used with permission.



OFEK06849

the two (Fig. 7–37). Fully vegetated areas define the "plane of vegetation," while bare soils data fall in a "plane of soils." Between the two are data with partially vegetated plots where both vegetation and soil are visible. Thus, during a growing season, an agricultural field is expected to begin on the plane of soils, migrate through the transition zone, arrive at the plane of vegetation near the end of crop development, and then move back toward the plane of soils during harvest or senescence.

In addition, they identified a third component that is related to soil features, including moisture status (Fig. 7–38). Thus an important new source of soil information may now be available through the inclusion of the middle-infrared bands of the thematic mapper. For example, the Charleston TM scene is decomposed into brightness, greenness, and moisture content images in Fig. 7–39, based on the use of the

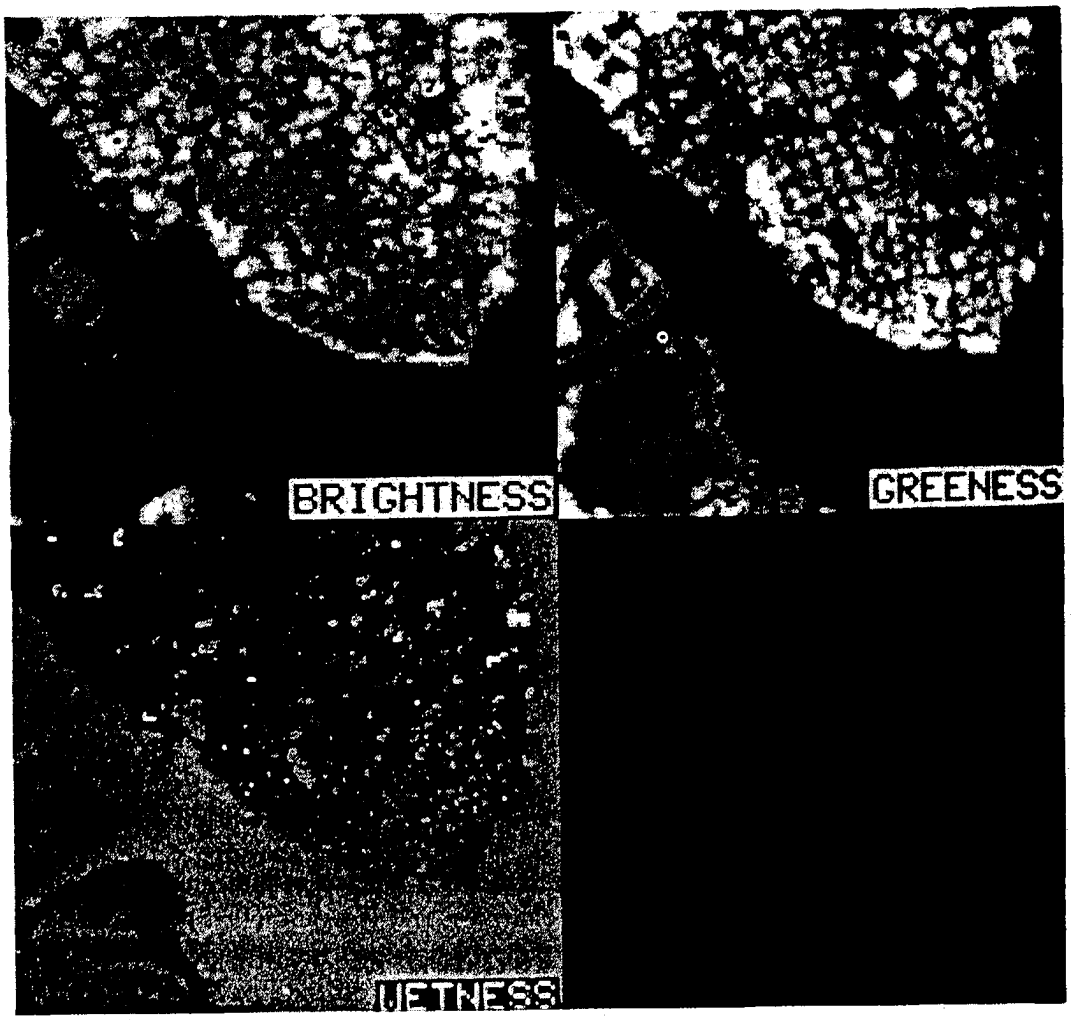



**Figure 7–39** Brightness, greenness, and wetness images derived by applying Kauth-Thomas tassled cap transformation coefficients to the Charleston, S.C., thematic mapper data (six bands). The coefficients used are summarized in Table 7–8.



OFEK06850

**TABLE 7–8** Coefficients for the Creation of Tassled Cap Features Using Thematic Mapper Data

| Feature | Thematic mapper band | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Brightness | 0.33183 | 0.33121 | 0.55177 | 0.42514 | 0.48087 | 0.25252 |
| Greenness | −0.24717 | −0.16263 | −0.40639 | 0.85468 | 0.05493 | −0.11749 |
| Third | 0.13929 | 0.22490 | 0.40359 | 0.25178 | −0.70133 | −0.45732 |

*Source:* Crist and Cicone (1984). © American Society of Photogrammetry and Remote Sensing. Used with permission.

TM tassled cap coefficients summarized in Table 7–8. The moisture component image provides subtle information concerning the moisture status of the wetland environment. As expected, the greater the moisture content, the brighter the response. A color composite using these three components enhances the separation between urban, water, and especially wetland classes [Fig. 7–40 (color section)].

*Principal components analysis* (described elsewhere in this chapter) can be applied to MSS data for vegetation applications. For example, Wheeler and Misra (1976) and Misra and Wheeler (1977) applied PCA to Landsat MSS data, producing results very similar to the Kauth–Thomas transformation:

$$\text{MSBI} = 0.406\text{MSS4} + 0.600\text{MSS5} + 0.645\text{MSS6} + 0.243\text{MSS7} \quad (7\text{-}67)$$

$$\text{MGVI} = -0.386\text{MSS4} - 0.530\text{MSS5} + 0.535\text{MSS6} + 0.532\text{MSS7} \quad (7\text{-}68)$$

$$\text{MYVI} = 0.723\text{MSS4} - 0.597\text{MSS5} + 0.206\text{MSS6} - 0.278\text{MSS7} \quad (7\text{-}69)$$

$$\text{MNSI} = 0.404\text{MSS4} - 0.039\text{MSS5} - 0.505\text{MSS6} + 0.762\text{MSS7} \quad (7\text{-}70)$$

The similarity of the Kauth–Thomas and Wheeler–Misra results was ironic in light of the fact that the logic and techniques underlying the two processes are different. Principal component analysis factors the total variation in the data into mutually orthogonal components. The order is established by the successive directions of maximum variation. With principal component analysis the analyst places *a priori order* on the principal directions. Conversely, [the Kauth–Thomas application of] Gram–Schmidt orthogonalization allows the analyst to freely establish a physical interpretation by choosing the order in which the calculations are performed (Perry and Lautenschlager, 1984).

The perpendicular distance to the "soil line" as an indicator of plant development (Fig. 7–41) was developed by Richardson and Wiegand (1977). Generally, the farther away from the soil line to the left, the greater the amount of vegetation present (Curran, 1983). Note that there exists a point on the line for both dry (A) and wet (B) soils. Points C and D are partially vegetated areas with relatively dry or wet soils, respectively. The soil line represents a two-dimensional variation of the Kauth–



OFEK06851

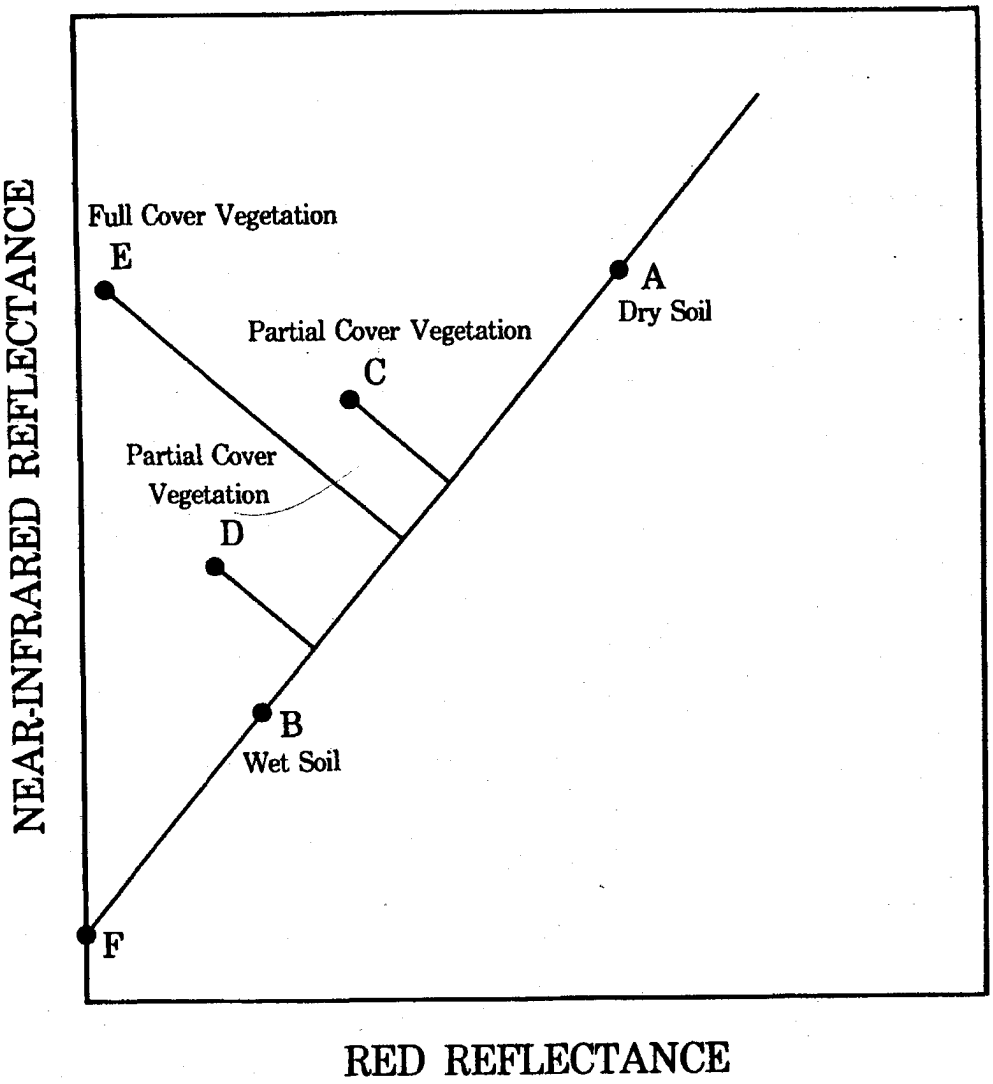



**Figure 7–41** Relationship of vegetation to soil background using a perpendicular vegetation index. (From Richardson and Wiegand, 1977.) © 1977 Elsevier Science Publishing Co. Inc. Used with permission.

Thomas brightness value index (SBI). Two *perpendicular vegetation indices* (PVIs) based on this logic were developed:

$$PVI7 = \sqrt{(0.355MSS7 - 0.149MSS5)^2 + (0.355MSS5 - 0.852MSS7)^2} \quad (7\text{-}71)$$

$$PVI6 = \sqrt{(-2.507 - 0.457MSS5 + 0.498MSS6)^2 + (2.734 + 0.498MSS5 - 0.543MSS6)^2} \quad (7\text{-}72)$$

Perry and Lautenschlager (1984) felt that the perpendicular vegetation indices described above were computationally inefficient and that they also did not distinguish



OFEK06852

right from left of the soil line (water from green vegetation). Therefore, they proposed perpendicular vegetation indices (PVIs) that took these factors into account:

$$PVI6 = \frac{1.091MSS6 - MSS5 - 5.49}{\sqrt{1.091^2 + 1}} \quad (7\text{-}73)$$

$$PVI7 = \frac{2.4MSS7 - MSS5 - 0.01}{\sqrt{2.4^2 + 1}} \quad (7\text{-}74)$$

Another vegetation index called *greenness above bare soil* (GRABS) was tested by Hay et al. (1979). It was computed using the Kauth–Thomas transformation on Landsat MSS data that had been corrected for sun-angle effects and atmospheric haze:

$$GRABS = GVI - (0.09178SBI) + 5.58959 \quad (7\text{-}75)$$

Finally, Kanemasu et al. (1977) regressed wheat leaf area measurements with MSS band ratios, resulting in the *ELAI index*:

$$\begin{aligned} ELAI = {} & 2.68 - 3.69R45 - 2.31R46 + 2.88R47 \\ & + 0.43R56 - 1.35R57 + 3.07\,[R45 - (0.5R47)(R45)] \end{aligned} \quad (7\text{-}76)$$

where R45 stands for a ratio of MSS bands 4 and 5. Work on a larger data set resulted in another regression equation (Pollock and Kanemasu, 1979):

$$\begin{aligned} CLAI = {} & 0.366 - 2.265R46 - 0.431(R45 - R47)(R45) \\ & + 1.745R45 + 0.57PV17 \end{aligned} \quad (7\text{-}77)$$

Perry and Lautenschlager (1984) used cluster analysis to study the empirical relationships among the vegetation indices described above and others. Clustering separated the VIs into two large groups and a few smaller ones. One large group contained VIs based on MSS5 and MSS7, including AVI, PVI7, TVI7, R75, and ND7. The other large group contained VIs involving MSS5 and MSS6 and a few VIs involving three or all four MSS channels, including GRABS, CLAI, R65, TVI6, ND6, GVI, MGVI, PVI6, and SGVI. The VIs within these two groups had absolute simple correlations greater than 0.90, with most greater than 0.95. The elements of these two large groups were correlated at 0.8 or higher, suggesting that redundancy was rampant. Three small groups, however, were off by themselves, suggesting that they have unique information not contributed by the other indices. The three groups were composed of the following indices: (NSI,R76), (R64,R74), and (SBI, MSBI, and SNSI).

This condition encouraged Perry and Lautenschlager (1984) to further evaluate several indices to determine their "equivalence" using alarm models and graphic display techniques. Their results demonstrated that for practical purposes several of the most widely used vegetation indices are completely equivalent. For example, all transformed vegetation indices were found to be equivalent to their corresponding



OFEK06853

band ratios. Therefore, redundant information may be obtained when two or more of the vegetation indices described above are applied to the same data set.

## Texture Transformations

When human beings visually interpret remotely sensed imagery, they synergistically take into account the following characteristics of the data: (1) context, (2) edges, (3) texture, and (4) tonal variation or color. Conversely, most digital image processing classification algorithms are based only on the use of the spectral (tonal) information (i.e., brightness values). Thus, it is not surprising that there has been considerable activity in trying to incorporate some of these other characteristics into the digital classification procedure.

Haralick (1973) defines a *discrete tonal feature* as being a connected set of pixels all of which have the same or almost the same gray shade (brightness value). When a small area of the image (e.g., a 3 × 3 area) has little variation of discrete tonal features, the dominant property of that area is gray shade. Conversely, when a small area has a wide variation of discrete tonal features, the dominant property of that area is *texture*. Most researchers trying to incorporate texture into the classification process have attempted to create a new texture image which can then be used as another "feature" or band in the classification. Thus, each new pixel of the texture image has a brightness value that represents the texture at that location (i.e., $BV_{i,j,\text{texture}}$).

In a comparative study of texture measures for level I terrain classification of remote sensor data, Weszka et al. (1976) identified three standard approaches to automatic texture classification. These approaches include features based on first- and second-order gray-level statistics and on the Fourier power spectrum.

**First-order statistics in the spatial domain.** One class of picture properties that can be used for texture synthesis is first-order statistics of local areas (i.e., the means, variances, and standard deviations). For example, Mauer (1974) used standard deviation as one of the texture features in the classification of crop types in Switzerland. The analysis was based on large extended blocks (50 × 50 pixels) of digitized 1:7,800 aerial photography. Such blocks also allowed the use of three other first-order measures, including average distance between pixels belonging to the same density class, average run length along a scan line of pixels of similar value, and a measure of the probability that a pixel has a neighboring pixel with a value in a particular density class. With the exception of run length, these texture parameters were useful for discriminating crop types.

Hsu (1978) proposed 17 first-order texture measures to classify level I land cover from digitized aerial photography. The study concluded that gray-level run-length statistics (called *waveform statistics*) were superior, although all texture measures except kurtosis were statistically significant. Based on these results, Irons and Petersen (1981) applied 11 of Hsu's local texture transforms to Landsat MSS data



OFEK06854

using a 3 × 3 moving window (mean, variance, skewness, kurtosis, range, Pearson's second coefficient of skewness, absolute value of mean norm length differences, mean of squared norm length differences, maximum of squared norm length differences, mean Euclidian distance, and maximum Euclidean distance). Unfortunately, none of these texture measures used in either supervised or unsupervised classifications (discussed in Chapter 8) resulted in the derivation of a texture transform class that corresponded to a land-cover category. The data sets and methods in these two studies were not identical; therefore, care should be exercised when comparing them.

A min-max texture operator based on an analysis of the brightness values found within the following five-element spatial moving window has been proposed:

$$\begin{array}{ccc} & A & \\ B & C & D \\ & E & \end{array}$$

where

$$\text{texture at pixel } C = \begin{pmatrix}\text{intensity of brightest}\\ \text{pixel at}\\ A,B,C,D,E\end{pmatrix} - \begin{pmatrix}\text{intensity of darkest}\\ \text{pixel at}\\ A,B,C,D,E\end{pmatrix} \tag{7-78}$$

Shih and Showengerdt (1983) devised another min-max algorithm to include information on the local edge amplitude and density (referred to as the LEAD texture feature). The main difference between their algorithm and that of Mitchell et al. (1977) is that all local pixel deviations within a specified threshold range were measured, not just local extrema. Their algorithm is based on the following three steps: (1) convolution of Landsat MSS band 5 data using a 3 × 3 high-pass spatial filter; (2) two absolute value thresholds are applied to the high-pass image, which has been histogram equalized so that each pixel is classified into one of three mutually exclusive threshold ranges, $T_1$, $T_2$, and $T_3$; and (3) a count image is produced by convolving the thresholded high-pass image with a window that counts the number of pixels in each threshold range within the window area. The number of pixels within each range is a measure of a specific amplitude ($T_1$, $T_2$, and $T_3$ range limits) and density (number of points within each range) of image activity, that is, edges, lines, points, and other elements of texture within the neighborhood defined by the window. Their window sizes were 5 × 5, 9 × 9, and 11 × 11. They reported that the inclusion of textural features significantly improved the overall classification accuracy of geologic/geomorphic classes within the training sites from an average of 73% in the "spectral only" classification to 96% in the "spectral plus textural" classification.

**Second-order brightness value statistics in the spatial domain.** A higher-order set of texture measures was proposed by Haralick and Shanmugam (1974) and Haralick (1979) based on brightness value spatial-dependency matrices. If c =



OFEK06855

$(\Delta x, \Delta y)$ is considered a vector in the $(x,y)$ image plane, for any such vector and for any image $f(x,y)$ it is possible to compute the joint probability density of the pairs of brightness values that occur at pairs of points separated by c. If the brightness values in the image can take upon themselves any value from 0 to the brightness value range, $BV_r$ (e.g., 0 to 255), this joint density takes the form of an array $h_c$, where $h_c(i,j)$ is the probability that the pairs of brightness values $(i,j)$ occur at separation c. This array, $h_c$, is $BV_r$ by $BV_r$ in size. It is easy to compute the $h_c$ array for $f(x,y)$, where $\Delta x$ and $\Delta y$ are integers, by simply counting the number of times each pair of brightness values occurs at separation c($\Delta x$ and $\Delta y$) in the image. For example, consider the following image, which has just five lines and five columns and contains brightness values ranging from only 0 to 3:

$$\text{original image} = \begin{matrix} 0 & 1 & 1 & 2 & 3 \\ 0 & 0 & 2 & 3 & 3 \\ 0 & 1 & 2 & 2 & 3 \\ 1 & 2 & 3 & 2 & 2 \\ 2 & 2 & 3 & 3 & 2 \end{matrix}$$

If ($\Delta x$ and $\Delta y$) = (1, 0), these numbers are represented by the brightness value spatial-dependency matrix $h_c$:

$$h_c = \begin{matrix} \\ 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{matrix} \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{bmatrix} 1 & 2 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 3 & 5 \\ 0 & 0 & 2 & 2 \end{bmatrix} \end{matrix}$$

where the entry in row $i$ and column $j$ of this matrix is the number of times brightness value $i$ occurs to the left of brightness value $j$ (Weszka et al., 1976). For example, brightness value 1 is adjacent to brightness value 2 a total of three times in this simplified image [i.e., $h_c(1, 2) = 3$]. It is assumed that all textural information is contained in the brightness value spatial-dependency matrices that are developed for angles of 0°, 45°, 90°, and 135° (Fig. 7–42). Generally, the greater the number found in the diagonal of the spatial dependency matrices, the more homogeneous the texture is for that part of the image being analyzed.

Haralick proposed a variety of measures to extract useful textural information from the $h_c$ matrices. In an empirical test of these measures with Landsat MSS data, Haralick and Shanmugan (1974) used extended blocks (64 × 64) of imagery quantized to just 16 levels. When four textures features and eight spectral Landsat MSS features (bands) were used, a 7% increase in level I land-cover classification accuracy was reported. Jensen (1979) and Jensen and Toll (1982) report on the use of Haralick's angular second moment (ASM) for use as an additional feature in the supervised classification of remotely sensed data obtained at the urban fringe and in urban change-detection mapping. The ASM is computed according to the formula



OFEK06856

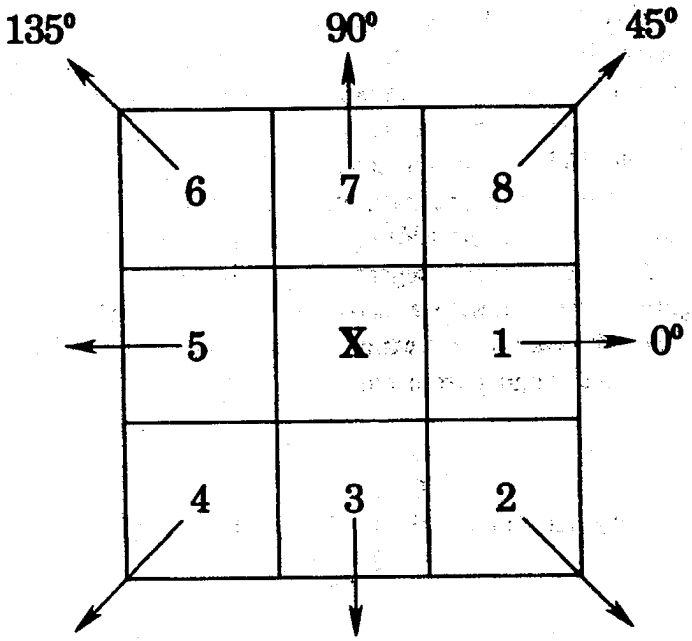



Figure 7–42 Eight nearest neighbors of pixel $X$ according to angle $\phi$ used in the creation of spatial dependency matrices for the measurement of image texture. (After Haralick and Shanmugam, 1973.)

$$\text{ASM} = \sum_{i=1}^{BV_r} \sum_{j=1}^{BV_r} \frac{h_c(i,j)^2}{R} \tag{7-79}$$

where $BV_r$ is the range of possible brightness values, $R$ the number of pixel pairs, and $h_c(i,j)$ the $(i,j)$th entry in one of the angular brightness value spatial-dependency matrices discussed previously. During computation, four brightness value spatial-dependency matrices (0, 45, 90, and 135) are derived for each pixel based on neighboring pixel values. The average of these four measures is normally output as the texture value for the pixel under consideration.

Sun and Wee (1983) built on the idea of the neighboring gray-level dependency matrix. They developed a new texture transformation (called NGLDM) that is (1) easily computed, (2) essentially invariant under spatial rotation (i.e., the same texture value will result even if the brightness values are rotated), and (3) invariant even when the brightness value range is transformed.

**Fourier power spectrum texture measures.** It has been shown that texture features developed in the Fourier domain are generally inferior to statistically based texture measures for use in terrain classification (Weszka et al., 1976).

Although texture features have been increasingly incorporated into multispectral classifications, no single algorithm combining efficiency and effectiveness has yet to be widely adopted. Since many of the features are based on the convolution of the original spectral data with a window of pixels, spatial detail is often lost at the edges of important phenomena (Townshend, 1981). This can be of significance especially in the urban and suburban environment. Also, the texture features derived for one type of application (e.g., land-use classification at the urban fringe) are not necessarily useful when applied to another geographic problem (e.g., identification of selected



OFEK06857

geomorphic classes). Finally, some parameters central to the computation of the texture features are still derived empirically (e.g., the size of the window or the location of certain "thresholds"). This makes it difficult to compare and contrast studies when so many variables in the creation of the texture features are not held constant.

## REFERENCES


ANUTA, P. E. 1977. "Computer-Assisted Analysis Techniques for Remote Sensing Data Interpretation," *Geophysics*, Vol. 42, pp. 468–481.

ASHBURN, P. 1978. "The Vegetative Index Number and Crop Identification," *Proceedings*, Technical Session, LACIE Symposium, Houston, Tex.: National Aeronautics and Space Administration, pp. 843–856.

BERNSTEIN, R. 1983. "Image Geometry and Rectification," Chapter 21 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: The American Society of Photogrammetry and Remote Sensing, Vol. 1, pp. 887–888.

BYRNE, G. F., P. F. CRAPPER, and K. K. MAYO. 1980. "Monitoring Land-Cover Change by Principal Component Analysis of Multitemporal Landsat Data," *Remote Sensing of Environment*, Vol. 10, pp. 175–184.

CHAVEZ, P. C., and B. BAUER. 1982. "An Automatic Optimum Kernel-Size Selection Technique for Edge Enhancement," *Remote Sensing of Environment*, Vol. 12, pp. 23–38.

CHAVEZ, P. C., S. C. GUPTILL, and J. A. BOWELL. 1984. "Image Processing Techniques for Thematic Mapper Data," *Technical Papers*, 50th Annual Meeting of the American Society of Photogrammetry, Vol. 2, pp. 728–743.

CHIN, R. T., and C. YEH. 1983. "Quantitative Evaluation of Some Edge-Preserving Noise-Smoothing Techniques," *Computer Vision, Graphics, and Image Processing*, Vol. 23, pp. 67–91.

CHITTINENI, C. B. 1983. "Edge and Line Detection in Multidimensional Noisy Imagery Data," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. GE-21, pp. 163–174.

COOLEY, W. W., and P. R. LOHNES. 1971. *Multivariate Data Analysis*. New York: John Wiley & Sons, Inc.

CRIST, E. P. 1983. "The Thematic Mapper Tassled Cap . . . A Preliminary Formulation," *Proceedings*, Machine Processing of Remotely Sensed Data, West Lafayette, Ind.: Laboratory for the Applications of Remote Sensing, pp. 357–363.

CRIST, E. P. 1984. "Comparison of Coincident Landsat-4 MSS and TM Data over an Agricultural Region," *Technical Papers*, 50th Annual Meeting of the American Society of Photogrammetry, Vol. 2, pp. 508–517.

CRIST, E. P., and R. C. CICONE. 1984. "Application of the Tasseled Cap Concept to Simulated Thematic Mapper Data," *Photogrammetric Engineering and Remote Sensing*, Vol. 50, pp. 343–352.

CRIST, E. P. and R. J. KAUTH. 1986. "The Tassled Cap De-Mystified," *Photogrammetric Engineering and Remote Sensing*, Vol. 52, 81–86.

CURRAN, P. J. 1980. "Multispectral Photographic Remote Sensing of Green Vegetation Biomass and Productivity," *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 243–250.

CURRAN, P. J. 1983. "Estimating Green LAI from Multispectral Aerial Photography," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 1709–1720.




OFEK06858

DAVIS, L. S. 1975. "A Survey of Edge Detection Techniques," *Computer Graphics and Image Processing*, Vol. 4, pp. 248–270.

DEERING, D. W., J. W. ROUSE, R. H. HAAS, and J. A. SCHELL. 1975. "Measuring Forage Production of Grazing Units from Landsat MSS Data," *Proceedings*, 10th International Symposium on Remote Sensing of Environment, Vol. 2, pp. 1169–1178.

DUDA, R. O., and P. E. HART. 1973. *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons, Inc.

EBERLEIN R. B., and J. S. WESZKA. 1975. "Mixtures of Derivative Operators as Edge Detectors," *Computer Graphics and Image Processing*, Vol. 4, pp. 180–183.

EZRA, C. E., L. R. TINNEY, and R. D. JACKSON. 1984. "Considerations for Implementing Vegetation Indices for Agricultural Applications," *Technical Papers*, 50th Annual Meeting of the American Society of Photogrammetry, pp. 526–536.

FORD, G. E., V. R. ALGAZI, and D. I. MEYER. 1983. "A Noninteractive Procedure for Land-Use Determination," *Remote Sensing of Environment*, Vol. 13, pp. 1–16.

FREI, W. 1977. "Image Enhancement by Histogram Hyperbolization," *Computer Graphics and Image Processing*, Vol. 6, pp. 286–294.

FRIEDMAN, S. Z. 1980. *Mapping Urbanized Area Expansion through Digital Image Processing of Landsat and Conventional Data*, Pasadena, Calif.: Jet Propulsion Laboratory Publication 79–113.

GIL, B., A. MITICHE, and J. K. AGGARWAL. 1983. "Experiments in Combining Intensity and Range Edge Maps," *Computer Vision, Graphics, and Image Processing*, Vol. 21, pp. 395–411.

GONZALEZ, R. C., and P. WINTZ. 1977. *Digital Image Processing*, Reading, Mass.: Addison-Wesley Publishing Company, Inc., 123–126.

GUNTHER, F. J. 1982. "A New Principal Components Procedure to Aid the Analysis of Landsat MSS Digital Data," *Proceedings*, Conference on Pattern Recognition and Image Processing, Las Vegas, Nev.: IEEE Computer Society.

HARALICK, R. M. 1973. "Glossary and Index to Remotely Sensed Image Pattern Recognition Concepts," *Pattern Recognition*, Vol. 5, pp. 391–403.

HARALICK, R. M. 1979. "Statistical and Structural Approaches to Texture," *Proceedings of the IEEE*, Vol. 67, pp. 786–804.

HARALICK, R. M., and K. FU. 1983. "Pattern Recognition and Classification," Chapter 18 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: The American Society of Photogrammetry and Remote Sensing, Vol. 1, pp. 793–805.

HARALICK, R. M., and K. S. SHANMUGAM. 1974. "Combined Spectral and Spatial Processing of ERTS Imagery Data," *Remote Sensing of Environment*, Vol. 3, pp. 3–13.

HAY, C. M., C. A. KURETZ, J. B. ODENWELLER, E. J. SCHEFFNER, and B. WOOD. 1979. *Development of AI Procedures for Dealing with the Effects of Episodal Events on Crop Temporal Spectral Response*, AGRISTARS Report SR-B9-00434.

HORD, R. M. 1982. *Digital Image Processing of Remotely Sensed Data*. New York: Academic Press, Inc.

HSU, S. 1978. "Texture-Tone Analysis for Automated Landuse Mapping," *Photogrammetric Engineering and Remote Sensing*, Vol. 44, pp. 1393–1404.

HUETE, A. R., D. F. POST and R. D. JACKSON. 1984. "Soil Spectral Effects on 4-Space Vegetation Discrimination," *Remote Sensing of Environment*, Vol. 14, pp. 155–165.

IRONS, J. R., and G. W. PETERSEN. 1981. "Texture Transforms of Remote Sensing Data," *Remote Sensing of Environment*, Vol. 11, pp. 359–370.

JACKSON, R. D. 1983. "Spectral Indices in N-Space," *Remote Sensing of Environment*, Vol. 13, pp. 409–421.



OFEK06859

JENSEN, J. R. 1979. "Spectral and Textural Features to Classify Elusive Land Cover at the Urban Fringe," *The Professional Geographer*, Vol. 31, pp. 400–409.

JENSEN, J. R. 1983. "Biophysical Remote Sensing," *Annals*, Association of American Geographers, Vol. 73, pp. 111–132.

JENSEN, J. R., and D. L. TOLL. 1982. "Detecting Residential Land-Use Development at the Urban Fringe," *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 629–643.

JENSEN, J. R., P. J. PACE, and E. J. CHRISTENSEN. 1983. "Remote Sensing Temperature Mapping: The Thermal Plume Example," *The American Cartographer*, Vol. 10, pp. 111–127.

KANEKO, T. 1978. "Color Composite Pictures from Principal Axis Components of Multispectral Scanner Data," *IBM Journal of Research and Development*, Vol. 22, pp. 386–392.

KANEMASU, E. T., J. L. HEILMAN, J. O. BAGLEY, and W. L. POWERS. 1977. "Using Landsat Data to Estimate Evapotranspiration of Winter Wheat," *Environmental Management*, Vol. 1, pp. 515–520.

KAUTH, R. J., and G. S. THOMAS. 1976. "The Tasseled Cap—A Graphic Description of the Spectral–Temporal Development of Agricultural Crops As Seen by Landsat," *Proceedings*, Symposium on Machine Processing of Remotely Sensed Data. West Lafayette, Ind.: Laboratory for the Applications of Remote Sensing, pp. 41–51.

KAUTH, R. J., P. F. LAMBECK, W. RICHARDSON, G. S. THOMAS, and A. P. PENTLAND. 1979. "Feature Extraction Applied to Agricultural Crops As Seen by Landsat," *Proceedings*, Technical Session, LACIE Symposium. Houston, Tex.: National Aeronautics and Space Administration, pp. 705–721.

LULLA, K. 1983. "The Landsat Satellites and Selected Aspects of Physical Geography," *Progress in Physical Geography*, Vol. 7, pp. 1–45.

MAUER, H. 1974. "Quantification of Textures: Textural Parameters and Their Significance for Classifying Agricultural Crop Types from Colour Aerial Photographs," *Photogrammetria*, Vol. 30, pp. 21–40.

MISRA, P. N., and S. G. WHEELER. 1977. "Landsat Data from Agricultural Sites—Crop Signature Analysis," *Proceedings*, 11th International Symposium on Remote Sensing of the Environment.

MITCHELL, O. R., C. R. MYERS, and W. BOYNE. 1977. "A Min–Max Measure for Image Texture Analysis," *IEEE Transactions on Computers*, Vol. C-26, pp. 408–414.

PELI, T., and D. MALAH. 1982. "A Study of Edge Detection Algorithms," *Computer Graphics and Image Processing*, Vol. 20, pp. 1–21.

PERRY, C. R., and L. F. LAUTENSCHLAGER. 1984. "Functional Equivalence of Spectral Vegetation Indices," *Remote Sensing of Environment*, Vol. 14, pp. 169–182.

POLLOCK, R. B., and E. T. KANEMASU. 1979. "Estimating Leaf-Area Index of Wheat with Landsat Data," *Remote Sensing of Environment*, Vol. 8, pp. 307–312.

PRATT, W. K. 1978. *Digital Image Processing*. New York: John Wiley & Sons, Inc., pp. 478–490.

RICHARDSON, A. J., and C. L. WEIGAND. 1977. "Distinguishing Vegetation from Soil Background Information," *Remote Sensing of Environment*, Vol. 8, pp. 307–312.

ROBINOVE, C. J. 1982. "Computation with Physical Values from Landsat Digital Data," *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 781–784.

ROBINSON, G. S. 1977. "Edge Detection by Compass Gradient Masks," *Computer Graphics and Image Processing*, Vol. 6, pp. 492–501.

ROSENFELD, A., and A. C. KAK. 1976. *Digital Picture Processing*. New York: Academic Press, Inc., Section 8.2.

ROUSE, J. W., R. H. HAAS, J. A. SCHELL, and D. W. DEERING. 1973. "Monitoring Vegetation



OFEK06860

Systems in the Great Plains with ERTS, *Proceedings*, Third ERTS Symposium, Vol. 1, pp. 48–62.

SATTERWHITE, M. B. 1984. "Discriminating Vegetation and Soils Using Landsat MSS and Thematic Mapper Bands and Band Ratios," *Technical Papers*, 50th Annual Meeting of the American Society of Photogrammetry, Vol. 2, pp. 479–485.

SHAW, G. B. 1979. "Local and Regional Edge Detectors: Some Comparisons," *Computer Graphics and Image Processing*, Vol. 9, pp. 135–149.

SHIH, E. H. H., and R. A. SCHOWENGERDT. 1983. "Classification of Arid Geomorphic Surfaces Using Landsat Spectral and Textural Features," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 337–347.

SHORT, N. 1982. "Principles of Computer Processing of Landsat Data," Appendix A in *The Landsat Tutorial Workbook*, NASA Technical Reference Publication 1078. Washington, D.C.: NASA, pp. 421–453.

SIVERTSON, W. E., R. G. WILSO, G. F. BULLOCK, and R. T. S. CHAPPELL. 1982. "Feature Identification and Location Experiment," *Science*, Vol. 218, pp. 1031–1033.

SPATIAL DATA INC. 1975. *Handbook on Image Processing*. Goleta, Calif.: Spatial Data, Inc.

SULLIVAN, G., L. TROTTIER, B. SAITTA, and C. IANNUZZI. 1984. "Multibus Image Processor Supports 512 × 512 Displays," *Electronic Imaging*, October, pp. 42–47.

SUN, C., and W. G. WEE. 1983. "Neighboring Gray Level Dependence Matrix for Texture Classification," *Computer Vision, Graphics, and Image Processing*, Vol. 23, pp. 341–352.

TARPLEY, D. 1983. "New Indexes Aid in Vegetation Monitoring," *Landsat Data Users Notes*, Vol. 26, pp. 2–3.

THOMPSON, D. R., and O. A. WEHMANEN. 1980. "Using Landsat Digital Data to Detect Moisture Stress in Corn–Soybean Growing Regions," *Photogrammetric Engineering and Remote Sensing*, Vol. 46, pp. 1082–1089.

TOWNSHEND, J. R. G. 1981. "Image Analysis and Interpretation for Land Resources Survey," Chapter 4 in *Terrain Analysis and Remote Sensing*, J. R. G. Townshend, Ed. London: George Allen & Unwin Ltd., pp. 59–108.

TUCKER, C. J. 1979. "Red and Photographic Infrared Linear Combinations for Monitoring Vegetation," *Remote Sensing of Environment*, Vol. 8, pp. 127–150.

VANDERBRUG, G. J. 1976. "Line Detection in Satellite Imagery," *IEEE Transactions on Geoscience Electronics*, Vol. GE-14, pp. 37–44.

WANG, D. C. C., A. H. VAGNUCCI, and C. C. LI. 1983. "Digital Image Enhancement: A Survey," *Computer Vision, Graphics, and Image Processing*, Vol. 24, pp. 363–381.

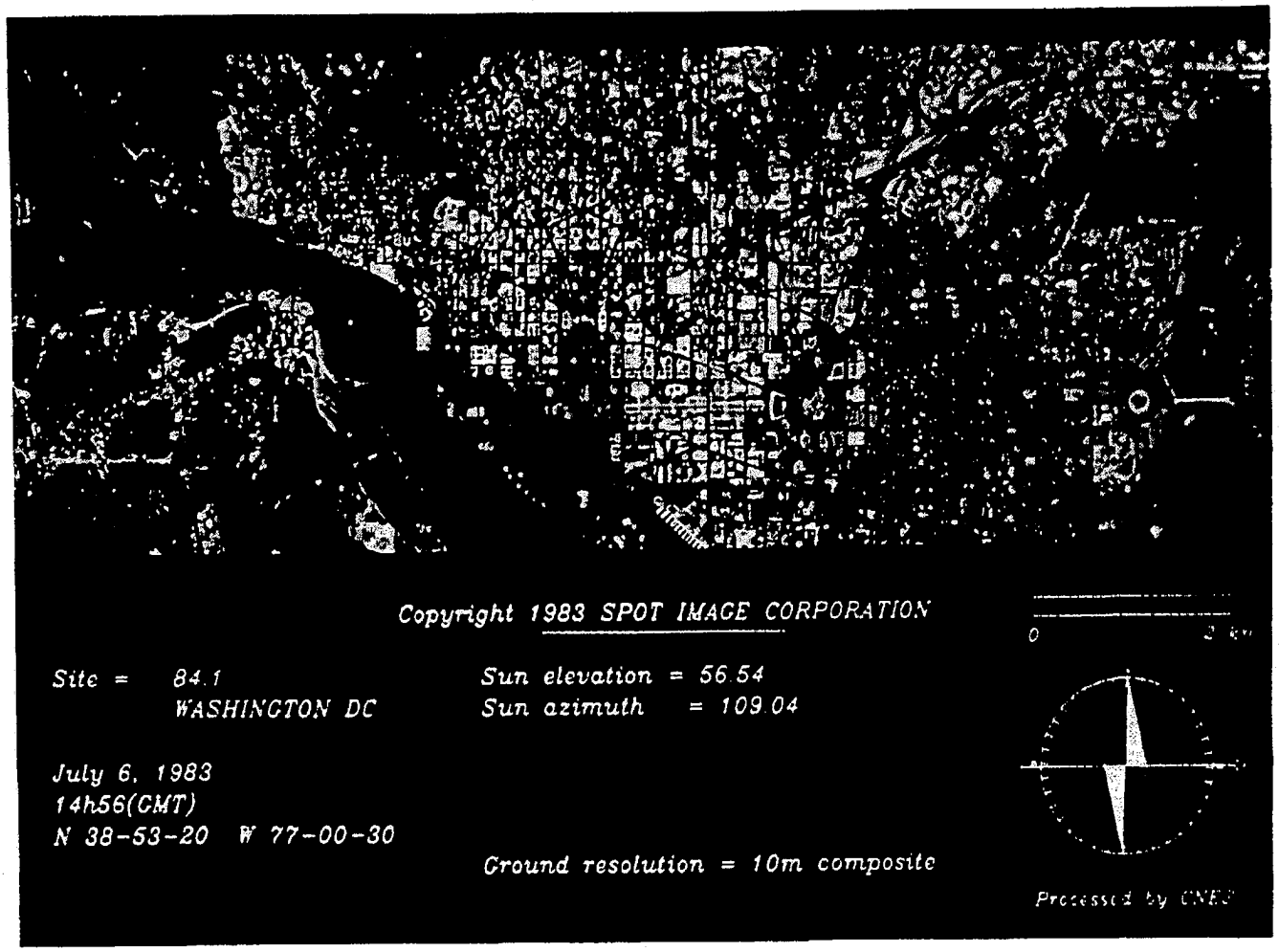
WESZKA, J., C. DYER, and A. ROSENFELD. 1976. "A Comparative Study of Texture Measures for Terrain Classification," *IEEE Transactions on System, Man and Cybernetics*, SMC-6, pp. 269–285.

WHEELER, S. G., and P. N. MISRA. 1976. "Linear Dimensionality of Landsat Agricultural Data with Implications for Classification," *Proceedings*, Symposium on Machine Processing of Remotely Sensed Data. West Lafayette, Ind.: Laboratory for the Applications of Remote Sensing.

WIEGAND, C. L., H. W. GAUSMANN, J. A. CUELLAR, A. H. GUBERMAN, and A. J. RICHARDSON. 1974. "Vegetation Density as Deduced from ERTS-1 MSS Response, *Proceedings*, Third ERTS-1 Symposium, Vol. 1.



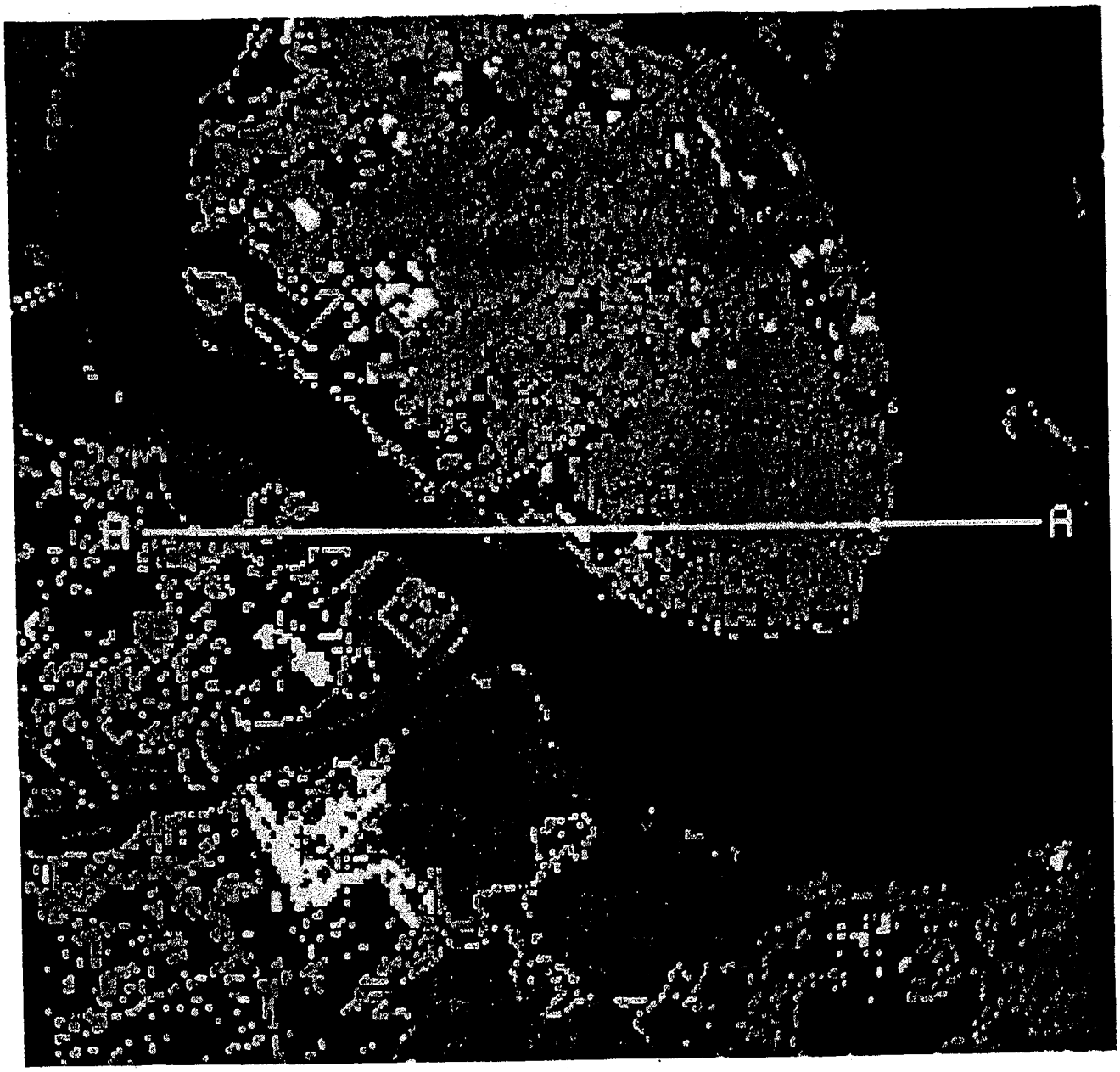
OFEK06861




**Color Fig. 2-25.** This is SPOT simulation data of the Washington, D.C. area obtained on 6 July 1983 using a Daedalus multi-spectral scanning system. [©1983 SPOT Image Corporation].



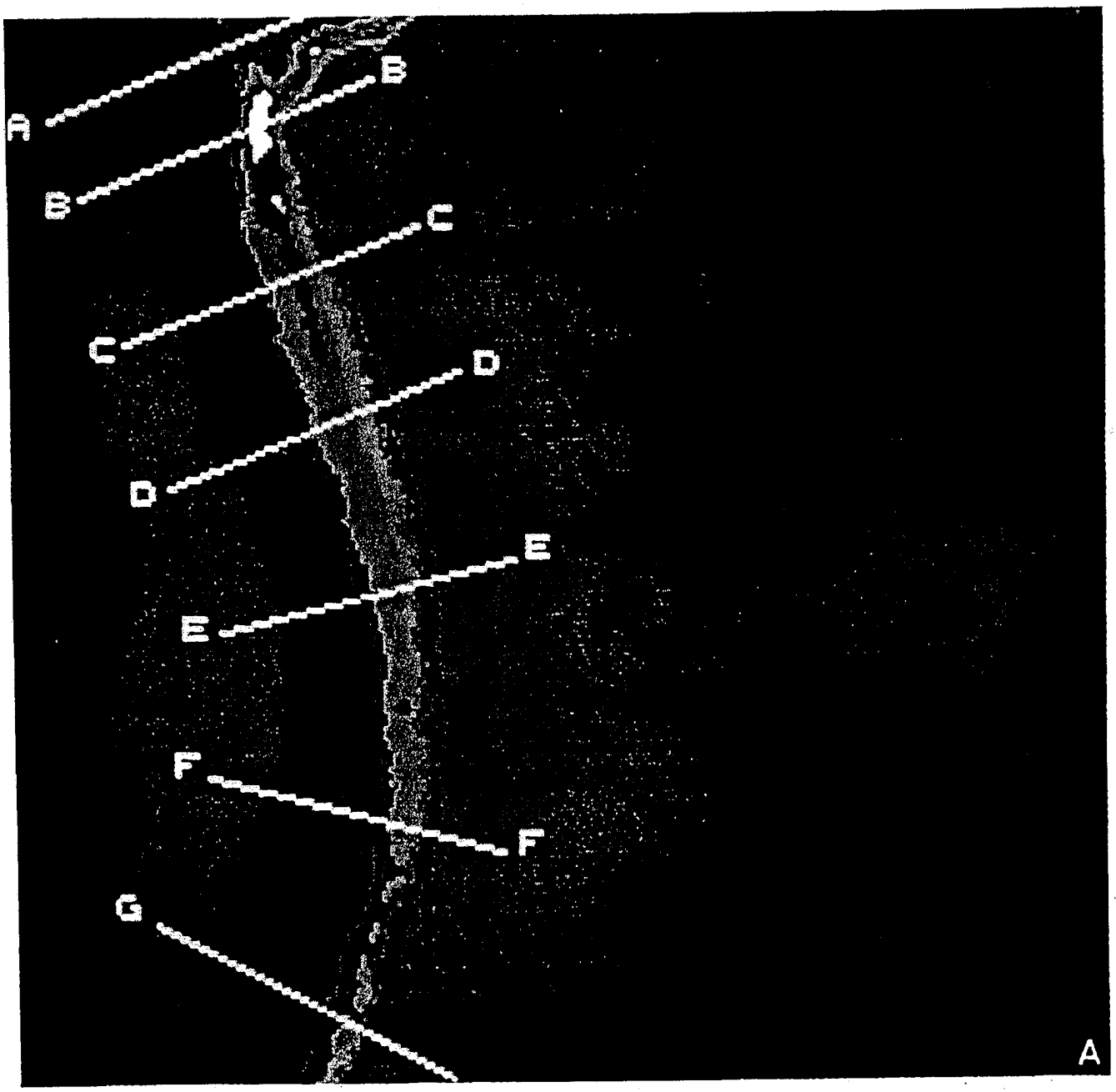
OFEK06862




**Color Fig. 5-12.** Color density slice of the Charleston, South Carolina Thematic Mapper band 4 image shown in Figure 5-10. The color codes and associated class intervals are summarized in Table 5-2. Transect A is discussed in Chapter 7. [Author].



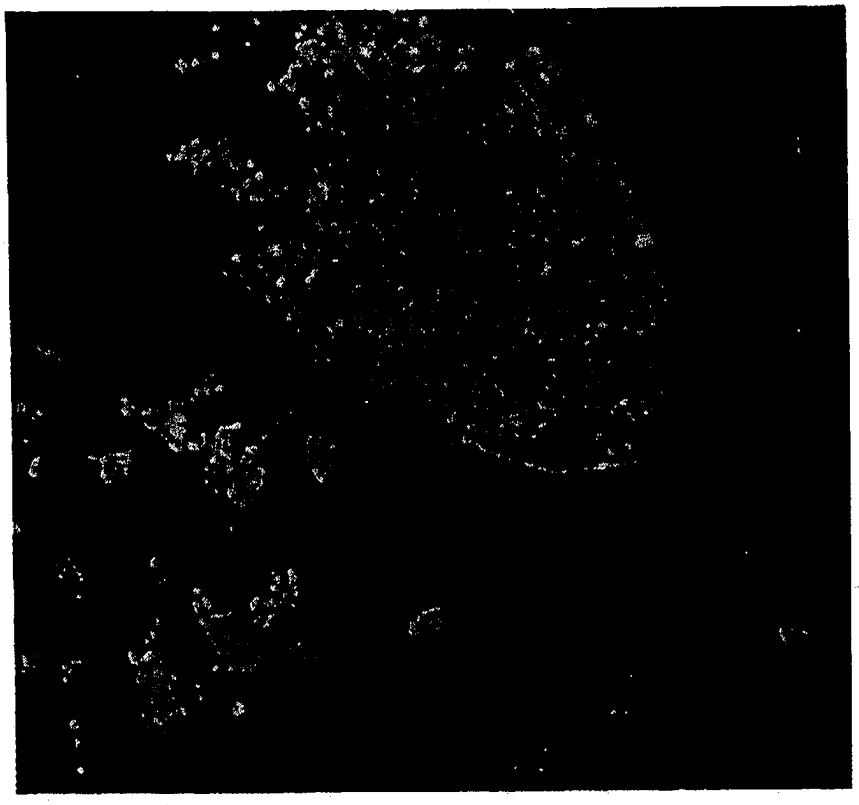
OFEK06863



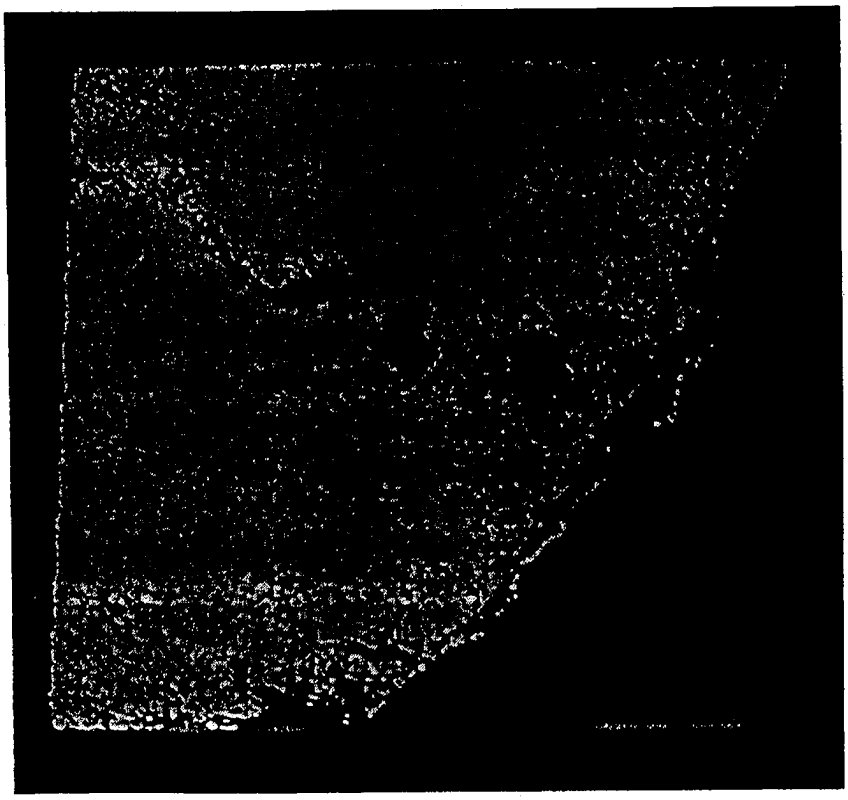

**Color Fig. 5-13.** Color density slice of the thermal plume in the Savannah River shown in Figure 5-11. The color codes and associated class intervals are summarized in Table 5-3. The transects are discussed in Chapter 7. [Jensen et al., 1983].



OFEK06864



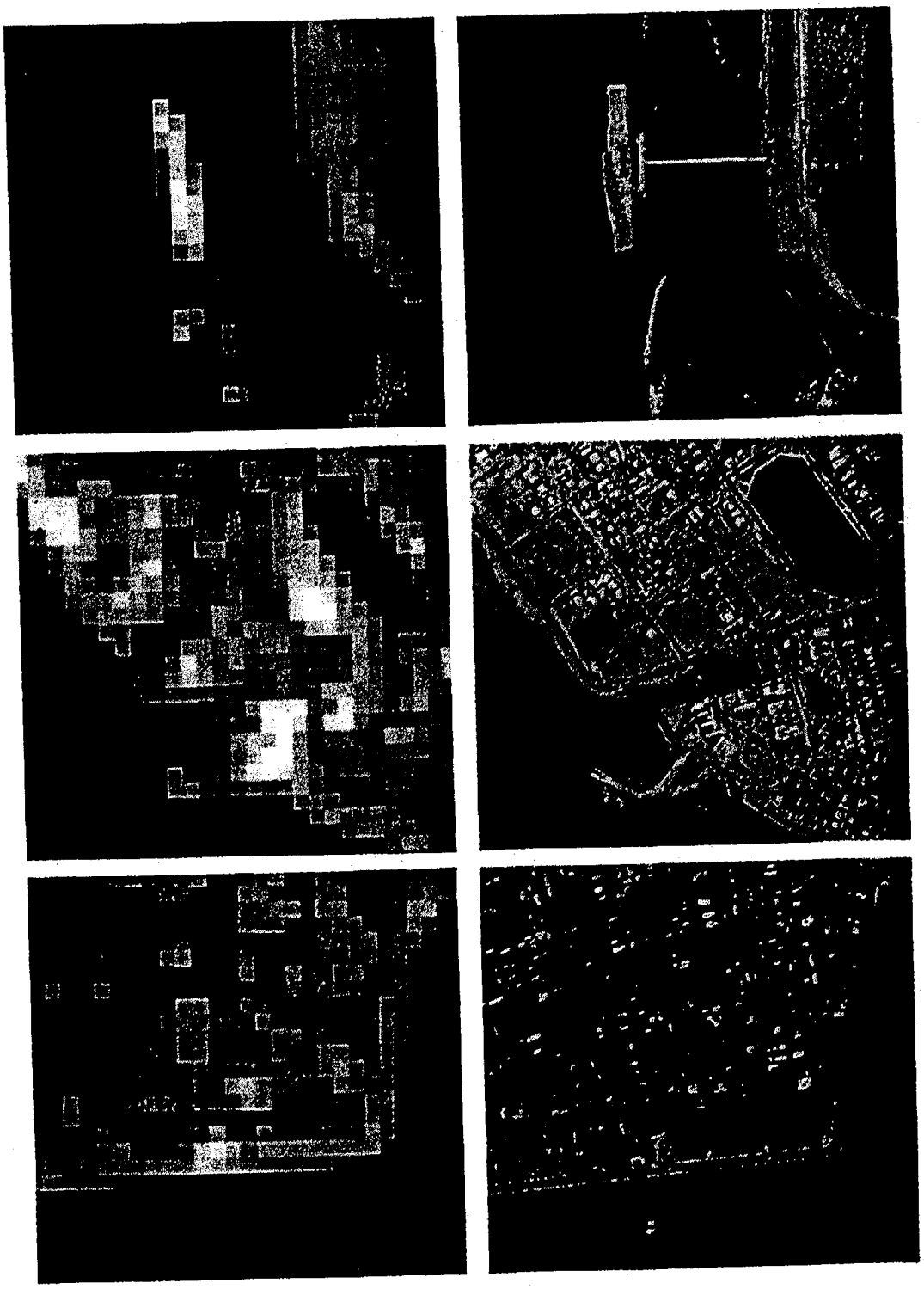
**Color Fig. 5-15.** A color infrared color composite of the Charleston, South Carolina Thematic Mapper image. TM bands 2, 3, and 4 were placed in the blue, green, and red banks of the image processor memory, respectively. [Author].



**Color Fig. 7-3.** A 198 row by 232 column color infrared color composite of bands 2, 3, and 4 of the entire Charleston, South Carolina, Thematic Mapper scene. Every 30th row and 30th column of the original 5940 rows and 6960 columns were sampled. The Charleston peninsula is the steel gray area in the lower center of the image. Note the turbidity off the South Carolina coast. [Author].



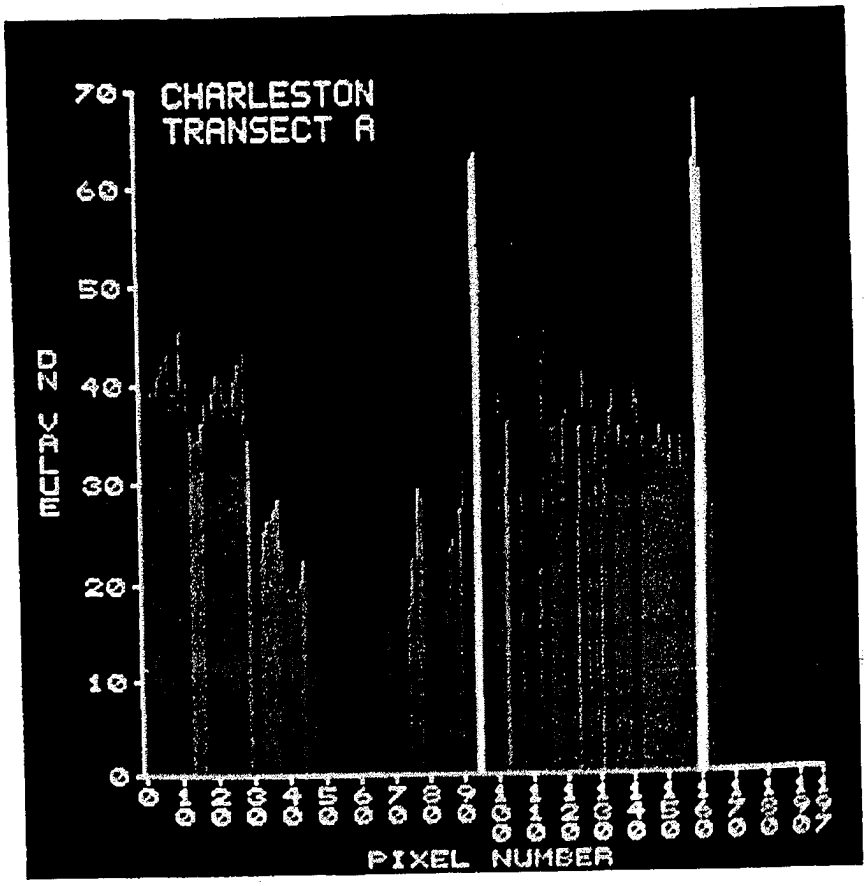
OFEK06865



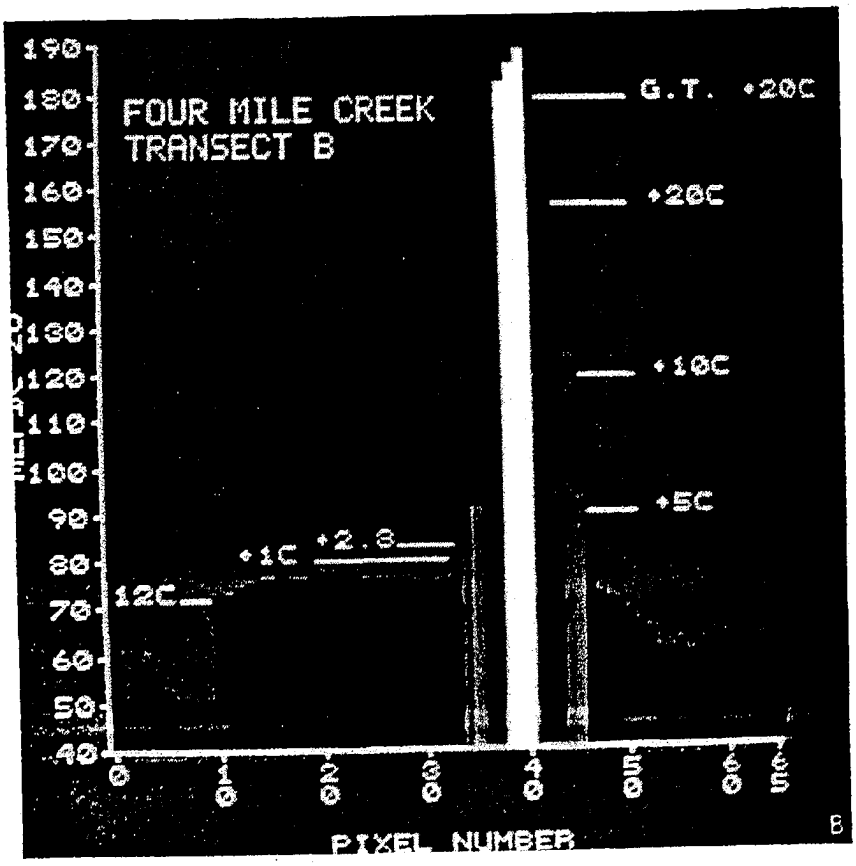
**Color Fig. 7-7.** Selected areas within the Charleston, South Carolina Thematic Mapper scene have been magnified and displayed in a color composite format using TM bands, 2, 3, and 4. [Author].



OFEK06866




**Color Fig. 7-8.** A color coded transect through the heart of the Charleston, South Carolina Thematic Mapper data. Refer to Color Figure 5-12 for the location of this transect in the density sliced band 4 TM image. The color codes and class intervals are summarized in Table 5-2. [Author].




**Color Fig. 7-9.** A color coded version of Transect B previously identified in Color Figure 5-13. The original thermal infrared image was calibrated to be within ±.2°C of the apparent temperature of objects. Therefore, once the transect was rotated so that the beginning and ending points of the transect fell on the same line in the image, statistical information could be extracted. Refer to Table 7-1 for a summary of the color codes and class intervals. [Jensen et al., 1983].



OFEK06867



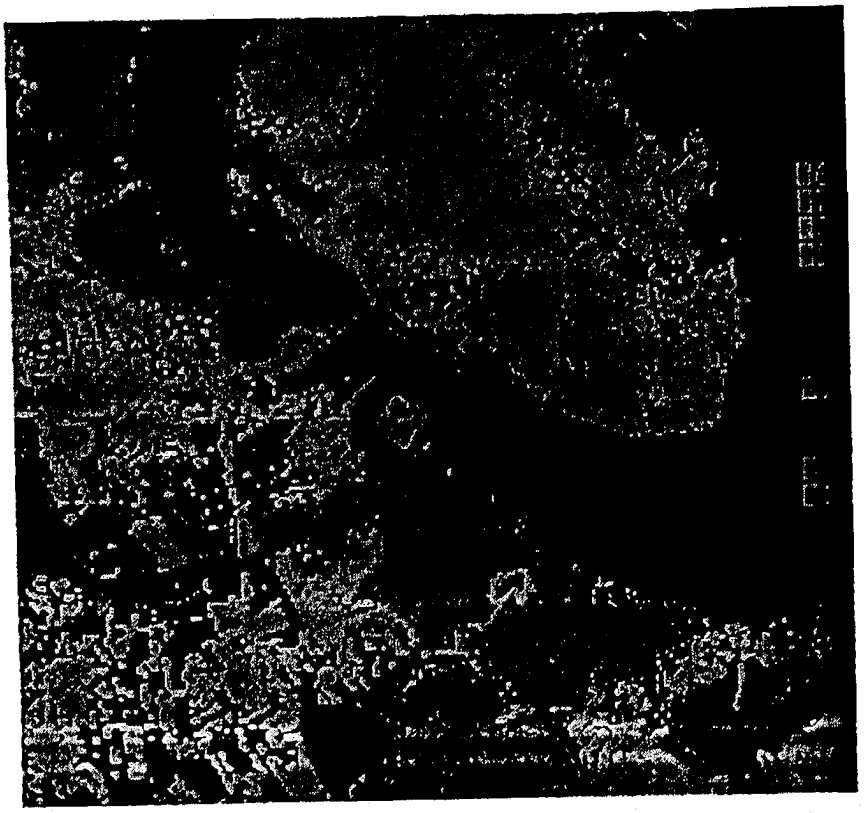
**Color Fig. 7-40.** A color composite produced by projecting the Brightness image in Figure 7-39 with red light, the Greeness image with green light, and the Wetness image with blue light. Note how the wetland areas show up in hues of blue and green which are entirely appropriate. Commercial and bare soil areas reflect visible and infrared light very similarly, therefore they both show up in shades of red. If desired, this three band vegetation index dataset could be used as the basis for a supervised or unsupervised classification. [Author].



OFEK06868



**Color Fig. 8-15.** The result of applying a **Minimum Distance to Means** classification algorithm to the training data summarized in Table 8-3. Bands 4 and 5 of the Thematic Mapper were used. The categories of this **Supervised Classification** are (1) Yellow = Residential land cover, (2) Red = Commercial, (3) Light Green = Wetland, (4) Dark Green = Forest, and (5) Blue = Water. See Table 8-7 and Table 8-8 for a summary of the number of pixels in each class and an evaluation of classification error. [Author].



**Color Fig. 8-21.** The result of performing an unsupervised classification of the Charleston, South Carolina Thematic Mapper image using bands 3, 4, and 5. Twenty clusters were extracted and relabeled according to the criteria shown in Figure 8-24 and Table 8-9. [Author].



OFEK06869

# 8

# Thematic Information Extraction

## INTRODUCTION

It is possible to analyze remotely sensed data of the earth and extract useful thematic information. Notice that data are transformed into information. One of the most often used methods of information extraction is *multispectral classification*. This procedure assumes that imagery of a specific geographic area is collected in multiple regions of the electromagnetic spectrum and that the images are in good registration (Fig. 8–1). For example, seven registered bands of thematic mapper data of a portion of Charleston, S.C., are displayed in Fig. 8–2. Most of the information extraction techniques rely on analysis of the spectral reflectance properties of such imagery and employ special algorithms designed to perform various types of "spectral analysis." The process of multispectral classification may be performed using either of two methods: supervised or unsupervised.

In a *supervised classification*, the identity and location of some of the land-cover types, such as urban, agriculture, wetland, and forest, are known a priori through a combination of field work, analysis of aerial photography, maps, and personal experience (Heaslip, 1975). The analyst attempts to locate specific sites in the remotely sensed data that represent homogeneous examples of these known land-cover types (Townshend, 1981). These areas are commonly referred to as *training sites* because the spectral characteristics of these known areas are used to "train" the classification algorithm for eventual land cover mapping of the remainder of the image. Multivariate statistical parameters (means, standard deviations, covariance matrices, correlation matrices, etc.) are calculated for each training site. Every pixel both within and outside





OFEK06870

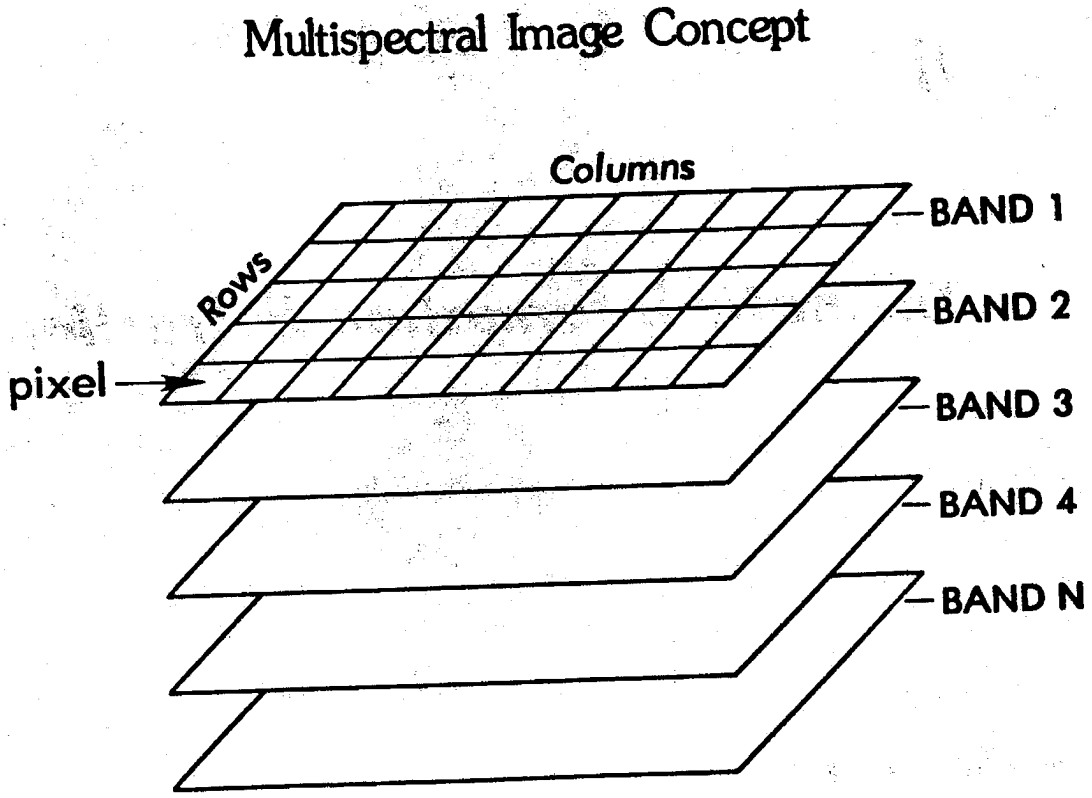



**Figure 8–1** To perform multispectral classification the $n$-bands of remotely sensed data must be in perfect registration.

these training sites is then evaluated and assigned to the class of which it has the highest likelihood of being a member.

In an *unsupervised classification*, the identities of land cover types to be specified as classes within a scene are not generally known a priori because ground truth is lacking or surface features within the scene are not well defined. The computer is required to group (cluster) pixel data into different spectral classes according to some statistically determined criteria. It is then the responsibility of the analyst to label these clusters. This chapter describes aspects of these two approaches.

## SUPERVISED CLASSIFICATION

The following are important aspects of conducting a rigorous and hopefully useful supervised classification of remote sensor data:

1. An appropriate classification scheme must be adopted.
2. Representative training sites must be selected, including an appreciation for signature extension factors, if possible.



OFEK06871

3. Statistics must be extracted from the training site spectral data.
4. The statistics are analyzed to select the appropriate features (bands) to be used in the classification process; this may involve both computer graphic and/or statistical methods of evaluating the degree of between-class separability.
5. Select the appropriate classification algorithm.
6. Classify the imagery into $m$ classes.
7. Statistically evaluate the classification accuracy.

If these considerations are carefully addressed and if there are real differences between the spectral reflectance characteristics of the various land cover classes under investigation, there is every hope that a useful thematic map will result. It is instructive to review these parameters individually to understand their significance.

## The Classification Scheme

The categories of interest must be carefully selected and defined to successfully perform digital image classification. It is important for the analyst to realize, however, that there is a fundamental difference between what we will call "information classes" and "spectral classes" (Jensen et al., 1983; Campbell, 1983). *Information classes* are those that man defines. Conversely, *spectral classes* are those that are inherent in the remote sensor data and must be identified and then labeled by the analyst. For example, in a remotely sensed image of an urban area there is likely to be single-family residential housing. A relatively high spatial resolution (30 × 30 m) remote sensing device such as the thematic mapper might be able to identify a few pure pixels of vegetation and a few pure pixels of asphalt road or shingles. However, it is much more likely that in this residential area the pixel brightness values will be a function of the reflectance from mixtures of vegetation and concrete. Unfortunately, few planners or administrators want to see a map labeled with such classes as (1) concrete, (2) vegetation, and (3) mixture of vegetation and concrete. Rather, they much prefer the analyst to rename the mixture class as single-family residential. The analyst should only do this if, in fact, there is a good association between the mixture class and single-family residential housing. Thus, we see that an analyst must often translate spectral classes into information classes to satisfy bureaucratic requirements. An analyst should understand well the spatial and spectral characteristics of the sensor system and be able to relate these system parameters to the types and proportions of materials found within the scene and within pixel IFOVs. If these parameters are understood, spectral classes can be thoughtfully relabeled as information classes when necessary.

Certain classification schemes have been developed that can readily incorporate land use and/or land cover data obtained by interpreting remotely sensed data. Only a few will be discussed, including the U.S. Geological Survey Land Use/Land Cover



OEEK06872

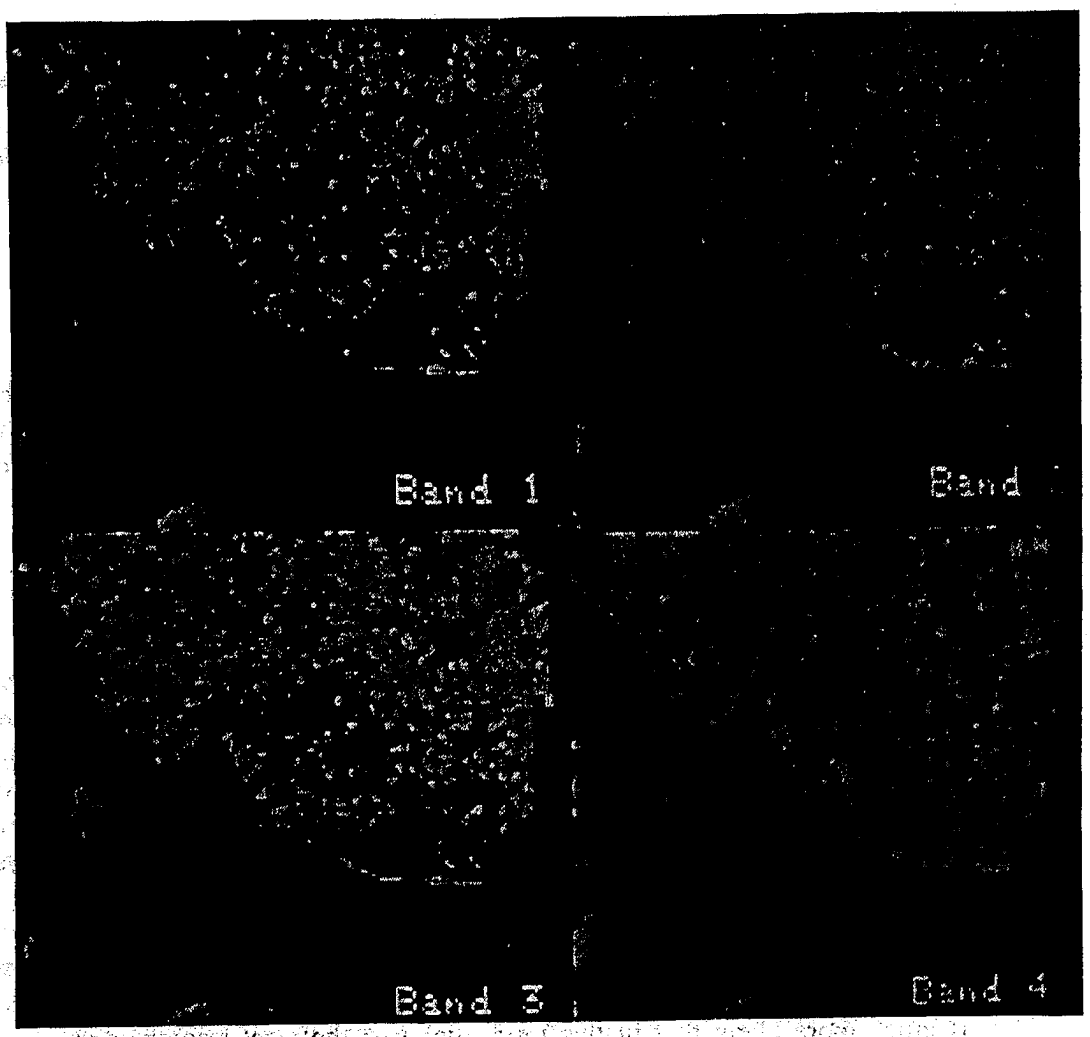



**Figure 8–2** Seven bands of Landsat thematic mapper data for an area centered on the Charleston, S.C., peninsula. Bands 1 to 6 have a spatial resolution of 30 × 30 m while the thermal infrared band 7 has a spatial resolution of 120 × 120 m.

Classification System, the Michigan Classification System, and the Cowardin Wetland Classification System.

Major points of difference between various classification schemes are their (1) emphasis, and (2) ability to incorporate information obtained using remote sensing. For example, the U.S. Geological Survey classification system (Anderson et al., 1976), is "resource" oriented in contrast with various "people or activity" oriented systems such as the *Standard Land Use Coding (SLUC) Manual* (Urban Renewal Administration, 1965). The USGS rationale is that "although there is an obvious need for an urban-oriented land-use classification system, there is also a need for a resource-oriented classification system whose primary emphasis would be the remaining 95 percent of the United States land area." The system addresses this need with eight of the nine level I categories treating land area that is not in urban or built-up categories



OFEK06873

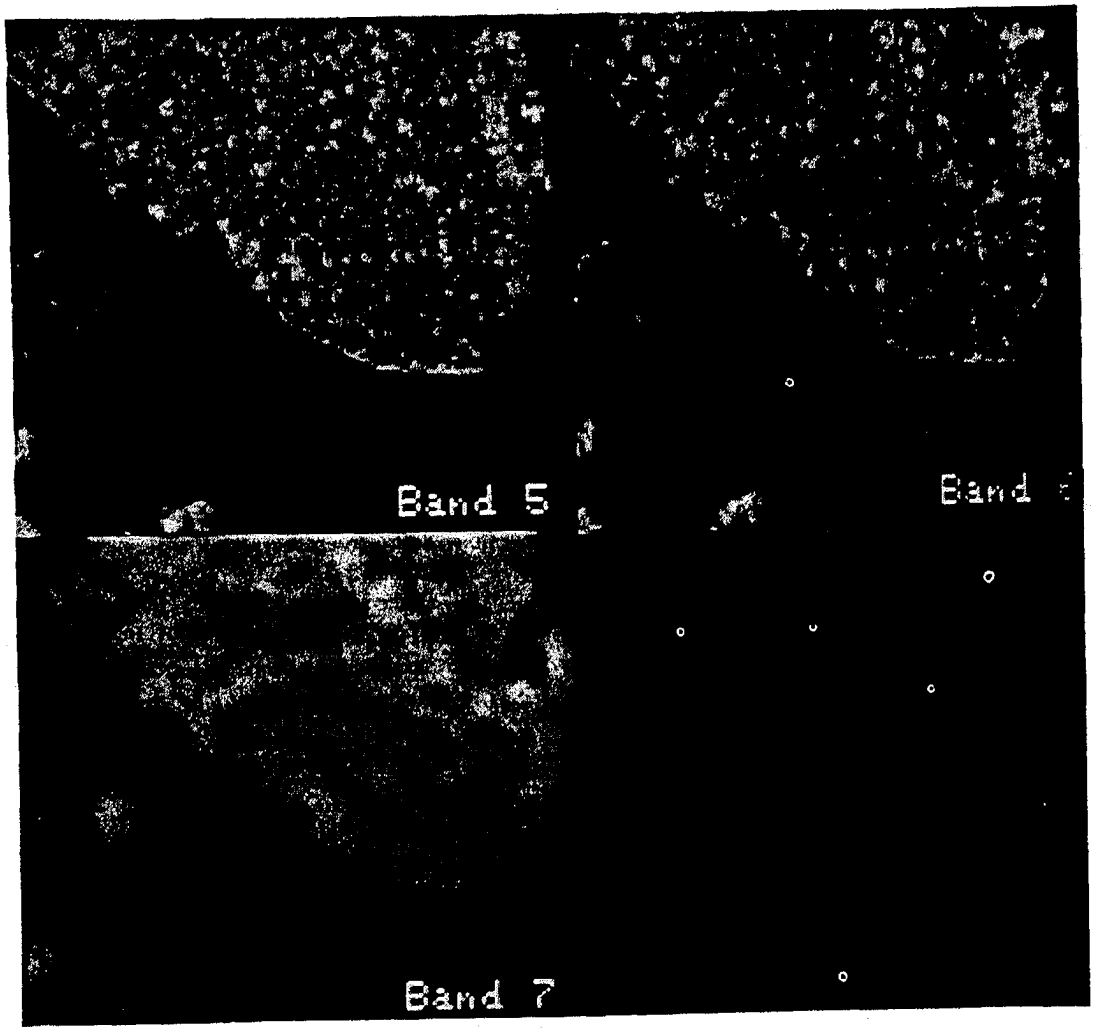



Figure 8–2 (See facing page for caption)

(Table 8–1). The system is designed to be driven primarily by the interpretation of remote sensor data obtained at various scales and resolutions (Table 8–2) and not data collected in situ. It was initially developed to include land-use data that was visually interpreted, although it has been widely used for digital multispectral classification as well. Land use/land cover data classified according to this system is compatible with the land use coding used in the U.S. Geological Survey's Geographic Information Retrieval and Analysis System (GIRAS) (Mitchell et al., 1977).

The Standard Land Use Coding Manual (SLUC), on the other hand, is land-use "activity" oriented and is dependent primarily on in situ observation to obtain remarkably specific land-use information even in regard to the contents of buildings (Rhind and Hudson, 1980). Obviously, there exists the need to merge the two approaches to produce a hybrid classification system that incorporates both land use interpreted from remote sensor data and very precise (and expensive) land-use information obtained in situ when necessary.



OFEK06874

**TABLE 8-1** U.S. Geological Survey Land Use/Land Cover Classification System for Use with Remote Sensor Data

| Level I | | Level II | |
|---|---|---|---|
| 1 | Urban or built-up land | 11 | Residential |
| | | 12 | Commercial and services |
| | | 13 | Industrial |
| | | 14 | Transportation, communications, and services |
| | | 15 | Industrial and commercial complexes |
| | | 16 | Mixed urban or built-up land |
| | | 17 | Other urban or built-up land |
| 2 | Agricultural land | 21 | Cropland and pasture |
| | | 22 | Orchards, groves, vineyards, nurseries, and ornamental horticultural areas |
| | | 23 | Confined feeding operations |
| | | 24 | Other agricultural land |
| 3 | Rangeland | 31 | Herbaceous rangeland |
| | | 32 | Shrub and brush rangeland |
| | | 33 | Mixed rangeland |
| 4 | Forest land | 41 | Deciduous forest land |
| | | 42 | Evergreen forest land |
| | | 43 | Mixed forest land |
| 5 | Water | 51 | Streams and canals |
| | | 52 | Lakes |
| | | 53 | Reservoirs |
| | | 54 | Bays and estuaries |
| 6 | Wetland | 61 | Forested wetland |
| | | 62 | Nonforested wetland |
| 7 | Barren land | 71 | Dry salt flats |
| | | 72 | Beaches |
| | | 73 | Sandy areas other than beaches |
| | | 74 | Bare exposed rocks |
| | | 75 | Strip mines, quarries, and gravel pits |
| | | 76 | Transitional areas |
| | | 77 | Mixed barren land |
| 8 | Tundra | 81 | Shrub and brush tundra |
| | | 82 | Herbaceous tundra |
| | | 83 | Bare ground |
| | | 84 | Mixed tundra |
| 9 | Perennial snow and ice | 91 | Perennial snowfields |
| | | 92 | Glaciers |

*Source:* Anderson et al. (1976).



OFEK06875

**TABLE 8-2** The Four Levels of the U.S. Geological Survey Land Use/Land Cover Classification System and the Type of Remotely Sensed Data Typically Used to Provide the Information

| Classification level | Typical data characteristics |
|---|---|
| I | Landsat (formerly ERTS) type of data |
| II | High-altitude data acquired at 40,000 ft (12,400 m) or above; results in imagery that is less than 1:80,000 scale |
| III | Medium-altitude data acquired between 10,000 and 40,000 ft (3100 and 12,400 m); results in imagery that is between 1:20,000 and 1:80,000 scale |
| IV | Low-altitude data acquired below 10,000 ft (3100 m); results in imagery that is larger than 1:20,000 scale |

*Source:* Anderson et al. (1976) and Jensen et al. (1983). © American Society of Photogrammetry and Remote Sensing. Used with permission.

Such a system was prepared and adopted for the state of Michigan (Michigan Land Use Classification and Referencing Committee, 1975). Although optimized for Michigan land use, it serves as a guide for the preparation of detailed urban/suburban classification schemes developed elsewhere. Like the USGS system, the Michigan system permits aggregation from the lowest level upward to successively higher levels, and in certain instances, the disaggregation of any higher-level data into lower levels. This reduces cost and permits increased benefits as multidisciplinary users access the same data base and avoid duplication of effort. Basically, the nine level I categories of the USGS system were adopted without change. In addition, however, the Michigan system includes detailed categories in levels II, III, and IV consistent with Michigan environmental and cultural conditions. It is land cover oriented up to and including level III, making it amenable to remote sensing data analysis. It is both land cover and activity oriented at level IV so as to better meet user needs at the local level. A complete list of this detailed classification system is given in Jensen et al. (1983).

The United States has tremendous wetland resources. The U.S. Fish and Wildlife Service is responsible for supervising the mapping of all wetland in the United States. The interpretation of remotely sensed data plays an important role in this endeavor. Cowardin et al. (1979) developed a wetland classification system that incorporates information extracted from remote sensor data and in situ measurement. It describes ecological taxa, arranges them in a system useful to resource managers, and provides uniformity of concepts and terms. Wetlands are classified based on plant characteristics, soils, and frequency of flooding. Ecologically related areas of deep water, traditionally not considered wetlands, are included in the classification as deep-water habitats. Five systems form the highest level of the classification hierarchy, including marine, estuarine, riverine, lacustrine, and palustrine (Fig. 8–3). Marine and estuarine systems each have two subsystems, subtidal and intertidal; the riverine system has four subsystems, tidal, lower perennial, upper perennial, and intermittent; the lacustrine has



OFEK06876

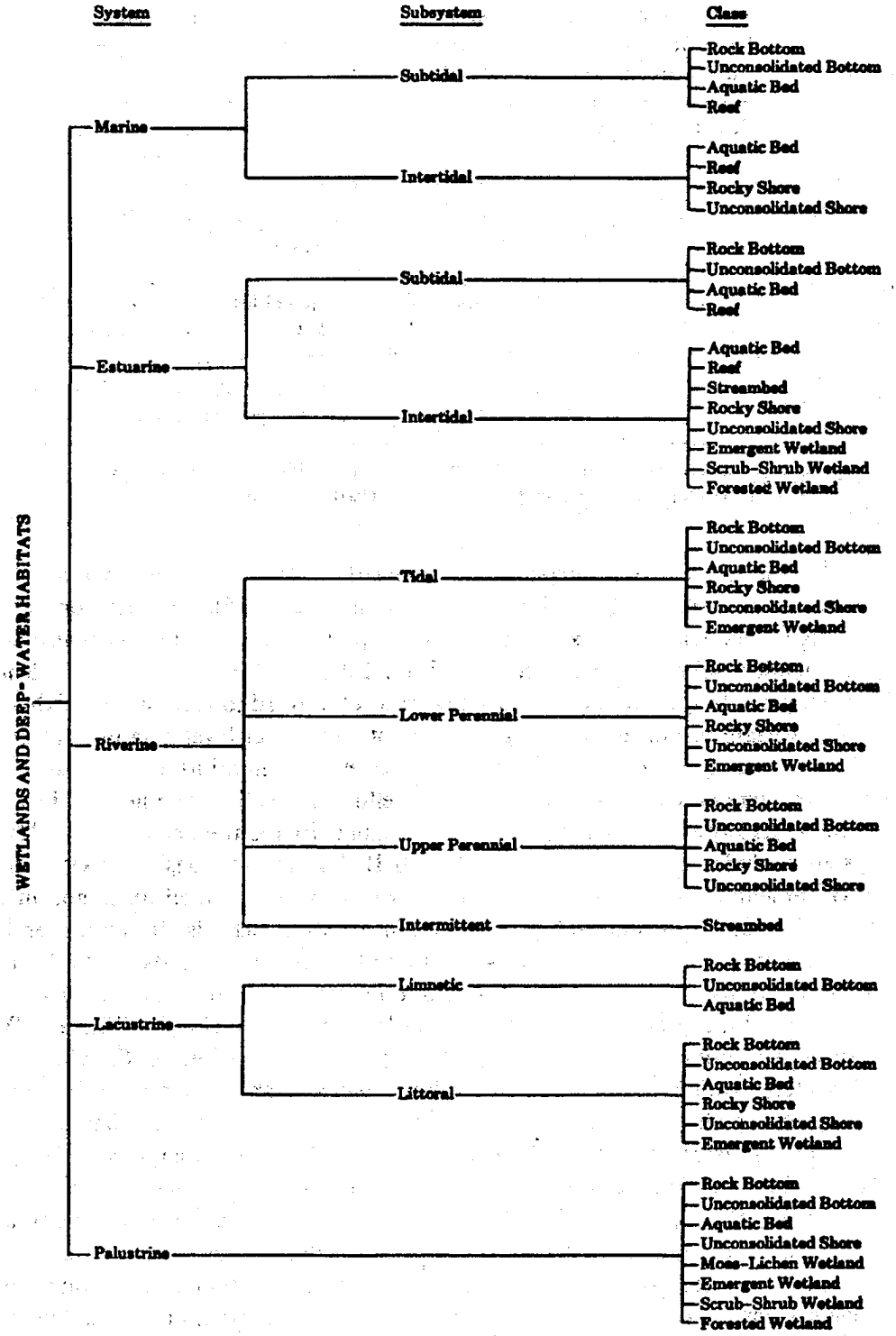



**Figure 8–3** The Cowardin et al. (1979) classification hierarchy of wetlands and deep-water habitats, showing systems, subsystems, and classes. The palustrine system does not include deep-water habitats.





OFEK06877

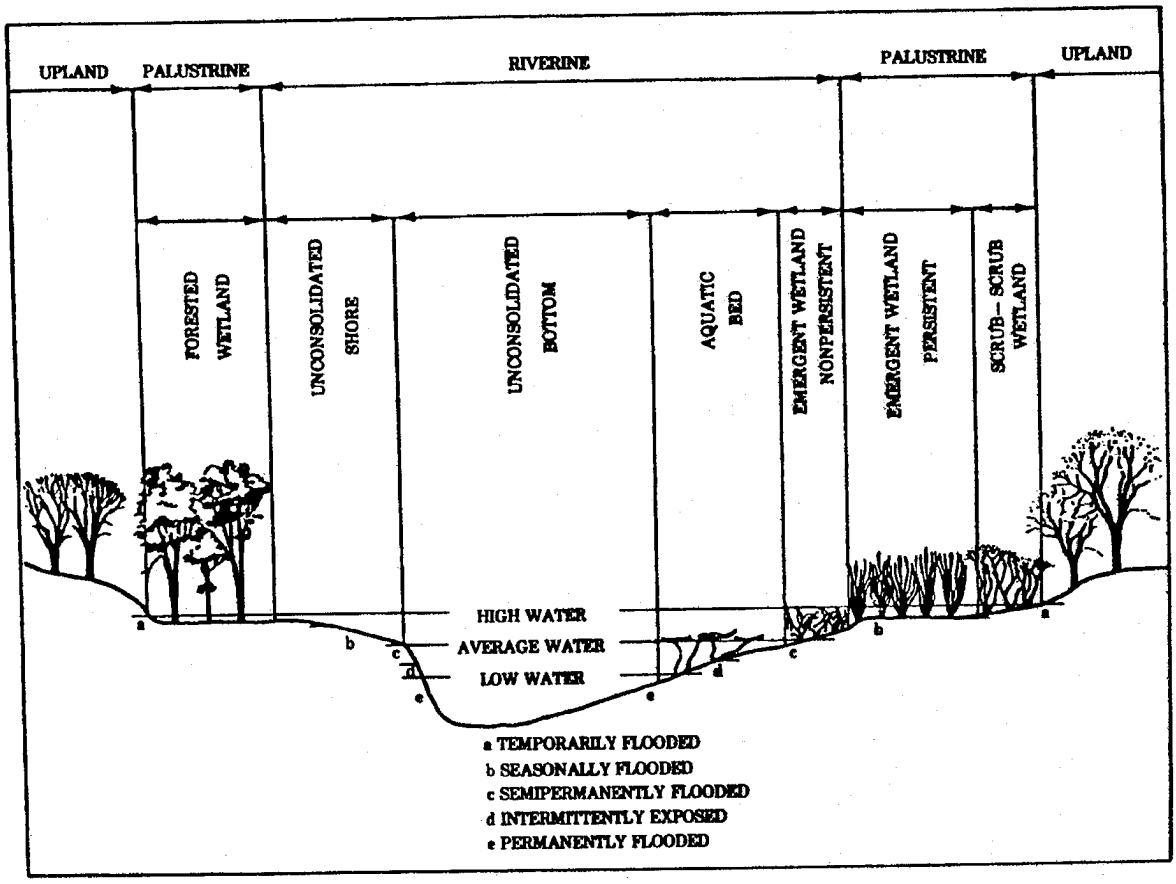



**Figure 8–4** Distinguishing features and examples of habitats in the riverine system. (Cowardin et al., 1979.)

two, littoral and limnetic; and the palustrine has no subsystem. Within the subsystems, classes are based on substrate material and flooding regime or on vegetative life form. The same classes may appear under one or more of the systems or subsystems. The distinguishing features of the riverine system are shown in Fig. 8–4. This is the only nationally recognized wetland classification scheme.

Robinove (1981) suggests that classification schemes such as those just described are rigid, based on a priori knowledge and are "difficult to develop and difficult to use. Nevertheless, they are widely employed because they theoretically provide a framework which can be used in the same way by many individuals whose classification results will be comparable." This brings us to an important consideration. If there is already a reputable classification system for the phenomena we are interested in, it is foolish to start anew, reinventing another system which will probably only be used by ourselves. It is better to adopt or modify existing nationally recognized classification systems. This allows us to interpret the significance of our classification results in light of other studies and makes it easier to share data (Rhind and Hudson, 1980).

Finally, it should be noted that there is a relationship between the level of



OFEK06878

detail in a classification scheme and the spatial resolution of remote sensor systems used to provide information. For example, Welch (1982) summarized this relationship for the mapping of urban/suburban land use and land cover in the United States (Fig. 8–5). A similar relationship exists when mapping vegetation (NASA, 1983; Botkin et al., 1984). For example, the sensor systems and spatial resolutions useful for discriminating vegetation from a global to an in situ perspective are summarized in Fig. 8–6. This suggests that the level of detail in the desired classification system dictates the spatial resolution of the remote sensor data that should be used. Spectral resolution is also an important consideration. However, it is not as critical a parameter as spatial resolution since most of the sensor systems (e.g., Landsat MSS, TM, SPOT, etc.) record energy in approximately the same visible and near-infrared portions of the electromagnetic spectrum (refer to Chapter 2).

### *Training Site Selection and Statistics Extraction*

With the classification scheme adopted, an analyst may select sites within the image that are representative of the land cover classes of interest. These sites should not be atypical but ones that represent the norm for each class. The image coordinates of these sites are then identified and used to extract statistics from the spectral data for each of these areas. Training data should be of value if the environment from

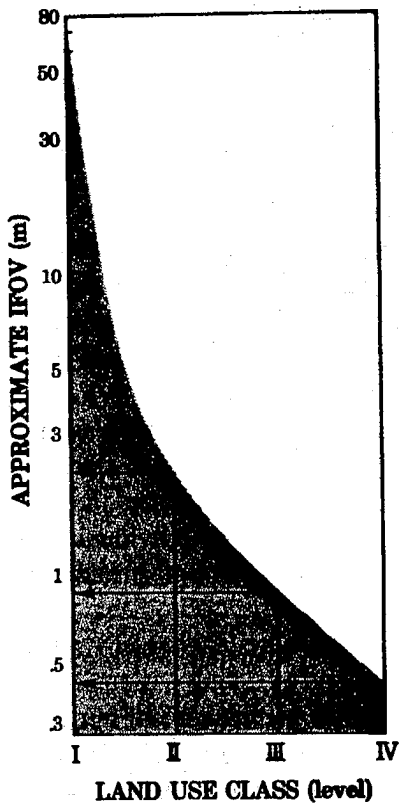
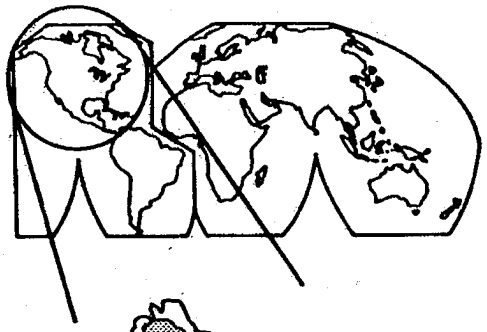



**Figure 8–5** Spatial resolution (IFOV) requirements as a function of the mapping requirements for levels I to IV land-use classes in the United States (based on Anderson et al., 1976). Levels I, II, III, and IV information are normally derived from satellite, high-, medium-, and low-altitude image data, respectively. Note the dramatic increase in resolution required to map level II classes. (From Welch, 1982; Jensen et al., 1983.) © American Society of Photogrammetry. Used with permission.

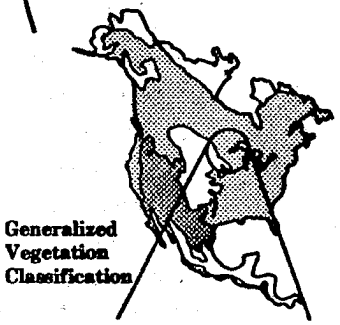


OFEK06879

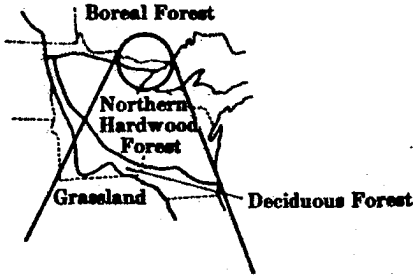
**LEVEL I : Global**
AVHRR
*resolution: 1.1 km*



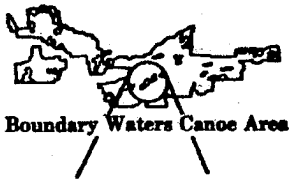
**LEVEL II : Continental**
AVHRR
Landsat Multispectral Scanner
*resolution: 1.1 km — 80 m*



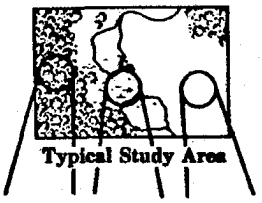

**LEVEL III : Biome**
Landsat Multispectral Scanner
Thematic Mapper
Synthetic Aperture Radars
*resolution : 80 m — 30 m*




**LEVEL IV : Region**
Thematic Mapper
High Altitude Aircraft
Large Format Camera
SPOT
*resolution : 30 m — 3 m +*




**LEVEL V : Plot**
High and Low Altitude Aircraft
*resolution : 3 m + — 1 m +*




**LEVEL VI : In Situ Sample Site**
Surface Measurements
and Observations

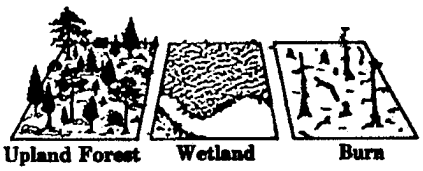



**Figure 8–6** Relationship between the level of detail required and the spatial resolution of various remote sensing systems for vegetation inventories. (From NASA, 1983; Botkin et al., 1984.)





OFEK06880

which they were obtained is relatively homogeneous. For example, if all the soils in a given region are of approximately the same association, it is likely that training data for wheat classification will be useful throughout the region. However, if the soil conditions should change dramatically as the study progresses across the study area, it is likely that the training data acquired at one end of the study area will not be representative of the spectral conditions found in the other side of the study area. Thus, we have a *signature extension* problem, meaning that it is not possible to extend our training data through $x$, $y$ space. The easiest way to remedy this situation is during the preliminary stages of a project. At this time all significant environmental factors that could contribute to signature extension problems are identified, such as differences in soil type; water turbidity; crop species (e.g., two strains of wheat); unusual soil moisture conditions, possibly caused by a thundershower that did not uniformly deposit its precipitation; scattered patches of atmospheric haze; and so on. Such environmental conditions should be carefully annotated on the imagery and the selection of training sites made based on the geographic stratification of these data. In such cases, it may be necessary to retrain the classifier over relatively short geographic distances. However, if environmental conditions are homogeneous or can be held constant (e.g., through band ratioing, atmospheric correction, etc.), it may be possible to extend signatures vast distances in space, significantly reducing the cost and effort involved with retraining (Carr et al., 1983). Much additional research is required before the concept of spatial and temporal (through time) signature extension is fully understood.

Once signature extension factors have been evaluated, the analyst selects representative training sites for each of the classes and collects the spectral statistics for each pixel found within each "training" site. Each site is usually composed of many pixels. The general rule is that if training data are being extracted from $n$ bands, the minimum number of pixels in a class should be $n + 1$. This condition will allow the inverse of the covariance matrix for each class to be calculated, which is important in some classification algorithms. Ideally, $>10n$ pixels of training data are collected for each class.

Program TRAINING in Appendix A accepts the row and column coordinates of training sites for up to 25 classes from as many as six bands of remote sensor data. The program then computes training data statistics to be used for the feature selection and classification algorithms to be discussed.

If the analyst trains on six bands of Landsat thematic mapper data, then each pixel in each training site is represented by a *measurement vector*, $X_c$, such that:

$$X_c = \begin{Bmatrix} BV_{ij,1} \\ BV_{ij2} \\ BV_{ij3} \\ BV_{ij4} \\ \vdots \\ BV_{ijk} \end{Bmatrix} \tag{8-1}$$



OFEK06881

where $BV_{ijk}$ is the brightness value for the $(i,j)$th pixel in band $k$. The brightness values for each pixel in each band in each training class can then be analyzed statistically to yield a mean measurement vector, $M_c$, for each class:

$$M_c = \begin{Bmatrix} \mu_{c1} \\ \mu_{c2} \\ \mu_{c3} \\ \mu_{c4} \\ \vdots \\ \mu_{ck} \end{Bmatrix} \tag{8-2}$$

where $\mu_{ck}$ represents the mean value of the data obtained for class $c$ in band $k$. The raw measurement vector can also be analyzed to yield the covariance matrix for each class $c$:

$$V_{ckl} = \begin{Bmatrix} \text{Cov}_{c11} & \text{Cov}_{c12} & \cdots & \text{Cov}_{ckl} \\ \text{Cov}_{c21} & \text{Cov}_{c22} & \cdots & \text{Cov}_{c2n} \\ \vdots & \vdots & \cdots & \vdots \\ \text{Cov}_{cn1} & \text{Cov}_{cn2} & \cdots & \text{Cov}_{cnn} \end{Bmatrix} = V_c \tag{8-3}$$

where $\text{Cov}_{ckl}$ is the covariance of class c between bands $k$ through $l$. For brevity, the notation for the covariance matrix for class $c$ (i.e., $V_{ckl}$) will be shortened to $V_c$. The same will be true for the covariance matrix of class $d$ (i.e., $V_{dkl} = V_d$).

The mean, standard deviation, variance, minimum value, maximum value, variance-covariance matrix, and correlation matrix for the training statistics of five Charleston land cover classes (residential, commercial, wetland, forest, and water) are listed in Table 8–3. These represent fundamental information on the spectral characteristics of the five classes. However, it is very difficult to use just this information to select the optimum bands for supervised classification. For this reason analysts often use methods of feature (band) selection.

**Feature selection.** Once the training statistics have been systematically collected from each band for each class of interest, a judgment must be made to determine those bands that are most effective in discriminating each class from all others. This process is commonly called *feature selection*. The goal is to delete from the analysis those bands that provide only redundant spectral information. In this way the *dimensionality* (i.e., the number of bands to be processed) in the data set may be reduced. This process minimizes the cost of the digital image classification (but hopefully, not the accuracy). Feature selection may involve both statistical and/or graphical analysis to determine the degree of between-class separability in the remote sensor training data. Combinations of bands are normally ranked according to their potential ability to discriminate each class from all others using $n$ bands at a time. Statistical measures such as divergence will be discussed shortly.



OFEK06882

**TABLE 8–3** Univariate and Multivariate Training Statistics for Five Land Cover Classes Using Six Bands of Landsat Thematic Mapper Data Obtained over Charleston, S.C.

a. Statistics for Residential

| | Band: 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| | | | *Univariate statistics* | | | |
| Mean | 70.6 | 28.8 | 29.8 | 36.7 | 55.7 | 28.2 |
| Std. dev. | 6.90 | 3.96 | 5.65 | 4.53 | 10.72 | 6.70 |
| Variance | 47.6 | 15.7 | 31.9 | 20.6 | 114.9 | 44.9 |
| Minimum | 59 | 22 | 19 | 26 | 32 | 16 |
| Maximum | 91 | 41 | 45 | 52 | 84 | 48 |
| | | | *Variance-covariance matrix* | | | |
| 1 | 47.65 | | | | | |
| 2 | 24.76 | 15.70 | | | | |
| 3 | 35.71 | 20.34 | 31.91 | | | |
| 4 | 12.45 | 8.27 | 12.01 | 20.56 | | |
| 5 | 34.71 | 23.79 | 38.81 | 22.30 | 114.89 | |
| 6 | 30.46 | 18.70 | 30.86 | 12.99 | 60.63 | 44.92 |
| | | | *Correlation matrix* | | | |
| 1 | 1.00 | | | | | |
| 2 | 0.91 | 1.00 | | | | |
| 3 | 0.92 | 0.91 | 1.00 | | | |
| 4 | 0.40 | 0.46 | 0.47 | 1.00 | | |
| 5 | 0.47 | 0.56 | 0.64 | 0.46 | 1.00 | |
| 6 | 0.66 | 0.70 | 0.82 | 0.43 | 0.84 | 1.00 |

b. Statistics for Commercial

| | Band: 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| | | | *Univariate Statistics* | | | |
| Mean | 112.4 | 53.3 | 63.5 | 54.8 | 77.4 | 45.6 |
| Std. dev. | 5.77 | 4.55 | 3.95 | 3.88 | 11.16 | 7.56 |
| Variance | 33.3 | 20.7 | 15.6 | 15.0 | 124.6 | 57.2 |
| Minimum | 103 | 43 | 56 | 47 | 57 | 32 |
| Maximum | 124 | 59 | 72 | 62 | 98 | 57 |
| | | | *Variance-covariance matrix* | | | |
| 1 | 33.29 | | | | | |
| 2 | 11.76 | 20.71 | | | | |
| 3 | 19.13 | 11.42 | 15.61 | | | |
| 4 | 19.60 | 12.77 | 14.26 | 15.03 | | |
| 5 | −16.62 | 15.84 | 2.39 | 0.94 | 124.63 | |
| 6 | −4.58 | 17.15 | 6.94 | 5.76 | 68.81 | 57.16 |
| | | | *Correlation matrix* | | | |
| 1 | 1.00 | | | | | |
| 2 | 0.45 | 1.00 | | | | |
| 3 | 0.84 | 0.64 | 1.00 | | | |
| 4 | 0.88 | 0.72 | 0.93 | 1.00 | | |
| 5 | −0.26 | 0.31 | 0.05 | 0.02 | 1.00 | |
| 6 | −0.10 | 0.50 | 0.23 | 0.20 | 0.82 | 1.00 |





OFEK06883

**TABLE 8–3** Cont.

c. Statistics for Wetland

| | Band: 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| | | | *Univariate statistics* | | | |
| Mean | 59.0 | 21.6 | 19.7 | 20.2 | 28.2 | 12.2 |
| Std. dev. | 1.61 | 0.71 | 0.80 | 1.88 | 4.31 | 1.60 |
| Variance | 2.6 | 0.5 | 0.6 | 3.5 | 18.6 | 2.6 |
| Minimum | 54 | 20 | 18 | 17 | 20 | 9 |
| Maximum | 63 | 25 | 21 | 25 | 35 | 16 |
| | | | *Variance-covariance matrix* | | | |
| 1 | 2.59 | | | | | |
| 2 | 0.14 | 0.50 | | | | |
| 3 | 0.22 | 0.15 | 0.63 | | | |
| 4 | −0.64 | 0.17 | 0.60 | 3.54 | | |
| 5 | −1.20 | 0.28 | 0.93 | 5.93 | 18.61 | |
| 6 | −0.32 | 0.17 | 0.40 | 1.72 | 4.53 | 2.55 |
| | | | *Correlation matrix* | | | |
| 1 | 1.00 | | | | | |
| 2 | 0.12 | 1.00 | | | | |
| 3 | 0.17 | 0.26 | 1.00 | | | |
| 4 | −0.21 | 0.12 | 0.40 | 1.00 | | |
| 5 | −0.17 | 0.09 | 0.27 | 0.73 | 1.00 | |
| 6 | −0.13 | 0.15 | 0.32 | 0.57 | 0.66 | 1.00 |

d. Statistics for Forest

| | Band: 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| | | | *Univariate statistics* | | | |
| Mean | 57.5 | 21.7 | 19.0 | 39.1 | 35.5 | 12.5 |
| Std. dev. | 2.21 | 1.39 | 1.40 | 5.11 | 6.41 | 2.97 |
| Variance | 4.9 | 1.9 | 1.9 | 26.1 | 41.1 | 8.8 |
| Minimum | 53 | 20 | 17 | 25 | 22 | 8 |
| Maximum | 63 | 28 | 24 | 48 | 54 | 22 |
| | | | *Variance-covariance matrix* | | | |
| 1 | 4.89 | | | | | |
| 2 | 1.91 | 1.93 | | | | |
| 3 | 2.05 | 1.54 | 1.95 | | | |
| 4 | 5.29 | 3.95 | 4.06 | 26.08 | | |
| 5 | 9.89 | 5.30 | 5.66 | 13.80 | 41.13 | |
| 6 | 4.63 | 2.34 | 2.22 | 3.22 | 16.59 | 8.84 |
| | | | *Correlation matrix* | | | |
| 1 | 1.00 | | | | | |
| 2 | 0.62 | 1.00 | | | | |
| 3 | 0.66 | 0.80 | 1.00 | | | |
| 4 | 0.47 | 0.56 | 0.57 | 1.00 | | |
| 5 | 0.70 | 0.59 | 0.63 | 0.42 | 1.00 | |
| 6 | 0.70 | 0.57 | 0.53 | 0.21 | 0.87 | 1.00 |





OFEK06884

**TABLE 8–3** Cont.

c. Statistics for Water

| | Band: 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| | | | *Univariate statistics* | | | |
| Mean | 61.5 | 23.2 | 18.3 | 9.3 | 5.2 | 2.7 |
| Std. dev. | 1.31 | 0.66 | 0.72 | 0.56 | 0.71 | 1.01 |
| Variance | 1.7 | 0.4 | 0.5 | 0.3 | 0.5 | 1.0 |
| Minimum | 58 | 22 | 17 | 8 | 4 | 0 |
| Maximum | 65 | 25 | 20 | 10 | 7 | 5 |
| | | | *Variance-covariance matrix* | | | |
| 1 | 1.72 | | | | | |
| 2 | 0.06 | 0.43 | | | | |
| 3 | 0.12 | 0.19 | 0.51 | | | |
| 4 | 0.09 | 0.05 | 0.05 | 0.32 | | |
| 5 | −0.26 | −0.05 | −0.11 | −0.07 | 0.51 | |
| 6 | −0.21 | −0.05 | −0.03 | −0.07 | 0.05 | 1.03 |
| | | | *Correlation matrix* | | | |
| 1 | 1.00 | | | | | |
| 2 | 0.07 | 1.00 | | | | |
| 3 | 0.13 | 0.40 | 1.00 | | | |
| 4 | 0.12 | 0.14 | 0.11 | 1.00 | | |
| 5 | −0.28 | −0.10 | −0.21 | −0.17 | 1.00 | |
| 6 | −0.16 | −0.08 | −0.04 | −0.11 | 0.07 | 1.00 |

If statistical techniques provide all the information necessary to select features for classification, why use graphical methods alone or in conjunction with statistical techniques? The reason is simple. An analyst may base a decision solely on the statistic, yet never obtain a fundamental understanding of the spectral nature of the data being analyzed. In effect, without ever visualizing where the spectral measurements cluster in $n$-dimensional feature space, each new supervised classification finds the analyst beginning anew, relying totally on abstract statistical analysis. Many of the practitioners of remote sensing are by necessity very graphically literate (i.e., they can readily interpret graphs) (Boardman, 1983). Therefore, a graphic display of the statistical data might be useful and often necessary for a thorough analysis of multispectral training data and feature selection. Several graphic feature selection methods have been developed for this purpose.

*Graphic methods of feature selection* may be grouped as being either two-dimensional or synthetic three-dimensional displays of training class statistics. One method of two-dimensional display is the "spectral plots" available in LARSYS and other image processing systems (Anuta, 1977), where statistics are displayed in a bar graph format for each band (Fig. 8–7). The "radiometric interpretive legend" developed by Robinove (1977) is another attempt to communicate visually the relative and absolute location of training class statistics. This method rearranges the data displayed in the spectral plots (Fig. 8–8). Still other two-dimensional graphic methods plot



OFEK06885

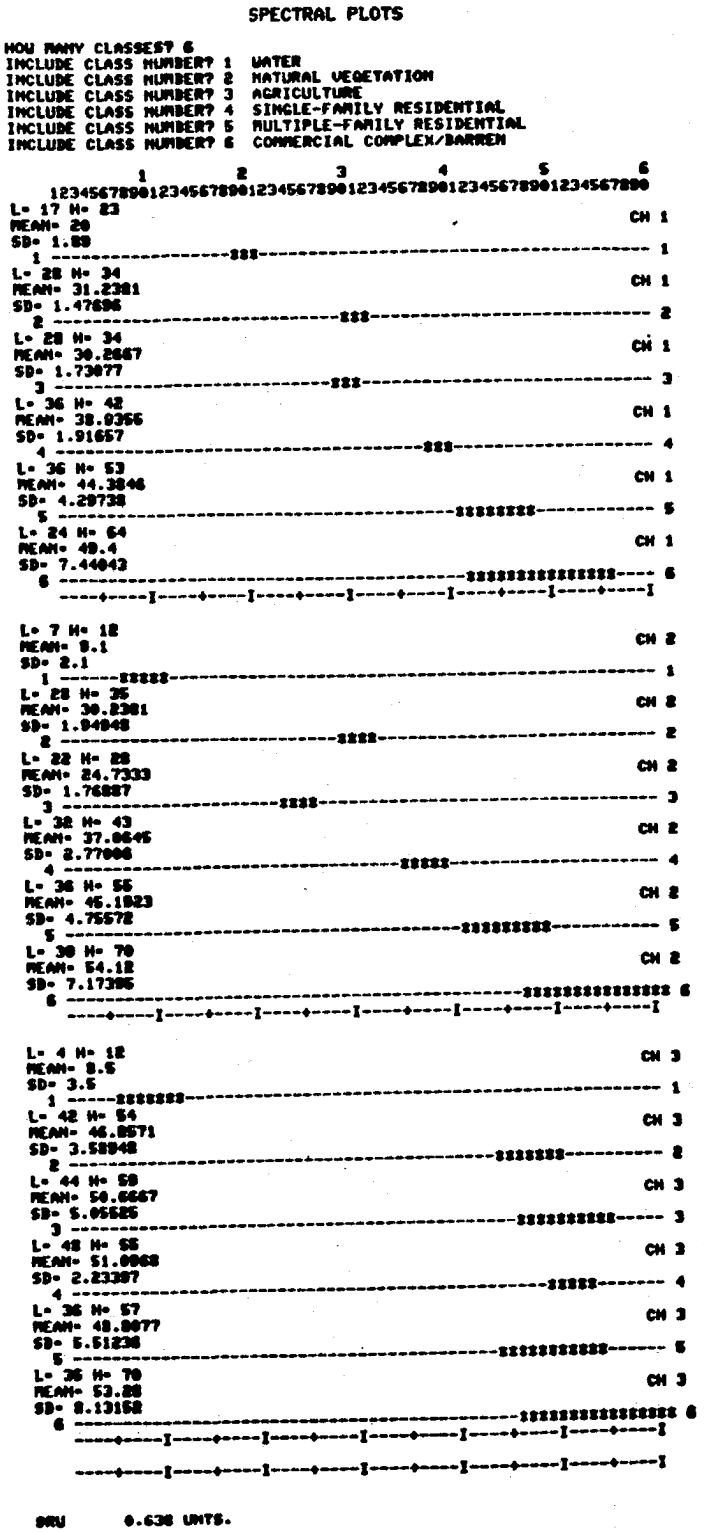



Figure 8–7 Cospectral plots of data analyzed by Jensen (1979). Training statistics (the mean ±1 standard deviation) for six land-cover classes are displayed for three Landsat bands. The display can be used to identify between-class separation for each class and band.



OFEK06886

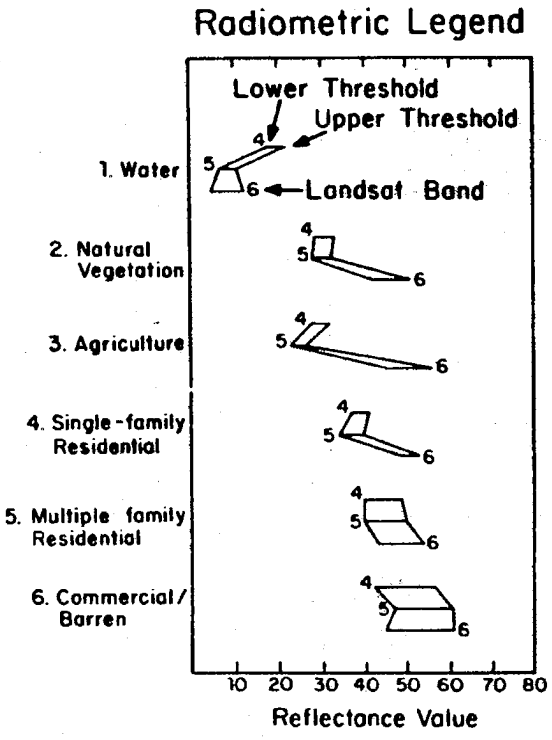



**Figure 8–8** Radiometric interpretive legend, although originally developed as a thematic map key (Robinove, 1977), is useful for assessing the separability of training class statistics. The relationship between spectral band and land cover training statistics is shown by a two-dimensional polygon with the remotely sensed bands plotted on the ordinate and reflectance values on the abscissa. (From Jensen, 1979.)

"ellipses or boxes" of training class statistics (Downs and Faust, 1978). Statistics in two bands are compared at one time until all possible combinations are exhausted. The statistics for the five Charleston land-cover classes are portrayed in this manner using bands 4 and 5 in Fig. 8–9. The mean $\pm$ 1 standard deviation was used to obtain the lower and upper dimension of the boxes. The histogram of each class is also plotted on each axis in this example. Notice the degree of overlap, particularly between classes 1 and 4 using band 4 data and classes 3 and 4 using band 5 data. The major drawback of such displays is that the analyst must overlay one set of graphs onto another in order to visualize the feature separability of three or more bands at one time. Driscoll and Walker (1982) improved on this type of two-dimensional presentation by plotting the per pixel training class statistics in a bivariate scatterplot which was color-coded. This method is effective but again suffers from the problem that it is only possible to visualize the training statistics using two bands at a time.

Graphic displays of training class statistics that facilitate analysis of three bands at once often take the form of parallelepipeds displayed in three-dimensional feature space (Fig. 8–10). Jensen (1979) presented a method of displaying these parallelepipeds in synthetic three-dimensional space and of interactively varying the viewpoint azimuth and elevation angles to enhance feature analysis and selection. The following discussion on the creation of the three-dimensional parallelepipeds is also pertinent to the section on parallelepiped classification to be presented.

To display three-dimensional parallelepipeds in feature space, the analyst must obtain fundamental training class statistical data and make several decisions. First,



OFEK06887

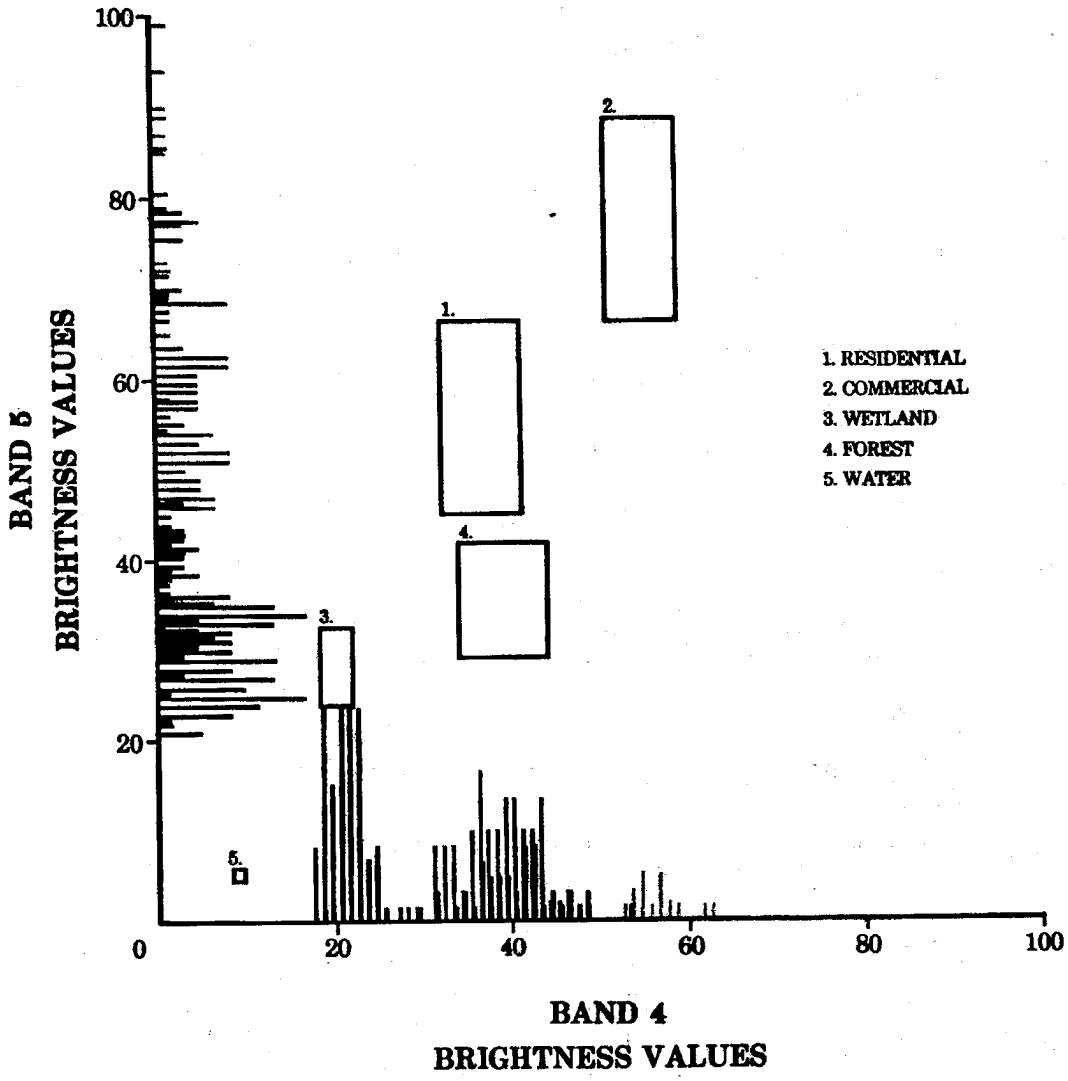



**Figure 8–9** Plot of the Charleston, S.C., thematic mapper training statistics for five classes measured in bands 4 and 5. The upper and lower limit of each box is ±1 standard deviation. The actual histogram for each class is shown on each axis. In some instances it is difficult to differentiate between the histograms because they overlap. Note the overlap especially for classes 1 and 4 on the band 4 axis and classes 3 and 4 on the band 5 axis.

the mean $\mu_{ck}$, and standard deviation, $s_{ck}$, of training class statistics for each class $c$ and band $k$ are required. These data are routinely provided by image processing systems. Using the mean and standard deviation statistics, the analyst determines the lower and upper threshold values for each class and band. Next, the analyst selects a combination of three bands to portray initially because it is not possible to use all six bands at once in a three-dimensional display. Landsat TM bands 4, 5, and 6 are used in the following example; however, the method is applicable to any three-band subset of a multispectral data set. Using this information, a parallelepiped



OFEK06888

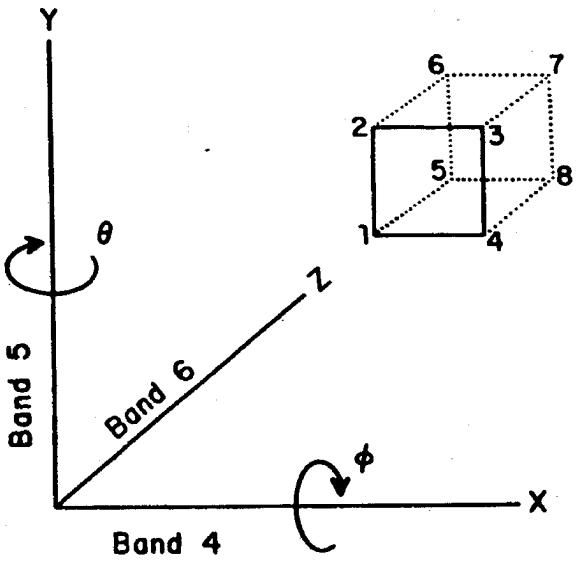



**Figure 8–10** Simple parallelepiped displayed in pseudo-three-dimensional space. Each of the eight corners represents a unique $X, Y, Z$ coordinate corresponding to a lower or upper threshold value of the training data. For example, the original coordinates of point 4 are associated with (1) the upper threshold value of band 4, (2) the lower threshold value of band 5, and (3) the lower threshold value of band 6. Rotation matrix transformations cause the original coordinates to be rotated about the $Y, Z$ axis some $\theta°$ and the $X, Z$ axis some $\phi°$. (From Jensen, 1979.)

for each land cover class may be derived. Each corner of a parallelepiped is identifiable by a unique set of $X, Y, Z$ coordinates corresponding to either the lower or upper threshold value for the three bands under investigation (Fig. 8–10).

By applying a rotation matrix transformation to the original $X, Y$, and $Z$ coordinates defined as transpose vector $P^T$, it is possible to provide a synthetic three-dimensional view of parallelepipeds for numerous land cover classes. A viewing azimuth rotation, $A$, causes the parallelepipeds to be rotated about the $Y, Z$ axis some $t$ radians (Fig. 8–10) using the formula

$$\overset{A}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos t & -\sin t \\ 0 & \sin t & \cos t \end{bmatrix}} \overset{P^T}{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}} = \overset{P^{T\prime}}{\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}} \tag{8-4}$$

A viewing elevation rotation, $B$, causes the parallelepiped to be rotated about the $X, Z$ axis some $s$ radians using the formula

$$\overset{B}{\begin{bmatrix} \cos s & 0 & -\sin s \\ 0 & 1 & 0 \\ \sin s & 0 & \cos s \end{bmatrix}} \overset{P^T}{\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}} = \overset{P^{T\prime}}{\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}} \tag{8-5}$$

In practice, these two matrix transformations, $A$ and $B$, are concatenated (Newman and Sproull, 1978; Gasson, 1983) using matrix algebra to provide a final matrix transformation $C$ that is applied to the original $P^T$ vector values, yielding $P^{T\prime}$. This transformation causes the original $X, Y$, and $Z$ coordinates of $P^T$ to be shifted about and contain depth information as vector $P^{T\prime}$ (MacDougall, 1976). Display devices are normally two-dimensional (e.g., plotter surfaces or cathode-ray-tube screens); con-



OFEK06889

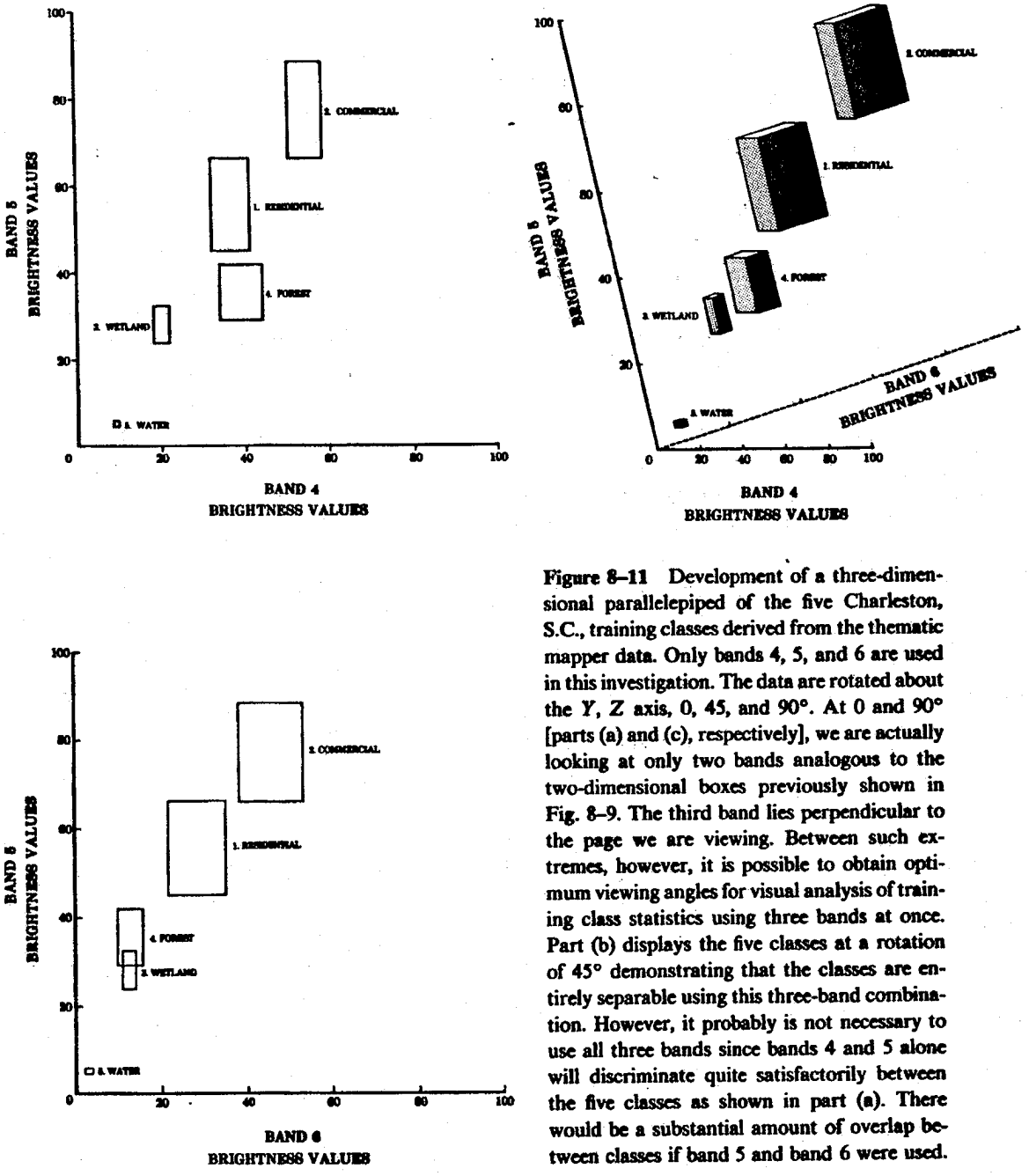



**Figure 8–11** Development of a three-dimensional parallelepiped of the five Charleston, S.C., training classes derived from the thematic mapper data. Only bands 4, 5, and 6 are used in this investigation. The data are rotated about the $Y$, $Z$ axis, 0, 45, and 90°. At 0 and 90° [parts (a) and (c), respectively], we are actually looking at only two bands analogous to the two-dimensional boxes previously shown in Fig. 8–9. The third band lies perpendicular to the page we are viewing. Between such extremes, however, it is possible to obtain optimum viewing angles for visual analysis of training class statistics using three bands at once. Part (b) displays the five classes at a rotation of 45° demonstrating that the classes are entirely separable using this three-band combination. However, it probably is not necessary to use all three bands since bands 4 and 5 alone will discriminate quite satisfactorily between the five classes as shown in part (a). There would be a substantial amount of overlap between classes if band 5 and band 6 were used.

sequently, only the $X'$ and $Y'$ elements of the transformed matrix $P^{T\prime}$ are used to draw the parallelepipeds.

Manipulation of the transformed coordinates of the Charleston training statistics using program PIPED in Appendix A provides a useful presentation (Fig. 8–11). In Fig. 8–11a, all three bands (4, 5, and 6) are being displayed except that the band 6 statistics are perpendicular (orthogonal) to the sheet of paper. Analysis of the



OFEK06890

upper and lower boundary for each class reveals that separation in band 4 is good for all classes at ± 1 standard deviation except for residential (class 1) and forest (4), which would overlap. Band 5 data exhibit fairly good separation between the individual classes except for wetland (3) and forest (4), which have some overlap. Notice how the use of two bands instead of just a single band completely separates the boxes. For example, a classification based on density slicing either band 4 or band 5 to classify these five classes would probably produce relatively poor results. However, if both bands 4 and 5 are used, there should be adequate separation to produce a useful thematic map of these five classes.

By rotating the display 45°, the contribution of band 6 becomes apparent (Fig. 8–11b). This represents a pseudo three-dimensional display of the parallelepipeds. As the display is rotated another 45° to 90°, band 6 data collapse onto what was the band 4 axis (Fig. 8–11c). The band 4 axis is now perpendicular to the page, just as band 6 was originally. The band 6, band 5 plot (Fig. 8–11c) displays some overlap between wetland (3) and forest (4). By systematically specifying various azimuth and elevation angles it is possible to display the parallelepipeds for optimum visual examination. This allows the analyst to obtain insight as to the consistent location of the spectra in three-dimensional feature space.

In this example, it may be evident that just two bands, 4 and 5, provide as good if not better separation than do all three bands used together. However, this may not be the very best set of two or three bands to use. It might be useful to evaluate other two- or three-band combinations. In fact, a certain combination of perhaps four or five bands used all at one time might be superior. The only way to determine this is through statistical feature selection.

*Statistical methods of feature selection* are used to quantitatively select the subset of bands (or features) that provides the greatest degree of statistical separability between any two classes, $c$ and $d$. The basic problem of spectral pattern recognition is: Given a spectral distribution of data in $n$ bands of remotely sensed data, find a discrimination technique that will allow separation of the major land cover categories with a minimum of error and a minimum number of bands. This problem is demonstrated diagrammatically using just 1 band and 2 classes in Fig. 8–12. Generally, the more bands analyzed in a classification, the greater the cost and perhaps the greater the amount of redundant spectral information being used. Examining the histograms in Fig. 8–9 suggests that there is substantial overlap between classes 1 and 4 in band 4 and between classes 3 and 4 in band 5. When there is overlap, any decision rule that one could use to separate or distinguish between two classes must be concerned with two types of error (Fig. 8–12):

1. A pixel may be assigned to a class to which it does not belong (an error of commission).
2. A pixel is not assigned to its appropriate class (an error of omission).

The goal is to select an optimum subset of bands and apply appropriate classification techniques to minimize both types of error in the classification process. If the training



OFEK06891

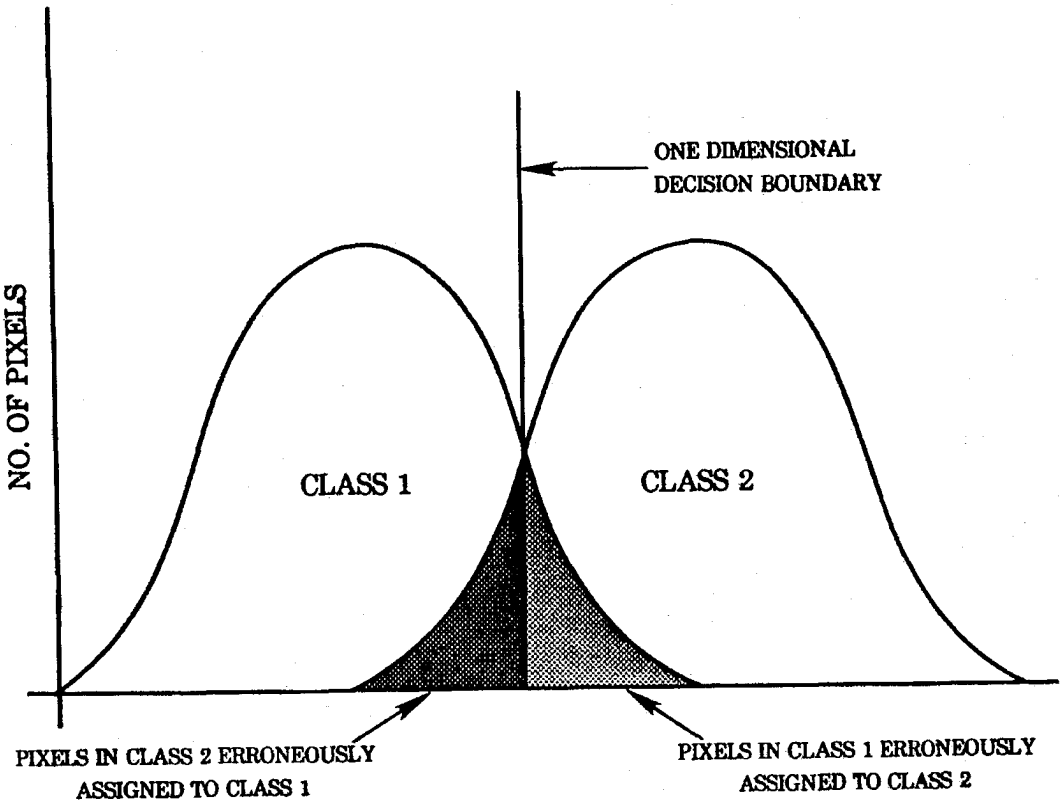



**Figure 8–12** The basic problem in remote sensing pattern recognition classification is: Given a spectral distribution of data in $n$ bands (here just one band) find an $n$-dimensional decision boundary that will allow the separation of the major classes (just 2 in this example) with a minimum of error and a minimum number of bands being evaluated. The dark areas of both distributions identifies potential classification error.

data for each class from each band are normally distributed as suggested in Fig. 8–12, it is possible to use either a divergence or transformed divergence equation to identify the optimum subset of bands to use in the classification procedure.

*Divergence* was one of the first measures of statistical separability used in the machine processing of remote sensor data, and it is still widely used as a method of feature selection (Swain and Wacker, 1971; Haack, 1983). It addresses the basic problem of deciding what is the best $q$-band subset of $n$ bands for use in the supervised classification process. The number of combinations, $C$, of $n$ bands taken $q$ at a time is

$$C\left(\frac{n}{q}\right)=\frac{n!}{q!(n-q)!} \tag{8-6}$$

Thus, if there are six thematic mapper bands and we are interested in the three best bands to use in the classification of the Charleston scene, this results in 20 combinations that must be evaluated:



OFEK06892

$$C\left(\frac{6}{3}\right) = \frac{6!}{3!(6-3)!}$$
$$= \frac{720}{6(6)} \qquad (8\text{-}7)$$
$$= 20 \text{ combinations}$$

If the best two-band combinations were desired, it would be necessary to evaluate 15 possible combinations.

Divergence is computed using the mean and covariance matrices of the class statistics collected in the training phase of the supervised classification. We will initiate the discussion by concerning ourselves with the statistical separability between just two classes, $c$ and $d$. The degree of divergence or "separability" between $c$ and $d$, $\text{Diverg}_{cd}$, is computed according to the formula

$$\begin{aligned}\text{Diverg}_{cd} = {} & 0.5 \text{ Tr } [(V_c - V_d)(V_d^{-1} - V_c^{-1})] \\ & + 0.5 \text{ Tr } [(V_c^{-1} + V_d^{-1})(M_c - M_d)(M_c - M_d)^T)]\end{aligned} \qquad (8\text{-}8)$$

where Tr $[\cdot]$ is the trace of a matrix (i.e., the sum of the diagonal elements), $V_c$ and $V_d$ are the covariance matrices for the two classes, $c$ and $d$, under investigation, and $M_c$ and $M_d$ are the mean vectors for classes $c$ and $d$. It should be remembered that the size of the covariance matrices $V_c$ and $V_d$ is a function of the number of bands used in the training process (i.e., if six bands were trained upon, both $V_c$ and $V_d$ would be matrices $6 \times 6$ in dimension). Divergence in this case would be used to identify the statistical separability of the two training classes using six bands of training data. However, this is not the usual goal of applying divergence. What we actually want to know is the optimum subset of $q$ bands. For example, if $q = 3$, what subset of three bands provides the best separation between these two classes? Therefore, in our example, we would proceed to systematically apply the algorithm to the 20 three-band combinations computing the divergence for our two classes of interest and eventually identifying the subset of bands, perhaps bands 2, 3, and 6, that results in the largest divergence value.

But what about the case where there are more than two classes? In this instance, the most common solution is to compute the *average divergence*, $\text{Diverg}_{avg}$. This involves computing the average over all possible pairs of classes, $c$ and $d$, while holding the subset of bands, $q$ constant. Then another subset of bands, $q$ is selected for the $m$ classes and analyzed. The subset of features (bands) having the maximum average divergence may be the superior set of bands to use in the classification algorithm. This can be expressed as:

$$\text{Diverg}_{avg} = \frac{\sum_{c=1}^{m-1} \sum_{d=c+1}^{m} \text{Diverg}_{cd}}{C} \qquad (8\text{-}9)$$



OFEK06893

Using this, the band subset, $q$, with the highest average divergence would be selected as the most appropriate set of bands for classifying the $m$ classes.

Kumar and Silva (1977) suggest that it is possible to take the divergence logic one step further and compute transformed divergence, $\text{Diverg}_{cd}^{T}$, expressed as:

$$\text{Diverg}_{cd}^{T} = 2000\left(1 - \exp\frac{(-\text{Diverg}_{cd})}{8}\right) \tag{8-10}$$

This statistic gives an exponentially decreasing weight to increasing distances between the classes. It also scales the divergence values to lie between 0 and 2000.

Program DIVERGENCE listed in Appendix A analyzes the covariance matrices and mean vectors for pairs of classes and computes the divergence and transformed divergence for $m$ classes evaluating $q$ bands at a time. For example, Table 8–4 demonstrates which bands are most useful when taken 1, 2, 3, 4, or 5 at a time. There is no need to compute the divergence using all six bands, since this represents the totality of the data set. It is useful, however, to calculate divergence with individual channels ($q = 1$) since a single channel might adequately discriminate between all classes of interest.

A transformed divergence value of 2000 suggests excellent between-class separation. Above 1900 provides good separation, while below 1700 is poor. It can be seen that for the Charleston study, using any single band (Table 8–4a) would not produce as acceptable results as will using bands 3 and 4 together (Table 8–4b). There are several three-band combinations that should yield good between-class separation for all classes. Most of them understandably include bands 3 and 4. But why should we use three, four, five, or six bands in the classification when divergence statistics suggest that very good between-class separation is possible using just two bands? We probably should not if the dimensionality of the data set can be reduced by a factor of 3 (from 6 to 2) and classification results appear promising using just the two bands.

There are other methods of feature selection also based on determining the separability between two classes at a time. For example, the Bhattacharyya distance assumes that the two classes, $c$ and $d$, are Gaussian in nature and that the means, $M_c$, $M_d$ and covariance matrices $V_c$ and $V_d$ are available. It is computed as:

$$\begin{aligned}\text{Bhat}_{cd} = &\frac{1}{8}(M_c - M_d)' \frac{V_d + V_d}{2}(M_c - M_d) \\ &+ \frac{1}{2}\text{Ln}\left[\frac{\det[(V_c + V_d)/2]}{\sqrt{\det(V_c)}\sqrt{\det(V_d)}}\right]\end{aligned} \tag{8-11}$$

To select the best $q$ features (i.e., combination of bands) from the original $n$ bands in an $m$-class problem, the Bhattacharyya distance is calculated between each of the $m(m-1)/2$ pairs of classes for each of the possible ways of choosing $q$ features from $n$ dimensions. The best $q$ features are those dimensions whose sum of the Bhattacharyya distance between the $m(m-1)/2$ classes is highest (Haralick and



OFEK06894

**TABLE 8-4** Divergence Statistics for the Five Charleston, S.C., Land Cover Classes Evaluated Using 1, 2, 3, 4, and 5 Thematic Mapper Band Combinations at One Time

| Band combinations | Average divergence | Divergence[a] and Transformed Divergence[b] Class combinations 1 2 | 1 3 | 1 4 | 1 5 | 2 3 | 2 4 | 2 5 | 3 4 | 3 5 | 4 5 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | a. *One band at a time* | | | | | | |
| 1 | 1583 | 45[a] 1993[b] | 36 1977 | 23 1889 | 38 1982 | 600 2000 | 356 2000 | 803 2000 | 1 198 | 3 651 | 7 1145 |
| 2 | 1588 | 34 1970 | 67 2000 | 18 1786 | 54 1998 | 1036 2000 | 286 2000 | 1090 2000 | 1 246 | 5 988 | 5 890 |
| 3 | 1525 | 54 1998 | 107 2000 | 39 1985 | 160 2000 | 1591 2000 | 576 2000 | 2071 2000 | 1 286 | 3 642 | 1 339 |
| 4 | 1748 | 19 1809 | 47 1994 | 0 70 | 1238 2000 | 209 2000 | 13 1603 | 3357 2000 | 60 1999 | 210 2000 | 1466 2000 |
| 5 | 1636 | 4 779 | 26 1920 | 7 1194 | 2645 2000 | 77 2000 | 29 1947 | 5300 2000 | 2 523 | 556 2000 | 961 2000 |
| 6 | 1707 | 6 1061 | 61 1999 | 18 1795 | 345 2000 | 238 2000 | 74 2000 | 940 2000 | 1 213 | 63 1999 | 56 1998 |
| | | | | | b. *Two bands at a time* | | | | | | |
| 1 2 | 1709 | 51 1997 | 92 2000 | 26 1919 | 85 2000 | 1460 2000 | 410 2000 | 1752 2000 | 2 463 | 8 1256 | 10 1457 |
| 1 3 | 1709 | 56 1998 | 125 2000 | 40 1987 | 182 2000 | 1888 2000 | 589 2000 | 2564 2000 | 2 418 | 7 1196 | 11 1490 |
| 1 4 | 1996 | 55 1998 | 100 2000 | 32 1962 | 1251 2000 | 941 2000 | 446 2000 | 3799 2000 | 66 1999 | 219 2000 | 1525 2000 |
| 1 5 | 1896 | 54 1998 | 71 2000 | 28 1939 | 3072 2000 | 778 2000 | 497 2000 | 7838 2000 | 6 1029 | 585 2000 | 1038 2000 |
| 1 6 | 1852 | 52 1997 | 107 2000 | 28 1939 | 426 2000 | 944 2000 | 421 2000 | 2065 2000 | 3 586 | 63 1999 | 76 2000 |
| 2 3 | 1749 | 57 1998 | 140 2000 | 42 1990 | 170 2000 | 2099 2000 | 593 2000 | 2345 2000 | 2 524 | 13 1599 | 9 1382 |
| 2 4 | 1992 | 35 1976 | 103 2000 | 28 1941 | 1256 2000 | 1136 2000 | 356 2000 | 3985 2000 | 65 1999 | 228 2000 | 1529 2000 |
| 2 5 | 1856 | 35 1976 | 86 2000 | 20 1826 | 2795 2000 | 1068 2000 | 328 2000 | 6932 2000 | 4 760 | 560 2000 | 979 2000 |

1097

OFEK06895

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 6 | 37 | 111 | 24 | 423 | 1148 | 292 | 2192 | 2 | 69 | 66 |
| | 1980 | 2000 | 1902 | 2000 | 2000 | 2000 | 2000 | 405 | 2000 | 1999 |
| 3 4 | 101 | 124 | 61 | 1321 | 1606 | 905 | 4837 | 80 | 210 | 1487 |
| | 2000 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 3 5 | 59 | 114 | 45 | 3206 | 1609 | 740 | 9142 | 5 | 597 | 1024 |
| | 1999 | 2000 | 1992 | 2000 | 2000 | 2000 | 2000 | 964 | 2000 | 2000 |
| 3 6 | 63 | 131 | 41 | 525 | 1610 | 606 | 3122 | 2 | 65 | 59 |
| | 1999 | 2000 | 1989 | 2000 | 2000 | 2000 | 2000 | 469 | 1999 | 1999 |
| 4 5 | 21 | 52 | 11 | 4616 | 231 | 37 | 10376 | 98 | 889 | 2902 |
| | 1851 | 1997 | 1468 | 2000 | 2000 | 1981 | 2000 | 2000 | 2000 | 2000 |
| 4 6 | 20 | 76 | 21 | 1742 | 309 | 79 | 4740 | 86 | 285 | 1599 |
| | 1844 | 2000 | 1857 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 5 6 | 6 | 62 | 24 | 2870 | 246 | 97 | 5956 | 5 | 588 | 989 |
| | 1074 | 1999 | 1900 | 2000 | 2000 | 2000 | 2000 | 978 | 2000 | 2000 |

*c. Three bands at a time*

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 2 3 | 59 | 154 | 44 | 191 | 2340 | 613 | 2821 | 3 | 16 | 17 |
| | 1999 | 2000 | 1992 | 2000 | 2000 | 2000 | 2000 | 643 | 1745 | 1774 |
| 1 2 4 | 95 | 142 | 40 | 1266 | 1662 | 675 | 4381 | 68 | 236 | 1573 |
| | 2000 | 2000 | 1986 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 2 5 | 58 | 118 | 32 | 3201 | 1564 | 604 | 9281 | 7 | 589 | 1045 |
| | 1999 | 2000 | 1964 | 2000 | 2000 | 2000 | 2000 | 1129 | 2000 | 2000 |
| 1 2 6 | 57 | 146 | 30 | 493 | 1653 | 494 | 3176 | 4 | 69 | 80 |
| | 1998 | 2000 | 1953 | 2000 | 2000 | 2000 | 2000 | 732 | 2000 | 2000 |
| 1 3 4 | 117 | 150 | 64 | 1329 | 1905 | 985 | 5120 | 86 | 219 | 1534 |
| | 2000 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 3 5 | 60 | 137 | 51 | 3569 | 1902 | 863 | 11221 | 7 | 622 | 1088 |
| | 1999 | 2000 | 1997 | 2000 | 2000 | 2000 | 2000 | 1202 | 2000 | 2000 |
| 1 3 6 | 63 | 157 | 45 | 580 | 1935 | 669 | 3879 | 4 | 66 | 79 |
| | 1999 | 2000 | 1993 | 2000 | 2000 | 2000 | 2000 | 731 | 1999 | 2000 |
| 1 4 5 | 82 | 105 | 36 | 4923 | 978 | 635 | 12361 | 104 | 906 | 2955 |
| | 2000 | 2000 | 1979 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 4 6 | 82 | 129 | 37 | 1777 | 1055 | 610 | 5452 | 93 | 288 | 1669 |
| | 2000 | 2000 | 1980 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 5 6 | 56 | 109 | 37 | 3405 | 956 | 508 | 8948 | 8 | 627 | 1077 |
| | 1998 | 2000 | 1982 | 2000 | 2000 | 2000 | 2000 | 1261 | 2000 | 2000 |
| 2 3 4 | 117 | 156 | 63 | 1331 | 2119 | 956 | 4971 | 81 | 229 | 1530 |
| | 2000 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 2 3 5 | 62 | 147 | 47 | 3221 | 2120 | 749 | 9480 | 6 | 605 | 1034 |
| | 1999 | 2000 | 1994 | 2000 | 2000 | 2000 | 2000 | 1082 | 2000 | 2000 |



OFEK06896

**TABLE 8-4** Cont.

| Band combinations | Average divergence | Divergence[a] and Transformed Divergence[b] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Class combinations | | | | | | | | | |
| | | 1 2 | 1 3 | 1 4 | 1 5 | 2 5 | 2 4 | 2 5 | 3 4 | 3 5 | 4 5 |
| 2 3 6 | | 66 | 160 | 46 | 541 | 2113 | 617 | 3480 | 3 | 74 | 69 |
| | 1865 | 1999 | 2000 | 1994 | 2000 | 2000 | 2000 | 2000 | 661 | 2000 | 2000 |
| 2 4 5 | | 38 | 108 | 31 | 4674 | 1158 | 385 | 11402 | 103 | 896 | 2946 |
| | 1994 | 1984 | 2000 | 1956 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 2 4 6 | | 40 | 125 | 34 | 1771 | 1191 | 367 | 5511 | 90 | 300 | 1668 |
| | 1996 | 1986 | 2000 | 1970 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 2 5 6 | | 38 | 113 | 33 | 3050 | 1157 | 365 | 7757 | 7 | 594 | 1006 |
| | 1906 | 1982 | 2000 | 1968 | 2000 | 2000 | 2000 | 2000 | 1113 | 2000 | 2000 |
| 3 4 5 | | 106 | 129 | 65 | 5031 | 1622 | 1037 | 13505 | 120 | 914 | 2935 |
| | 2000 | 2000 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 3 4 6 | | 111 | 144 | 63 | 1841 | 1644 | 955 | 6309 | 102 | 285 | 1626 |
| | 2000 | 2000 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 3 5 6 | | 66 | 134 | 63 | 3453 | 1648 | 823 | 9900 | 8 | 631 | 1054 |
| | 1927 | 1999 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 1268 | 2000 | 2000 |
| 4 5 6 | | 22 | 83 | 26 | 5003 | 362 | 114 | 11477 | 105 | 944 | 2994 |
| | 1979 | 1870 | 2000 | 1923 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| | | *d. Four bands at a time* | | | | | | | | | |
| 1 2 3 4 | | 167 | 177 | 65 | 1339 | 2361 | 1151 | 5259 | 87 | 238 | 1575 |
| | 2000 | 2000 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 2 3 5 | | 63 | 165 | 54 | 3582 | 2355 | 876 | 11525 | 8 | 630 | 1095 |
| | 1929 | 1999 | 2000 | 1998 | 2000 | 2000 | 2000 | 2000 | 1294 | 2000 | 2000 |
| 1 2 3 6 | | 67 | 182 | 49 | 595 | 2369 | 683 | 4222 | 5 | 75 | 87 |
| | 1888 | 2000 | 2000 | 1996 | 2000 | 2000 | 2000 | 2000 | 885 | 2000 | 2000 |
| 1 2 4 5 | | 115 | 147 | 46 | 4971 | 1696 | 901 | 13287 | 108 | 913 | 2987 |
| | 1999 | 2000 | 2000 | 1994 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 2 4 6 | | 110 | 165 | 45 | 1801 | 1731 | 868 | 6161 | 96 | 303 | 1725 |
| | 1999 | 2000 | 2000 | 1993 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |

1099

OFEK06897

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 2 5 6 | 61 | 148 | 41 | 3564 | 1665 | 614 | 10579 | 9 | 633 | 1085 |
| 1932 | 1999 | 2000 | 1989 | 2000 | 2000 | 2000 | 2000 | 1331 | 2000 | 2000 |
| 1 3 4 5 | 133 | 156 | 74 | 5293 | 1931 | 1283 | 15187 | 127 | 928 | 2976 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 3 4 6 | 134 | 172 | 69 | 1863 | 1955 | 1184 | 6814 | 110 | 289 | 1682 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 3 5 6 | 66 | 159 | 66 | 3919 | 1954 | 901 | 12411 | 10 | 665 | 1129 |
| 1940 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 1397 | 2000 | 2000 |
| 1 4 5 6 | 88 | 135 | 42 | 5422 | 1105 | 659 | 13950 | 112 | 970 | 3068 |
| 1999 | 2000 | 2000 | 1990 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 2 3 4 5 | 122 | 161 | 67 | 5040 | 2133 | 1093 | 13663 | 121 | 933 | 2981 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 2 3 4 6 | 132 | 173 | 65 | 1848 | 2143 | 1023 | 6509 | 103 | 302 | 1670 |
| 2000 | 2000 | 2000 | 1999 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 2 3 5 6 | 69 | 163 | 68 | 3476 | 2144 | 837 | 10308 | 9 | 639 | 1062 |
| 1937 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 1370 | 2000 | 2000 |
| 2 4 5 6 | 41 | 131 | 38 | 5079 | 1229 | 397 | 12641 | 110 | 951 | 3037 |
| 1997 | 1987 | 2000 | 1983 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 3 4 5 6 | 112 | 148 | 74 | 5436 | 1665 | 1066 | 14688 | 125 | 971 | 3030 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| | | | e. *Five bands at a time* | | | | | | | |
| 1 2 3 4 5 | 176 | 183 | 75 | 5302 | 2384 | 1422 | 15334 | 128 | 947 | 3019 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 2 3 4 6 | 176 | 196 | 71 | 1871 | 2393 | 1316 | 7015 | 111 | 305 | 1726 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 2 3 5 6 | 70 | 184 | 72 | 3940 | 2386 | 919 | 12798 | 11 | 673 | 1135 |
| 1948 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 1479 | 2000 | 2000 |
| 1 2 4 5 6 | 117 | 171 | 50 | 5487 | 1770 | 920 | 15021 | 115 | 977 | 3101 |
| 2000 | 2000 | 2000 | 1996 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 1 3 4 5 6 | 138 | 176 | 80 | 5803 | 1979 | 1294 | 16829 | 132 | 994 | 3089 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |
| 2 3 4 5 6 | 134 | 177 | 77 | 5443 | 2161 | 1130 | 14893 | 126 | 987 | 3072 |
| 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 | 2000 |

Class numbers: 1, residential; 2, commercial; 3, wetland; 4, forest; 5, water.



1100

OFEK06898

Fu, 1983). Kalayeh et al. (1983) describe another statistical method of feature selection based on the two-class problem.

### *Select the Appropriate Classification Algorithm*

Various supervised classification methods may be used to assign an unknown pixel to one of a number of classes. The choice of a particular classifier or decision rule depends on the nature of the input data and the desired output (Friedman, 1980). *Parametric* classification algorithms assume that the observed measurement vectors $X_c$ obtained for each class in each spectral band during the training phase of the supervised classification are Gaussian in nature (i.e., they are normally distributed). *Nonparametric* classification algorithms make no such assumption. It is instructive to review the logic of several of the classifiers. Among the most frequently used classification algorithms are the parallelepiped, minimum distance, and maximum likelihood decision rules.

**The parallelepiped classification algorithm.** This is a widely used decision rule (Addington, 1975). It is based on simple Boolean "and/or" logic. Training data in $n$ spectral bands are used in performing the classification. Brightness values from each pixel of the multispectral imagery are used to produce an $n$-dimensional mean vector, $M_c = (\mu_{ck}, \mu_{c2}, \mu_{c3}, \ldots, \mu_{cn})$, with $\mu_{ck}$ being the mean value of the training data obtained for class $c$ in band $k$ out of $m$ possible classes as previously defined. $s_{ck}$ is the standard deviation of the training data class $c$ of band $k$ out of $m$ possible classes. In this discussion we evaluate all five Charleston classes using just bands 4 and 5 of the training data.

Using a one-standard deviation threshold (as shown in Fig. 8–13), a parallelepiped algorithm decides $BV_{ijk}$ is in class $c$ if, and only if,

$$\mu_{ck} - s_{ck} \leq BV_{ijk} \leq \mu_{ck} + s_{ck} \tag{8-12}$$

where $c = 1, 2, 3, \ldots, m$, number of classes
$k = 1, 2, 3, \ldots, n$, number of bands

Therefore, if the low and high decision boundaries are defined as

$$\text{Low}_{ck} = \mu_{ck} - s_{ck} \tag{8-13}$$

and

$$\text{High}_{ck} = \mu_{ck} + s_{ck} \tag{8-14}$$

the parallelepiped algorithm becomes

$$\text{Low}_{ck} \leq BV_{ijk} \leq \text{High}_{ck} \tag{8-15}$$

These decision boundaries form an $n$-dimensional parallelepiped in feature space. If the pixel value lies above the lower threshold and below the high threshold for all $n$ bands evaluated, it is assigned to that class (see point $a$ in Fig. 8–13). When an unknown pixel does not satisfy any of the Boolean logic criteria (point $b$ in Fig. 8–


OFEK06899

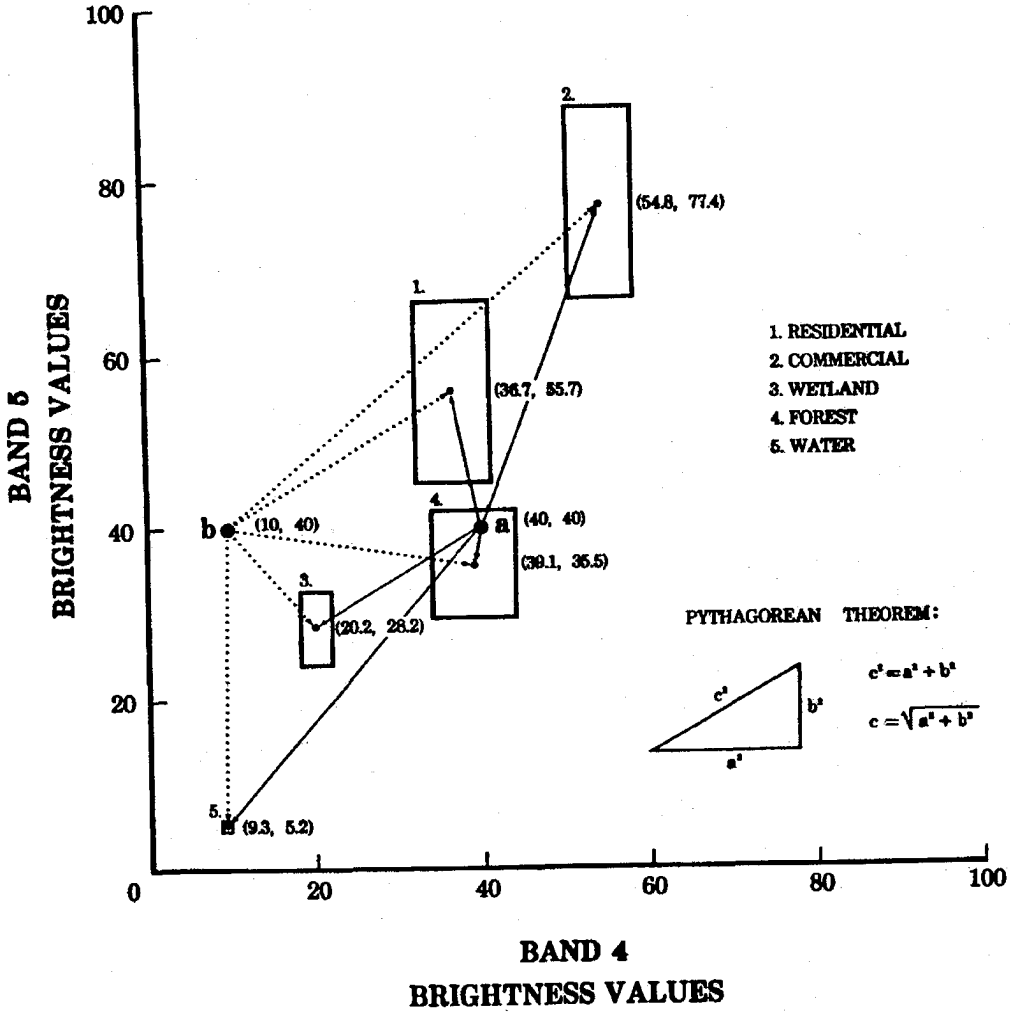



**Figure 8–13** Points *a* and *b* are pixels in the image to be classified. Pixel *a* has a brightness value of 40 in band 4 and 40 in band 5. Pixel *b* has a brightness value of 10 in band 4 and 40 in band 5. The boxes represent the parallelepiped decision rule associated with a ±1 standard deviation classification. The vectors (arrows) represent the distance from *a* and *b* to the mean of all classes in a minimum distance to means classification algorithm. Refer to Table 8–5 and Table 8–6 for the results of classifying points *a* and *b* using both classification techniques.

13), it is assigned to an unclassified category. Although it is only possible to analyze visually up to three dimensions as described in the section on computer graphic feature analysis, it is possible to create an *n*-dimensional parallelepiped for classification purposes.

We will review how unknown pixels *a* and *b* are assigned to the forest and unclassified categories in Fig. 8–13. The computations are summarized in Table 8–5. First, the standard deviation is subtracted and added to the mean of each class and for each band to identify the lower ($L_{ck}$) and upper ($H_{ck}$) edge of the parallelepiped. In this case only two bands are used, 4 and 5, resulting in a two-dimensional box.



OFEK06900

**TABLE 8–5** Example of Parallelepiped Classification Logic for Pixels *a* and *b* in Fig. 8–13

| Class | Lower threshold, $L_{ck}$ | Upper threshold, $U_{ck}$ | Does pixel $a(40, 40)$ satisfy criteria for this class in this band? $L_{ck} \le a \le U_{ck}$ | Does pixel $b(10, 40)$ satisfy criteria for this class in this band? $L_{ck} \le b \le U_{ck}$ |
|---|---|---|---|---|
| 1. Residential | | | | |
| Band 4 | $36.7 - 4.53 =$ **31.27** | $36.7 + 4.53 =$ **41.23** | Yes | No |
| Band 5 | $55.7 - 10.72 =$ **44.98** | $55.7 + 10.72 =$ **67.42** | No | No |
| 2. Commercial | | | | |
| Band 4 | $54.8 - 3.88 =$ **50.92** | $54.8 + 3.88 =$ **58.68** | No | No |
| Band 5 | $77.4 - 11.16 =$ **75.4** | $77.4 + 11.6 =$ **88.56** | No | No |
| 3. Wetland | | | | |
| Band 4 | $20.2 - 1.88 =$ **18.32** | $20.2 + 1.88 =$ **22.08** | No | No |
| Band 5 | $28.2 - 4.31 =$ **23.89** | $28.2 + 4.31 =$ **32.51** | No | No |
| 4. Forest | | | | |
| Band 4 | $39.1 - 5.11 =$ **33.99** | $39.1 + 5.11 =$ **44.21** | Yes | No |
| Band 5 | $35.5 - 6.41 =$ **29.09** | $35.5 + 6.41 =$ **41.91** | Yes, assign pixel to class 4, forest STOP | No |
| 5. Water | | | | |
| Band 4 | $9.3 - 0.56 =$ **8.74** | $9.3 + 0.56 =$ **9.86** | — | No |
| Band 5 | $5.2 - 0.71 =$ **4.49** | $5.2 + 0.71 =$ **5.91** | — | No, assign pixel to unclassified category STOP |

This, of course, could be extended to $n$ dimensions or bands. With the lower and upper thresholds for each box identified, it is possible to determine if the brightness value of an input pixel in each band, $k$, satisfies the criteria of any of the five parallelepipeds. For example, pixel $a$ has a value of 40 in both band 4 and band 5. It satisfies the band 4 criteria of class 1 (i.e., $31.27 \le 40 \le 41.23$) but does not satisfy the band 5 criteria. Therefore, the process continues by evaluating the parallelepiped criteria of classes 2 and 3, which are also not satisfied. However, when the brightness values of $a$ are compared with class 4 thresholds, we find that it satisfies the criteria



OFEK06901

for band 4 (i.e., $33.99 \leq 40 \leq 44.21$) and band 5 ($29.09 \leq 40 \leq 41.91$). Thus the pixel is assigned to class 4, forest.

This same logic is applied to classify unknown pixel *b*. Unfortunately, its brightness value of 10 in band 4 and 40 in band 5 never fall within the thresholds of any of the parallelepipeds. Therefore, it is assigned to an unclassified category. Increasing the size of the thresholds to $\pm$ 2 or 3 standard deviations would increase the size of the parallelepipeds. This might result in point *b* being assigned to one of the classes. However, this same action might also introduce a significant amount of overlap among many of the parallelepipeds, resulting in classification error. Perhaps point *b* really belongs to a class that has not been trained upon (e.g., dredge spoil).

The parallelepiped algorithm is a computationally efficient method of classifying remote sensor data. Unfortunately, because some parallelepipeds overlap, it is possible that an unknown candidate pixel might satisfy the criteria of more than one class. In such cases it is usually assigned to the first class for which it meets all criteria. A more elegant solution is to take the pixel that can be assigned to more than one class and use a minimum distance to means decision rule to assign it to just one class.

**The minimum-distance to means classification algorithm.** This decision rule is is computationally simple and commonly used. It can result in classification accuracies comparable to other more computationally intensive algorithms, such as the maximum likelihood algorithm (Hixson et al., 1980). Like the parallelepiped algorithm, it requires that the user provide the mean vectors for each class in each band $\mu_{ck}$, from the training data. To perform a minimum distance classification, a program must calculate the distance to each mean vector, $\mu_{ck}$, from each unknown pixel ($BV_{ijk}$). It is possible to calculate this distance using Euclidian distance based on the pythagorean theorem or "around the block" distance measures (Swain and Davis, 1978; Fig. 8–14). In this discussion we demonstrate the method of minimum distance classification using Euclidian distance measurements applied to the two unknown points (*a* and *b*) shown in Fig. 8–13.

The computation of the Euclidian distance from point *a* (40, 40) to the mean of class 1 (36.7, 55.7) measured in bands 4 and 5 relies on the equation:

$$\text{Dist} = \sqrt{(BV_{ijk} - \mu_{ck})^2 + (BV_{ijl} - \mu_{cl})^2} \tag{8-16}$$

where $\mu_{ck}$ and $\mu_{cl}$ represent the mean vectors for class *c* measured in bands *k* and *l*. In our example this would be:

$$\text{Dist}_{a \text{ to class } 1} = \sqrt{(BV_{ij\,4} - \mu_{1,4})^2 + (BV_{ij\,5} - \mu_{1,5})^2} \tag{8-17}$$

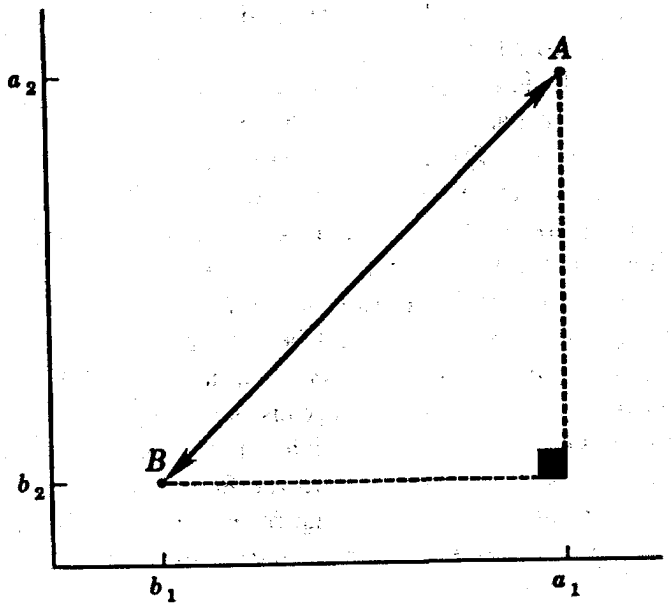
To compute the distance from point *a* to the mean of class 2 in these same two bands would be:

$$\text{Dist}_{a \text{ to class } 2} = \sqrt{(BV_{ij\,4} - \mu_{2,4})^2 + (BV_{ij\,5} - \mu_{2,5})^2} \tag{8-18}$$

Notice that the subscript that stands for class, *c*, is incremented from 1 to 2. By calculating the Euclidian distance from point *a* to the mean of all five classes it is possible to determine which distance is shortest. Table 8–6 has a complete listing



OFEK06902




EUCLIDEAN DISTANCE:

$$D_{AB} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}$$

"ROUND THE BLOCK" DISTANCE:

$$D_{AB} = \sum_{i=1}^{n} |a_i - b_i|$$

**Figure 8–14** The distance used in a minimum distance to means classification algorithm can take two forms: the "Euclidian" distance based on the Pythagorean theorem and the "round the block" distance. The Euclidian distance is more computationally intensive. (Adapted from Swain and Davis, 1978 © 1978 McGraw-Hill Book Co. Used with permission.)

**TABLE 8–6** Example of Minimum Distance to Means Classification Logic for Pixels *a* and *b* in Fig. 8–13

| Class | Distance from pixel *a* (40, 40) to the mean of each class | Distance from pixel *b* (10, 40) to the mean of each class |
|---|---|---|
| 1. Residential | $\sqrt{(40-36.7)^2+(40-55.7)^2} = 16.04$ | $\sqrt{(10-36.7)^2+(40-55.7)^2} = 30.97$ |
| 2. Commercial | $\sqrt{(40-54.8)^2+(40-77.4)^2} = 40.22$ | $\sqrt{(10-54.8)^2+(40-77.4)^2} = 58.35$ |
| 3. Wetland | $\sqrt{(40-20.2)^2+(40-28.2)^2} = 23.04$ | $\sqrt{(10-20.2)^2+(40-28.2)^2} = 15.75$ Assign pixel *b* to this class; it has the minimum distance |
| 4. Forest | $\sqrt{(40-39.1)^2+(40-35.5)^2} = 4.59$ Assign pixel *a* to this class; it has the minimum distance | $\sqrt{(10-39.1)^2+(40-35.5)^2} = 29.45$ |
| 5. Water | $\sqrt{(40-9.3)^2+(40-5.2)^2} = 46.4$ | $\sqrt{(10-9.3)^2+(40-5.2)^2} = 34.8$ |



OFEK06903

of the mathematics associated with the computation of distances for the five land cover classes. It reveals that pixel *a* should be assigned to class 4 (forest) because it obtained the minimum distance of 4.59. The same logic can be applied to evaluating the unknown pixel *b*. It is assigned to class 3 (wetland) because it obtained the minimum distance of 15.75. It should be obvious that any unknown pixel will definitely be assigned to one of the five training classes using this algorithm. There will be no unclassified pixels.

Many minimum-distance algorithms let the analyst specify a distance or threshold from the class means beyond which a pixel will not be assigned to a category even though it is nearest to the mean of that category. For example, if a threshold of 10.0 was specified, point *a* would still be classified as class 4 (forest) because it had a minimum distance of 4.59 which was below the threshold. Conversely, point *b* would not be assigned to class 3 (wetland) because its minimum distance of 15.75 was greater than the 10.0 threshold. Instead, point *b* would be assigned to an unclassified category.

When more than two bands are evaluated in a classification, it is possible to extend the logic of computing the distance between just two points in *n*-space using the equation

$$D_{AB} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{8-19}$$

Figure 8–14 demonstrates how this algorithm is implemented.

Program CLASSIFY in Appendix A performs minimum-distance to means classification for up to 25 classes at one time using as many as six bands of remote sensor data. The program was run on the Charleston, S.C., thematic mapper data set using the training data described previously. The results are displayed as a color-coded thematic map in Fig. 8–15 (color section). The total number of pixels in each class are found in Table 8–7. Error associated with the classification is summarized in Table 8–8. Classification error is discussed later in this chapter in a section on land-use classification accuracy assessment. Program CLASSIFY will store the classification map matrix which may then be plotted on an electrostatic plotter using the BRIGHTNESS MAP program. A sample of the classified study area is shown in Fig. 8–16.

**TABLE 8–7** Total Number of Pixels Classified into Each of the Five Charleston, S.C., Land-Cover Classes Shown in Fig. 8–15

| Class | Total number of pixels |
|---|---|
| 1. Residential | 14,398 |
| 2. Commercial | 4,088 |
| 3. Wetland | 10,772 |
| 4. Forest | 11,673 |
| 5. Water | 20,509 |


OFEK06904

**TABLE 8-8** Evaluation of the Charleston, S.C., Thematic Mapper Test Data

a. Error Matrix

| Actual land-use class | Interpreted land use 1 | 2 | 3 | 4 | 5 | Total | Ommission error (%) | Commission error (%) | Accuracy[a] (%) |
|---|---|---|---|---|---|---|---|---|---|
| 1. Residential | 70 | 5 | 0 | 13 | 0 | 88 | 20.5 | 4.5 | 79.5 |
| 2. Commercial | 3 | 55 | 0 | 0 | 0 | 58 | 5.2 | 8.6 | 94.8 |
| 3. Wetland | 0 | 0 | 99 | 0 | 0 | 99 | 0.0 | 4.0 | 100.0 |
| 4. Forest | 1 | 0 | 4 | 37 | 0 | 42 | 11.9 | 30.9 | 88.0 |
| 5. Water | 0 | 0 | 0 | 0 | 121 | 121 | 0.0 | 0.0 | 100.0 |
| Total | 74 | 60 | 103 | 50 | 121 | 408 | | | |

b. Derivation of Omission and Commission Errors

| Land-use class | Omission errors Total | % | Commission errors Total | % | Correct Total | % |
|---|---|---|---|---|---|---|
| 1. Residential | 18/88 | 20.5 | 4/88 | 4.5 | 70/88 | 79.5 |
| 2. Commercial | 3/58 | 5.2 | 5/58 | 8.6 | 55/58 | 94.8 |
| 3. Wetland | 0/99 | 0.0 | 4/99 | 4.0 | 99/99 | 100.0 |
| 4. Forest | 5/42 | 11.9 | 13/42 | 30.9 | 37/42 | 88.0 |
| 5. Water | 0/121 | 0.0 | 0/121 | 0.0 | 121/121 | 100.0 |

[a] Overall accuracy = 382/408 pixels = 93.6%.

**The maximum likelihood classification algorithm.** This decision rule assigns each pixel having pattern measurements or features $X$ to the class $c$ whose units are most probable or likely to have given rise to feature vector $X$. It assumes that training data statistics for each class in each band are normally distributed (i.e., Gaussian in nature). In other words, training data with bi- or trimodal histograms in a single band are not ideal. In such cases the individual modes probably represent individual classes that should be trained upon individually and labeled as separate classes. This would then produce unimodal, Gaussian training class statistics that would fulfill the normal distribution requirement.

Maximum likelihood classification makes use of the statistics already computed and discussed in previous sections, including the mean measurement vector, $M_c$, for each class and the covariance matrix of class $c$ for bands $k$ through $l$, $V_c$. The decision rule applied to the unknown measurement vector, $X$, is (Anuta, 1977): Decide $X$ is in class $c$ if, and only if,

$$p_c \geq p_i, \text{ where } i = 1, 2, 3, \ldots, m \text{ possible classes} \tag{8-20}$$

and

$$p_c = [-0.5 \log_e (\det(V_c))] - [0.5(X - M_c)^T(V_c^{-1})(X - M_c)] \tag{8-21}$$



OFEK06905

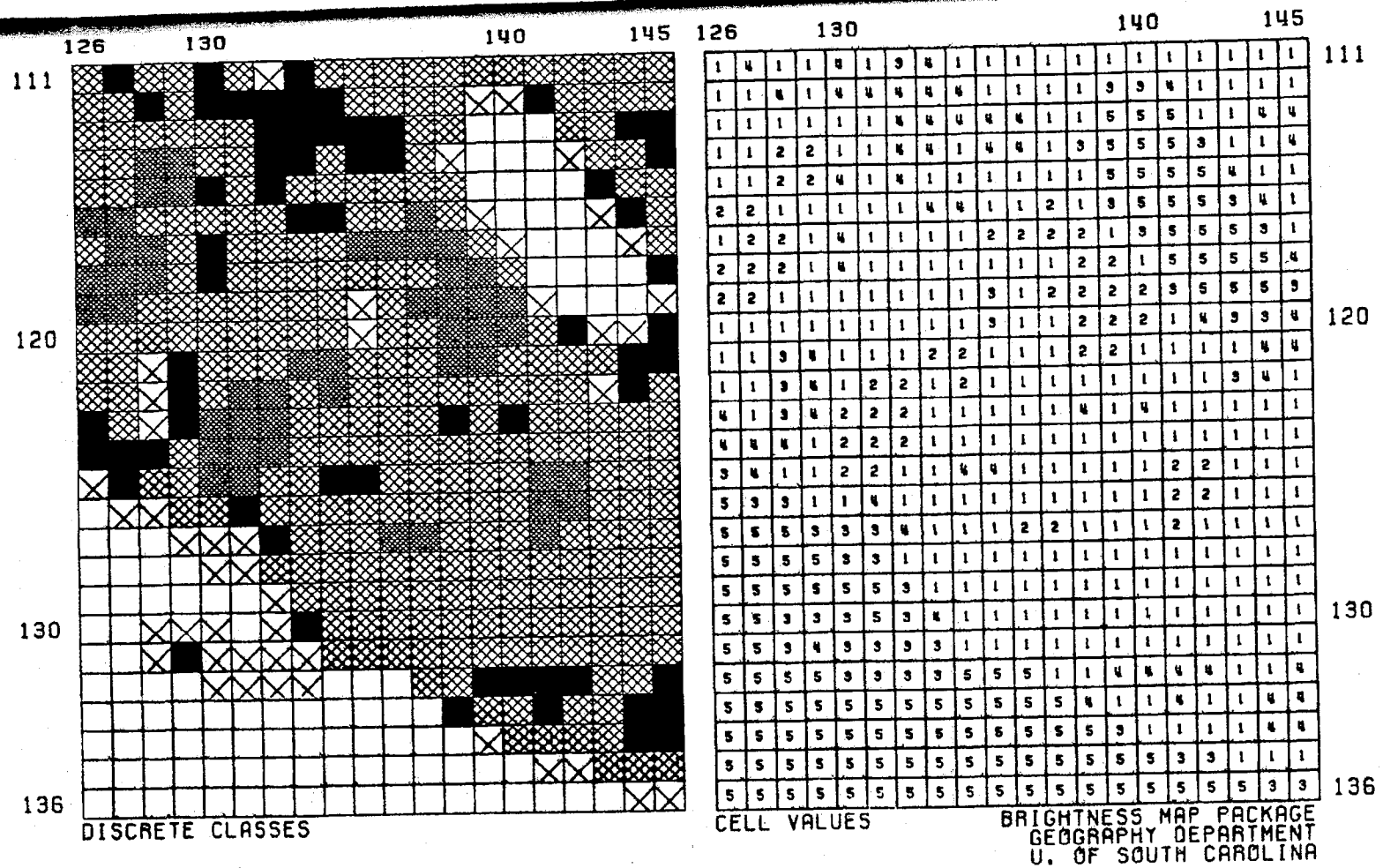



**Figure 8–16** Output from the **CLASSIFY** program for a small portion of the Charleston, S.C., supervised classification. The thematic map is on the left with the cell values for the discrete classes on the right.



1108
OFEK06906

and det ($V_c$) is the determinant of the covariance matrix $V_c$. Therefore, to classify the measurement vector $X$ of an unknown pixel into a class, the maximum likelihood decision rule computes the value $p_c$ for each class. Then it assigns the pixel to the class that has the largest (or maximum) value.

Now let us consider the computations required. In the first pass, $p_1$ is computed with $V_1$ and $M_1$ being the covariance matrix and mean vectors for class 1. Next, $p_2$ is computed using $V_2$ and $M_2$. This would continue for all $m$ classes. The pixel or measurement vector $X$ is assigned to the class that produces the largest or maximum $p_c$. The measurement vector $X$ used in each step of the calculation consists of $n$ elements (the number of bands being analyzed). For example, if all six bands were being analyzed, each unknown pixel would have a measurement vector $X$, of

$$X = \begin{bmatrix} BV_{(i,j,1)} \\ BV_{(i,j,2)} \\ BV_{(i,j,3)} \\ BV_{(i,j,4)} \\ BV_{(i,j,5)} \\ BV_{(i,j,6)} \end{bmatrix} \tag{8-22}$$

Equation (8-21) assumes that each class has an equal probability of occurring in the terrain. Common sense reminds us that in most remote sensing applications there is a high probability of encountering some classes more often than others. For example, in the Charleston scene the probability of encountering residential land use is approximately 20% (i.e., 0.2), commercial (0.1), wetland (0.3), forest (0.1), and water (0.3). Thus, we would expect more pixels to be classified as water simply because it is more prevelant in the terrain. It is possible to include this valuable a priori (prior knowledge) information in the classification decision. We can do this by weighting each class, $c$, by its appropriate a priori probability, $a_c$. The equation then becomes: Decide $X$ is in class $c$ if, and only if,

$$p_c(a_c) \geq p_i(a_c) \qquad \text{where } i = 1, 2, 3, \ldots, m \text{ possible classes} \tag{8-23}$$

and

$$p_c(a_c) = \log_e(a_c) - [0.5 \log_e(\det(V_c))] - [0.5(X - M_c)^T(V_c^{-1})(X - M_c)] \tag{8-24}$$

This Bayes' decision rule is identical to the maximum likelihood decision rule except that it does not assume that each of the classes is equally probable (Hord, 1982). A priori probabilities have been used successfully as a way of incorporating the effects of relief and other terrain characteristics in improving classification accuracy (Strahler 1980). Haralick and Fu (1983) provide an in-depth discussion of the probabilities and mathematics of the maximum likelihood and Bayes decision rules. Obviously the maximum likelihood and Bayes' classifications require many more computation per pixel than either the parallelepiped or minimum-distance classification algorithms. They do not always produce superior results.



OFEK06907

## UNSUPERVISED CLASSIFICATION

In contrast to supervised classification, *unsupervised classification* requires only a minimal amount of initial input from the analyst. It is a process whereby numerical operations are performed that search for "natural" groupings of the spectral properties of pixels, as examined in multispectral feature space. The user allows the computer to select the class means and covariance matrices to be used in the classification. Once the data are classified, the analyst attempts, a posteriori (after the fact) to assign these "natural" or *spectral* classes to the *information* classes of interest (Robinove, 1981). This may not be easy. Some of the clusters may be meaningless because they represent mixed classes of earth surface materials. It takes interaction by the analyst to unravel such mysteries. Hopefully, the analyst understands the spectral characteristics of the terrain well enough to label certain clusters as representing information classes.

Hundreds of methods of clustering have been developed for a wide variety of purposes apart from pattern recognition in remote sensing. For example, Sokal and Sneath (1963) developed clustering procedures for assessing plant taxonomy, and Berry (1967) applied clustering to geographic regionalization. Clustering algorithms used for the unsupervised classification of remotely sensed data generally vary according to the efficiency with which the clustering takes place. Different criteria of efficiency lead to different approaches (Duda and Hart, 1973; Hartigan, 1975; Goldberg and Shlien, 1976; Bryant, 1979; Haralick and Fu, 1983). An example of a conceptually simple but not necessarily efficient clustering algorithm will be used to demonstrate the fundamental logic of unsupervised classification.

The clustering algorithm discussed here operates in a two-pass mode (i.e., it passes through the registered multispectral data set two times). In the first pass, the program reads through the data set and sequentially builds clusters (groups of points in spectral space). There is a mean vector associated with each cluster. In the second pass, a minimum-distance classification to means algorithm similar to the one described previously is applied to the whole data set on a pixel-by-pixel basis, where each pixel is assigned to one of the mean vectors created in pass 1. The first pass, therefore, automatically creates the cluster signatures to be used by the classifier.

### Pass 1: Cluster Building

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

1. $R$, a radius in spectral space used to determine when a new cluster should be formed (e.g., 15 units)
2. $C$, a spectral space distance parameter used when merging clusters (e.g., 30 units)



OFEK06908

3. $N$, the number of pixels to be evaluated between each merging of the clusters (e.g., 2000 pixels)
4. $C_{max}$, the maximum number of clusters to be identified by the algorithm (e.g., 20 clusters)

These can be set to default values if no initial human interaction is desired.

Starting at the origin of the multispectral data set (i.e., line 1, column 1), pixels are evaluated sequentially from left to right. After one line is processed, the next line of data is considered. We will only analyze the clustering of the first three pixels in a hypothetical image and label them pixel 1, pixel 2, and pixel 3. The pixels have brightness values in just two bands, 4 and 5. Their spatial relationships in two-dimensional spectral space are shown in Fig. 8–17.

First, we let the brightness values associated with pixel 1 in the image represent the mean data vector of cluster 1 (i.e., $M_1 = \{10,10\}$). Remember, it is an $n$-dimensional mean data vector with $n$ being the number of bands used in the unsupervised classification. In our example, just two bands are being used, so $n = 2$. Because we have not identified all 20 spectral clusters ($N_{max}$) as yet, pixel 2 will be considered as the mean data vector of cluster 2 (i.e., $M_2 = \{20, 20\}$). If the spectral distance ($D$) between cluster 2 to cluster 1 is greater than $R$, cluster 2 will remain as cluster 2. However, if the spectral distance ($D$) is less than $R$, the mean data vector of cluster 1 becomes the average of the first and second pixel brightness values and the weight (or count) of cluster 1 becomes 2 (Fig. 8–18). In our example, the distance ($D$) between cluster 1 (actually, pixel 1) and pixel 2 is 14.14. If the radius, $R$, was initially set at 15.0,

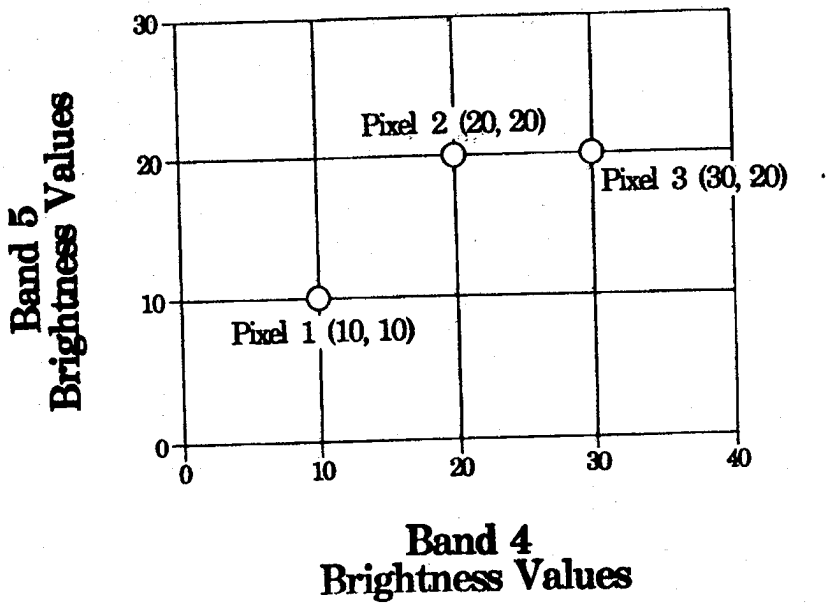



Figure 8–17 Original values of pixels 1 to 3 as measured in bands 4 and 5 of the hypothetical remotely sensed data.



OFEK06909

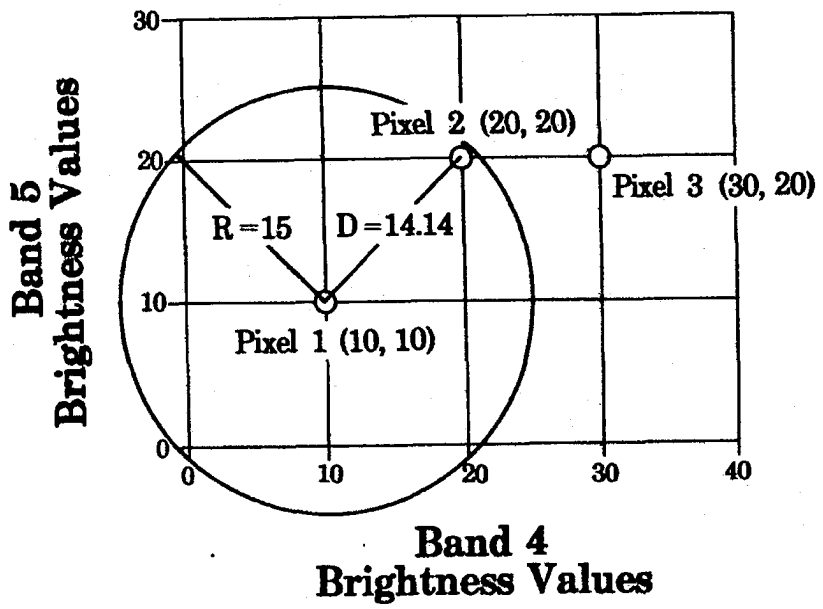



Figure 8–18 The distance ($D$) in two-dimensional spectral space between pixel 1 (cluster 1) and pixel 2 (cluster 2) in the first iteration is computed and tested against the value of $R$, the minimum acceptable radius. In this case, $D$ does not exceed $R$; therefore, we merge clusters 1 and 2 as shown in Fig. 8–19.

pixel 2 does not satisfy the criteria for being cluster 2 because its distance from cluster 1 is $< 15$. Therefore, the mean data vectors of cluster 1 and pixel 2 are averaged, yielding the new location of cluster 1 at $M_1 = \{15, 15\}$, as shown in Fig. 8–19. The spectral distance ($D$) is computed using the Pythagorean theorem as discussed previously.

Next, pixel 3 is considered as the mean data vector of cluster 2 (i.e., $M_2 = \{30, 20\}$). The distance from pixel 3 to the revised location of cluster 1, $M_1 = \{15, 15\}$, is 15.81 (Fig. 8–19). Because it is $> 15$, the mean data vector of pixel 3 becomes the mean data vector of cluster 2.

This cluster accumulation continues until the number of pixels evaluated is greater than $N$. At that point, the program stops evaluating individual pixels and looks closely at the nature of the clusters obtained thus far. It calculates the distance between each cluster and every other cluster. Any two clusters separated by a spectral distance less than $C$ are merged. Such a new cluster mean vector is the weighted average of the two original clusters and the weight is the sum of the two individual weights. This proceeds until there are no clusters with a separation distance of less than $C$. Then, the next pixel is considered. This process continues to iterate until the entire multispectral data set is examined.

Schowengerdt (1983) suggests that virtually all of the commonly used clustering algorithms use iterative calculations to find an optimum set of decision boundaries for the data set. It should be noted that some clustering algorithms allow the analyst to initially "seed" the mean vector for several of the important classes. The seed


OFEK06910

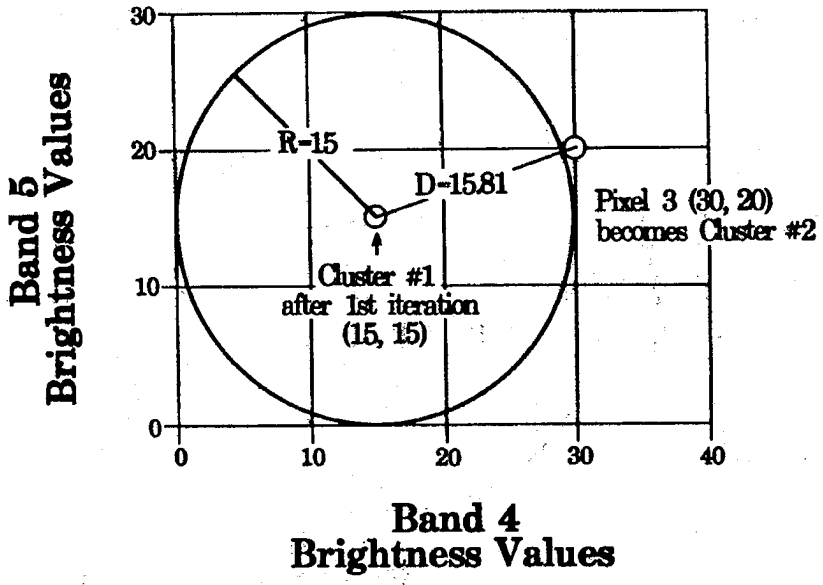



**Figure 8–19.** Pixels 1 and 2 now represent cluster 1. Note that the location of cluster 1 has migrated from 10, 10 to 15, 15 after the first iteration. Now, pixel 3 distance ($D$) is computed to see if it is greater than the minimum threshold, $R$. If is, so pixel location 3 becomes cluster 2. This process continues until all 20 clusters are identified. Then the 20 clusters are evaluated using a distance measure, $C$ (not shown), to merge the clusters that are closest to one another.

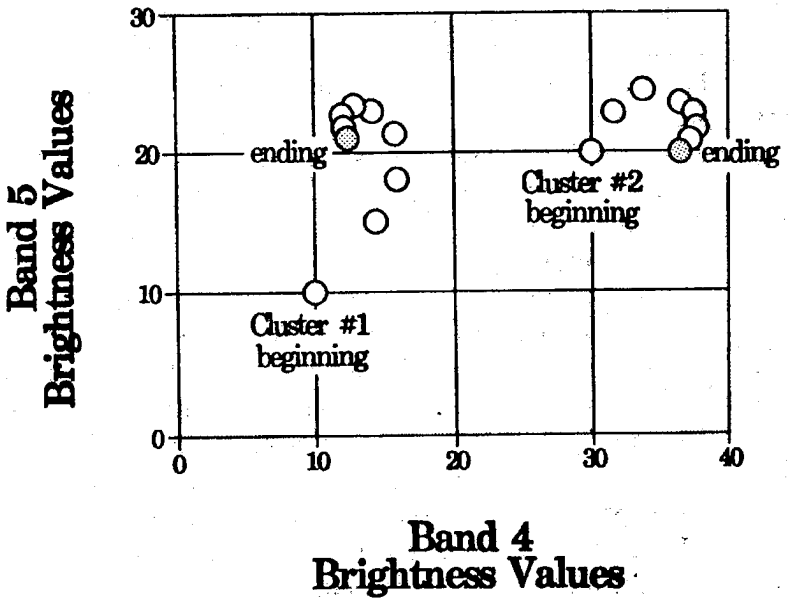



**Figure 8–20** This illustration demonstrates how the clusters migrate during the several iterations of a clustering algorithm. The final ending point represents the mean vector that would be used in phase 2 of the clustering process when the minimum distance classification is performed.



OFEK06911

data are usually obtained in a supervised fashion, as discussed previously. Others allow the analyst to utilize a priori information to direct the clustering process (Wharton and Turner, 1981).

Some programs do not evaluate every line and every column of the data when computing the mean vectors for the clusters. Instead, they may sample every $i$th row and $j$th column to identify the $C_{max}$ clusters. The decision to sample the data is usually left up to the analyst (Justice and Townshend, 1982). If computer resources are abundant, every pixel may be sampled. If resources are scarce, acceptable results can usually be obtained by sampling the data. Obviously, there are a tremendous number of computations being performed during this initial pass through the data set.

A hypothetical diagram showing the cluster migration for our two band data set is shown in Fig. 8–20. Note that as more points are added to a cluster, the mean shifts less dramatically since the new mean computed is weighted by the number of pixels currently in a cluster. The ending point is the spectral location of the final mean vector that is used as a signature in the minimum distance classifier applied in pass 2.

### *Pass 2: Assignment of Pixels to One of the $C_{max}$ Clusters Using Minimum-Distance Classification Logic*

The final cluster mean data vectors are used in a minimum-distance to means classification algorithm to classify all the pixels in the image into one of the $C_{max}$ clusters. The analyst usually produces a display depicting to which cluster each pixel was assigned. It is then necessary to evaluate the location of the clusters in the image, label them if possible, and see if any should be combined. It is usually necessary to combine some clusters. This is where an intimate knowledge of the terrain is critical.

An unsupervised classification of the Charleston, S.C., TM scene is displayed in Fig. 8–21 (color section). It was created using only TM bands 2, 3, and 4. The analyst stipulated that a total of 20 clusters ($C_{max}$) should be extracted from the data. The mean data vectors for each of the final 20 clusters are summarized in Table 8–9. These mean vectors represented the data used in the minimum-distance classification of every pixel in the scene into one of the 20 cluster categories.

Scatterplots of the mean data vectors for each of the 20 clusters using just bands 2 and 3 and bands 3 and 4 are displayed in Figs. 8–22 and 8–23, respectively. In the band 2–band 3 scatterplot, the 20 clusters lie on a diagonal extending from the origin of the plot. Unfortunately, the water cluster was located in the same spectral space as forest and wetland when using just bands 2 and 3. Therefore, this scatterplot was not used to *label* or assign the clusters to *information* classes. Conversely, the scatterplot of bands 3 and 4 mean data vectors is relatively easy to interpret and looks very much like the perpendicular vegetation index distribution shown earlier in Fig. 7–41. This is not surprising since this is a red (band 3) versus near-infrared (band 4) plot.

Cluster labeling is usually performed by interactively displaying all the pixels



OFEK06912

**TABLE 8–9** Results of Clustering on Thematic Mapper Bands 2, 3, and 4 of the Charleston, S.C., Scene

| Cluster | Percent of scene | Mean vector | | | Class description | Color assignment |
|---|---|---|---|---|---|---|
| | | Band 2 | Band 3 | Band 4 | | |
| 1 | 24.15 | 23.14 | 18.75 | 9.35 | Water | Dark blue |
| 2 | 7.14 | 21.89 | 18.99 | 44.85 | Forest 1 | Dark green |
| 3 | 7.00 | 22.13 | 19.72 | 38.17 | Forest 2 | Dark green |
| 4 | 11.61 | 21.79 | 19.87 | 19.46 | Wetland 1 | Bright green |
| 5 | 5.83 | 22.16 | 20.51 | 23.90 | Wetland 2 | Green |
| 6 | 2.18 | 28.35 | 28.48 | 40.67 | Residential 1 | Bright yellow |
| 7 | 3.34 | 36.30 | 25.58 | 35.00 | Residential 2 | Bright yellow |
| 8 | 2.60 | 29.44 | 29.87 | 49.49 | Parks, golf | Gray |
| 9 | 1.72 | 32.69 | 34.70 | 41.38 | Residential 3 | Yellow |
| 10 | 1.85 | 26.92 | 26.31 | 28.18 | Commercial 1 | Dark red |
| 11 | 1.27 | 36.62 | 39.83 | 41.76 | Commercial 2 | Bright red |
| 12 | 0.53 | 44.20 | 49.68 | 46.28 | Commercial 3 | Bright red |
| 13 | 1.03 | 33.00 | 34.55 | 28.21 | Commercial 4 | Red |
| 14 | 1.92 | 30.42 | 31.36 | 36.81 | Residential 4 | Yellow |
| 15 | 1.00 | 40.55 | 44.30 | 39.99 | Commercial 5 | Bright red |
| 16 | 2.13 | 35.84 | 38.80 | 35.09 | Commercial 6 | Red |
| 17 | 4.83 | 25.54 | 24.14 | 43.25 | Residential 5 | Bright yellow |
| 18 | 1.86 | 31.03 | 32.57 | 32.62 | Residential 6 | Yellow |
| 19 | 3.26 | 22.36 | 20.22 | 31.21 | Commercial 7 | Dark red |
| 20 | 0.02 | 34.00 | 43.00 | 48.00 | Commercial 8 | Bright red |

assigned to an individual cluster on a CRT screen. In this manner it is possible to identify their location and spatial association with the other clusters. This interactive visual analysis, in conjunction with the information provided in the scatterplot, allows the analyst to group the clusters into information classes as shown in Fig. 8–24 and Table 8–9. It is instructive to review some of the logic that resulted in the unsupervised classification (Fig. 8–21) (color section).

Cluster 1 was a very distinct cluster in spectral space (Fig. 8–24); therefore, it was not difficult to assign it to the information class "water." Clusters 2 and 3 had a high reflectance in the near-infrared (band 4) with low reflectance in the red (band 3) due to chlorophyll absorption. These two clusters were both assigned to the forest class and color-coded dark green (refer to Table 8–9).

Clusters 4 and 5 were situated alone in spectral space between the forest (2 and 3) and water (1) and evidently were comprised of a mixture of moist soil and abundant vegetation. Therefore, it was not difficult to assign both these clusters to a wetland class. They were given different color codes to demonstrate that, indeed, two separate classes of wetland were identified.

Six clusters were associated with residential housing. These clusters were situated between the forest and commercial clusters (to be discussed). This is not unusual since residential housing is composed of a mixture of vegetated and nonvegetated



OFEK06913

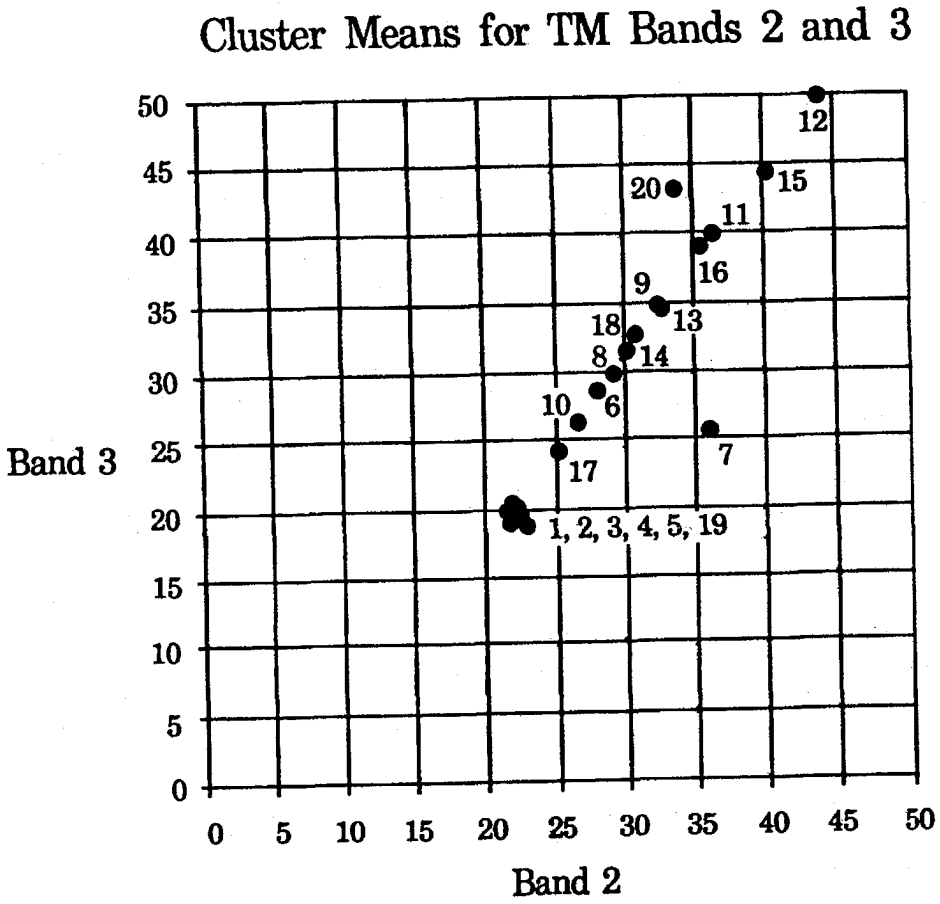



Figure 8–22 The mean vectors of the 20 clusters displayed in Fig. 8–21 are shown here using only bands 2 and 3. The mean vector values are summarized in Table 8–9. Notice the substantial amount of overlap among clusters 1, 2, 3, 4, 5, and 19.

(asphalt/concrete) surfaces, especially at TM spatial resolutions of 30 × 30 m. Based on where they were located in feature space, the six clusters were collapsed into just two: bright yellow (6, 7, 17) or yellow (9, 14, 18).

Eight clusters were associated with commercial land use. Four of the clusters (11, 12, 15, 20) reflected high amounts of both red and near-infrared energy, as commercial land use composed of concrete and bare soil often does. Two other clusters (13 and 16) were associated with commercial strip areas, particularly the downtown areas.

Finally, there were two clusters (10 and 19) that were definitely commercial in character but that had a substantial amount of associated vegetation. They were mainly found along major thoroughfares in residential areas, where vegetation is more plentiful. These three subgroups of commercial land use were assigned bright red, red, and dark red, respectively (Table 8–9).

Cluster 8 did not fall nicely into any group. It experienced very high near-



OFEK06914

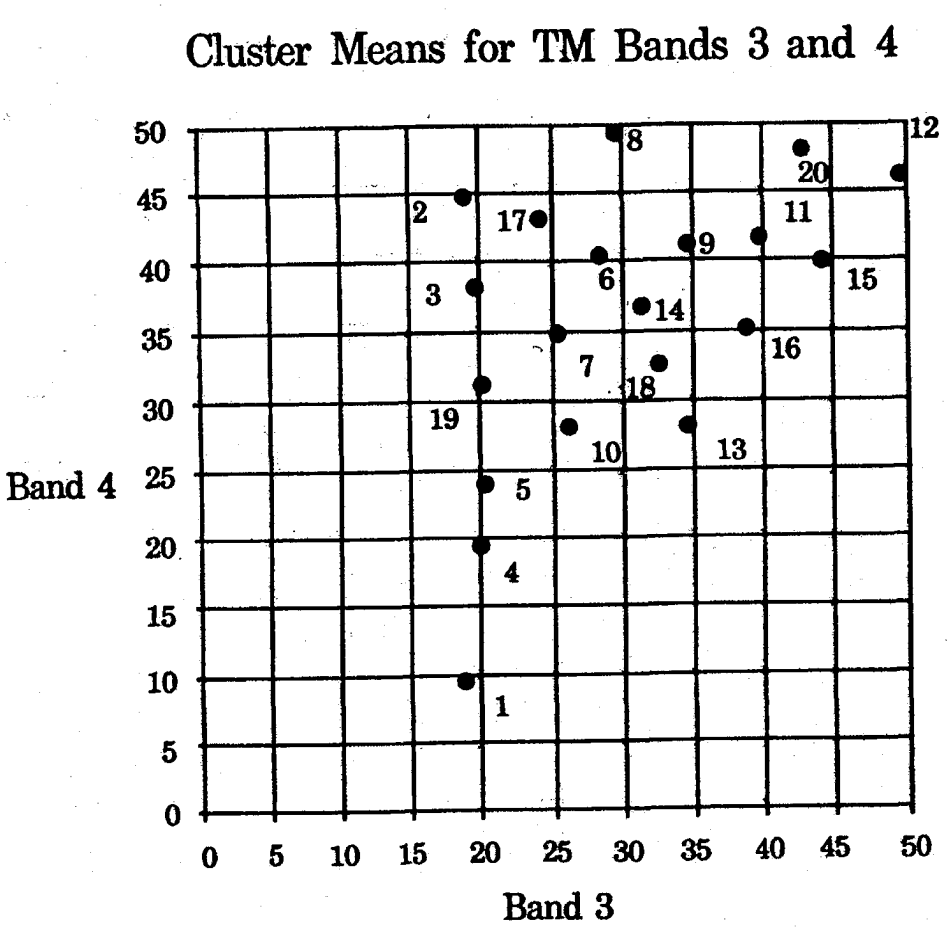



**Figure 8–23** The mean vectors of the 20 clusters displayed in Fig. 8–21 are shown here using only band 3 and 4 data. The mean vector values are summarized in Table 8–9. Compare the spatial distribution of these 20 clusters in the red and near-infrared feature space with what is expected in a typical perpendicular vegetation index as discussed in Chapter 7 and Fig. 7–41.

infrared reflectance and chlorophyll absorption, often associated with very well kept lawns or parks. In fact, this is precisely what it was labeled, "parks and golf."

The 20 clusters and their color assignments are shown graphically in Fig. 8–21. Obviously, there is more information present in this unsupervised classification than in the supervised classification. Except for water, there are at least two classes in each land-use category that could be successfully identified using the unsupervised technique. The supervised classification simply did not sample many of these classes during the training process.

## INCORPORATING ANCILLARY AND CONTEXTUAL DATA IN THE CLASSIFICATION PROCESS

*Ancillary data* is any type of information used in the classification process that is not directly obtainable from either the spatial or spectral characteristics of the remote


OFEK06915

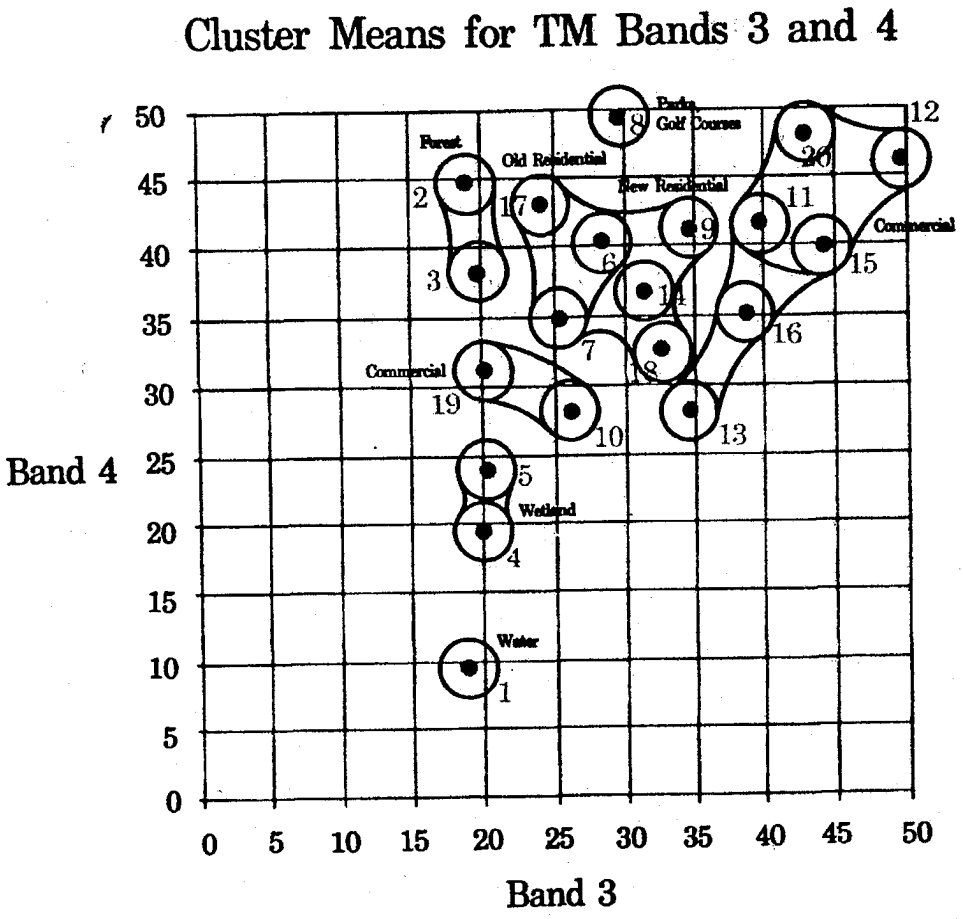



**Figure 8–24** Grouping (labeling) of the original 20 spectral clusters into information classes. The labeling was performed by analyzing the mean vector locations in bands 3 and 4.

sensor data itself (Tom et al., 1978; Strahler, 1980). For example, the amount and type of agricultural land use in the previous year should be of value in this year's agricultural crop-type inventory. Similarly, certain types of forest species are spatially distributed based solely on aspect and/or topographic characteristics. Ideally, land cover classifications based on the analysis of remote sensor data should incorporate such valuable ancillary information.

Several approaches have been developed to incorporate ancillary data into the remote sensing classification process. For example, one approach is to actually digitize the ancillary data into an image format that is in near-perfect registration with the remote sensor data. In this case the ancillary data become another "feature" in the measurement vector, (e.g., instead of perhaps seven Landsat TM features, there are eight, the eighth being an elevation feature). The digital terrain information can also be used to compute other ancillary variables, such as aspect and slope, to improve forest species identification. Such ancillary data can be trained on in a supervised



OFEK06916

classification or simply used as another plane of information in an unsupervised classification (Hoffer et al., 1979; Tom et al., 1978).

Campbell (1983) suggests that ancillary data are well suited to layered classifications. *Layered classification* refers to the hierarchial process whereby two or more decision rules are used in the classification process (Hauska and Swain, 1975; Jensen, 1978). The usual single-step classification algorithm uses all available information in a single process to produce the entire menu of information classes. Layered classification uses subsets of features in a series of separate decisions. For example, a simple layered classification involving ancillary data might (1) threshold all pixels in the scene that lie between 2600 and 3800 ft above sea level using digital terrain information, and (2) apply a traditional maximum likelihood classification only to those spectral data meeting this criterion to identify all spruce or fir in an area in the Colorado Rockies. This should reduce the number of spruce-fir pixels assigned to forested areas that cannot ecologically support spruce-fir.

Ancillary data can also be used to assist in the final stages of cluster labeling rather than as a feature in the classification process (Walsh, 1980; Logan and Strahler, 1980). Still others have used such ancillary data to develop pixel-specific a priori probabilities that can be used in the maximum likelihood classification algorithm discussed previously (Strahler et al., 1978, 1980).

Richards et al. (1982) developed an alternative means of incorporating information from ancillary data sources into the classification process. This method departs from the previous studies in several important respects. First, it is applied to the classification map after it has already been produced. Second, it can incorporate contextual information into the process. Whereas *texture* refers to the spatial variation within a contiguous group of pixels, the *context* of a pixel (or group of pixels) refers to its spatial relationships with pixels in the remainder of the scene. Thus contextual decision rules can, potentially at least, involve the use of any other pixel or group of pixels from throughout the whole scene (Gurney and Townshend, 1983). There are other classification algorithms that can incorporate contextual information, but none incorporate ancillary data and context as effectively as this procedure (Kettig and Landgrebe, 1976; Swain et al., 1980, 1981; Wharton, 1982).

The method is based on probabilistic label relaxation procedures, which are used for embedding spatial context data in image-labeling problems. In addition, ancillary information is used as a set of probabilities that are injected into a modified relaxation method called *supervised relaxation* labeling. This procedure is applied to the existing remote sensing classification map and strikes a balance between the spectral, spatial, and ancillary data sources. Richards et al. (1982) reported improvements of 68 to 81% in forest classification accuracy when topographic ancillary information was incorporated in this manner.

The ancillary data used in any of the foregoing procedures is seldom collected for the sole purpose of being used as additional information in a remote sensing classification. Rather, it is usually collected for an entirely different reason and then adapted for use in the remote sensing application. Therefore, ancillary data are seldom initially compatible with the remotely sensed data in scale, projection, minimum



OFEK06917

mapping unit, date, or accuracy. For example, U.S. Geological Survey digital elevation models (DEMS) and U.S. Soil Conservation Service soils maps are frequently used as ancillary data. The map scales, projections, and minimum mapping units of these two data types are totally different and initially incompatible with the format of the digital remote sensor data. Many differences can be minimized through radiometric and geometric preprocessing. Any unresolved incompatabilities, however, many detract from the usefulness of the ancillary data.

The incorporation of ancillary and contextual data in the remote sensing classification process is an important alternative to those studies based solely on the analysis of spectral information analyzed on a per pixel basis. However, the choice of variables to be included is critical. Common sense suggests that the analyst should thoughtfully select only those variables with conceptual and practical significance to the classification problem at hand. Incorporating layers of possibly redundant ancillary information can serve no useful purpose and can rapidly consume limited data analysis resources.

## LAND-USE CLASSIFICATION ACCURACY ASSESSMENT[1]

If remote sensing–derived land-use or land-cover maps and associated statistics are to be useful to the researcher or the user community, there must be a method for quantitatively assessing *classification accuracy*. This usually requires the collection of some in situ data or a priori knowledge about some parts of the terrain which can then be compared with the remote sensing–derived classification map. Thus, to assess classification accuracy it is necessary to compare two classification maps: (1) the *remote sensing–derived* map, and (2) what we will call the *assumed true* map (it may in fact contain error). The assumed true map may be derived from in situ investigation or quite often from the interpretation of remotely sensed data obtained at a larger scale or higher resolution. For example, researchers commonly compare a Landsat-supervised classification with an assumed true map produced by interpreting large-scale (e.g. 1:20,000) aerial photography. Since both maps are derived from the interpretation of remotely sensed data, it does not seem to be a rigorous foundation to build on. In any case, land-use data on the assumed true map must be in near-perfect registration with the remote sensing–derived map. Both maps must have exactly the same *nominal scale* classification scheme [i.e., codes such as 1, 2, 3 or A, B, C that represent land-use classes at pixel locations $(i,j)$]. If these criteria are met, it is possible to conduct two types of classification accuracy surveys: overall and site specific.

### Overall Classification Map Accuracy Assessment

To determine the *overall accuracy* of a remote sensing–derived classification map, it is necessary to ascertain whether the map meets or exceeds some predetermined classification accuracy criteria. For example, the overall accuracy of land-use maps

[1] This section contains information reprinted with permission from Jensen et al. 1983, "Urban–Suburban Landuse Analysis," Chapter 30 in *The Manual of Remote Sensing*, R. N. Colwell, ed., Vol. 2, 1571–1666. © 1983 The American Society of Photogrammetry and Remote Sensing.



OFEK06918

for earth resource management should generally be 85% and the accuracy must be approximately equal for most categories (Anderson et al., 1976; Milazzo, 1980). Procedures for evaluating two maps and accepting or rejecting such hypotheses are reviewed in Fitzpatrick-Lins (1978, 1980), Ginevan (1979), Aronoff (1982), and Jensen et al. (1983).

Overall accuracy assessments evaluate the agreement between the two maps in total area in each category. They usually do not evaluate compensating errors that occur in the various categories. For this reason, Aronoff (1982) suggests that "the comparative analysis of [site-specific] error matrices may provide better methods of comparing land-use classification mapping methods than a simple comparison of overall estimated map accuracy."

### Site-Specific Classification Map Accuracy Assessment

This type of error analysis compares the accuracy of the remote sensing–derived classification map pixel by pixel with the assumed true land use map. A number of strategies have been developed. We will constrain our discussion to the results of a supervised classification, although the techniques are equally applicable to unsupervised classification map comparisons.

First, it is possible to conduct a site-specific error evaluation based only on the *training* pixels used to train the classifier in a supervised classification. This simply means that those pixel locations $i, j$ used to train the classifier are carefully evaluated on both the remote sensing–derived classification map and the true map. If training samples are distributed randomly throughout the study area, this evaluation may be considered representative of the study area. Unfortunately, the locations of training sites are usually nonrandom. They are biased by the analyst's a priori knowledge of where certain land cover types exist in the scene. Because of this bias, the classification accuracies for pixels found within the training sites are generally higher than for the remainder of the map because these are the data locations that were used to "train" the classifier. Therefore, it would seem that this procedure is born of expediency and, as a result, can have little use in any serious attempt at accuracy assessment (Campbell, 1983).

Conversely, if other *test* locations in the study area are identified and correctly labeled prior to the classification and if these are not used in the training of the classification algorithm, they can be used to evaluate the accuracy of the classification map. This procedure generally yields a more credible classification accuracy assessment. However, additional "ground truth" is required for these test site areas, coupled with the problem of determining how many pixels are necessary in each test site class. Also, the method of identifying the location of the test sites prior to classification is important since many statistical tests require that locations be randomly selected (e.g., using a random number generator for the identification of unbiased row and column coordinates) so that the analyst does not again bias their selection.

The most rigorous error evaluation is when *every* pixel in the remote sensing–



OFEK06919

derived classification map has a companion pixel in the "assumed true" map. Such an error evaluation requires additional resources to identify, rectify, and compare every pixel in the assumed true map with the same pixel in the classification map. This procedure is unsuitable for practical projects concerned with land-use classification over vast areas, since it requires that duplicate inventories be made. However, for small studies, where the emphasis is on technique development or understanding the advantages and disadvantages of multispectral classification, this approach does have certain advantages.

Once the criterion for objectively identifying the location of specific pixels to be compared is determined, it is necessary to identify the class assigned to each pixel in both the remote sensing–derived map and the "assumed true" map. These data are tabulated and reported in a *contingency table*, where overall classification accuracy and misclassification between categories are identified. It usually takes the form of an $m \times m$ matrix, where $m$ is the number of classes under investigation. The rows in the matrix represent the "assumed true" classes, while the columns are associated with the remote sensing–derived land use. The entries in the contingency table represent the raw number of pixels encountered in each condition; however, they may be expressed as percentages if the number becomes too large. One of the most important characteristics of such matrices is their ability to summarize errors of omission and commission. Results from the supervised classification of the Charleston, S.C., thematic mapper data will be used to demonstrate how the contingency table can be used to identify these types of error.

Prior to the supervised classification of the Charleston scene with the training data summarized in Table 8–3, *test* sites that contained a total of 408 pixels were randomly identified. The ideal number of points to be tested in the land-use classification map was determined from the formulas for the binomial probability theory. The formula for the number of points selected was:

$$N = \frac{4(p)(q^{\sim})}{E^2} \tag{8-25}$$

where $p$ is the expected percent accuracy, $q^{\sim}$ the difference between 100 and $p$, $E$ the allowable error, and $N$ the number of points to be sampled (Fitzpatrick-Lins, 1980). For a sample where the expected accuracy was 85% at an allowable error of 5%, the number of points necessary for reliable results was:

$$N = \frac{4(85 \times 15)}{5^2} = \text{a minimum of 204 points}$$

With expected map accuracies of 85% and an acceptable error of 10%, the sample size for a map would be at least 50. The greater the allowable error, the fewer points that need to be collected to evaluate the classification accuracy. According to Van Genderen and Lock (1977), however, the smallest sample size for meaningful results is 20 points even if the sample is error-free.

The disposition of the 408 pixels in both the remote sensing–derived map and



OFEK06920

the "assumed true" map are summarized in the contingency table found in Tabl 8–8a. The total column on the right summarizes the "assumed true" number c pixels in each class. The total at the bottom of each column provides informatio on the number of pixels in each class found within the test sites on the remot sensing classification map. The diagonal elements of the matrix represent the numbe of pixels correctly classified. For example, 382 of the 408 pixels were correctly classified representing an overall classification accuracy of 93.6%.

The 93.6% accuracy statistic is useful but does not report how confident w should be that the map actually exceeded the 85% criterion originally established To determine this, the following methodology was used. The ratio, $p$ (expressed a a percent), of the number of points correct, $c$, to the total number of points, $n$, represented the accuracy for the map. As this value was the test value for comparison to the minimum standard of 85% accuracy, a one-tailed test was appropriate. The 95% one-tailed lower confidence limit for a binomial distribution was obtained from the equation (derived from Snedecor and Cochran, 1967, p. 211):

$$p = \tilde{p} - \left[1.645\sqrt{(\tilde{p})(\tilde{q})/n} + \frac{50}{n}\right] \tag{8-26}$$

where $p$ = accuracy of the map expressed as a percent

$\tilde{p}$ = value $c/n$ expressed as a percent
$\tilde{q} = 100 - \tilde{p}$
$n$ = sample size

If $p$ exceeds the 85% criterion at the lower confidence limit, it is possible to accept (with 95% confidence) that the map met or exceeded the accuracy standards. When the overall accuracy statistic ($\tilde{p} = 93.6\%$) and the total number of points ($n = 408$) are analyzed, the 95% one-tailed lower confidence limit is 91.48%. Thus this map exceeds the 85% overall accuracy criterion.

The off-diagonal entries provide information on errors of omission and commission. Errors of *commission* are found in the lower left half of the matrix, while errors of *omission* are found in the upper right half of the matrix. Errors of omission for each class are computed by summing the number of pixels assigned to incorrect categories along each row and dividing this number by the total number of true pixels in the category. For example, 18 pixels of residential land use were assigned to categories other than residential, primarily forest. This resulted in an error of omission for this category of 20.5%. Errors of commission are computed by summing the number of pixels assigned to incorrect categories along each column and dividing this number by the total number of pixels assigned to the column category. For example, 4 of the 88 residential land-use pixels on the classification map really belong to other categories. This resulted in a commission error of 4.5% for the residential land-use category.

Whereas a single accuracy statement with a given lower confidence limit was sufficient to describe the entire map, it is also important to determine the accuracy


OFEK06921

of individual categories. To determine the true 95% confidence limits for each category, the following two-tailed test was applied:

$$p = p^{\sim} \pm \left[ 1.96 \sqrt{(p^{\sim})(q^{\sim})/n} + \frac{50}{n} \right] \tag{8-27}$$

where $p^{\sim}$ is again the value of the true percent correct, $q^{\sim}$ is equal to $100 - p^{\sim}$, and $n$ is the number of points sampled *in each category*.

The results of the five classes examined for errors of commission and omission are shown in Table 8–10. Based on an evaluation of the errors of commission, all classes except forest and commercial exceeded the specified accuracy of 85% at the lower confidence limit. Commercial almost met the criteria at 84%. Residential land use and forest did not meet the 85% criterion when errors of omission are evaluated.

**TABLE 8–10** Individual Category Error Evaluation

| | | Commission | | | Omission | | |
|---|---|---|---|---|---|---|---|
| Class | Points correct | n | % correct | 95% confidence limits | n | % correct | 95% confidence limits |
| Residential | 70 | 74 | 95 | 89–100 | 88 | 80 | 71–89 |
| Commercial | 55 | 60 | 92 | 84–99 | 58 | 95 | 88–100 |
| Wetland | 99 | 103 | 96 | 92–100 | 99 | 100 | 98–100 |
| Forest | 37 | 50 | 74 | 65–83 | 42 | 90 | 81–99 |
| Water | 121 | 121 | 100 | 98–100 | 121 | 100 | 98–100 |

Such tables and statistics provide useful information on the type of confusion that took place in the classification. For example, some residential land use was misclassified as forest (13 pixels), but forest was rarely misclassified as residential (1 pixel). Perhaps more attention should be given to the selection of training sites for residential and forest which might improve the supervised classification of these two classes.

It is possible to produce remote sensing–derived classification maps by holding certain parameters constant and applying various treatments such as different classification algorithms, the use of human interpreters with different degrees of skill or expertise, analysis of different types of imagery or film/filter combinations, or classification maps obtained from different scales of imagery. Rosenfield (1982) and Congalton and Mead (1983) propose non-parametric and parametric techniques that the analyst can use to determine if the differences in the classification maps produced by the various treatments are significant.

Procedures such as those discussed allow land-use maps derived from remote sensing to be quantitatively evaluated to determine overall and individual category classification accuracy. Their proper use enhances the credibility of using remote sensing–derived land-use information.



OFEK06922

## REFERENCES

ADDINGTON, J. D. 1975. "A Hybrid Maximum Likelihood Classifier Using the Parallelepiped and Bayesian Techniques," *Technical Papers*, 50th Annual Meeting of the American Society of Photogrammetry, pp. 772–784.

ANDERSON, J. R., E. HARDY, J. ROACH, and R. WITMER. 1976. "A Land Use and Land Cover Classification System for Use with Remote Sensor Data," *U.S. Geological Survey Professional Paper 964*.

ANUTA, P. E. 1977. "Computer-Assisted Analysis Techniques for Remote Sensing Data Interpretation," *Geophysics*, Vol. 42, pp. 468–481.

ARONOFF, S. 1982. "Classification Accuracy: A User Approach," *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 1299–1307.

BERRY, B. J. L. 1967. "Grouping and Regionalization," in *Quantitative Geography*, Part 1, W. L. Garrison and D. Marble, Eds. Northwestern University Studies in Geography, No. 13.

BOARDMAN, D. 1983. *Graphicacy and Geography Teaching*. Beckenham, Kent, England: Croom Helm Ltd.

BOTKIN, D. B., J. E. ESTES, R. B. MACDONALD, and M. V. WILSON. 1984. "Studying the Earth's Vegetation from Space," *Bioscience*, Vol. 34, No. 8, pp. 508–514.

BRYANT, J. 1979. "On the Clustering of Multidimensional Pictorial Data," *Pattern Recognition*, Vol. 11, pp. 15–125.

CAMPBELL, J. B. 1983. *Mapping the Land: Aerial Imagery for Land Use Information*. Washington, D.C.: Association of American Geographers.

CARR, J. R., C. E. GLASS, and R. A. SCHOWENGERDT. 1983. "Signature Extension Versus Retraining for Multispectral Classification of Surface Mines in Arid Regions," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 1193–1199.

CONGALTON, R. G., and R. A. MEAD. 1983. "A Quantitative Method to Test for Consistency and Correctness in Photointerpretation," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 69–74.

COWARDIN, L. M., V. CARTER, F. C. GOLET, and E. T. LAROE. 1979. *Classification of Wetlands and Deepwater Habitats of the United States*, Washington, D.C.: U.S. Fish and Wildlife Service, FWS/OBS-79/31.

DOWNS, S. W., and N. L. FAUST. 1978. "The Use of Landsat Data to Inventory Cotton and Soybean Acreage in North Alabama," *Proceedings*, Remote Sensing of Earth Resources Conference, Tullahoma, Tenn.

DRISCOLL, T., and C. WALKER. 1982. *Evolution of Image Processing Algorithms from Software to Hardware*. Milpitas, Calif.: International Imaging Systems.

DUDA, R. D. and P. E. HART. 1973. *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, Inc.

FITZPATRICK-LINS, K. 1978. "Accuracy and Consistency Comparisons of Land Use and Land Cover Maps Made from High-Altitude Photographs and Landsat Multispectral Imagery," *Journal of Research*, U.S. Geological Survey, Vol. 6, pp. 23–40.

FITZPATRICK-LINS, K. 1980. "The Accuracy of Selected Land Use and Land Cover Maps at Scales of 1:250,000 and 1:100,000," *Journal of Research*, U.S. Geological Survey, Vol. 6, pp. 169–173.

FRIEDMAN, S. Z. 1980. *Mapping Urbanized Area Expansion through Digital Image Processing of Landsat and Conventional Data*. Publication 79–113. Pasadena, Calif.: Jet Propulsion Laboratory.



OFEK06923

GASSON, P. C. 1983. *Geometry of Spatial Forms: Analysis, Synthesis, Concept Formulation and Space Vision for CAD*. New York: Halsted Press, pp. 204–221.

GINEVAN, M. E. 1979. "Testing Land-Use Map Accuracy: Another Look," *Photogrammetric Engineering and Remote Sensing*, Vol. 45, pp. 1371–1377.

GOLDBERG, M., and S. SHLIEN. 1976. "A Four-Dimensional Histogram Approach to the Clustering of Landsat Data," *Canadian Journal of Remote Sensing*, Vol. 2, pp. 1–11.

GURNEY, C. M., and J. R. G. TOWNSHEND. 1983. "The Use of Contextual Information in the Classification of Remotely Sensed Data," *Photogrammetric Engineering and Remote Sensing*, Vol. 49, pp. 55–64.

HAACK, B. N. 1983. "An Analysis of Thematic Mapper Simulator Data for Urban Environments," *Remote Sensing of Environment*, Vol. 13, pp. 265–275.

HARALICK, R. M., and K. FU. 1983. "Pattern Recognition and Classification," Chapter 18 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: American Society of Photogrammetry and Remote Sensing, Vol. 1, pp. 793–805.

HARTIGAN, J. A. 1975. *Clustering Algorithms*. New York: John Wiley & Sons, Inc.

HAUSKA, H., and P. SWAIN. 1975. "The Decision Tree Classifier: Design and Potential," *Proceedings*, Symposium on Machine Processing of Remotely Sensed Data. West Lafayette, Ind.: Laboratory for the Applications of Remote Sensing, pp. 33–48.

HEASLIP, G. G. 1975. *Environmental Data Handling*. New York: John Wiley & Sons, Inc.

HIXSON, M., D. SCHOLZ, N. FUHS, and T. AKIYAMA. 1980. "Evaluation of Several Schemes for Classification of Remotely Sensed Data," *Photogrammetric Engineering and Remote Sensing*, Vol. 46, pp. 1547–1553.

HOFFER, R. M., M. D. FLEMING, L. A. BARTOLUCCI, S. M. DAVIS, and R. F. NELSON. 1979. "Digital Processing of Landsat MSS and Topographic Data to Improve Capabilities for Computerized Mapping of Forest Cover Types," *LARS Technical Report 011579*. West Lafayette, Ind.: Laboratory for the Applications of Remote Sensing.

HORD, R. M. 1982. *Digital Image Processing of Remotely Sensed Data*. New York: Academic Press, Inc.

JENSEN, J. R. 1978. "Digital Land Cover Mapping Using Layered Classification Logic and Physical Composition Attributes," *The American Cartographer*, Vol. 5, pp. 121–132.

JENSEN, J. R. 1979. "Computer Graphic Feature Analysis and Selection," *Photogrammetric Engineering and Remote Sensing*, Vol. 45, pp. 1507–1512.

JENSEN, J. R. et al. 1983. "Urban–Suburban Land Use Analysis," Chapter 30 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: American Society of Photogrammetry and Remote Sensing, Vol. 2, pp. 1571–1666.

JUSTICE, C., and J. TOWNSHEND. 1982. "A Comparison of Unsupervised Classification Procedures on Landsat MSS Data for an Area of Complex Surface Conditions in Basilicata, Southern Italy," *Remote Sensing of Environment*, Vol. 12, pp. 407–420.

KALAYEH, H. M., M. J. MUASHER, and D. A. LANDGREBE. 1983. "Feature Selection with Limited Training Samples," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. GE-21, pp. 434–437.

KETTIG, R. L., and D. A. LANDGREBE. 1976. "Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects," *IEEE Transactions on Geoscience Electronics*, Vol. GE-14, pp. 19–26.

KUMAR, R., and L. F. SILVA. 1977. "Separability of Agricultural Cover Types by Remote Sensing in the Visible and Infrared Wavelength Regions," *IEEE Transactions on Geoscience Electronics*, Vol. GE-15, pp. 42–49.

LOGAN, T. L., and A. H. STRAHLER. 1980. "Forest Cover Classification from Landsat Imagery:



OFEK06924

An Application for Image Processing," *1980 IEEE Workshop on Picture Data Descriptio and Management*, Santa Barbara, Calif.

MACDOUGALL, E. B. 1976. *Computer Programming for Spatial Problems*. London: Edwar Arnold.

Michigan Land Use Classification and Referencing Committee. 1975. *Michigan Land Cover Use Classification System*. Lansing: Office of Land Use Department of Natural Resource State of Michigan.

MILAZZO, V. A. 1980. "A Review and Evaluation of Alternatives for Updating U.S. Geologic Survey Land Use and Land Cover Maps," *U.S. Geological Survey Circular 826*.

MITCHELL, W. B., S. C. GUPTILL, K. E. ANDERSON, R. G. FEGEAS, and C. A. HALLAM 1977. *GIRAS: A Geographic Information Retrieval and Analysis System for Handling Lan Use and Land Cover Data*. Washington, D.C.: U.S. Geological Survey Professional Pap 1059.

NASA. 1983. *Land-Related Global Habitability Science Issues*, Washington, D.C.: Nation Aeronautics and Space Administration, Technical Memorandum 85841.

NEWMAN, W. M., and R. F. SPROULL. 1978. *Principles of Interactive Computer Graphic* New York: McGraw-Hill Book Company.

RHIND, D., and R. HUDSON. 1980. *Land Use*. New York: Methuen, Inc.

RICHARDS, J. A., D. A. LANDGREBE, and P. H. SWAIN. 1982. "A Means for Utilizing Ancilla Information in Multispectral Classification," *Remote Sensing of Environment*, Vol. 12, p 463–477.

ROBINOVE, C. J. 1977. "A Radiometric Interpretive Legend for Landsat Digital Themat Maps," *Photogrammetric Engineering and Remote Sensing*, Vol. 43, pp. 593–594.

ROBINOVE, C. J. 1981. "The Logic of Multispectral Classification and Mapping of Land *Remote Sensing of Environment*, Vol 11, pp. 231–244.

ROSENFIELD, G. H. 1982. "The Analysis of Areal Data in Thematic Mapping Experiments *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 1455–1462.

SCHOWENGERDT, R. A. 1983. *Techniques for Image Processing and Classification in Remo Sensing*. New York: Academic Press, Inc.

SNEDECOR, G. W., and W. G. COCHRAN. 1967. *Statistical Methods*, 6th ed, Ames: Iow State University Press.

SOKAL, R. R., and P. H. A. SNEATH. 1963. *Principles of Numerical Taxonomy*. San Francisc Freeman.

STRAHLER, A. H. 1980. "The Use of Prior Probabilities in Maximum Likelihood Classificatic of Remotely Sensed Data," *Remote Sensing of Environment*, Vol. 10, pp. 135–163.

STRAHLER, A. H., T. L. LOGAN, and N. A. BRYANT. 1978. "Improving Forest Cover Classific tion Accuracy from Landsat by Incorporating Topographic Information," *Proceedings*, 12 International Symposium on Remote Sensing of the Environment, pp. 927–942.

STRAHLER, A. H., J. E. ESTES, P. F. MAYNARD, F. C. MERTZ, and D. A. STOW. 198 "Incorporating Collateral Data in Landsat Classification and Modeling Procedures," *Proceedings*, 14th International Symposium on Remote Sensing of Environment, pp. 1009–102

SWAIN, P. H. and S. M. DAVIS (Eds). 1978. *Remote Sensing the Quantitative Approach*. Ne York, McGraw-Hill Book Co., p. 180.

SWAIN, P. H., and A. G. WACKER. 1971. "Comparison of the Divergence and B-Distan in Feature Selection." West Lafayette, Ind.: Laboratory for the Applications of Remo Sensing, Technical Report 020871.



OFEK06925

SWAIN, P. H., H. J. SIEGEL, and B. W. SMITH. 1980. "Contextual Classification of Multispectral Remote Sensing Data Using a Multiprocessor System," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. GE-18, No. 2, pp. 197–204.

SWAIN, P. H., S. B. VARDEMAN, and J. C. TILTON. 1981. "Contextual Classification of Multispectral Image Data," *Pattern Recognition*, Vol. 13, pp. 429–441.

TOM, C., L. D. MILLER and J. W. CHRISTENSON. 1978. "Spatial Land Use Inventory, Modeling, and Projection/Denver Metropolitan Area with Inputs from Existing Maps, Airphotos, and Landsat Imagery," *NASA Technical Memorandum 79710*. Greenbelt, Md.: National Aeronautics and Space Administration.

TOWNSHEND, J. R. G. 1981. "Image Analysis and Interpretation for Land Resources Survey," Chapter 4 in *Terrain Analysis and Remote Sensing*, J. R. G. Townshend, Ed. London: George Allen & Unwin Ltd., pp. 59–108.

Urban Renewal Administration, Housing and Home Finance Agency, and Bureau of Public Roads. 1965. *Standard Land Use Coding Manual: A Standard System for Identifying and Coding Land Use Activities*. Washington, D.C.

VAN GENDEREN, J. L., and B. F. LOCK. 1977. "Testing Land-Use Map Accuracy," *Photogrammetric Engineering*, Vol. 43, pp. 1135–1137.

WALSH, S. J. 1980. "Coniferous Tree Species Mapping Using Landsat Data," *Remote Sensing of Environment*, Vol. 9, pp. 11–26.

WELCH, R. A. 1982. "Spatial Resolution Requirements for Urban Studies," *International Journal of Remote Sensing*, Vol. 3, pp. 139–146.

WHARTON, S. W. 1982. "A Context-Based Land Use Classification Algorithm for High Resolution Remotely Sensed Data," *Journal of Applied Photographic Engineering*, Vol. 8, pp. 46–50.

WHARTON, S. W., and B. J. TURNER. 1981. "ICAP: An Interactive Cluster Analysis Procedure for Analyzing Remotely Sensed Data," *Remote Sensing of Environment*, Vol. 11, pp. 279–293.



OFEK06926

# 9

# Change Detection

## INTRODUCTION

Accurate land-use and land-cover information is needed at the local, regional, and national administrative levels for thoughtful land-use planning and management (Paul and Mascarenhas, 1981). For example, in areas where intensive urbanization and limited agriculture must coexist, local agencies often conduct surveys to monitor and regulate the conversion of prime agricultural land to suburban land, lest this renewable resource be lost indefinitely (Carlson and Rubingh, 1979).

This chapter identifies methods of digital change detection mapping using remote sensor data. Emphasis is on detection of new residential development at the urban fringe, although there are other equally important types of change detection that could be demonstrated. The nature of the change detection problem in general is first addressed to demonstrate the complexity of the digital change detection task. Change detection algorithms used with Landsat imagery are then reviewed.

## THE NATURE OF CHANGE DETECTION

How do we identify changes in imagery of the same geographic area obtained at different times? Change detection can be a fairly straightforward task for interpreters evaluating large-scale aerial photography, but visual comparison of photos can be slow, tiring, and is subject to numerous errors of omission (Shepard, 1964). These are some of the reasons that digital methods are used to correlate and compare two sets of imagery to identify changes. A change detection method should be based





OFEK06927

on a sensor system that (1) has a systematic period between overflights (e.g., 18 days), (2) records imagery of the same geographic area at the same time of day to minimize diurnal sun angle effects, (3) maintains the same scale and look-angle geometry, (4) reduces relief displacement as much as possible, and (5) records reflected radiant flux in consistent and useful spectral regions. If these conditions are satisfied, it may be possible to analyze the spatial, spectral, and temporal characteristics of the remote sensor data to obtain land-use change statistics and produce a land-use change map.

Accurate spatial registration of at least two images is essential for digital change detection. The images may be raw remote sensor data obtained on two different dates or two classification maps produced from imagery obtained on two different dates. This necessitates the use of geometric rectification algorithms that register the images to each other or, even better, to a standard map projection (refer to Chapter 6 for a discussion on image rectification). Rectification should result in the two images or maps being within one-fourth to one-half a pixel of registration one to another. If the misregistration is greater than 1 pixel, this may result in the identification of spurious areas of change between the two data sets.

The spatial resolution of the sensor system is important because the instantaneous field of view (IFOV) governs the size of the area over which radiant flux will be integrated and recorded. The size of the picture element represents the minimum mapping unit. For example, the 79 × 79 m ground resolution of the Landsat MSS precludes the identification of individual houses, trees, and so on. Therefore, one must be satisfied recording general levels of land cover such as rangeland or residential housing.

Having established that the sensor system should record an image of the same area at regular intervals and at the same time of day, a fundamental question arises. Which dates should be used? *Anniversary dates* are often used because they minimize differences in reflectance caused by seasonal vegetation changes, soil moisture variation, or sun angle differences. Indeed, there may even be optimum dates within the year for change detection for a given region. For example, anniversary spring dates were found to be superior for Richmond, Virginia, because the vegetation canopy was sparse enough to enable residential land use to be discriminated from natural vegetation (Christenson, 1977).

A fundamental assumption of digital change detection is that there exists a difference in the spectral response of a pixel on two dates if the land-use changes from one type of land cover to another. Ideally, the spectral resolution of the sensor system is sufficient to record reflected radiant flux in spectral regions that best capture the most descriptive spectral attributes of the object. Unfortunately, Landsat's MSS is relatively coarse being composed of four broad bands. Also, these bands were optimized for terrain analysis, not suburban, applications and there is redundancy in the information content of MSS bands 4 and 5 and in bands 6 and 7. Therefore, we find instances where the four-dimensional vector of the MSS data for each pixel may not provide sufficient information to allow correct classification, let alone change detection. For example, numerous reports identify signature overlap between agricul-


OFEK06928

tural and residential land use (Jensen, 1981). Fortunately, detection of change requires less spatial and spectral resolution than does identification of an object. Therefore, although it may be difficult to classify suburban land use to its most detailed class, it may be possible to at least identify that it has changed. Higher-resolution sensor systems (e.g., aerial photography) or field work can then be applied to the areas of change that have been detected.

Landsat MSS and even Thematic Mapper data are at best useful only for regional or national change detection. Even then the data may not constitute a primary source for such an update, but may rather function in a complementary role (Milazzo, 1980). It takes improved spatial and spectral resolution to achieve detailed local change detection. Even the Landsat thematic mapper with its 30 × 30 m spatial resolution and six spectral bands cannot be used to perform detailed change detection in urban environments. The 16- or 18-day temporal resolution, however, is probably sufficient for most urban change detection tasks.

## CHANGE DETECTION ALGORITHMS

The selection of an appropriate change detection algorithm is based on an analysis of several important factors. First, the analyst must know the cultural and biophysical characteristics of the study area. Next, it is imperative to know the precision with which the multiple-date imagery is registered. Finally, one should be aware of change detection algorithm alternatives, their degree of flexibility, and availability. There are at least five general classes of change detection algorithms, including image differencing, image ratioing, classification comparison, comparison of preprocessed imagery, and change vector analysis (Howarth and Wickware, 1981; Jensen, 1981; Jensen and Toll, 1982). There are a variety of alternatives within most of these classes, and it is not practical to provide examples of each change detection alternative. Rather, most algorithms will be discussed in light of previous research, with a select few demonstrated by example. In the latter case, a 6-mi$^2$ portion of the Fitzsimmons 7.5-minute quadrangle in Denver, Colorado, where substantial residential development has taken place between 1976 and 1978 will be evaluated. Landsat MSS imagery was obtained on October 1, 1976, and September 30, 1978 (Fig. 9–1). The data were registered to a universal transverse Mercator (UTM) projection using 32 ground control points. Pixels were resampled to 60 × 60 m using a nearest-neighbor algorithm that did not alter the original pixel values, in contrast to cubic-convolution resampling (refer to Chapter 6 for a discussion of these resampling methods).

To complement the digital data set, panchromatic aerial photography of the study area (0.4 to 0.7 $\mu$m) was obtained on October 8, 1976, and October 15, 1978 (Figs. 9–2 and 9–3). Ten stages of residential development were visually interpreted and overlayed on top of the aerial photography based on evidence of clearing, subdivision, transportation, buildings, and landscaping (Fig. 9–4). This detailed classification of relatively homogeneous phases of residential development were used as "ground truth" to assess the digital change detection performance.



OFEK06929

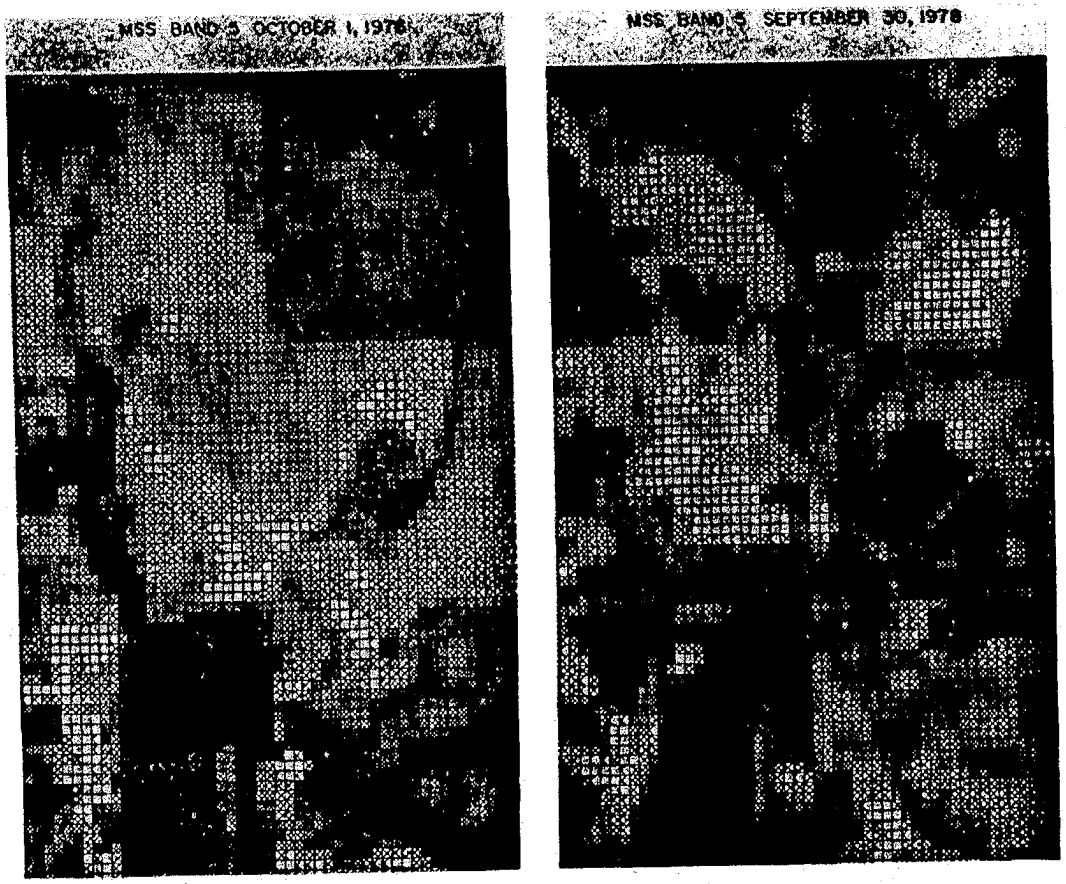



**Figure 9–1** Landsat MSS band 5 images of a portion of the Fitzsimmons 7.5-minute quadrangle study area obtained on October 1, 1976, and September 30, 1978. Raw data were used in all change detection procedures, whereas these figures have been contrast stretched. (From Jensen and Toll, 1982.)

## *Image Differencing*

Image differencing involves subtracting the imagery of one date from that of another. The subtraction results in positive and negative values in areas of radiance change and zero values in areas of no change. In an 8-bit ($2^8$) analysis with pixel values ranging from 0 to 255, the potential range of difference values is −255 to 255. The results are normally transformed into positive values by adding a constant, $c$. The operation is expressed mathematically as

$$\Delta x_{ijk} = BV_{ijk}(1) - BV_{ijk}(2) + c \tag{9-1}$$

where $\Delta x_{ijk}$ = change pixel value
$BV_{ijk}(1)$ = brightness value at time 1
$BV_{ijk}(2)$ = brightness value at time 2
$c$ = a constant (e.g., 255)


OFEK06930

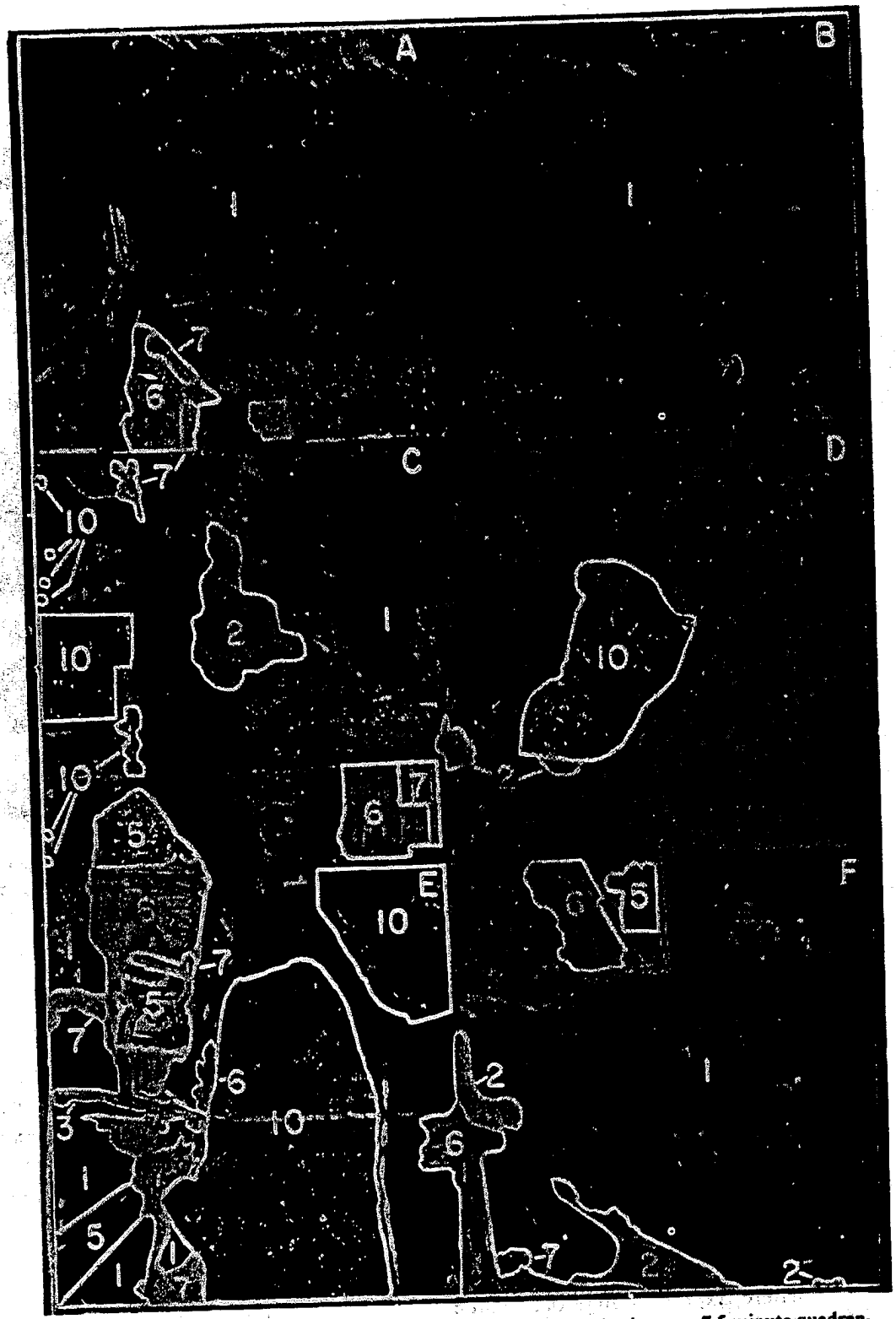



**Figure 9–2** Panchromatic aerial photography of a portion of the Fitzsimmons 7.5-minute quadrangle near Denver, Colorado, on October 8, 1976. Photography was obtained from Colorado Aerial Photo Service of Denver, Colorado. The original scale was 1:52,800. The land cover was visually photointerpreted and classified into 10 classes of residential development using the logic shown in Fig. 9–4. (From Jensen, 1981.)



OFEK06931

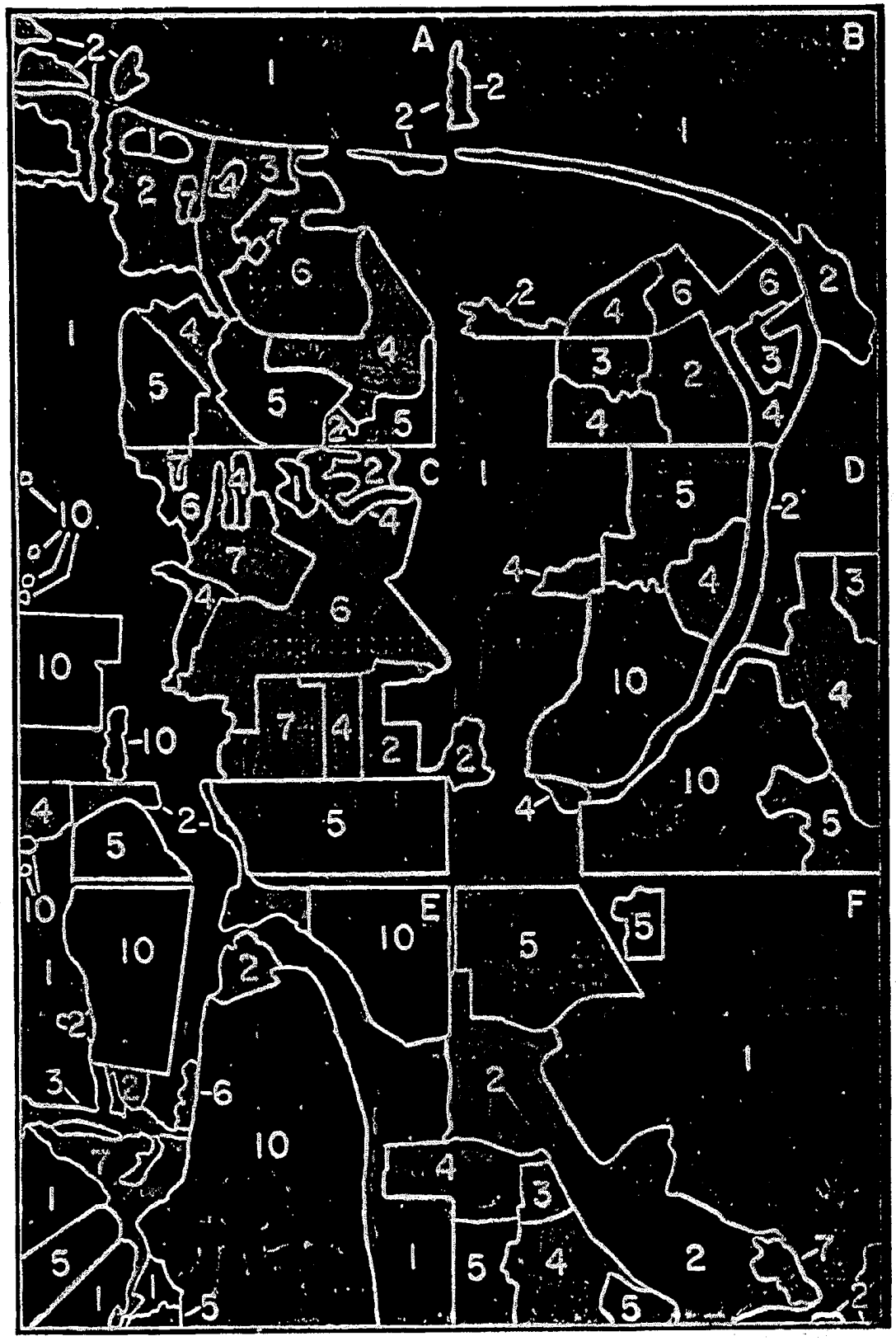



**Figure 9–3** Panchromatic aerial photography of the study area obtained on October 15, 1978. The original scale was 1:57,600. Comparison with Fig. 9–2 reveals substantial residential land use development since October 8, 1976. (From Jensen, 1981.)



OFEK06932

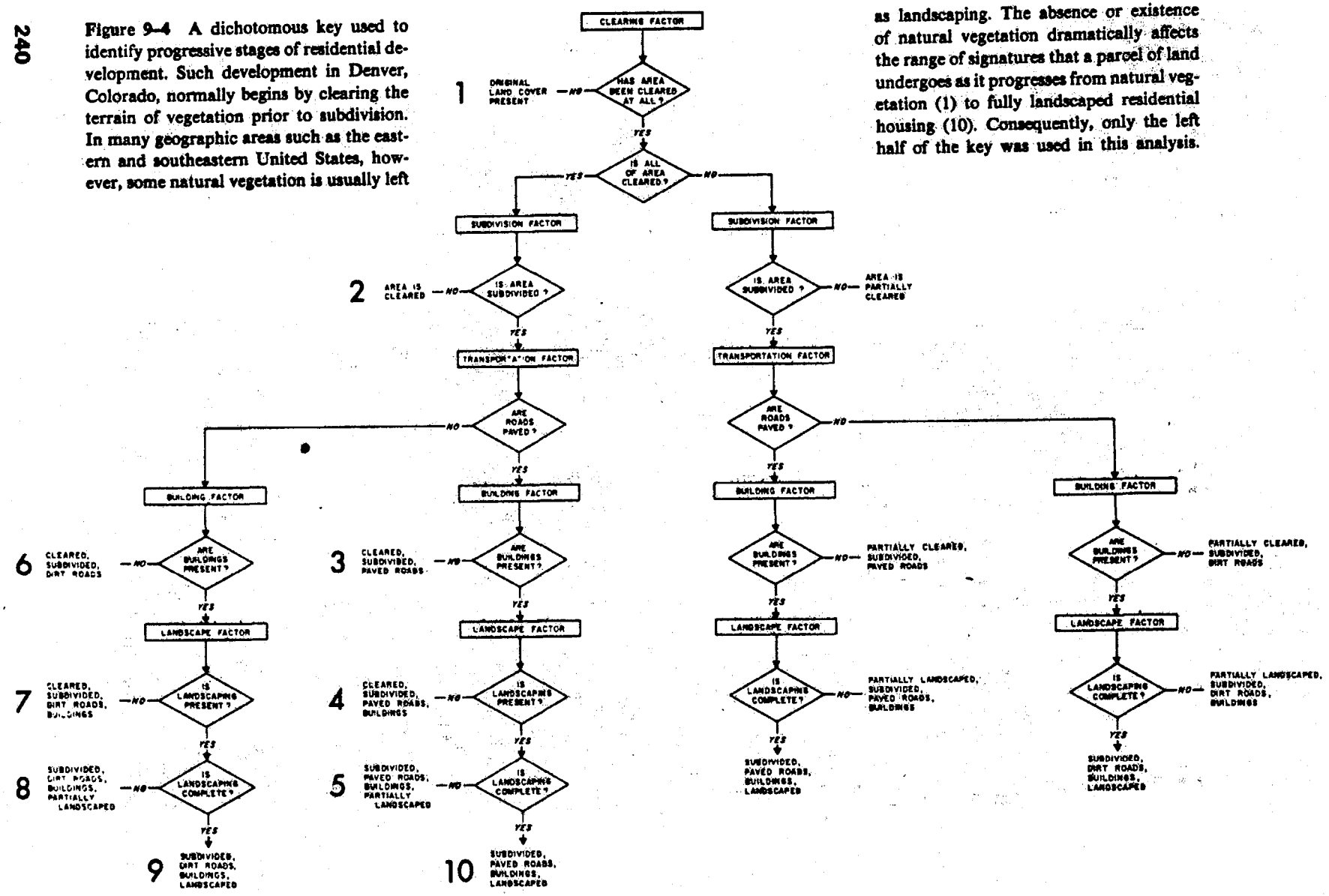



**Figure 9-4** A dichotomous key used to identify progressive stages of residential development. Such development in Denver, Colorado, normally begins by clearing the terrain of vegetation prior to subdivision. In many geographic areas such as the eastern and southeastern United States, however, some natural vegetation is usually left as landscaping. The absence or existence of natural vegetation dramatically affects the range of signatures that a parcel of land undergoes as it progresses from natural vegetation (1) to fully landscaped residential housing (10). Consequently, only the left half of the key was used in this analysis.

1135

OEK06933

$i = 1 \ldots$ number of line
$j = 1 \ldots$ number of column
$k =$ a single band (e.g., MSS 5)

This procedure yields a "differenced" distribution for each band (Fig. 9–5), approximately Gaussian in nature, where pixels of no brightness value change are distributed around the mean and pixels of change are found in the tails of the distribution. A critical element of the image differencing method is deciding where to place the *threshold* boundaries between change and no-change pixels displayed in the histogram. Often, a standard deviation from the mean is selected and tested empirically to determine if changes were accurately monitored. The procedure has also become interactive, wherein an analyst familiar with the area tries various thresholds until optimum ones are identified (Royal, 1980).

A change histogram was created for the Denver study area by differencing MSS band 5 data acquired on October 1, 1976, and September 30, 1978. Interactive analysis of the histogram produced a change map that was superimposed as precisely as possible onto the 1978 aerial photography (Fig. 9–6). Comparing manually interpreted categories (Figs. 9–2 and 9–3) with the digital change map (Fig. 9–6) revealed some interesting relationships. The digital method accurately identified land-use change from natural vegetation (category 1) to partially or fully landscaped residential development (categories 5 and 10, respectively). However, in sections A and C there were substantial errors of omission when change was from natural vegetation to categories 3, 4, and 6 (i.e., when development was initiated but not completed). The absolute accuracy of the classification was 77%, with the majority of the error being omission (statistics are summarized in Table 9–1a).

Akin to image differencing is the image overlay method described by Banner and Lynham (1981) and Howarth and Boasson (1983). They found that displaying the MSS band 5 image on one date in blue on the CRT screen and the MSS band 5 image on the next date in red on the CRT was quite useful for identifying change and did not necessitate taking the difference. After adjustment of the color balance, areas of no change appeared gray. Any areas that had changed were highlighted in one of the colors. The greater the change in tone from light to dark (or vice versa) between the two dates, the more dominant one of the colors (blue or red) appeared on the screen. This technique requires the availability of a digital image processing system.

### *Image Ratioing*

Changes in viewing conditions (e.g., shadows, seasonal reflectance differences due to sun angle, etc.) degrade the ability of a classifier to identify materials correctly. Fortunately, ratio transformations tend to remain invariant when changes result from such factors. These ratios are especially useful for change detection when several dates of imagery are used in an analysis because they can reduce the effect of environ-


OFEK06934

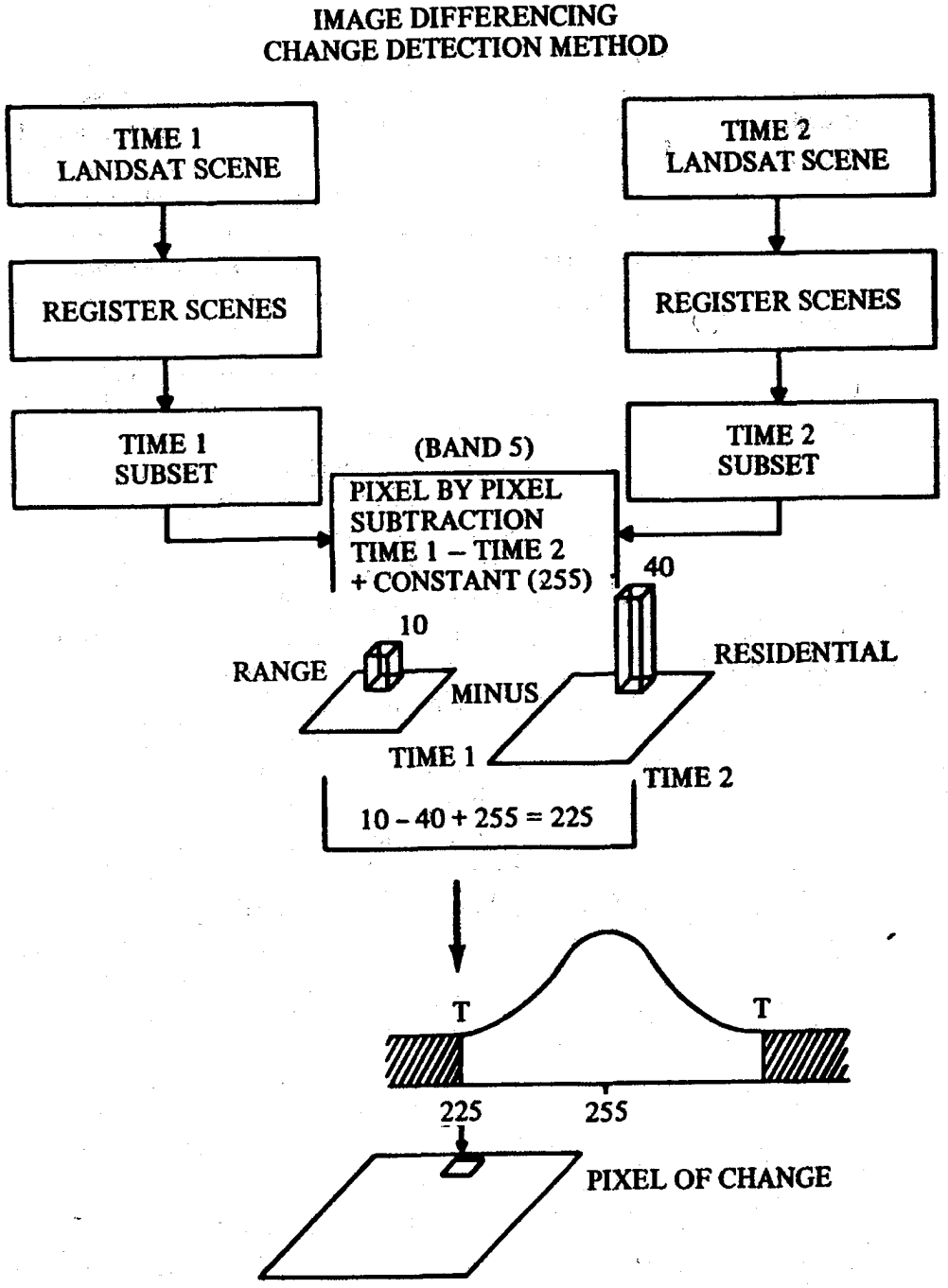



**Figure 9–5** Image differencing change detection method. (From Jensen, 1981.)





OFEK06935

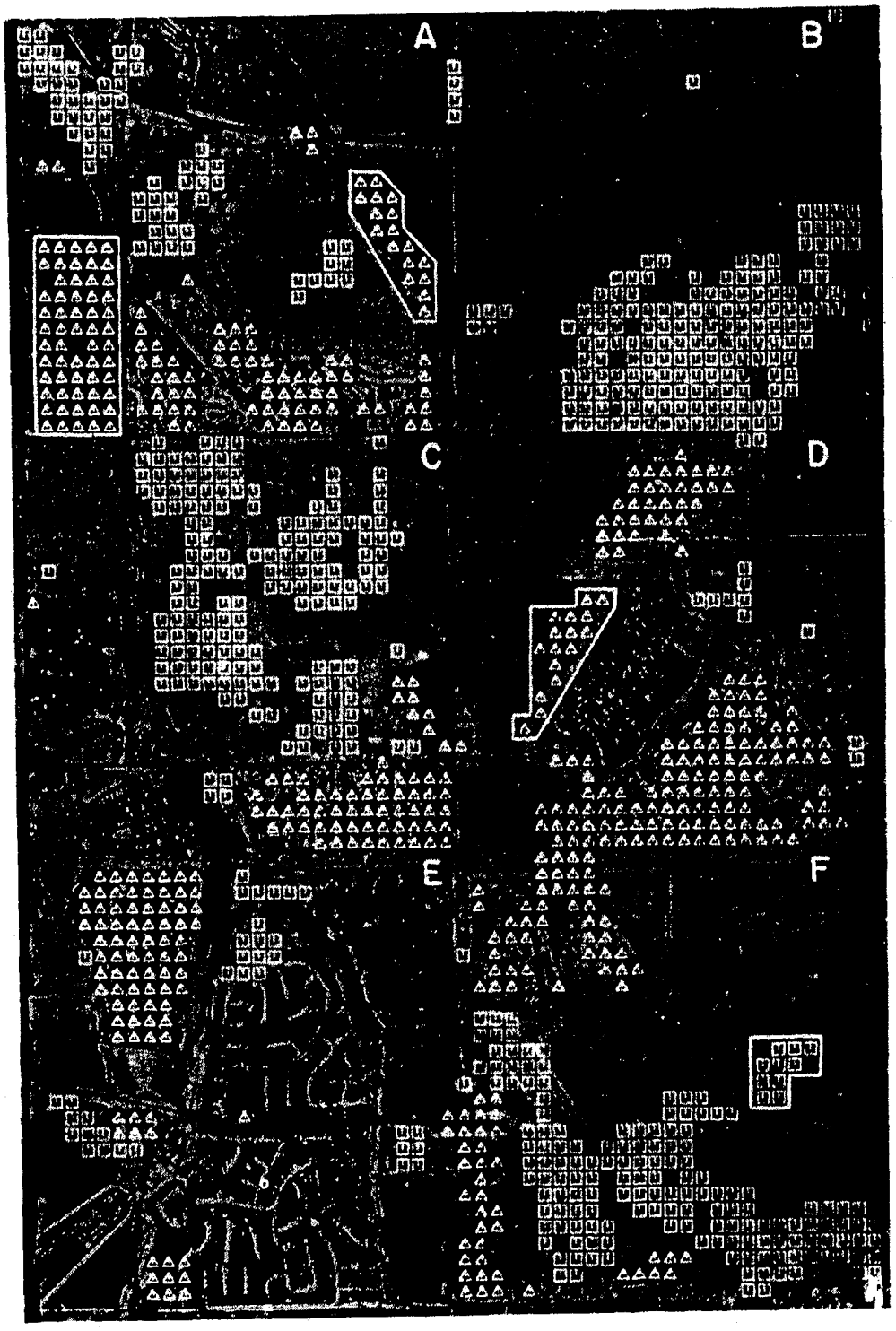



**Figure 9–6** Land-use change map produced using the MSS band 5 image differencing method. The results are superimposed on the October 15, 1978 aerial photography. Two symbols represent pixels found in the lower or upper tails of the change histogram. Some errors of commission are outlined. (From Jensen, 1981.)





OFEK06936

**TABLE 9-1** Change Detection Evaluation for an Area Near Denver, Colo. from October 1, 1976 to September 30, 1978 Using Landsat MSS Data[a]

a. *Band 5 spectral image differencing*

| | Change | No change | |
|---|---|---|---|
| Change | 973 | 810[b] | absolute accuracy: |
| No Change | 175[c] | 2362 | (973 + 2362)/4320 = 77% |

b. *Band 5 spectral image differencing plus texture differencing*

| | Change | No change | |
|---|---|---|---|
| Change | 1142 | 641[b] | absolute accuracy: |
| No Change | 180[c] | 2357 | (1142 + 2357)/4320 = 81% |

[a] These statistics were derived by comparing the results of Landsat digital change detection methods with a binary change/no-change map produced from the photointerpretations of the 1976 and 1978 panchromatic aerial photography.

[b] Errors of omission.

[c] Errors of commission.

*Source:* Jensen and Toll (1982).

mental and system multiplicative factors present. The better these factors are controlled, the higher the probability that accurate change detection will take place.

The mathematical expression of the ratio function is:

$$\Delta r_{ijk} = \frac{BV_{ijk}(1)}{BV_{ijk}(2)} \tag{9-2}$$

where $\Delta r_{ijk}$ = non-normalized change pixel value
$BV_{ijk}(1)$ = brightness value at time 1
$BV_{ijk}(2)$ = brightness value at time 2
$i$ = 1 . . . number of line
$j$ = 1 . . . number of column
$k$ = a single band (e.g., MSS 5)

Basically, a pixel that has not changed in land cover will have the same brightness value on both dates, yielding a ratio value of 1.0 that can be scaled to produce a midgray tone (the method of scaling the ratioed data is described in the section on band ratioing in Chapter 7). Areas of change in the multiple date imagery will have values either higher or lower than 1.0. Thus, as with the image differencing method, a change histogram is produced with the tails of the distribution containing the change information. The selection of thresholds is again based on empirical judgment (Friedman, 1978).

Ratio change detection using this logic was first reported by Todd (1977) for an area near Atlanta, Georgia, where 91% of the land-use change was identified correctly. Friedman and Angelici (1979) also used visual analysis of the ratio histograms to identify thresholds for mapping areas of urban and nonurban change in



OFEK06937

Los Angeles. Howarth and Boasson (1983) applied the ratio change detection technique in a CRT environment. Pixels that did not change in brightness value between dates were allocated a midgray tone (i.e., RGB = 127, 127, 127). Areas of change that had values either higher or lower than 1.0 were displayed in either light or dark tones that were directly proportional to the spectral intensity of the change that took place. Therefore, the lighter or darker a pixel appeared, the more the digital value changed between the two dates. They experienced good results for an area near Hamilton, Ontario, Canada, using MSS band 5 ratio data from 1974 and 1978. The band 5 ratio data were sensitive to cultural changes, while band 7 ratioed data were sensitive to changes in water conditions. Robinson (1979) criticized ratio change detection as being a statistically invalid method due to the non-Gaussian, bimodal distribution that is often produced when ratioing multiple-date data.

### *Classification Comparisons*

Several change detection alternatives evaluate land-use classifications produced either from each separate date of imagery or from the two dates of imagery as a set. Because accurate classification of urban land use on even a single date is difficult using Landsat MSS or TM data, care must be exercised in the application of these techniques.

**Postclassification comparison.** This alternative identifies change by comparing two independently produced classification maps. An algorithm simply compares the two classification maps utilizing class pairs specified by the analyst and generates a map indicating areas of change. By properly coding the classification results for date 1 and date 2, the analyst can produce change maps that show a complete matrix of changes. Therefore, it is possible to identify not only the pixels that have changed between the dates, but also the nature of that change (Howarth and Wickware, 1981).

The results of postclassification comparison in one study were considered a priori to be the superior change detection method and used as the standard for evaluating the results of other methods (Weismiller et al., 1977). Unfortunately, no ground truth was available to test the method. Conversely, others have found that this method produces poor results unless extremely accurate classifications of the individual dates are made (Stow et al., 1980). The accuracy of the method depends on the accuracy of the initial classifications, and any errors are compounded. Another study concluded that the heterogeneity of urban land use produced many "mixed" pixels, which lowered the accuracy of the individual classification maps. A symptom of the problem was that too much change was consistently identified (Toll et al., 1980).

**Spectral/temporal change classification.** This alternative detects change by performing a single classification on a multiple-date data set. If Landsat MSS data are used, the four bands of data for each of two dates (eight bands total) are analyzed at one time using pattern-recognition techniques. The method may employ supervised classification techniques, in which case the analyst selects "training" sites



OFEK06938

that are used to determine signatures for desired classes or unsupervised classification techniques, in which case like phenomena are clustered into similar classes based on their statistical properties. In either case, *change classes* should have significantly different statistics from nonchange classes.

Although this method is appealing because it requires only a single classification, it is a very complex classification. If all eight bands are used, some will be redundant in information content. Also, if clustering is performed, the cluster labeling is usually difficult. For example, 52 spectral classes were produced when a coastal Texas study area was analyzed using this method. Five information classes were synthesized from the 52 spectral classes, four of which represented change. The change map was judged unsatisfactory when compared to the reference data set (Weismiller et al., 1977).

## Preprocessing to Improve Change Detection

Many change detection studies preprocess the remotely sensed data prior to using it in one of the change detection methods just discussed.

**Low-frequency filtering.** One type of preprocessing is image smoothing where a spatial moving average enhances areas of homogeneity at the expense of high-frequency detail. For example, MSS band 7 data were (1) smoothed using a $3 \times 3$ moving-average filter, or (2) smoothed using a boundary preserving algorithm (Maxwell, 1976; Nagao and Matsuyama, 1979). The new images were then differenced and mapped. The methods were judged ineffective (Riordin, 1980). Conversely, other research found that data smoothed with a $3 \times 3$ spatial moving average performed almost as well as differenced MSS band 5 data (Royal, 1980).

**High-frequency filtering.** It is possible to apply a filter that enhances high-frequency detail in the scene. For example, a Laplacian enhancement may be applied to each local neighborhood. Such enhancements produce a sharp visual image but contain more noise than the original. Thus, there is a trade-off in deciding to what extent edges should be enhanced. In one study using such a procedure, the results were less accurate than using simple image differencing of MSS band 5 data (Royal, 1980).

**Texture transformations.** Texture is very important in the human interpretation of change in aerial photography and therefore offers the potential of improved accuracy when used as an element in machine-assisted change detection. Basically, it is possible to preprocess the remotely sensed imagery to yield multiple-date texture images. These images can then be analyzed using any of the change detection procedures described previously. Jensen (1982) analyzed the Denver, Colorado, Landsat data set to determine the utility of incorporating texture as well as spectral information in an image differencing change detection analysis of residential areas. He used the angular second moment (ASM) proposed by Haralick (1979) to create texture images from the 1976 and 1978 MSS band 5 images (see Chapter 7 for a discussion of


OFEK06939

these and other texture measures). The 1976 and 1978 texture images were differenced and the histogram analyzed to produce a change map showing areas where texture changed dramatically between 1976 and 1978 (Fig. 9–7). The texture change map complemented rather than duplicated information provided in the change map derived from MSS band 5 spectral characteristics alone (Fig. 9–6). The texture change map accurately identified many pixels of developing residential land use, especially conditions 4 and 6 in sections A, C, and D (outlines areas in Fig. 9–7). Unfortunately, change in land use from rangeland (category 1) to established residential (category 10) occasionally produced no significant change in texture, resulting in these areas not being identified. Thus, the texture change map complemented the spectral change map and should be used in conjunction with, but not alone for this study area. The differenced band 5 (Fig. 9–6) and the texture change map (Fig. 9–7) were combined into a single change map (Fig. 9–8). The absolute accuracy of this change map was 81%. The 4% increase was due to fewer omission errors (refer to Table 9–1b).

Analysis of where the stages of development cluster in three-space (band 6, $X$-axis; band 5, $Y$-axis; and texture on the $Z$-axis) provided insight into the nature of both spectral and textural change as a function of residential development stage. With the 1976 band 5 and 6 statistics displayed in Fig. 9–9a, notice the separation that exists between most stages, with the exception of rangeland (1) and partially landscaped residential (5). In 1978, there was still overlap between stages 1 and 5, however, the more serious problem was the complete overlap of stages 2 and 4 (Fig. 9–9a). Note in both dates the modest contribution of band 6 to separability. Band 5 accounted for the majority of the separation.

As the parallelepipeds are viewed from 0, 60, and 90°, the importance of the texture information becomes apparent. The edge-on view of 1976 MSS band 5 and texture data shown in Fig. 9–9c demonstrated that, although there was still some overlap between stages 1 and 5, there was substantial textural difference between these two classes. The most pronounced texture contribution, however, was shown in Fig. 9–9c; where the texture provided complete separation of stages 2 and 4 when used in conjunction with MSS band 5 data. Stages 1 and 5 were completely disjoint for the 1978 image.

In this study, Landsat spectral data and derived texture data provided complementary information, and when used together, increased change detection accuracy. The primary contribution of texture data was the measure of class heterogeneity. Specifically, texture data improved the detection of intermediate stages of residential development, particularly stages 4 and 6 in the model. Unfortunately, texture data alone cannot be used to adequately detect change from natural vegetation (1) to either partially (5) or fully landscaped (10) residential areas.

**Principal component analysis (PCA).** Principal component analysis applied to multispectral data produces a series of linear transformations of the observed variables that result in a new, smaller set of mutually orthogonal variables (refer to Chapter 7 for a discussion of how PCA is performed). Because each consecutive new variable accounts for the maximum amount of variability within the original



OFEK06940

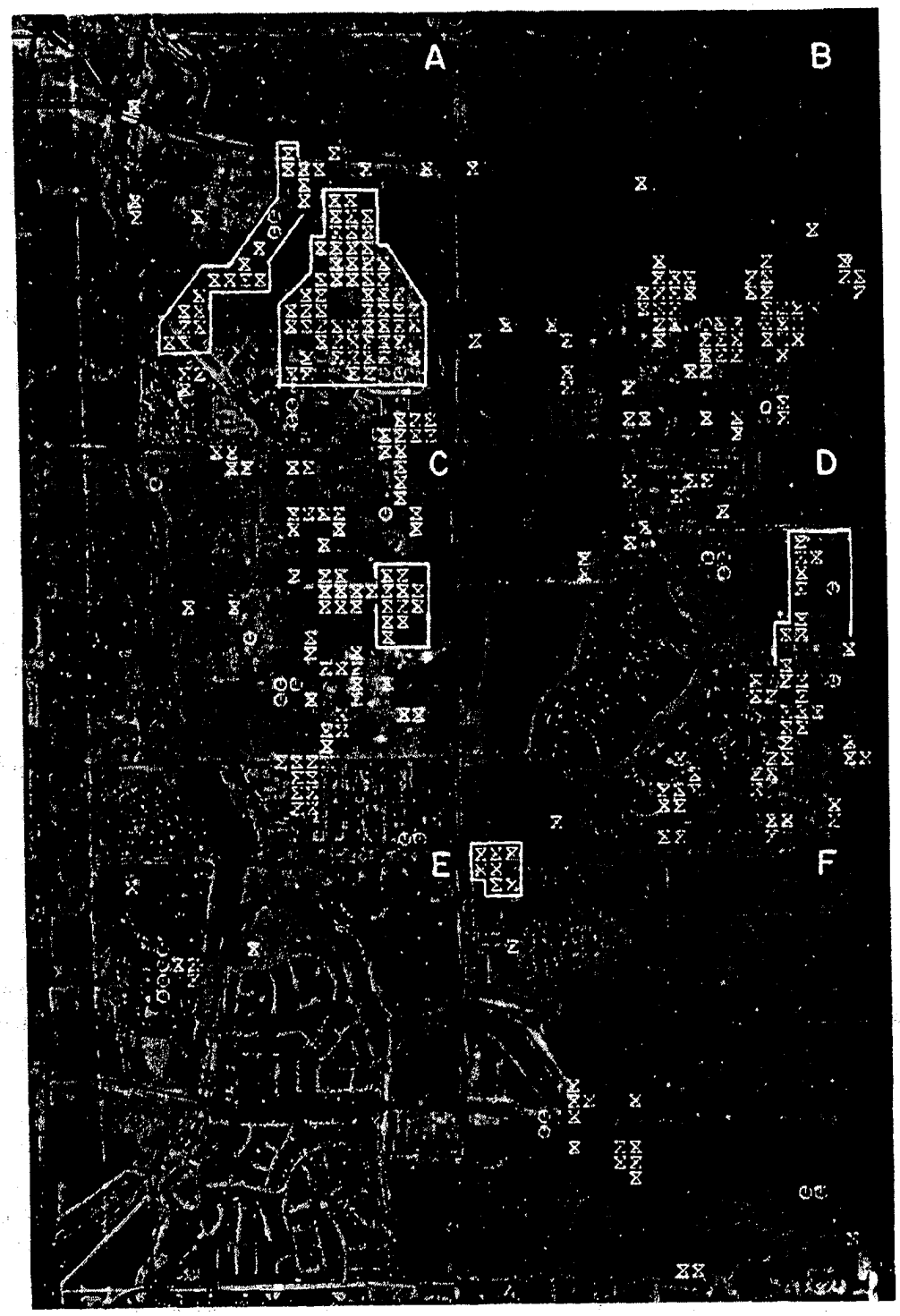



**Figure 9–7** Land-use change map overlay produced using the texture differencing method. Some new areas of residential development are correctly identified (outlined); however, considerable change from natural vegetation to partially or fully landscaped residential is omitted. Consequently, it is preferable to use this method in conjunction with other methods (see Fig. 9–8).





OFEK06941

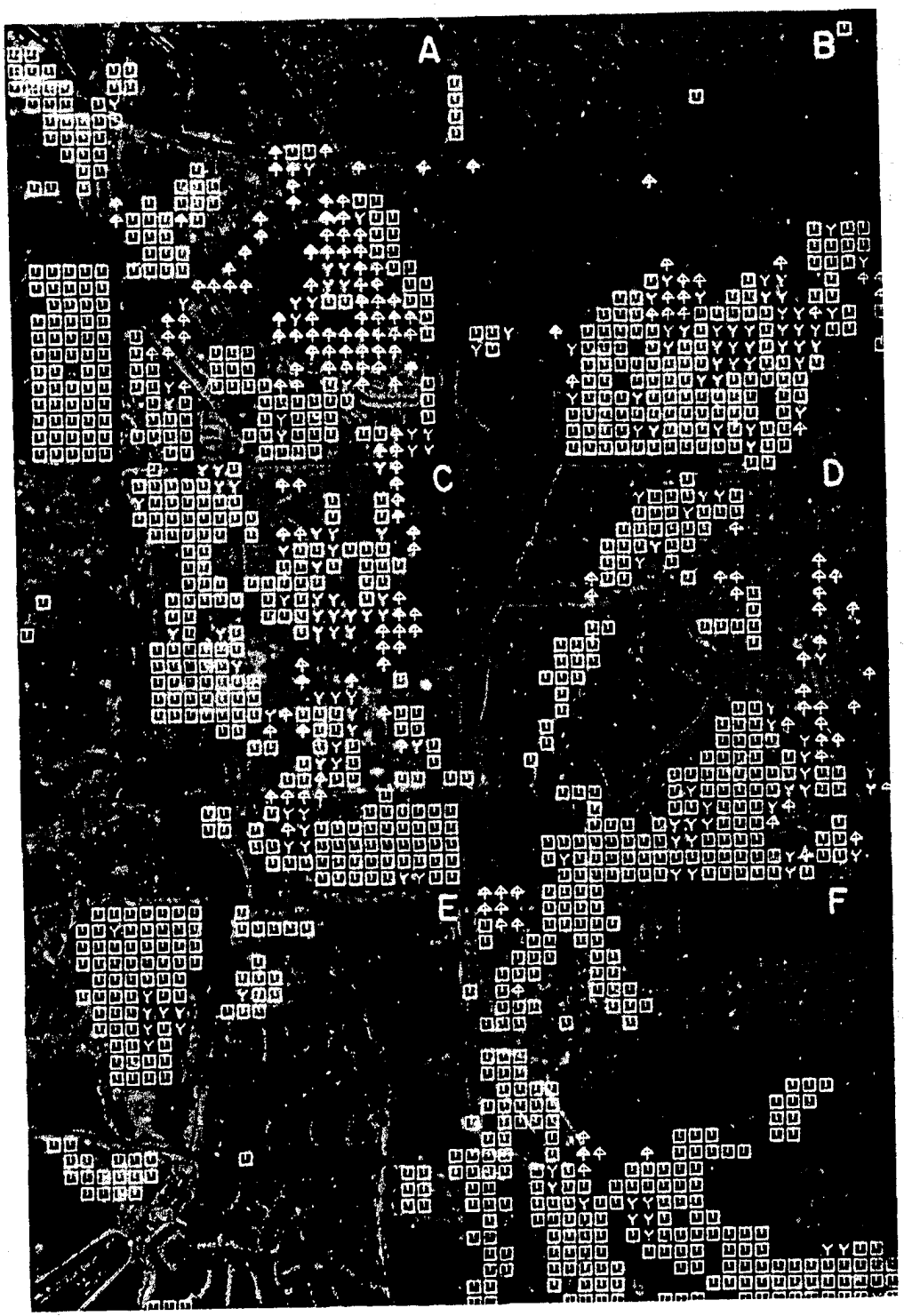



**Figure 9–8** Land-use change map produced by combining the MSS band 5 image differencing change map in Fig. 9–6 with the texture change map in Fig. 9–7. Pixels classified as change by the MSS band 5 image differencing are symbolized with a box, while those classified as change by the texture method are shown as an arrow. Pixels classified as change by both methods are symbolized with a "y" representing "yes." Note that the two change maps complement one another. (From Jensen, 1981.)





OFEK06942

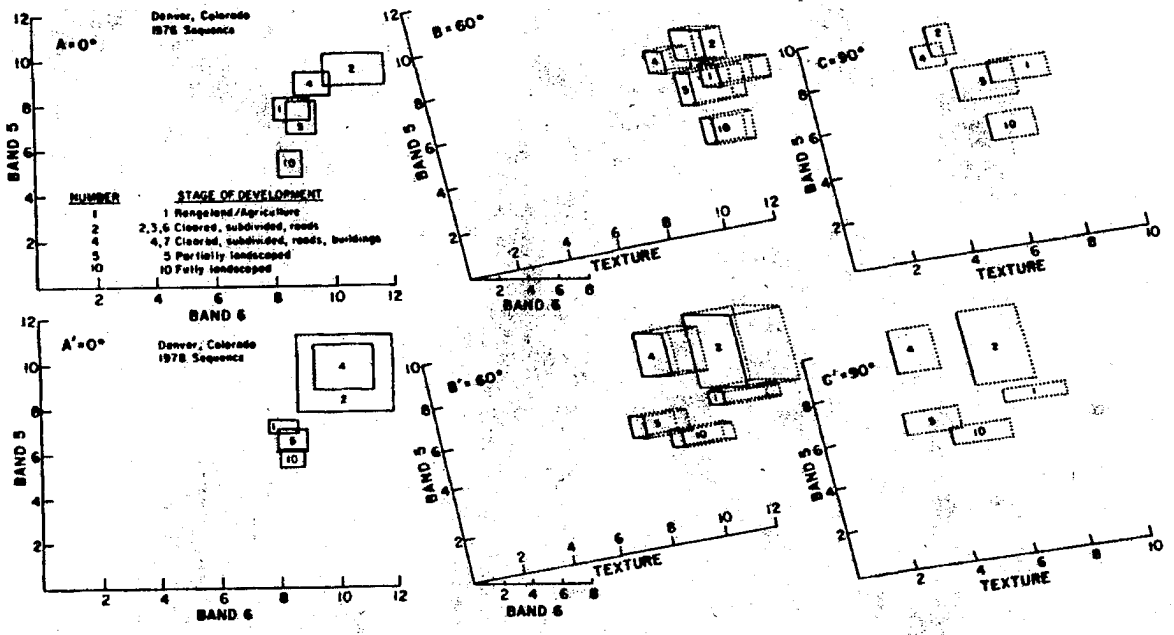



**Figure 9–9** Stages of residential development at the urban fringe displayed using parallelepipeds of spectral (MSS5, MSS6) and textural statistics obtained from the 1976 and 1978 Landsat MSS imagery of the Fitzsimmons 7.5-minute quadrangle near Denver, Colorado. The parallelepipeds were constructed using the mean ±1 standard deviation for each spectral or textural feature and each of the five stages of residential development. The top row provides the 1976 statistics rotated from 0 to 60, then 90°, and the bottom row presents the 1978 statistics. Of particular importance is the counterclockwise stage of development pattern found when band 5 and texture are plotted together in C and C'. Also, note the degree of separation between stages, especially in the 1978 data (i.e., in C'). (From Jensen and Toll, 1982.)

data, the transformation provides uncorrelated variables, with the first to enter component generally containing 90 to 95% of the variance. Thus, principal component analysis may reduce the dimensionality of the change detection problem without losing useful information.

In several change detection studies, the four-dimensional Landsat data of *each* date were preprocessed using principal component analysis. In each case, (1) the mean and variance of a sample of observations from a given subscene were obtained, (2) eigenvalues and eigenvectors were extracted, (3) the entire population was transformed using the matrix obtained in step 1, and (4) histograms were analyzed using image differencing techniques and a decision rule applied. In an analysis of imagery for Orlando, Florida, land was separated from water using component 1 and urban versus nonurban identified using component 2. This method was judged superior to simple ratio techniques also applied to the scene (Friedman, 1980). Conversely, Toll et al. (1980) applied principal component transformations to images of Denver, Colorado, and Richmond, Virginia, study areas yielding poor change detection results compared to simple image differencing of band 5 or 7 data.

Byrne et al. (1980) used two four-channel Landsat scenes of the same area of



OFFK06943

New South Wales that were recorded on different dates. These data were registered and treated as a single eight-dimensional data set as input to the principal component analysis. They found that component 1 represented primarily the unchanged land cover as measured by the infrared bands. It was also a good indicator of terrain topography and roughness. Component 2 represented unchanged land cover measured in the visible region. Components 3 to 5 contained change information. Components 6 to 8 contained random noise. They concluded that principal component analysis provides an effective way of identifying areas in which change has occurred between two four-channel multispectral scanner images.

### *Change Vector Analysis*

When land undergoes a change or disturbance, its spectral appearance may change. If two spectral variables are measured for the area both before and after change occurs and then are plotted on the same graphic, a diagram such as Fig. 9–10a might result. The vector describing the direction and magnitude of change from the first to the second date is a spectral change vector. The decision that a change has occurred is made if a threshold is exceeded. Figure 9–10b illustrates a case in which no change would be detected, while Figs. 9–10c and d illustrate the other half of the information contained in the change vector, namely its direction. Direction contains information about the type of change. For example, the direction of change due to clearing should be different than change due to regrowth of vegetation.

This method was applied successfully to forest change detection in northern Idaho (Malila, 1980). The data were preprocessed using an agricultural transformation that collapsed the four Landsat features of each date to just two features, brightness and greenness.

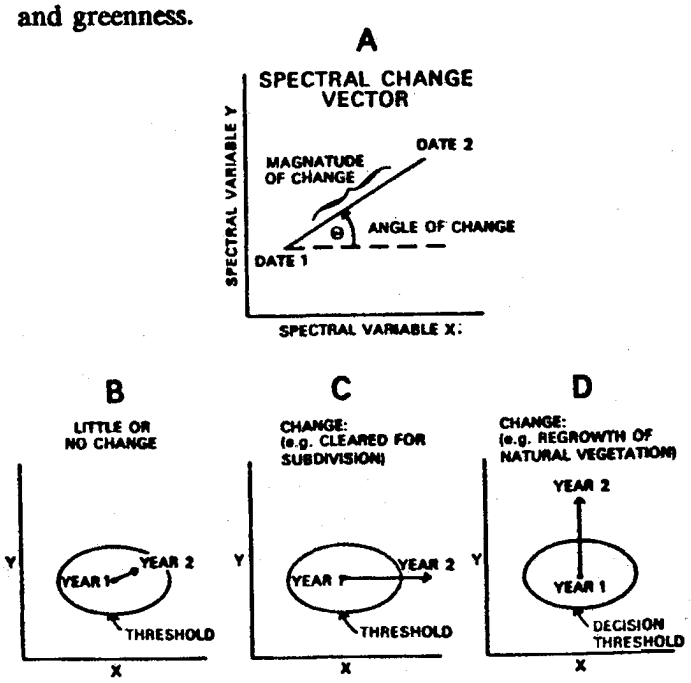



**Figure 9–10** A schematic diagram of the spectral change vector change detection method (after Malila, 1980).



OFEK06944

## CONCLUSIONS

Digital change detection is a difficult task to perform accurately. The results will not be as accurate as those produced by a photointerpreter analyzing large-scale aerial photography on individual dates and transferring this information onto a base map. Nevertheless, because manual change detection is exhausting, difficult to replicate, and incurs substantial data acquisition costs, researchers continue to seek improved digital change detection alternatives.

Spatial, spectral, and temporal constraints affect digital change detection. Assuming that the imagery can be placed in a configuration that allows digital analysis, the selection of an appropriate change detection algorithm takes on significance. There exist a variety of algorithms from which to choose. Image differencing or ratioing of spectral data is practical but may be too simple to identify the variety of change in a complex scene. Classification comparison methods are useful only if accurate land-use classifications can be obtained. Principal component analysis appears to offer the most promising change detection technique involving image preprocessing. Change vector analysis also appears promising.

Persons conducting digital change detection must be intimately familiar with the environment under study, the quality of the data set, and the characteristics of change detection algorithms. Also, much work remains to identify optimum change detection algorithms for specific geographic areas and problems.

## REFERENCES

BANNER, A., and T. LYNHAM. 1981. "Multi-temporal Analysis of Landsat Data for Forest Cutover Mapping—A Trial of Two Procedures," *Proceedings*, Seventh Canadian Symposium on Remote Sensing, Winnipeg, Manitoba, pp. 233–240.

BYRNE, G. F., P. F. CRAPPER, and K. K. MAYO. 1980. "Monitoring Land-Cover Change by Principal Component Analysis of Multitemporal Landsat Data," *Remote Sensing of Environment*, Vol. 10, pp. 175–184.

CARLSON, D., and J. RUBINGH. 1979. *Agricultural Land Conversion in Colorado*. Denver: Colorado Department of Agriculture.

CHRISTENSON, J. W. 1977. *Preliminary Design Requirements for a Census/Urbanized Area Application System Verification and Transfer*, Final Report NASA Contract NAS5–23412. Greenbelt, Md: Goddard Space Flight Center.

FRIEDMAN, S. Z. 1978. "Use of the Ratio Threshold Classifier in Mapping Urban/Nonurban Land Cover from Landsat MSS Imagery," unpublished Master's thesis, University of Wisconsin–Madison.

FRIEDMAN, S. Z. 1980. *Mapping Urbanized Area Expansion through Digital Image Processing of Landsat and Conventional Data*, Publication 79–113. Pasadena, Calif.: Jet Propulsion Laboratory.

FRIEDMAN, S. Z., and G. L. ANGELICI. 1979. "The Detection of Urban Expansion from Landsat Imagery," *The Remote Sensing Quarterly*, Vol. 5, pp. 3–17.



OFEK06945

HARALICK, R. M. 1979. "Statistical and Structural Approaches to Texture," *Proceedings of the IEEE*, Vol. 67, pp. 786–804.

HOWARTH, P. J., and E. BOASSON. 1983. "Landsat Digital Enhancements for Change Detection in Urban Environments," *Remote Sensing of Environment*, Vol. 13, pp. 149–160.

HOWARTH, P. J., and G. M. WICKWARE. 1981. "Procedures for Change Detection Using Landsat Digital Data," *International Journal of Remote Sensing*, Vol. 2, pp. 277–291.

JENSEN, J. R. 1981. "Urban Change Detection Mapping Using Landsat Digital Data," *The American Cartographer*, Vol. 8, pp. 127–147.

JENSEN, J. R., and D. L. TOLL. 1982. "Detecting Residential Land-Use Development at the Urban Fringe," *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 629–643.

MALILA, W. A. 1980. "Change Vector Analysis: An Approach for Detecting Forest Changes with Landsat," *Proceedings*, Machine Processing of Remotely Sensed Data. West Lafayette, Ind.: Laboratory for the Applications of Remote Sensing.

MAXWELL, E. L. 1976. "Multivariate Systems Analysis of Multispectral Imagery," *Photogrammetric Engineering and Remote Sensing*, Vol. 42, pp. 1173–1186.

MILAZZO, V. A. 1980. "A Review and Evaluation of Alternatives for Updating U.S. Geological Survey Land Use and Land Cover Maps," *U.S. Geological Survey Circular 826*.

NAGAO, M., and T. MATSUYAMA. 1979. "Edge Preserving Smoothing," *Computer Graphics and Image Processing*, Vol. 9, pp. 394–407.

PAUL, C. K., and A. C. MASCARENHAS. 1981. "Remote Sensing in Development," *Science*, Vol. 214, pp. 139–145.

RIORDIN, C. J. 1980. "Non-Urban to Urban Land Cover Change Detection Using Landsat Data," Report #1 NASA Contract NAS5–25696, Earth Resources Dept., Colorado State University, Fort Collins.

ROBINSON, J. W. 1979. *A Critical Review of the Change Detection and Urban Classification Literature*, Technical Memorandum 79/6235. Silver Springs, Md.: Computer Sciences Corporation.

ROYAL, J. A. 1980. *Change Detection Method Development: Census Urban Area Application Pilot Test*, Final Report Contract NAS5–35707. Beltsville, Md.: General Electric Company.

SHEPARD, J. R. 1964. "A Concept of Change Detection," *Photogrammetric Engineering*, Vol. 30, p. 649.

STOW, D. A., L. R. TINNEY, and J. E. ESTES. 1980. "Deriving Land Use Land Cover Change Statistics from Landsat: A Study of Prime Agricultural Land," *Proceedings*, 14th International Symposium on Remote Sensing of Environment. Ann Arbor, Mich.: ERIM, pp. 1227–1237.

TODD, W. J. 1977, "Urban and Regional Land Use Change Detected by Using Landsat Data," *Journal of Research*, U.S. Geological Survey, Vol 5, 529–534.

TOLL, D. L., J. A. ROYAL, and J. B. DAVIS. 1980. "Urban Area Update Procedures Using Landsat Data," *Technical Papers*, Annual Meeting of the American Society of Photogrammetry.

WEISMILLER, R. A., S. J. KRISTOF, D. K. SCHOLZ, P. E. ANUTA, and S. A. MOMIN. 1977. "Change Detection in Coastal Environments," *Photogrammetric Engineering and Remote Sensing*, Vol. 43, pp. 1533–1539.



OFEK06946

# 10

# The Interface of Remote Sensing and Geographic Information Systems

## INTRODUCTION

There exists in the world today a tremendous amount of cartographic, thematic information. More and more of this information is being maintained in *geographic information systems* (GIS) that are designed to accept large volumes of spatial data derived from a variety of sources. The GIS efficiently store, retrieve, manipulate, analyze, and display these data according to user-defined specifications (Marble et al., 1983). This cartographic information is usually very static in nature, with most being collected on a single occasion and then archived (Curran, 1985).

Remote sensing also produces a tremendous amount of information. However, remote sensing systems usually collect data not just on a single date, but on multiple dates, allowing the analyst to not only inventory, but also to monitor. The ability to monitor development through time provides valuable information about the processes at work. Furthermore, remote sensing often provides valuable information about certain biophysical measurements (e.g., object temperature, biomass, height) which could be of significant value in modeling the environment. Unfortunately, such valuable remote sensing information is not often used because it is very difficult to interrelate the remote sensing information with other types of spatially distributed (cartographic) information. Many observers feel that the full potential of both geographic information systems and remote sensing can best be achieved if the technologies are integrated (Shelton and Estes, 1981; Short, 1982). Some have suggested that the real utility of remotely sensed data is intimately related to whether or not the data can be associated with other spatial information, usually stored in a GIS format (Simonett et al., 1977).





OEEK06947

Over two decades ago, Garrison et al. (1965) described the significant potential for integrating remotely sensed information with other nominal or interval scale information in a GIS for modeling purposes. Since this time GIS technology has continued to develop. There now exist scores of geographic information systems, many of which are summarized in Calkins and Tomlinson (1977), Knapp and Rider (1979), Marble et al. (1983), and Hansen (1985). In fact, Tomlinson (1984) reported that approximately 1000 GIS and automatic cartography systems were in operation in North America in 1983 and 4000 are forecast to exist by the end of the decade. Some GIS can incorporate remotely sensed information readily, while others experience great difficulty incorporating it. It is beyond the scope of this chapter to review all GIS available and their attributes. Rather it is instructive to identify the fundamental generic attributes common to most GIS and suggest how the remotely sensed data can be incorporated and used within the GIS framework.

## FUNDAMENTAL GEOGRAPHIC INFORMATION SYSTEM CONCEPTS

All data that can be mapped has both locational $(x, y)$ and nonlocational (i.e., attribute) characteristics. For example, a feature may exist at an $x, y$ location and possess $z$ attributes. These attributes can be both qualitative (e.g., the land use at a location) or quantitative (e.g., the elevation at the same location). In addition, the attributes at the location can be monitored through time. These three components—location, attribute, and time—represent the content of most GIS (Fig. 10–1). How are these data stored (encoded) digitally?

### Data Encoding

There are four fundamental types of geographic data to be stored in a GIS; points, lines, polygons and surfaces (Deuker, 1979). A summary of three of these data types and their methods of representation is shown in Fig. 10–2. These data can be stored in a number of formats, including (1) traditional Cartesian $x, y$ coordinates; (2) in a topological format as nodes, links (line segments), and polygons; or (3) in a grid (raster) format.

The more traditional encoding of points, lines, and polygons is based on the use of Cartesian coordinates such as longitude/latitude founded on the principles of Euclidean geometry. This is often called *polygon* or *vector coding*. Figure 10–3 shows conceptually how a typical map composed of points, lines, and polygons is encoded from its original analog map format into a digital format through the process of digitization. Some of the more popular *polygonal GIS* in the United States are the Map Overlay System (MOSS; Lee, 1983), ARC-INFO (Dangermond, 1983), and Intergraph.

The *polygonal GIS* is ideal for the incorporation of thematic information derived from the visual analysis of aerial photography or other types of analog remote sensor



OFFK06948

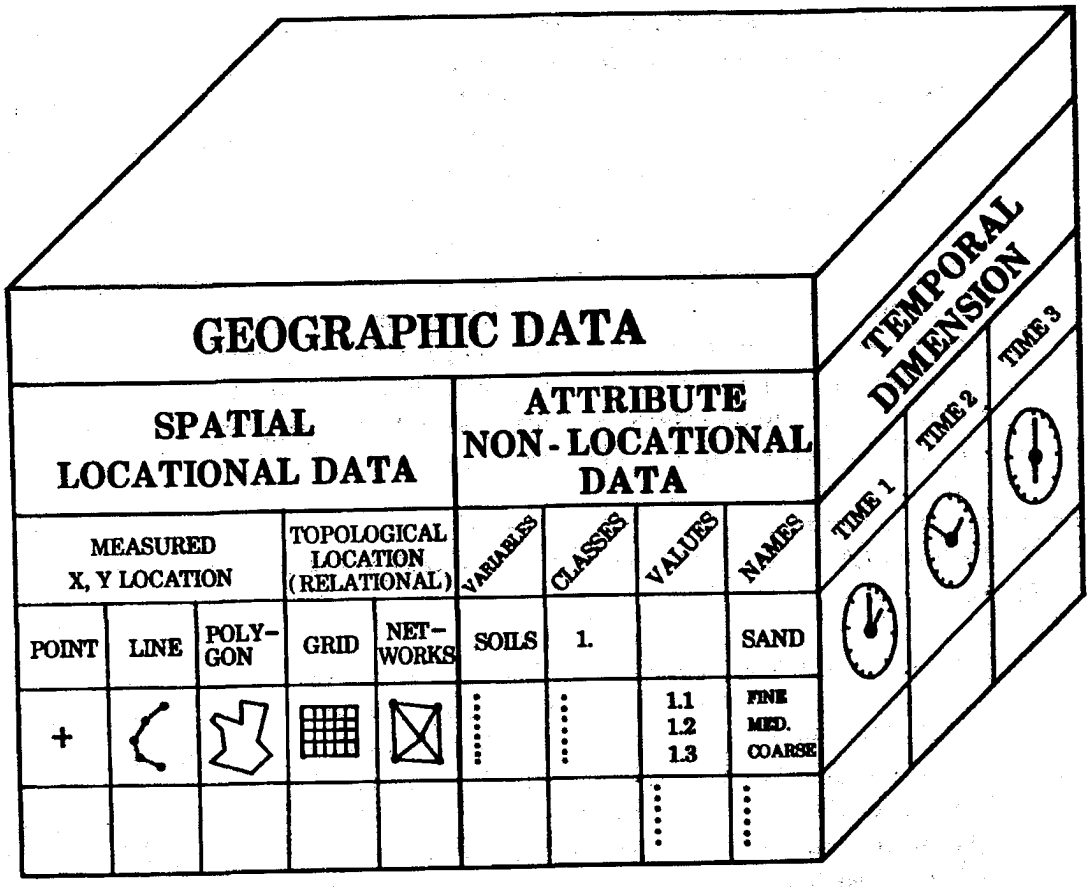



**Figure 10–1** Three conceptual components of a geographic information system (GIS). (After Dangermond, 1983.)

data. The overlays produced in the interpretation process become simply another type of map which can be systematically digitized into the GIS. It is much more difficult to incorporate remotely sensed information obtained in a digital raster format (such as land-use maps derived from digital Landsat MSS or TM data) into the polygon GIS format since the fundamental data structures are different. To accomplish this goal it is necessary to perform a raster-to-polygon conversion on the gridded remotely sensed information. This is no trivial matter, as it can consume considerable computer resources.

In addition to the expression of geographic phenomena using Cartesian coordinates, one can also use principles of graph theory involving *topological* relationships to express the relative location of various map elements. The first and most commonly used GIS system applying graph theory to maps is called dual independent map encoding (DIME) and was developed by the U.S. Bureau of the Census (Marble et al., 1984). DIME is actually a combined $x$, $y$ coordinate and topological encoding system. It involves topologically structuring the graph elements of a map (i.e., the



OFFK06949

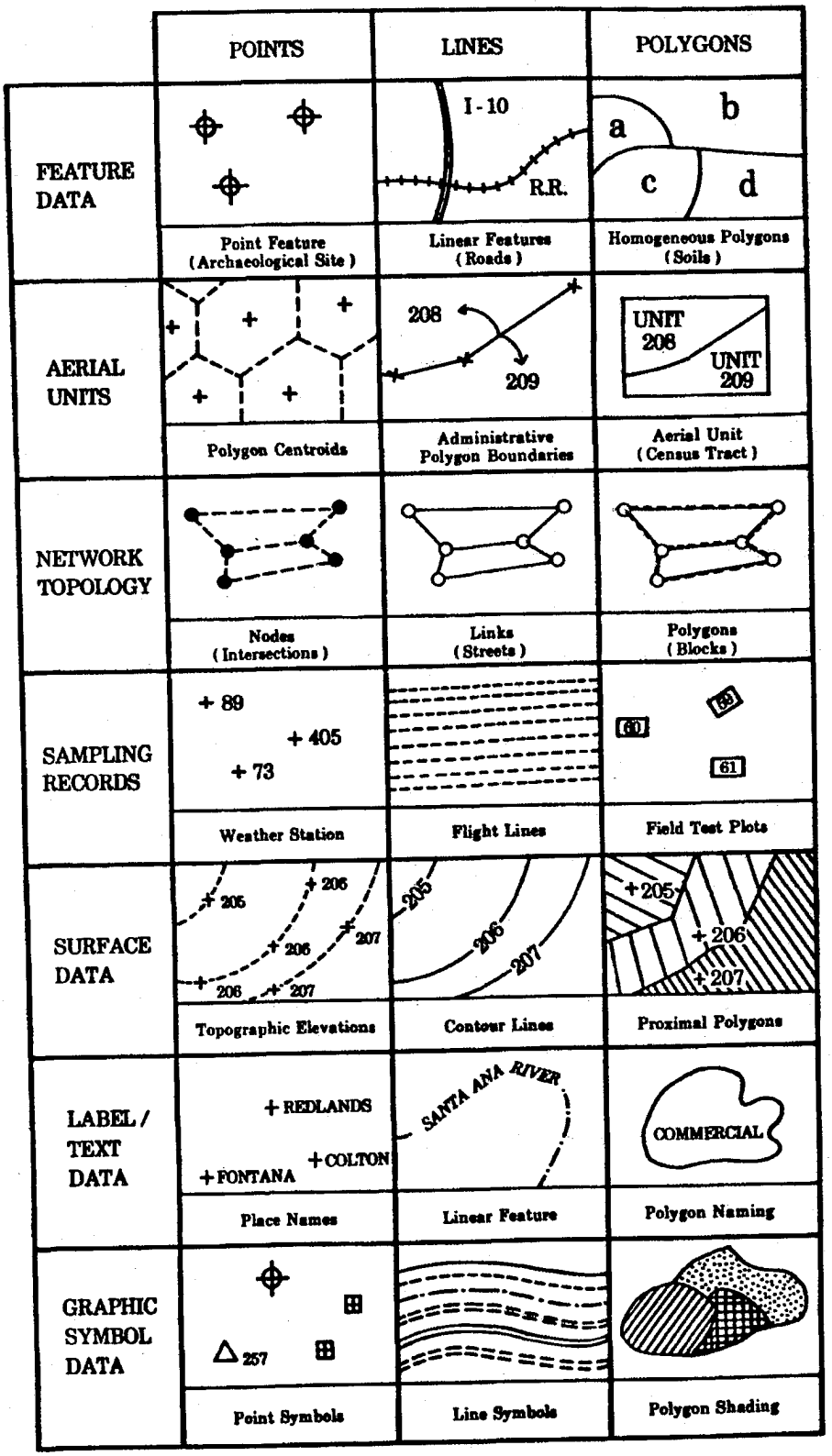
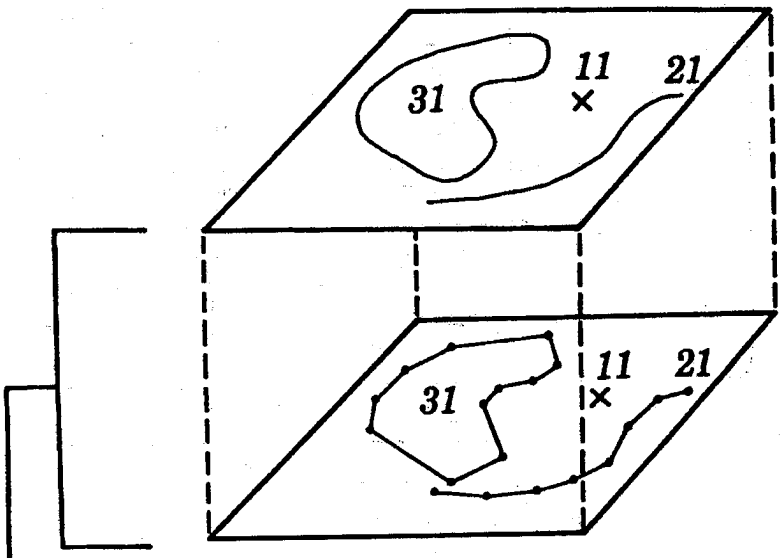



Figure 10–2 Geographic data types and various methods of representation. (After Dangermond, 1983.)





OFEK06950




| | FEATURE NUMBER | LOCATION |
|---|---|---|
| POINT | 11 | X, Y (SINGLE POINT) |
| LINE | 21 | $X_1Y_1, X_2Y_2, \ldots\ldots\ldots\ldots X_nY_n$ (STRING) |
| POLYGON | 31 | $X_1Y_1, X_2Y_2, \ldots\ldots\ldots\ldots X_1Y_1$ (CLOSED LOOP) |

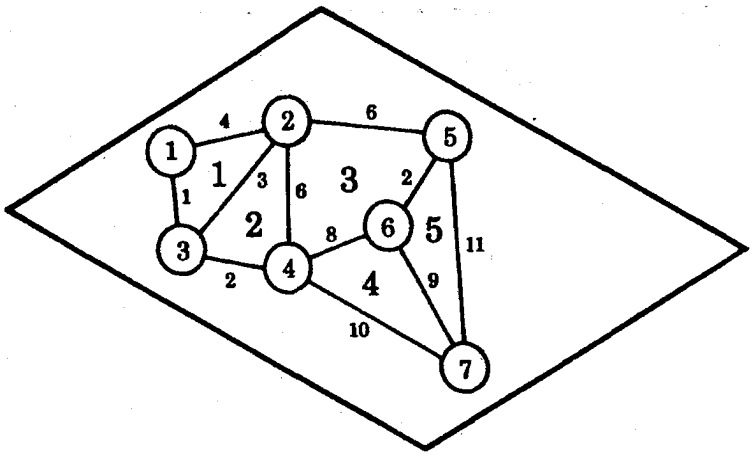
**Figure 10–3** Encoding of point, line, and polygon features using polygon (vector) Cartesian coordinate digitization techniques. (After Dangermond, 1983.)

map nodes, line segments, and polygons). Figure 10–4 shows how a typical polygon map can be abstracted into seven nodes, 11 links (line segments), bounding five basic polygons. By numbering these links and associating them with nodes as well as polygons on the right and left, a basic map notation system can be derived. By adding $x$, $y$ coordinate encoding at each of the nodes, one has a dual system for spatially identifying all elements of a map. This facilitates not only analysis using $x$, $y$ coordinates but also use of mathematics associated with graph theory involving networks, spatial aggregation, and so on (Dangermond, 1983).

Point, line, or polygonal data can also be encoded using a *grid* (often called a



OFEK06951




CODED NETWORK MAP

| LINK # | RIGHT POLYGON | LEFT POLYGON | NODE 1 | NODE 2 |
|---|---|---|---|---|
| 1 | 1 | 0 | 3 | 1 |
| 2 | 2 | 0 | 4 | 3 |
| 3 | 2 | 1 | 3 | 2 |
| 4 | 1 | 0 | 1 | 2 |
| 5 | 3 | 2 | 4 | 2 |
| 6 | 3 | 0 | 2 | 5 |
| 7 | 3 | 3 | 5 | 6 |
| 8 | 4 | 3 | 6 | 4 |
| 9 | 5 | 4 | 7 | 6 |
| 10 | 4 | 0 | 7 | 4 |
| 11 | 5 | 0 | 5 | 7 |

TOPOLOGICALLY CODED NETWORK & POLYGON FILE

| NODE # | X COORDINATE | Y COORDINATE |
|---|---|---|
| 1 | 23 | 8 |
| 2 | 17 | 17 |
| 3 | 29 | 15 |
| 4 | 26 | 21 |
| 5 | 8 | 26 |
| 6 | 22 | 30 |
| 7 | 24 | 38 |

X, Y COORDINATE NODE FILE

**Figure 10–4** Example of an $x$, $y$ coordinate node file created using a topological data structure. (After Dangermond, 1983.)





OFEK06952

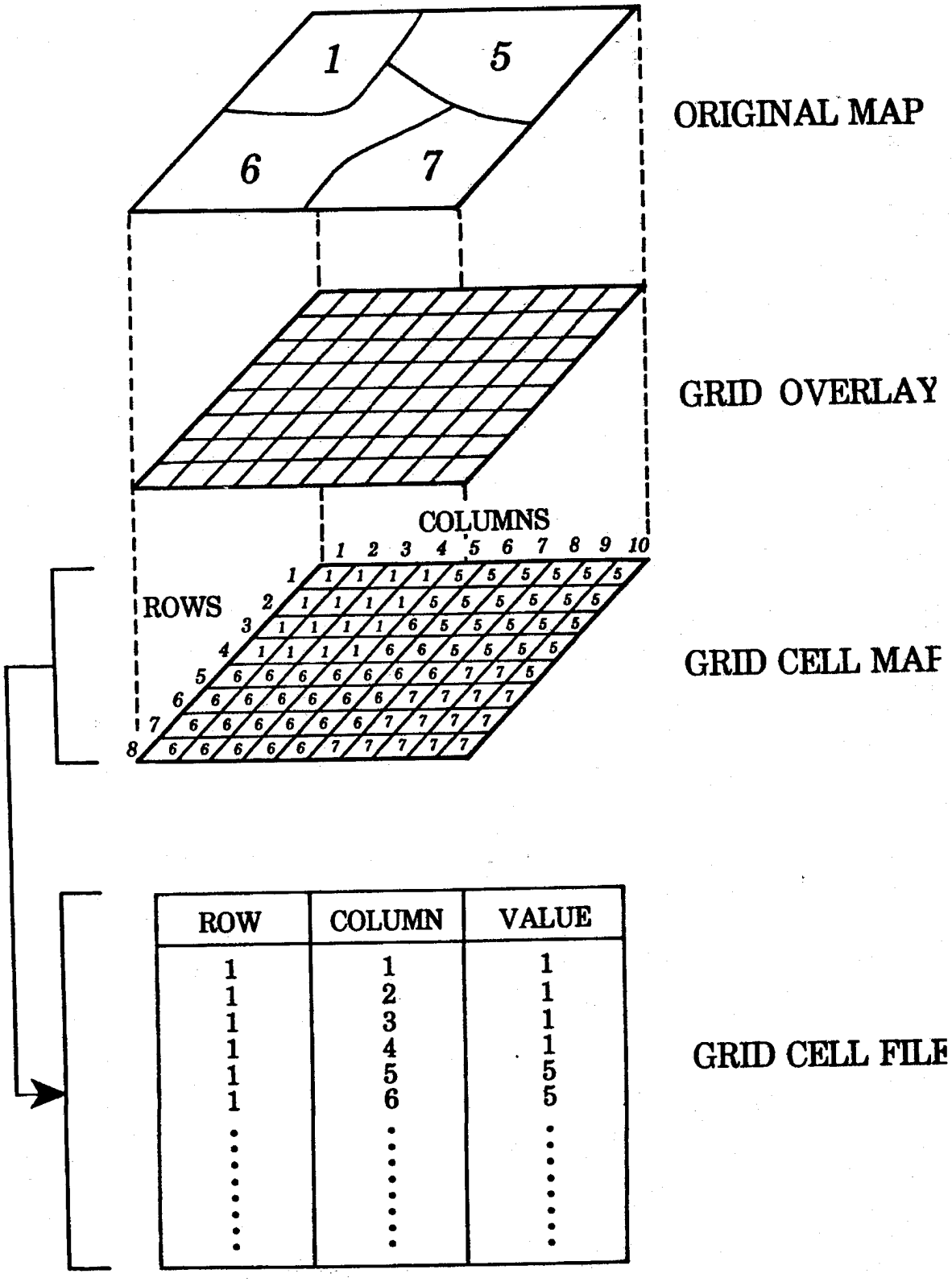



Figure 10–5 Basic structure of a grid-based GIS file. (After Dangermond, 1983.)





OEEK06953

*raster*) cartographic data structure (Fig. 10–5). Grid-cell coding actually creates a matrix that is superimposed over the terrain such that the attribute information is collected within a systematic array of grid cells. These cells are often referred to as picture elements (pixels) and analyzed and displayed as if they were images in a remote sensing data set. Grid-based geographic information systems usually have little difficulty incorporating information derived from digital remotely sensed data. All that is necessary is that the GIS files and the remote sensing–derived files be referenced to the exact coordinate location on the ground. Fig. 10–6 illustrates how a number of remote sensing–derived files can be placed in registration with other cartographic information using the raster format. To bring these polygon-based files into congruence with the raster remote sensor data, it is necessary to perform a vector-to-raster conversion. Again, this can consume computer resources. Some of the more popular grid-based GIS systems are MAP (Berry, 1981; Berry and Tomlin,

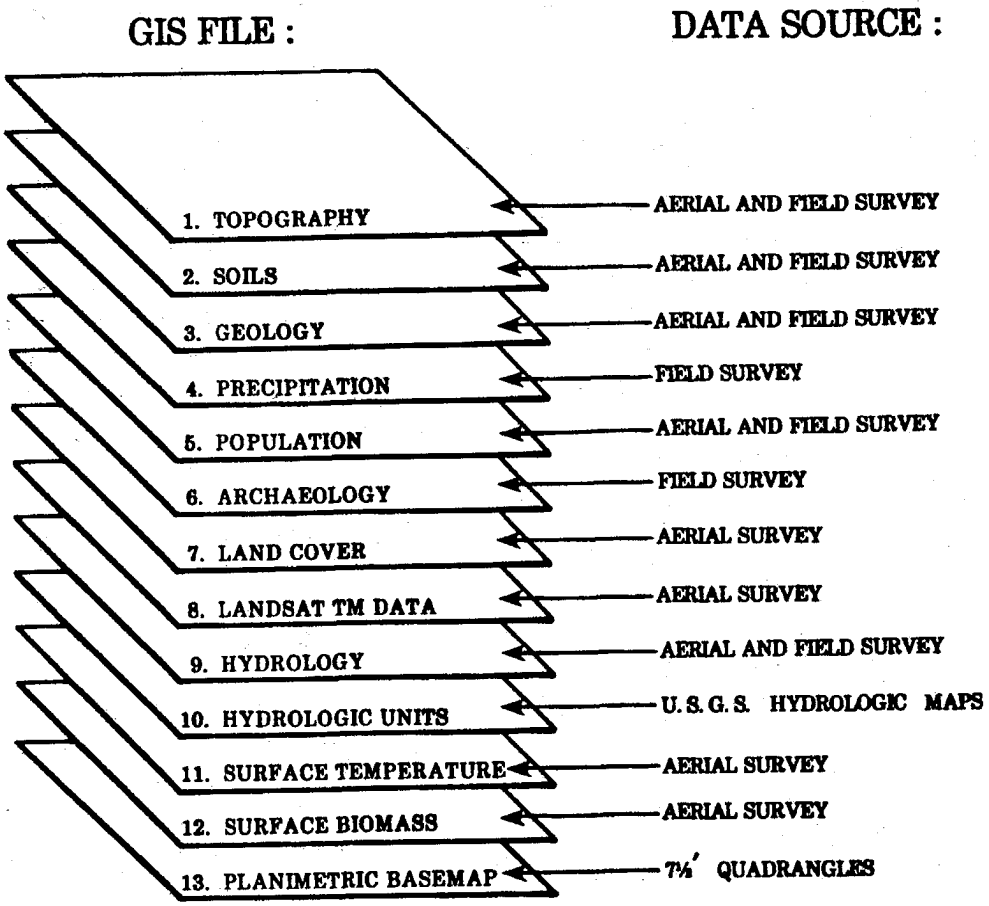



**Figure 10–6** A computerized geographic information system can be conceptualized as a base map accompanied by numerous registered overlays (files). This example illustrates a grid-based GIS where 13 matrices are in near-perfect registration. Information in the data base was derived from a variety of sources.



OFEK06954

1981), ELAS (Whitley et al., 1981), IBIS (Zobrist et al., 1978), and IDIMS (ESL, 1982).

Generally, *polygon* encoding more accurately defines the boundaries of point, line, and polygonal data compared with the *grid* method of digitization. However, if the grid mesh is extremely small in dimension over a given region, it is also possible to maintain very accurate polygon shape characteristics using this data structure.

Data can be encoded into one of the three cartographic data structures using either manual or automatic techniques. The most tedious is manual encoding involving an analyst who must physically encode each point, line, and polygon using a coordinate digitizing system. When the Cartesian method discussed previously is used, this may even entail "double digitizing" of certain line segments, especially those between adjacent polygons. This double digitizing may create serious error, often called "slivers," to be introduced into the data set if exactly the same points are not digitized. In fact, this problem of double digitization is precisely the reason many polygon-based GIS now incorporate topological digitizing schemes which do not require that the common boundary between adjacent polygons be digitized twice. Rather, the polygon coordinates are reconstructed by the computer using the node, line segment and polygon information.

There now exist expensive instruments that will automatically convert hard-copy map information into any of these formats. The two main approaches are usually referred to as automatic line following and scanning. Most users in North America currently prefer the scanning approach (Marble et al., 1983). The linear photodiode method of scan digitization described in Chapter 2 is among the most popular. In addition, very sophisticated software must be available to perform pattern recognition on the scanned information to appropriately label the points, nodes, and polygons and place them in one of the cartographic data structures discussed.

### Data Management

Once the spatial data have been encoded they must be stored, yet be ready for immediate access and manipulation. Each variable (often called a *GIS layer*) is archived in a computer-compatible digital format as a geographically referenced plane in the geographic information system data base. The data base can contain any type of information that is spatially distributed, ranging from socioeconomic (e.g., population density), to climatological (e.g., ppm ozone over the city), to fundamental biophysical variables (e.g., surface temperature). When digitally registered to one another, they form a data bank composed of $n$ layers which can be queried to answer questions (Fig. 10–6).

Because each polygon can have a unique shape, its storage, retrieval, and manipulation in a polygonal data structure is complex and expensive (Calkins and Tomlinson 1977). Therefore, Curran (1985) suggests that the grid is the most popular data structure, as it is cheap to store, retrieve, and manipulate and can be of any dimension from the 10 × 10 km unit used for the United Kingdom land characteristic data base (Ball et al., 1983) to the 30 × 30 m unit of a Landsat TM pixel. Conversely



OFFK06955

Avery and Berlin (1985) state that polygon coding requires less computer storage space than does the grid coding. Thus there continues to be debate over which is the most economical and efficient data format.

Ideally, the data base files reside in CPU memory (RAM) and are immediately accessible for computation and manipulation. However, because of the tremendous amount of information contained in the files, they are usually stored on relatively fast hard disk drives. The worst case is when the data are stored off-line on tape or other media and must be loaded onto the hard disk when needed. Improvements in optical disk storage during the next decade will make the economical storage and access of extremely large GIS data banks possible.

### *Data Manipulation*

To extract meaningful information from a GIS data base, one must be able to query it and ask logical questions, a process commonly referred to as *data manipulation*. Berry and Tomlin (1981) and Dangermond (1983) provide an extensive review of the editing and data manipulation alternatives available on sophisticated GIS systems. Here, we will only touch on the most fundamental aspects of data manipulation and provide diagrammetric examples whenever possible.

First, it is assumed that the GIS has the capability to make scale and projection changes, remove distortion, and perform coordinate rotation and translation if necessary (Fig. 10–7). When map scale is changed this might also require (1) line coordinate thinning, which reduces the number of coordinates defining a given line; (2) line dropping when the boundary between two adjacent polygons is deleted; or (3) edge matching when a large number of map sheets must be composited into one continuous map for analysis.

The analyst must be able to browse or roam within the data base, selectively windowing in on a region of interest. Then, within this window it should be possible

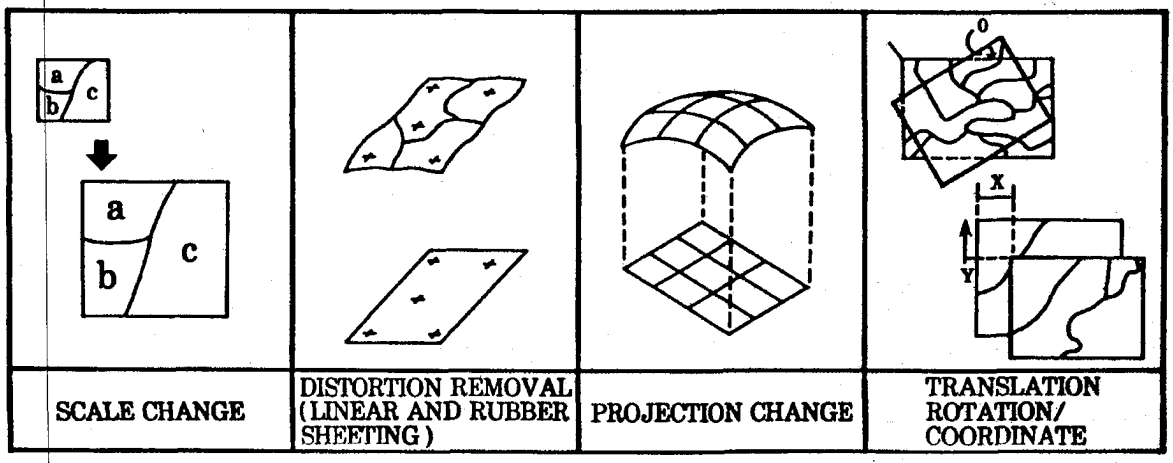



**Figure 10–7** Fundamental geometric manipulation of GIS data base files. (After Dangermond, 1983.)



OFEK06956

to define an even more specific subwindow within which more detailed questions will be asked. Ideally, this takes place in real-time while viewing the data on a high-resolution color CRT monitor.

Once the area of interest is identified, various types of analyses may be performed. Some of the most important include map overlay, map dissolve, polygon overlay for area calculation, measurement, and proximity searches. We will evaluate these based on *polygon* GIS logic and then look at them again using *grid* GIS logic.

*Polygon* overlay and dissolve techniques involve the compositing (integrating) or extracting (disintegration) of multiple maps (two or more) in order to create a new data set (Fig. 10–8). Normally, the new data set is created containing new polygons formed from the intersection of the boundaries of the two or more sets of separate polygon layers. In addition to creating new polygons based on the overlay of the multiple layers, these polygons are assigned multiple attributes (i.e., the attributes that were attributed to each separate overlay before the composition occurred). The mathematical overlay of these maps is performed for the purpose of area measurement as well as multiple attribute (multiple parameter) modeling exercises. This may involve the weighting of various parameters and their subclassification in order to create "model interpretations"; an example of such an interpretation is the generation of land suitability/capability assessments.

Map dissolve is the inverse of polygon overlay (Fig. 10–8). It extracts from a multiple attribute polygon file single attributes, both by attribute description and locational definition. The coordinates of the lines dropped can be deleted from the data base if desired.

A second type of polygon overlay is performed when the areas for a given data layer (e.g., land use) need to be calculated and summarized within a second layer of polygons such as census tracts (Fig. 10–8). The resulting output is the summary of statistics (i.e., land-use areas by census tracts).

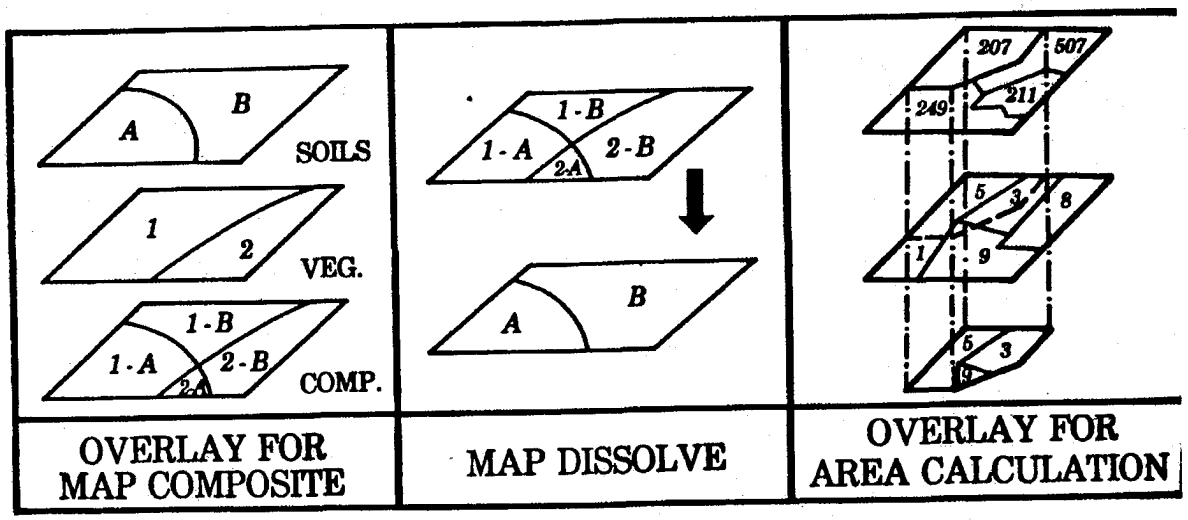



Figure 10–8 Data manipulation using polygon overlay and dissolve techniques. (After Dangermond, 1983.)



OFEK06957

The four most common types of measurement tasks involve points, line, polygons, and volumes (Fig. 10–9). The two typical measurement activities associated with points are enumeration of total number of points and also enumeration of total number of points falling within polygons. The latter technique involves the use of a "point-in-polygon" routine which counts the number of the various types of points falling within selected polygons (e.g., number of wells within a water district). The two basic forms of line measurement are point to point and measurement along a curvilinear line. The two basic types of area measurement are the area of a polygon and the perimeter of a polygon. The fourth category of measurement involves volumetric measurement, which is performed either through a cross-section technique or through overlays of multiple surfaces (i.e., before grading, after grading, and difference computation.) The latter procedure requires that the data be in a grid format.

*Grid cell analysis* analytical activities are similar to the types of map analysis that are performed using a polygon-based GIS, but they have more generalized spatial resolution. However, it should be pointed out that the grid cell technique for map manipulation is typically much more efficient both in data storage and in the operation of analytic tasks (Dangermond, 1983).

Grid cell overlay involves the development of Boolean logic map overlays in order to create composite maps. Typically, this involves the weighting of selected classifications for multiple map layers, resulting in a numeric surface of composite data, expressing the user's interpretation (Fig. 10–10). The example in Fig. 10–11 demonstrates how weighted soils, slope, and access data can be used to compute a suitability index map.

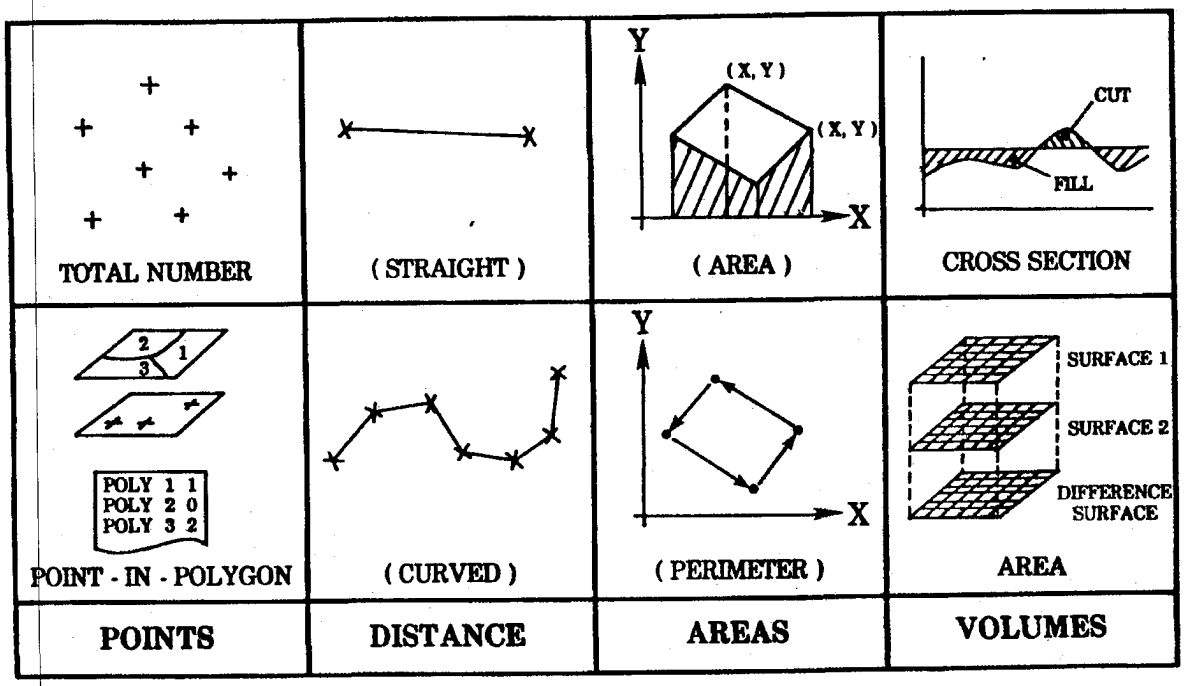



Figure 10–9 Measurement of points, distance, area, and volume in a GIS. (After Dangermond, 1983.)



OFEK06958

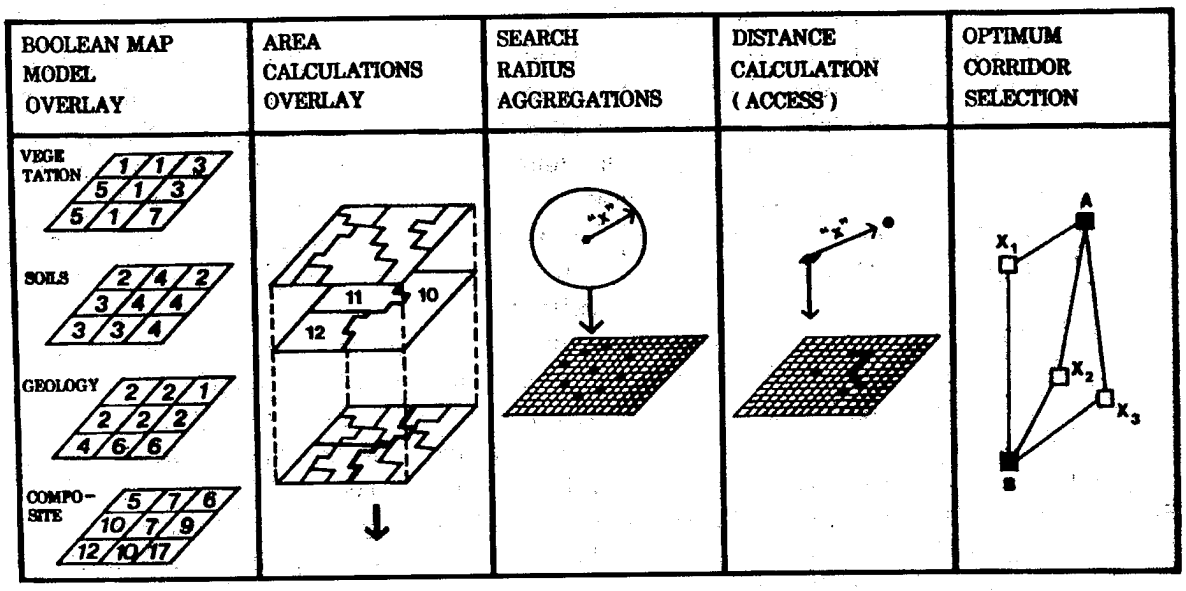



Figure 10–10 Grid-based GIS analytical techniques. (After Dangermond, 1983.)

Grid cell point, distance, and area calculation are similar to polygon overlay for calculations except that they are based on a gridded data. Grid cell search radius aggregation is used to count the number of events, activities, or related geographic phenomena that fall within a specified distance from each grid cell. The information is stored in a new grid array and represents a frequency/accessibility surface. The computation is usually based on a moving circle with a user-specified search radius moving across the grid on a cell-by-cell basis.

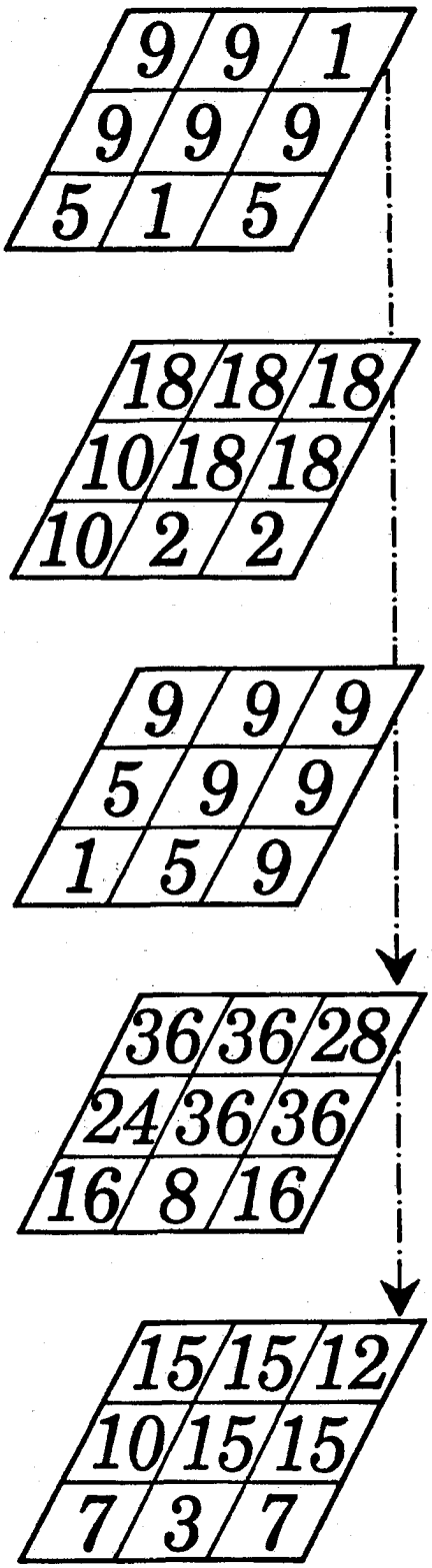
Grid cell distance calculation involves the computation applied to the nearest event or geographic characteristic, which is defined by the user. A typical application is calculation of the distance from each cell to a stream network, road network, and so on. The output from the analysis is the creation of an accessibility file (Fig. 10–10).

Using a grid-based GIS it is also possible to compute an optimal corridor (Fig. 10–10). This involves the computation of a least cost between points. Specifically, a friction surface is computed away from the origin and also away from the destination. These friction surfaces are numerical summaries of cost based on an original input grid of costs computed using the grid weighted overlay technique described above. This original map (actually surface of grid data) expresses for each cell the cost to move across it. These costs can be economic costs, environmental costs, social costs, and so on, and are summarized based on multiple parameter overlay. After computing the two friction maps (origin and destination), they are overlaid and the resulting surface is an optimum route selection with gradient computations of increased costs away from the optimum.

Another type of *grid*-based GIS operation that is very useful is matrix analysis. This function allows the analyst to select two files that are to be combined by assigning



OFEK06959




| SOILS | WEIGHTING IMPORTANCE |
|---|---|
| 9 GOOD | x1 |
| 5 FAIR | x1 |
| 1 POOR | x1 |

| SLOPES | WEIGHTING IMPORTANCE |
|---|---|
| 9 GOOD | x2 |
| 5 FAIR | x2 |
| 1 POOR | x2 |

| ACCESS | WEIGHTING IMPORTANCE |
|---|---|
| 9 GOOD | x1 |
| 5 FAIR | x1 |
| 1 POOR | x1 |

INTERMEDIATE VALUES CALCULATED

NORMALIZED TO 0 – 15 ;

HIGHEST VALUES ARE BEST

**Figure 10–11** Logic of grid cell overlay using soils and vegetation files stored in a grid-based GIS.





OFEK06960

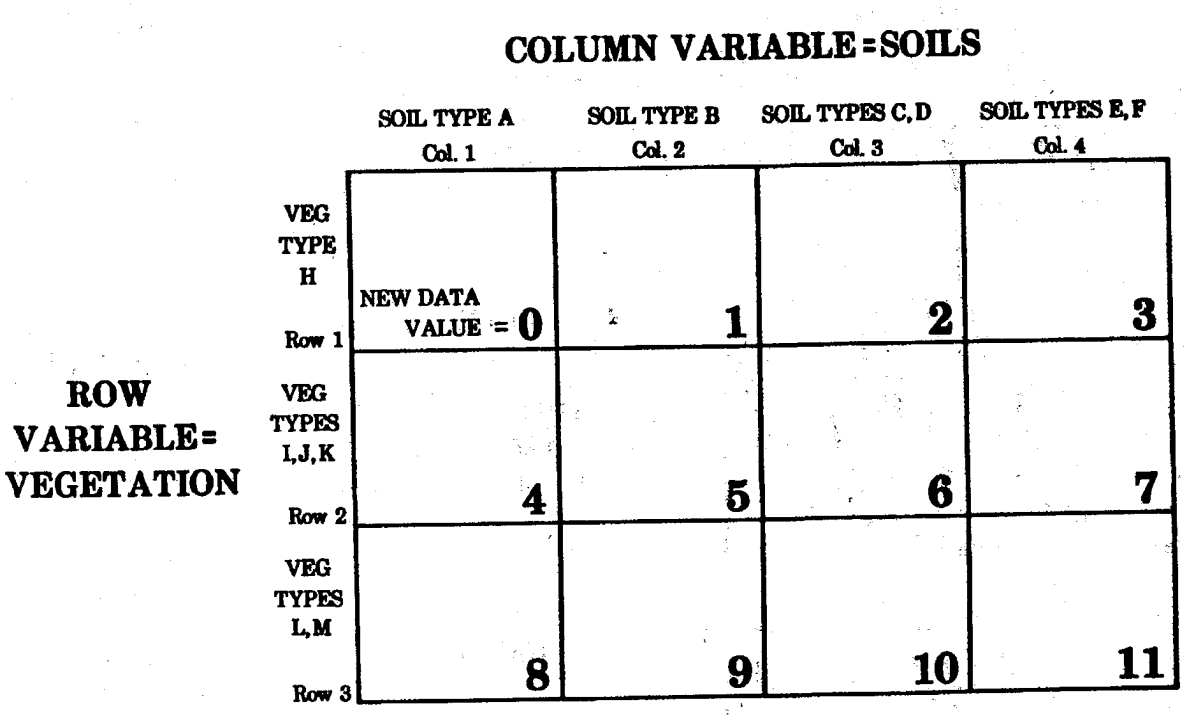



**Figure 10–12** Example of grid cell overlay using two attribute files to identify the most suitable area for development.

one variable to the columns of a matrix and the other variable to the rows of a matrix. The logic is shown digrammatically in Fig. 10–12. Note that the data subclasses can be assigned selectively or combined within each row or column. The numbers shown in the corners of the matrix elements are the resultant data values in the output file. For example, data value 9 represents all grid cells that consist of both soil type B and either vegetation type L or M. The user defines how many rows and columns will be used to create the matrix.

Many types of data are best manipulated in a grid format, especially those which are continuous over the landscape, such as elevation, slope, temperature, population density, and so on. For this reason most of the GIS work performed on such surfaces require that this type of data be in a grid format. Examples of applying this information include the computation of the viewshed for a proposed subdivision on a hillside, the determination of sun intensity and duration if large buildings are to be built, and the modeling of surface relief, slope, and aspect relative to other socioeconomic and environmental parameters.

## *DATA OUTPUT*

The output from GIS operations may be a hard-copy display of the spatial distribution of important thematic information on a line plotter, an ephemeral display of the same data on a CRT screen, or a listing of statistics. Since the data were entered



OFEK06961

using known coordinate locations associated with a rigorous base map and projection, the output map files may be scaled to desired dimensions.

## CONCLUSION

Some remote sensing specialists cannot understand why many GIS practictioners do not incorporate more remote sensing information into their GIS data base. Goodchild (1978) pointed out that those who continually work with polygon format GIS feel that the remote sensor data are poor in positional and categorical accuracy. Furthermore, a common oversight of those attempting to promote the incorporation of remote sensing data into GIS is to assume that the flow of data should be unidirectional (i.e., from the remote sensing system to the GIS). They should realize that the reverse flow, from the GIS to the remote sensing system, is desirable but used only infrequently. Why should the "backward" flow of data be so important? One example is that remote sensing analysts can use these "ancillary data" to improve their classification accuracy. For example, forest cover mapping from Landsat was improved by incorporating topographic and other ancillary information (Strahler et al., 1978; Maw and Brass, 1981).

At present the interface between geographic information systems and remote sensing systems is functional but weak (Hutchinson, 1982; Junkin, 1982; Jensen, 1984). Each side suffers from a lack of critical support of a type that could be provided by the other. The GIS has a continuing need for timely, accurate updates of the various spatial data entities, whereas the remote sensing systems could benefit from access to highly accurate ancillary information to extract more useful information from the imagery (Calkins and Marble, 1980; Marble et al., 1983). Generally, if the data in the GIS are improved, everyone can address more important questions. Indications suggest that the relationship is improving.

## REFERENCES

AVERY, T. E., and G. L. BERLIN. 1985. "Geographic Information Systems and Land-Use Land-Cover Mapping," Chapter 8 in *Interpretation of Aerial Photographs*. Minneapolis, Minn.: Burgess Publishing Company, pp. 235–250.

BALL, D. F., G. L. RADFORD, and W. M. WILLIAMS. 1983. *A Land Characteristic Data Bank for Great Britain*, Bangor Occasional Paper 13, Institute of Terrestrial Ecology, Bangor.

BERRY, J. K. 1981. "Spatial Information Systems: 'Instant' Maps for Analyzing Natural Resources Data," *Special Libraries*, Vol. 72, pp. 261–269.

BERRY, J. K., and C. DANA TOMLIN. 1981. "Fundamental Procedures of Geographic Information Analysis," presented at *In-Place Resource Inventories: Principles and Practices, A National Workshop*, University of Maine, Orono, Maine.

CALKINS, H. W., and D. F. MARBLE. 1980. *Computer Software for Spatial Data Handling*. Ottawa, Ontario: International Geographical Union.



OEEK06962

CALKINS, H. W., and R. F. TOMLINSON. 1977. *Geographic Information Systems: Methods and Equipment for Land Use Planning*. Washington, D.C.: The International Geographical Union, Commission on Geographical Data Sensing and Processing, and the U.S. Geological Survey, 2 vols.

CURRAN, P. J. 1985. "Image Processing," Chapter 6 in *Principles of Remote Sensing*. New York: Longman, Inc., pp. 223–226.

DANGERMOND, J. 1983. "A Classification of Software Components Commonly Used in Geographic Information Systems," in *Design and Implementation of Computer-Based Geographic Information Systems*, D. Peuquet, and J. O'Callaghan, Eds. Amherst, N.Y.: International Geographical Union Commission on Geographic Data Sensing and Processing.

DEUKER, K. J. 1979. "Land Resource Information Systems: A Review of Fifteen Years Experience," *Geo-Processing*, Vol. 1, pp. 105–128.

ESL. 1982. *IDIMS Functional Guide*. Sunnyvale, Calif.: Electromagnetic Systems Laboratory.

GARRISON, W. L., R. ALEXANDER, W. BAILEY, M. F. DACEY, and D. F. MARBLE. 1965. "Data Systems Requirements for Geographic Research," in *Scientific Experiments for Manned Orbital Flight*, Proceedings of the American Astronautical Society's Third Goddard Memorial Symposium.

GOODCHILD, M. F. 1978. "Statistical Aspects of the Polygon Overlay Problem," *Proceedings*, First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems, Vol. 6: Spatial Algorithms.

HANSEN, D. B. 1985. "GENERIC GIS: An Evaluation of FWS and BLM Geographic Data Processing Capabilities," *Proceedings*, Annual Meeting of the American Society of Photogrammetry. Falls Church, Va.: The American Society of Photogrammetry, pp. 712–727.

HUTCHINSON, C. F. 1982. "Techniques for Combining Landsat and Ancillary Data for Digital Classification Improvement," *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 123–130.

JENSEN, J. R. 1984. "Recent Developments in the Use of Remote Sensing for Earth Resource Mapping," Section 9 in U.S. National Report to the ICA, *The American Cartographer*, pp. 89–100.

JUNKIN, B. G. 1982. "Development of Three-Dimensional Spatial Displays Using A Geographical Based Information System," *Photogrammetric Engineering and Remote Sensing*, Vol. 48, pp. 577–586.

KNAPP, E. M., and D. RIDER. 1979. Automated Geographic Information Systems and Landsat Data: A Survey," *Computer Mapping in Natural Resources and Environment*, Cambridge, Mass.: Harvard Library of Computer Graphics, pp. 57–68.

LEE, J. R. 1983. *MOSS User's Manual*. Fort Collins, Colo.: Western Energy and Land Use Team.

MARBLE, D. F., D. J. PEUQUET, A. R. BOYLE, N. BRYANT, H. W. CALKINS, T. JOHNSON, and A. ZOBRIST. 1983. "Geographic Information Systems and Remote Sensing," Chapter 22 in *The Manual of Remote Sensing*, R. N. Colwell, Ed. Falls Church, Va.: American Society of Photogrammetry, Vol. 1, pp. 923–958.

MARBLE, D. F., H. S. CALKINS, and D. J. PEUQUET. 1984. "Technical Description of the DIME System," in *Basic Readings in Geographic Information Systems*. New York: SPAD Systems Ltd., pp. 2–57 to pp. 2–78.

MAW, K. D., and J. A. BRASS. 1981. "Forest Management Applications of Landsat Data in a Geographic Information System," *Proceedings*, Pecora VII Symposium, Sioux Falls, S.D., pp. 330–340.



OEEK06963

SHELTON, R. C., and J. E. ESTES. 1981. "Remote Sensing and Geographical Information Systems: An Unrealized Potential," *Geo-Processing*, Vol. 1, pp. 395–420.

SHORT, N. M. 1982. "Geographic Information Systems," Activity 7 in *The Landsat Tutorial Workbook—Basics of Satellite Remote Sensing*. Washington, D.C.: NASA Scientific and Technical Information Branch, pp. 277–307.

SIMONETT, D. S., T. R. SMITH, W. TOBLER, D. G. MARKS, J. E. FREW, and J. C. DOZIER (Eds.). 1977. *Geobase Information System Impacts on Space Image Formats*. Santa Barbara, Calif.: Remote Sensing Unit Technical Report 3.

STRAHLER, A. H., T. L. LOGAN, and N. A. BRYANT. 1978. "Improving Forest Cover Classification Accuracy from Landsat by Incorporating Topographic Information," *Proceedings*, Twelfth International Symposium on Remote Sensing of the Environment. Ann Arbor, Mich.: Environmental Research Institute of Michigan, pp. 927–942.

TOMLINSON, R. F. 1984. "Geographic Information Systems—A New Frontier," *Proceedings*, International Symposium on Spatial Data Handling, Zurich, Switzerland, pp. 1–14.

WHITLEY, S. L., R. W. PEARSON, B. R. SEYFARTH, and M. H. GRAHAM. 1981. "ELAS: A Geobased Information System That Is Transferable to Several Computers," *Proceedings*, Fifteenth International Symposium on Remote Sensing of Environment. Ann Arbor, Mich.: Environmental Research Institute of Michigan, pp. 1091–1099.

ZOBRIST, A. L., N. A. BRYANT, S. Z. FRIEDMAN, and G. L. ANGELICI. 1978. *Image Based Information System (IBIS) System Guide*. Publication 900–909. Pasadena, Calif.: Jet Propulsion Laboratory. (Available through COSMIC at the University of Georgia in Athens, Ga.)



OEEK06964

# APPENDIX A

# The SENSOR Digital Image Processing Programs*

## THE SENSOR PROGRAMS

The SENSOR programs listed in this appendix may be used to perform several of the digital image processing operations described in the text. They were conceived and developed by John R. Jensen and Michael E. Hodgson of the Department of Geography of the University of South Carolina. They may be freely copied and used for educational and research purposes. When the use of these programs results in publishable research, please cite the following reference:

> Jensen, J. R., and M. E. Hodgson. 1986. "Appendix A: The SENSOR Digital Image Processing Programs," in *Introductory Digital Image Processing: A Remote Sensing Perspective*. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

The programs were written in ANSI FORTRAN 66 on an IBM mainframe computer system. Programs written using this compiler are very transportable. They will also compile on an ANSI FORTRAN 77 compiler with minor modification. All the programs use direct-access files that are defined by a single statement (DEFINE FILE). To run the programs on other computer systems using a FORTRAN compiler, simply change the pertinent DEFINE FILE statements and make modifications in the READ or WRITE input/output statements.

These programs were written to operate in batch mode primarily because most students have access to a mainframe computer with a version of ANSI FORTRAN

* This appendix was written by John R. Jensen and Michael E. Hodgson. The authors assume no liability for the accuracy of these programs, although great care was taken in their preparation. Please contact the authors if you have suggestions regarding the SENSOR digital image processing software.





OFFK06965

that will run in batch mode. However, the programs can easily be made to operate in an interactive environment. All that is required is that the programs be modified so that they accept input from the terminal rather than from unit 5, route output to the terminal instead of unit 6, and use the appropriate compiler file definition syntax.

### Program Limitations

The programs were written to use very small amounts of computer random access memory (RAM). The maximum amount of image data held in RAM at one time is one row of data $i$, for each of the $k$, possible bands. The default maximum number of columns in a row of image data is 2100 pixels. There is no limit to the number of rows. The default maximum number of bands in a remote sensor dataset is 6. Therefore, the analyst can process quite a significant matrix of remote sensor data. When performing supervised training and classification the maximum number of classes is 25. Any of these default maximum values may be increased or decreased by simply changing the DATA and DIMENSION statements at the beginning of the programs. It is important to remember, however, that all the programs must have the same limits. This means that if a change is made in the maximum number of bands (e.g., from 6 to 7), this same change should be made in all the programs.

### The Format of a SENSOR Image File

The raw SENSOR image file (and the output file for a classification map) are stored in direct-access files in a band sequential (BSQ) format as discussed in Chapter 2. However, prior to each bank of band sequential data, there exist two header records. The first header record contains the information on the number of rows, number of columns, and so on, of the data set. The second header record contains a title for the image file. These two header records are then followed by the image data. The image data are found in record 3 and onward. Here the complete matrix of band 1 data are present followed by band 2 data and so on. The input/output routines that access the image files use a record length and block size of 100 bytes. Internal program documentation in the first few programs describes how to change the blocksize if desired.

Three input/output subroutines (RWHEAD, RWLINE, RWBLK) perform all the reading and writing to the image data. While the image data are stored as byte data (to minimize storage), the input/output routines decompress the byte data to two-byte INTEGER values before use in the other routines of each program.

### The STATISTICS File

Prior to performing supervised training and classification it is necessary to initialize a direct-access file by running the CREATE STATS program. This file will contain one header record plus three records of statistics data for each of the 25 maximum



OEEK06966

possible classes. The variance, covariance, the means, and standard deviation vectors are all stored. Also, the number of samples, number of training sites, and coordinates of the training sites for each class are stored. Two input/output routines (RWSTHE and RWSTAT) are used to read to or write from the STATISTICS file. Blocksize for the statistics file should be $\geq 90 + (16 * \text{number of bands})$.

## AN OVERVIEW OF THE SENSOR SYSTEM PROGRAMS

The SENSOR software will perform the image processing operations shown in the accompanying flowchart (Fig. A–1). It is instructive to review briefly the major functions of these programs.

The LANDREAD and TMREAD programs can be used to extract a user-specified rectangular submatrix from either a Landsat multispectral scanner (MSS) data set or a Landsat thematic mapper (TM) data set. The LANDREAD program assumes that the data have had standard geometric correction applied at the EROS Data Center and are written in band sequential (BSQ) format on two 1600-bpi tapes. The TMREAD program assumes that the P-level data are written in band sequential format (BSQ) and reside on either a 1600- or 6250-bpi tape drive. The multiple-band submatrix of data extracted from the tapes are then placed in a SENSOR direct-access file in the BSQ format described previously.

The STATISTICS program computes the mean, standard deviation, minimum value, maximum value, covariance matrix, and correlation matrix of the user-specified bands of data for a subimage found in a SENSOR data set. These data are then printed out in the format described in Chapter 4.

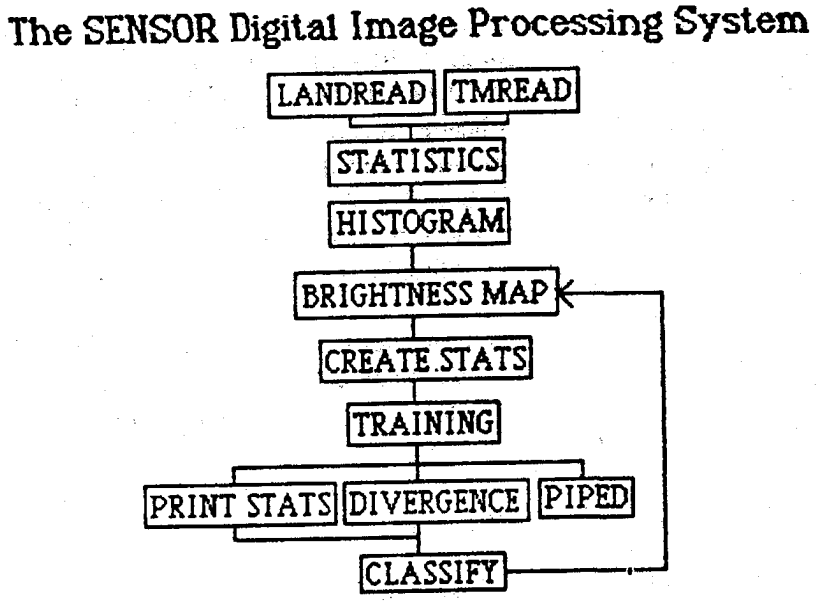



Figure A–1 SENSOR Digital Image Processing System programs and the typical sequence of program operation.



OFEK06967

The HISTOGRAM program produces histograms of the individual bands of remotely sensed data stored in the SENSOR data set. The user specifies the beginning and ending rows and columns of the subimage to be histogrammed, the band number, and the type of histogram to be displayed. It is possible to display raw histograms of the data or histograms that reflect the distribution of the data after a minimum-maximum or percentage linear contrast stretch has been applied. The analyst specifies the length of the axes and annotation of the histograms. Output is to a pen plotter or electrostatic plotter. The listing provides JCL for output to a Gould 100-dot-per-inch electrostatic plotter using CALCOMP pen plotter emulation routines.

The BRIGHTNESS MAP program will display a pseudo-continuous-tone image of a single band of imagery stored in a SENSOR file. The analyst identifies the row and column dimensions of a submatrix within the data set to be displayed and the band number. Raw brightness values may be mapped or the data may be contrast stretched using a minimum-maximum or percentage linear contrast stretch algorithm. In addition, some compensation for the nonlinear response of the human eye to stimuli can be made. Output is to an electrostatic plotter using CALCOMP pen plotter emulation routines. The program is also used to display the output from a matrix produced in the supervised classification map process.

The CREATE STATS program initializes (places "zeros") in files that are used in the collection of training statistics and in the supervised classification of the remotely sensed data. This program must be run prior to running the TRAINING program.

The TRAINING program allows the analyst to identify the row and column coordinates of specific rectangular training sites located within a SENSOR file. The user determines the number of classes to be trained on, the number of training sites to be identified, their coordinates, and the bands to be evaluated. The training statistics, including the minimum, maximum, mean, variance, covariance, and correlation matrices of the various training classes, are then written to the printer and to the statistics direct-access file for subsequent use in other SENSOR programs. Program PRINTSTATS can be used to print the contents of the statistics file at a later date without having to rerun the TRAINING program.

The PIPED program displays two- or pseudo-three-dimensional parallelepipeds of the training statistics for classes of interest that were collected in the TRAINING program. This allows the analyst visually to see if there is substantial overlap among the training class statistics while evaluating up to three bands at one time. The user specifies the classes to be plotted, the number of standard deviations to be used to create the parallelepipeds, the bands to be evaluated, and the azimuth and rotation angles. Output is to an electrostatic plotter using CALCOMP pen plotter emulation routines.

The DIVERGENCE program allows the analyst to statistically evaluate the training class statistics obtained in the TRAINING program. The program will determine the "best" $q$ band subset out of $k$ bands for classifying $m$ classes. For example, we might want to identify the best three-band combination out of six bands for classifying five classes. The optimum subset of bands (called features) may then be used in the supervised classification process.



OEEK06968

The CLASSIFY program allows the analyst to perform a minimum distance to means classification based on the statistics collected in the TRAINING program and stored in the statistics file. The user identifies the bands and classes to be used in the analysis. A minimum-distance to means classification is then performed and routed to a file for eventual display on the electrostatic plotter using program BRIGHTNESS MAP. If desired, the program will produce a contingency table showing errors of omission and commission for the pixels located within the class training sites.



OFEK06969

# LANDREAD

```
//BOOKLN JOB (D6600217,9),'VF-LANREAD',TIME=(,59),MSGLEVEL=(1,1),
// MSGCLASS=A,REGION=128K,USER=SBS01,PASSWORD=??????
/*ROUTE PRINT MVS.R8
// EXEC DSUTIL
//SYSIN DD *
 DELETE DSN=SBS.MEH1B.LAMS4C.R149C177
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*********************************************************************
C  LANREAD PROGRAM
C  THIS PROGRAM READS LANDSAT DIGITAL TAPES(MSS ONLY) WHICH HAVE BEEN
C  GEOMETRICALLY CORRECTED AND ARE WRITTEN IN THE BAND SEQUENTIAL
C  FORMAT(BSQ) ONTO TWO 1600 B.P.I. TAPES.
C
C INPUT PARAMETER LINES:
C  COLUMNS       DESCRIPTION
C
C ......LINE 1
C  1- 5      PUT A 'FIRST' IF THIS IS THE FIRST BAND TO BE PUT IN
C            THE FILE.  ANYTHING ELSE FOR THE 2ND OR SECEEDING BANDS.
C
C  6-10      NUMBER OF ROWS TO BE STORED IN THE SENSOR IMAGE FILE
C 11-15      NUMBER OF COLUMNS TO BE STORED IN THE SENSOR IMAGE FILE
C 16-20      TOTAL NUMBER OF BANDS TO BE PLACED IN THE IMAGE FILE
C
C ......LINE 2
C  1- 5      BEGINNING ROW OF SUBMATRIX ON TM TAPE
C  6-10      ENDING ROW OF SUBMATRIX
C 11-15      BEGINNING COLUMN OF SUBMATRIX
C 16-20      ENDING COLUMN OF SUBMATRIX
C 21-25      BAND # WHICH THIS DATA WILL BE STORED AS
C 26-30      SAMPLING INTERVAL ( 1 MEANS TAKE EVERY PIXEL)
C
C ......LINE 3
C  1-80      ALPHANUMERIC TITLE FOR SENSOR IMAGE FILE
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NCMAX - MAXIMUM # OF COLUMNS OR PIXELS IN A LINE
C          THIS VALUE MUST NOT BE LARGER THAN THE 'LINE.' ARRAY SIZES.
C  NBLK  - # OF PIXELS PER LOGICAL RECORD IN IMAGE FILE.
C           (MUST BE CONSISTENT WITH OTHER PROGRAMS AND IMPORTANTLY
C            WITH THE ARRAY 'IBYTE' SIZE IN 'RWLINE' SUBROUTINE.
C
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS IN THE IMAGE FILE
C MUST BE SET IN THE 'DEFINE FILE 9(# STATMENT.
C
C    THE PROGRAM BYPASSES THE DIRECTORY AND HEADER FILES ON EACH TAPE
C  AND GOES DIRECTLY TO THE BRIGHTNESS VALUES IN THE IMAGE
C  FILES. THIS IS ACCOMPLISHED THROUGH THE APPROPRIATE INPUT FILE
C  INFORMATION(E.G. OF IBM JCL - LABEL=(3,BLP,,IN) DIRECTS THE PROGRAM
C  TO BEGIN READING BAND 4 IMAGE FILE DATA IF TAPE 1 IS LOADED).
C    THE REQUESTED DATA MATRIX IS PLACED IN THE SENSOR IMAGE FILE
C  FORMAT.
```





OFEK06970

```
C
C
C THE INPUT UNIT #(LANDSAT TAPE) IS UNIT 8.
C   THE MSS BANDS ON TWO 1600 B.P.I. TAPES ARE IDENTIFIED
C   AS FOLLOWS:
C   BAND 4    -   TAPE 1   FILE 3
C   BAND 5    -   TAPE 1   FILE 6
C   BAND 6    -   TAPE 2   FILE 3
C   BAND 7    -   TAPE 2   FILE 6
C
C LOGICAL UNITS USED:
C  5 - USER INPUT FROM LINES
C  6 - LINE PRINTER OR TERMINAL
C  8 - TAPE FILE
C  9 - IMAGE FILE
C
C*********************************************************************
      DIMENSION FILTIT(20)
      INTEGER*2 LINE1(2100)
      DATA NBLK/100/, NCMAX/2100/, RFIRST/'FIRS'/, LINE1/2100*0/
      DATA IERR/0/
C  ....MAXIMUM # OF COLS AND ROWS IN MSS BAND-SEQUENTIAL IMAGE
      DATA NCOLMS/3548/, NROWMS/2983/
C
C-----> 1ST ARGUMENT SHOULD BE SET THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> ((INTEGER(NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(1195,100,L,IREC)
C
C.......CHECK TO SEE IF THE USER IS SETTING UP THE IMAGE FILE FOR
C    ....THE FIRST BAND.
      CALL READIN(IBR,IER,IBC,IEC,IBAND,NROWS,NCOLS,NBAND,FILTIT,
     &            ISAMP,RBAND,NCOLMS,NROWMS,IERR)
      IF(IERR.EQ.1) GOTO 9999
C
C  ....COMPUTE # OF RECORDS PER ROW
      NACC=(NCOLS/NBLK) + 1
      IREM=MOD(NCOLS,NBLK)
      IF(IREM.EQ.0) NACC=NACC-1
C
C .....WRITE OUT HEADER IF FIRST TIME
      IF(RBAND.EQ.RFIRST) CALL SETUP(NROWS,NCOLS,NBAND,NACC,NBLK,
     &                               FILTIT,LINE1,NCMAX,IERR)
      IF(IERR.EQ.1) GOTO 9999
C
      IF(RBAND.NE.RFIRST) CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
C
C......PRINT BANNER
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C
      CALL WRMAT(NROWS,NACC,NBLK,IBR,IER,IBC,IEC,IBAND,LINE1,
     &           ISAMP,NCMAX,IERR)
C
C
```



OFEK06971

```
 9999 STOP
      END
C*******************************************************************
      SUBROUTINE READIN(IBR,IER,IBC,IEC,IBAND,NROWS,NCOLS,NBAND,FILTIT,
     &                  ISAMP,RBAND,NCOLMS,NROWMS,IERR)
C*******************************************************************
C  READS THE PARAMETERS SPECIFIED BY THE USER AND ECHOS THEM OUT.
C  IF A CONFLICT IN THE IMAGE FILE AND THE SUBMATRIX SPECIFIED
C  IS IDENTIFIED, THE PROGRAM TERMINATES BY SETTING IERR=1.
C*******************************************************************
      DIMENSION FILTIT(20)
      DATA RFIRST/'FIRS'/
C
C .....INPUT USER PARAMETERS
      READ(5,11,END=9400) RBAND,NROWS,NCOLS,NBAND
   11 FORMAT(A4,1X,3I5)
      READ(5,12,END=9400) IBR,IER,IBC,IEC,IBAND,ISAMP
   12 FORMAT(6I5)
      READ(5,13,END=9400) FILTIT
   13 FORMAT(20A4)
C
      NROWS2=(IER-IBR+1)/ISAMP
      NCOLS2=(IEC-IBC+1)/ISAMP
C
C
C  ....ECHO OUT PARAMETERS READ IN
      WRITE(6,31) NROWS,NCOLS,NBAND,ISAMP,IBR,IER,IBC,IEC,IBAND,RFIRST
   31 FORMAT(1X//1X,'--> PARAMETERS FOR SENSOR IMAGE FILE:'/
     &           1X,'  # ROWS= ',I5/
     &           1X,'  # COLS= ',I5/
     &           1X,'  # BANDS=',I5/
     &           1X,'  SAMPLING INTERVAL=',I5/
     &           1X,'    SUBMATRIX SPECIFIED'/
     &           1X,'  - ROWS -    - COLUMNS -        '/
     &           1X,' BEGIN   END  BEGIN  END     BAND'/
     &           1X,4I6,5X,I2,//
     &           1X,' FIRST ? =',A4//)
C
C  .....CHECK FOR PROGRAM AND IMAGE LIMITATIONS
      IF(IBAND.LT.1 .OR. IBAND.GT.4 .OR. IBAND.GT.NBAND) GOTO 9000
      IF(IBR.LT.1 .OR. IBC.LT.1 .OR. IBR.GT.NROWMS .OR. IBC.GT.NCOLMS)
     &    GOTO 9100
      IF(IER.LT.1 .OR. IEC.LT.1 .OR. IER.GT.NROWMS .OR. IEC.GT.NCOLMS)
     &    GOTO 9100
      IF(NROWS2.NE.NROWS .OR. NCOLS2.NE.NCOLS) GOTO 9200
      IF(ISAMP.LT.1) GOTO 9300
      RETURN
C
C .........ERROR MESSAGES.................
 9000 WRITE(6,9001) IBAND,NBAND
 9001 FORMAT(1X,//1X,'*** YOU SPECIFIED AN ILLEGAL BAND # OF ',I2,';'/
     &           1X,'    THERE ARE',I2,' BANDS IN SENSOR IMAGE FILE.')
      IERR=1
      RETURN
 9100 WRITE(6,9101)
```



OFEK06972

```
 9101 FORMAT(1X,//1X,'*** SUBMATRIX REQUESTED OUTSIDE OF IMAGE '/
     &           1X,'    FILE LIMITS')
      IERR=1
      RETURN
 9200 WRITE(6,9201)
 9201 FORMAT(1X,//1X,'*** TOTAL NUMBER OF ROWS AND COLUMNS IN IMAGE'/
     &           1X,'    FILE DO NOT MATCH SUBIMAGE DIMENSIONS.')
      IERR=1
      RETURN
 9300 WRITE(6,9301) ISAMP
 9301 FORMAT(1X,//1X,'*** YOU SPECIFIED AN ILLEGAL SAMPLING INTERVAL'/
     &           1X,'    OF ',I3)
      IERR=1
      RETURN
 9400 WRITE(6,9301)
 9401 FORMAT(1X,//1X,'*** PREMATURELY RAN OUT OF PARAMETER CARDS')
      IERR=1
      RETURN
      END
C*******************************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*******************************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*******************************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          LANDREAD PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=    ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*******************************************************************************
      SUBROUTINE SETUP(NROWS,NCOLS,NBAND,NACC,NBLK,FILTIT,LINE1,NCMAX,
     &                 IERR)
C*******************************************************************************
C WRITES THE HEADER AND DUMMY DATA FOR THE FILE.
C*******************************************************************************
      DIMENSION FILTIT(20)
      INTEGER*2 LINE1(2100)
C
C  ....WRITE HEADER RECORD
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,1)
C
C  ....WRITE 0'S FOR ALL IMAGE DATA LINES
      DO 200 IBAND=1,NBAND
        DO 100 IROW=1,NROWS
          CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,IROW,IBAND,NBLK,1,
     &                IERR)
          IF(IERR.EQ.1) RETURN
```



OFEK06973

```
  100      CONTINUE
  200    CONTINUE
C
      WRITE(6,901)
  901 FORMAT(1X///1X,'---> SENSOR IMAGE FILE INITIALIZED')
      RETURN
      END
C*********************************************************************
      SUBROUTINE WRMAT(NROWS,NACC,NBLK,IBR,IER,IBC,IEC,IBAND,LINE1,
     &                 ISAMP,NCMAX,IERR)
C*********************************************************************
C READS THE TAPE AND CREATES THE SENSOR IMAGE FILE.
C*********************************************************************
      INTEGER*2 LINE1(NCMAX)
      LOGICAL*1 REC(3596)
C
C......READ AND WRITE BRIGHTNESS VALUES IN SUBMATRIX
C ......READ TO FIRST OF DATA MATRIX
        IF(IBR.EQ.1) GOTO 50
        NSKIP=IBR-1
        DO 20 I=1,NSKIP
          READ(8,11) REC
   11     FORMAT(100(100A1),100(100A1),100(100A1))
   20     CONTINUE
C
C  ......READ AND WRITE REQUESTED SUBMATRIX
   50   IROW=NSKIP
        KROW=0
        DO 100 I=IBR,IER
          READ(8,11) REC
C
          IROW=IROW+1
          IR= IROW-1
          IREM= MOD(IR,ISAMP)
          IF(IREM.NE.0) GOTO 100
          CALL MOVBYT(LINE1,NCMAX,IBC,IEC,ISAMP,REC)
          KROW=KROW+1
          CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,KROW,IBAND,NBLK,1,
     &                IERR)
          IF(IERR.EQ.1) RETURN
   80     CONTINUE
C
  100   CONTINUE
C
 9999 RETURN
      END
C*********************************************************************
      SUBROUTINE MOVBYT(LINE1,NCMAX,IBC,IEC,ISAMP,REC)
C*********************************************************************
C  MOVES A GROUP OF BYTES INTO THE LINE1 ARRAY.
C  COLUMNS IBC TO IEC ARE MOVED TO THE 1 THROUGHT NTH BYTE IN LINE1.
C
C  MAJOR VARIABLES:
C  REC   - 4 LINES OF A DATA FROM A TM TAPE
C  LINE1 - 2 BYTE ARRAY HOLDING 1 LINE OF IMAGE DATA
```



OFEK06974

```
C  K     - SPECIFIES 1 - 4TH LINE OF IMAGE DATA TO MOVE
C*************************************************************************
      INTEGER*2 I2BYTE(2), LINE1(NCMAX)
      LOGICAL*1 I1BYTE(2), REC(3596)
      EQUIVALENCE (I1BYTE(1),I2BYTE(1))
C
      IB=1
C.....OFFSET FOR LOCATION OF PIXELS IN RECORD
      IBEGP= IBC + 12
      IENDP= IEC + 12
C
C .....DECOMPRESS A LINE OF DATA
      DO 100 IP=IBEGP,IENDP,ISAMP
        I1BYTE(2)=REC(IP)
        LINE1(IB)=I2BYTE(1)
        IB=IB+1
  100   CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE RWHEAD(IUNIT,NROWS,NCOLS,NBAND,FILTIT,IRW)
C*************************************************************************
C  READS OR WRITES A HEADER RECORD FOR THE DIRECT ACCESS FILE
C  IUNIT - SPECIFIES THE FILE TO ACCESS (9-IMAGE FILE, 11-CLASSIFIED
C          FILE)
C  NROWS - # OF ROWS IN IMAGE FILE
C  NCOLS - # OF COLUMNS IN IMAGE FILE
C  NBAND - # OF BANDS IN THE IMAGE FILE
C  IRW   - SPECIFIES WHETHER TO READ(0) OR WRITE(1) A LINE
C*************************************************************************
      DIMENSION FILTIT(20)
      IREC=1
      IF(IRW.EQ.1) GOTO 10
      READ(IUNIT'IREC) FILTIT
      IREC=2
      READ(IUNIT'IREC) NROWS,NCOLS,NBAND
      RETURN
   10 WRITE(IUNIT'IREC) FILTIT
      IREC=2
      WRITE(IUNIT'IREC) NROWS,NCOLS,NBAND
      RETURN
      END
C*************************************************************************
      SUBROUTINE RWLINE(IUNIT,LINE,NCMAX,NROWS,NACC,IROW,IBAND,NBLK,
     &                  IRW,IERR)
C*************************************************************************
C  READS OR WRITES A ROW OF DATA OF A SPECIFIED CHANNEL FROM
C  A DIRECT ACCESS FILE.
C
C  MAJOR VARIABLES:
C   IUNIT - SPECIFIES THE FILE TO ACCESS (9-IMAGE FILE, 11-CLASSIFIED
C           FILE)
C   LINE  - ARRAY OF DN VALUES FOR THE ROW
C           (NOTE: THIS IS PRESENTLY SET FOR 1 BYTE PER PIXEL)
C   NCMAX - MAXIMUM # OF COLUMNS OR PIXELS POSSIBLE IN A LINE
```



OFEK06975

```
C   IBYTE - A RECORD OF DATA DIMENSIONED TO 'NBLK'
C   NROWS - TOTAL NUMBER OF ROWS FOR A CHANNEL
C   NACC  - THE NUMBER OF BLOCKS PER ROW OF DATA
C   IROW  - THE REQUESTED ROW
C   IBAND - THE CHANNEL NUMBER
C   NBLK  - NUMBER OF PIXELS PER BLOCK
C   IRW   - SPECIFIES WHETHER TO READ(0) OR WRITE(1) A LINE
C   IERR  - RETURNS A 1 IF THERE IS A CONFLICT IN THE # BYTES PER
C           RECORD
C
C NOTE: THIS IO ROUTINE USES A RECORD LENGTH OF 100 BYTES.
C       IF A DIFFERENT RECORD LENGTH IS DESIRABLE ONE MUST REDIMENSION
C       THE ARRAY 'IBYTE' AND RESET THE DATA STATEMENT FOR NB.
C*************************************************************************
      INTEGER*2 LINE(NCMAX)
      LOGICAL*1 IBYTE(100)
      DATA NB/100/
C
      IF(NBLK.NE.NB) GOTO 9000

      IREC2=((IBAND-1)*NROWS*NACC) + ((IROW-1)*NACC) + 2
      IBA=1
      DO 50 I=1,NACC
        IREC=IREC2+I
        CALL READLI(IUNIT,NBLK,IREC,IRW,IBYTE,LINE(IBA))
        IBA=IBA+NBLK
   50   CONTINUE
      RETURN
 9000 WRITE(6,9001) NBLK,NB
 9001 FORMAT(1X,'*** SORRY, CANNOT USE THE IMAGE FILE AS YOU ARE'/
     &       1X,'    SUGGESTING A BLOCK SIZE OF ',I5,' BUT THE '/
     &       1X,'    RWLINE SUBROUTINE IS PRESET TO ',I5,' BYTES')
      IERR=1
      RETURN
      END
C*************************************************************************
      SUBROUTINE READLI(IUNIT,NBLK,IREC,IRW,IBYTE,IBLK)
C*************************************************************************
C  COMPANION ROUTINE OF 'RWLINE' FOR READING AND WRITING
C  A RECORD OF DATA
C*************************************************************************
      INTEGER*2 I2BYTE(1),IBLK(NBLK)
      LOGICAL*1 I1BYTE(2),IBYTE(NBLK)
      EQUIVALENCE (I1BYTE(1),I2BYTE(1))
C
      IF(IRW.EQ.1) GOTO 100
C
C  ....GET A BLOCK AND DECOMPRESS
      READ(IUNIT'IREC) IBYTE
      DO 20 IP=1,NBLK
        I1BYTE(2)=IBYTE(IP)
   20   IBLK(IP)=I2BYTE(1)
      RETURN
C
C  ....COMPRESS A BLOCK AND WRITE OUT
```



OFEK06976

```
  100 DO 50 IP=1,NBLK
        I2BYTE(1)=IBLK(IP)
   50   IBYTE(IP)=I1BYTE(2)
       WRITE(IUNIT'IREC) IBYTE
       RETURN
       END
//GO.FT05F001 DD *
FIRST  149  177    4
    1 2983    1 3548    1   20
LOS ANGELES (SAMPLED EVERY 20)
//GO.FT08F001 DD UNIT=TAPED,DISP=OLD,DCB=(RECFM=F,LRECL=3596,
//     BLKSIZE=3596,DEN=3),VOL=SER=LSAT01,LABEL=(3,BLP,,IN)
//GO.FT09F001 DD DISP=(NEW,DELETE,UNCATLG),UNIT=3330V,
//     DSN=SBS.MEH1B.LAMS4C.R149C177,SPACE=(100,(1195),RLSE),
//     VOL=SER=SBS102,DCB=(RECFM=F,LRECL=100,BLKSIZE=100)
```



OFEK06977

# TMREAD

```
//BOOKTM JOB (D6600217,9),'VF-BOOKTM',TIME=(8,59),MSGLEVEL=(1,1),
// MSGCLASS=A,PASSWORD=??????,USER=SBS01
/*ROUTE PRINT MVS.R8
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
C*************************************************************************
C  TMREAD PROGRAM
C  REFORMATS A SEGMENT OF A RAW THEMATIC MAPPER TAPE (P LEVEL) INTO
C  A 'SENSOR' IMAGE FILE.
C
C INPUT PARAMETER LINES:
C  COLUMNS       DESCRIPTION
C
C ......LINE 1
C  1- 5      PUT A 'FIRST' IF THIS IS THE FIRST BAND TO BE PUT IN
C            THE FILE.  ANYTHING ELSE FOR THE 2ND OR SECEEDING BANDS.
C
C  6-10      NUMBER OF ROWS TO BE STORED IN THE SENSOR IMAGE FILE
C 11-15      NUMBER OF COLUMNS TO BE STORED IN THE SENSOR IMAGE FILE
C 16-20      TOTAL NUMBER OF BANDS TO BE PLACED IN THE IMAGE FILE
C
C ......LINE 2
C  1- 5      BEGINNING ROW OF SUBMATRIX OF TM TAPE
C  6-10      ENDING ROW OF SUBMATRIX
C 11-15      BEGINNING COLUMN OF SUBMATRIX
C 16-20      ENDING COLUMN OF SUBMATRIX
C 21-25      BAND # WHICH THIS DATA WILL BE STORED AS
C 26-30      SAMPLING INTERVAL (1 MEANS TAKE EVERY PIXEL)
C
C ......LINE 3
C  1-80      ALPHANUMERIC TITLE FOR IMAGE FILE
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NCMAX - MAXIMUM # OF COLUMNS OR PIXELS IN A LINE
C          THIS VALUE MUST NOT BE LARGER THAN THE 'LINE.' ARRAY SIZES.
C  NBLK  - # OF PIXELS PER LOGICAL RECORD IN IMAGE FILE.
C           (MUST BE CONSISTENT WITH OTHER PROGRAMS AND IMPORTANTLY
C            WITH THE ARRAY 'IBYTE' SIZE IN 'RWLINE' SUBROUTINE.
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS IN THE IMAGE FILE
C MUST BE SET IN THE 'DEFINE FILE 9(# STATMENT.
C ALSO, THE USER MUST SET THE LABEL PARAMETER ON THE TAPE.
C
C LOGICAL UNITS USED:
C  5 - USER INPUT FROM LINES
C  6 - LINE PRINTER OR TERMINAL
C  8 - TAPE FILE
C  9 - IMAGE FILE
C
C*************************************************************************
      DIMENSION FILTIT(20)
      INTEGER*2 LINE1(2100)
      DATA NBLK/100/, NCMAX/2100/, RFIRST/'FIRS'/, LINE1/2100*0/
      DATA IERR/0/
C  ....MAXIMUM # OF COLS AND ROWS IN TM P-LEVEL IMAGE
```





OFEK06978

```
      DATA NCOLIM/6967/, NROWIM/5965/
C
C-----> 1ST ARGUMENT SHOULD BE SET THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> ((INTEGER(NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4161,100,L,IREC)
C
C.......CHECK TO SEE IF THE USER IS SETTING UP THE IMAGE FILE FOR
C    ....THE FIRST BAND.
      CALL READIN(IBR,IER,IBC,IEC,IBAND,NROWS,NCOLS,NBAND,FILTIT,
     &            ISAMP,RBAND,NCOLIM,NROWIM,IERR)
      IF(IERR.EQ.1) GOTO 9999
C
C  ....COMPUTE # OF RECORDS PER ROW
      NACC=(NCOLS/NBLK) + 1
      IREM=MOD(NCOLS,NBLK)
      IF(IREM.EQ.0) NACC=NACC-1
C
C .....WRITE OUT HEADER IF FIRST TIME
      IF(RBAND.EQ.RFIRST) CALL SETUP(NROWS,NCOLS,NBAND,NACC,NBLK,
     &                               FILTIT,LINE1,NCMAX,IERR)
      IF(IERR.EQ.1) GOTO 9999
C
      IF(RBAND.NE.RFIRST) CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
C
C......PRINT BANNER
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C
      CALL WRMAT(NROWS,NACC,NBLK,IBR,IER,IBC,IEC,IBAND,ISAMP,LINE1,
     &           NCMAX,IERR)
C
C
 9999 STOP
      END
C*********************************************************************
      SUBROUTINE READIN(IBR,IER,IBC,IEC,IBAND,NROWS,NCOLS,NBAND,FILTIT,
     &                  ISAMP,RBAND,NCOLIM,NROWIM,IERR)
C*********************************************************************
C  READS THE PARAMETERS SPECIFIED BY THE USER AND ECHOS THEM OUT.
C  IF A CONFLICT IN THE IMAGE FILE AND THE SUBMATRIX SPECIFIED
C  IS IDENTIFIED, THE PROGRAM TERMINATES BY SETTING IERR=1.
C*********************************************************************
      DIMENSION FILTIT(20)
      DATA RFIRST/'FIRS'/
C
C .....INPUT USER PARAMETERS
      READ(5,11,END=9400) RBAND,NROWS,NCOLS,NBAND
   11 FORMAT(A4,1X,3I5)
      READ(5,12,END=9400) IBR,IER,IBC,IEC,IBAND,ISAMP
   12 FORMAT(6I5)
      READ(5,13,END=9400) FILTIT
   13 FORMAT(20A4)
C
      NROWS2=(IER-IBR+1)/ISAMP
```



OFEK06979

```
      NCOLS2=(IEC-IBC+1)/ISAMP
C
C
C  ....ECHO OUT PARAMETERS READ IN
      WRITE(6,31) NROWS,NCOLS,NBAND,ISAMP,IBR,IER,IBC,IEC,IBAND,RFIRST
   31 FORMAT(1X//1X,'--> PARAMETERS FOR SENSOR IMAGE FILE:'/
     &           1X,'  # ROWS= ',I5/
     &           1X,'  # COLS=',I5/
     &           1X,'  # BANDS=',I5/
     &           1X,'  SAMPLING INTERVAL=',I5///
     &           1X,'    SUBMATRIX SPECIFIED'/
     &           1X,'  - ROWS -    - COLUMNS -       '/
     &           1X,' BEGIN   END  BEGIN  END     BAND'/
     &           1X,4I6,5X,I2,//
     &           1X,' FIRST ? =',A4//)
C
C  .....CHECK FOR PROGRAM AND IMAGE LIMITATIONS
       IF(IBAND.LT.1 .OR. IBAND.GT.7 .OR. IBAND.GT.NBAND) GOTO 9000
       IF(IBR.LT.1 .OR. IBC.LT.1 .OR. IBR.GT.NROWIM .OR. IBC.GT.NCOLIM)
     &     GOTO 9100
       IF(IER.LT.1 .OR. IEC.LT.1 .OR. IER.GT.NROWIM .OR. IEC.GT.NCOLIM)
     &     GOTO 9100
       IF(NROWS2.NE.NROWS .OR. NCOLS2.NE.NCOLS) GOTO 9200
       IF(ISAMP.LT.1) GOTO 9300
       RETURN
C
C ..........ERROR MESSAGES..................
 9000  WRITE(6,9001) IBAND,NBAND
 9001  FORMAT(1X,//1X,'*** YOU SPECIFIED AN ILLEGAL BAND # OF ',I2,';'/
     &           1X,'    THERE ARE',I2,' BANDS IN SENSOR IMAGE FILE.')
       IERR=1
       RETURN
 9100  WRITE(6,9101)
 9101  FORMAT(1X,//1X,'*** SUBMATRIX REQUESTED OUTSIDE OF IMAGE '/
     &           1X,'    FILE LIMITS')
       IERR=1
       RETURN
 9200  WRITE(6,9201)
 9201  FORMAT(1X,//1X,'*** TOTAL NUMBER OF ROWS AND COLUMNS IN IMAGE'/
     &           1X,'    FILE DO NOT MATCH SUBIMAGE DIMENSIONS.')
       IERR=1
       RETURN
 9300  WRITE(6,9301) ISAMP
 9301  FORMAT(1X,//1X,'*** YOU SPECIFIED AN ILLEGAL SAMPLING INTERVAL'/
     &           1X,'    OF ',I3)
       IERR=1
       RETURN
 9400  WRITE(6,9401)
 9401  FORMAT(1X,//1X,'*** PREMATURELY RAN OUT OF PARAMETER CARDS')
       IERR=1
       RETURN
       END
C*********************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*********************************************************************
```



OFEK06980

```
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*************************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '       TM TAPE TO SENSOR IMAGE FILE PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=   ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*************************************************************************
      SUBROUTINE SETUP(NROWS,NCOLS,NBAND,NACC,NBLK,FILTIT,LINE1,NCMAX,
     &                 IERR)
C*************************************************************************
C WRITES THE HEADER AND DUMMY DATA FOR THE FILE.
C*************************************************************************
      DIMENSION FILTIT(20)
      INTEGER*2 LINE1(2100)
C
C  ....WRITE HEADER RECORD
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,1)
C
C  ....WRITE 0'S FOR ALL IMAGE DATA LINES
      DO 200 IBAND=1,NBAND
        DO 100 IROW=1,NROWS
          CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,IROW,IBAND,NBLK,1,
     &                IERR)
          IF(IERR.EQ.1) RETURN
  100     CONTINUE
  200   CONTINUE
C
      WRITE(6,901)
  901 FORMAT(1X///1X,'---> SENSOR IMAGE FILE INITIALIZED')
      RETURN
      END
C*************************************************************************
      SUBROUTINE WRMAT(NROWS,NACC,NBLK,IBR,IER,IBC,IEC,IBAND,ISAMP,
     &                 LINE1,NCMAX,IERR)
C*************************************************************************
C READS THE TAPE AND CREATES THE SENSOR IMAGE FILE.
C*************************************************************************
      INTEGER*2 LINE1(NCMAX)
      LOGICAL*1 REC(28672)
C
C
C......READ AND WRITE BRIGHTNESS VALUES IN SUBMATRIX
C ......READ TO FIRST OF DATA MATRIX
        NSKIP=0
        IF(IBR.LE.4) GOTO 50
        NSKIP=((IBR-1)/4)
```



OFEK06981

```
      DO 20 I=1,NSKIP
        READ(8,11) REC
 11     FORMAT(100(100A1),100(100A1),100(100A1))
 20     CONTINUE
C
C  .....READ AND WRITE REQUESTED SUBMATRIX
 50   IROW=NSKIP*4
      KROW=0
      DO 100 I=1,1492
        READ(8,11) REC
        IF(IROW.GT.IER) RETURN
C
        DO 80 K=1,4
          IROW=IROW+1
          IF(IROW.LT.IBR .OR. IROW.GT.IER) GOTO 80
          IR=IROW-1
          IREM=MOD(IR,ISAMP)
          IF(IREM.NE.0) GOTO 80
          CALL MOVBYT(LINE1,NCMAX,K,IBC,IEC,ISAMP,REC)
          KROW=KROW+1
          CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,KROW,IBAND,NBLK,1,
     &                IERR)
          IF(IERR.EQ.1) RETURN
 80       CONTINUE
C
 100    CONTINUE
C
9999  RETURN
      END
C*********************************************************************
      SUBROUTINE MOVBYT(LINE1,NCMAX,K,IBC,IEC,ISAMP,REC)
C*********************************************************************
C  MOVES A GROUP OF BYTES INTO THE LINE1 ARRAY.
C  COLUMNS IBC TO IEC OF REC ARE MOVED TO THE 1 THROUGHT NTH
C  LOCATION IN LINE1.
C
C  MAJOR VARIABLES:
C  REC   - 4 LINES OF DATA FROM A TM TAPE (BYTE VALUES)
C  LINE1 - 2 BYTE ARRAY HOLDING 1 LINE OF IMAGE DATA (2-BYTE VALUES)
C  K     - SPECIFIES 1 - 4TH LINE OF IMAGE DATA TO MOVE
C*********************************************************************
      INTEGER*2 I2BYTE(2), LINE1(NCMAX)
      LOGICAL*1 I1BYTE(2), REC(28672)
      EQUIVALENCE (I1BYTE(1),I2BYTE(1))
C
      IB=1
      IBEGP= ((K-1) * 7168) + IBC
      IENDP= IBEGP + IEC-1
C
C .....DECOMPRESS A LINE OF DATA
      DO 100 IP=IBEGP,IENDP,ISAMP
        I1BYTE(2)=REC(IP)
        LINE1(IB)=I2BYTE(1)
        IB=IB+1
 100    CONTINUE
```



OFEK06982

```
      RETURN
      END
C*************************************************************************
      SUBROUTINE RWHEAD(IUNIT,NROWS,NCOLS,NBAND,FILTIT,IRW)
C*************************************************************************
C  READS OR WRITES A HEADER RECORD FOR THE DIRECT ACCESS FILE
C  IUNIT - SPECIFIES THE FILE TO ACCESS (9-IMAGE FILE, 11-CLASSIFIED
C          FILE)
C  NROWS - # OF ROWS IN IMAGE FILE
C  NCOLS - # OF COLUMNS IN IMAGE FILE
C  NBAND - # OF BANDS IN THE IMAGE FILE
C  IRW   - SPECIFIES WHETHER TO READ(0) OR WRITE(1) A LINE
C*************************************************************************
      DIMENSION FILTIT(20)
      IREC=1
      IF(IRW.EQ.1) GOTO 10
      READ(IUNIT'IREC) FILTIT
      IREC=2
      READ(IUNIT'IREC) NROWS,NCOLS,NBAND
      RETURN
   10 WRITE(IUNIT'IREC) FILTIT
      IREC=2
      WRITE(IUNIT'IREC) NROWS,NCOLS,NBAND
      RETURN
      END
C*************************************************************************
      SUBROUTINE RWLINE(IUNIT,LINE,NCMAX,NROWS,NACC,IROW,IBAND,NBLK,
     &                  IRW,IERR)
C*************************************************************************
C  READS OR WRITES A ROW OF DATA OF A SPECIFIED CHANNEL FROM
C  A DIRECT ACCESS FILE.
C
C  MAJOR VARIABLES:
C   IUNIT - SPECIFIES THE FILE TO ACCESS (9-IMAGE FILE, 11-CLASSIFIED
C           FILE)
C   LINE  - ARRAY OF DN VALUES FOR THE ROW
C           (NOTE: THIS IS PRESENTLY SET FOR 1 BYTE PER PIXEL)
C   NCMAX - MAXIMUM # OF COLUMNS OR PIXELS POSSIBLE IN A LINE
C   IBYTE - A RECORD OF DATA DIMENSIONED TO 'NBLK'
C   NROWS - TOTAL NUMBER OF ROWS FOR A CHANNEL
C   NACC  - THE NUMBER OF BLOCKS PER ROW OF DATA
C   IROW  - THE REQUESTED ROW
C   IBAND - THE CHANNEL NUMBER
C   NBLK  - NUMBER OF PIXELS PER BLOCK
C   IRW   - SPECIFIES WHETHER TO READ(0) OR WRITE(1) A LINE
C   IERR  - RETURNS A 1 IF THERE IS A CONFLICT IN THE # BYTES PER
C           RECORD
C
C NOTE: THIS IO ROUTINE USES A RECORD LENGTH OF 100 BYTES.
C       IF A DIFFERENT RECORD LENGTH IS DESIRABLE ONE MUST REDIMENSION
C       THE ARRAY 'IBYTE' AND RESET THE DATA STATEMENT FOR NB.
C*************************************************************************
      INTEGER*2 LINE(NCMAX)
      LOGICAL*1 IBYTE(100)
      DATA NB/100/
```



OFEK06983

```
C
      IF(NBLK.NE.NB) GOTO 9000

      IREC2=((IBAND-1)*NROWS*NACC) + ((IROW-1)*NACC) + 2
      IBA=1
      DO 50 I=1,NACC
        IREC=IREC2+I
        CALL READLI(IUNIT,NBLK,IREC,IRW,IBYTE,LINE(IBA))
        IBA=IBA+NBLK
   50   CONTINUE
      RETURN
 9000 WRITE(6,9001) NBLK,NB
 9001 FORMAT(1X,'*** SORRY, CANNOT USE THE IMAGE FILE AS YOU ARE'/
     &       1X,'    SUGGESTING A BLOCK SIZE OF ',I5,' BUT THE '/
     &       1X,'    RWLINE SUBROUTINE IS PRESET TO ',I5,' BYTES')
      IERR=1
      RETURN
      END
C*******************************************************************************
      SUBROUTINE READLI(IUNIT,NBLK,IREC,IRW,IBYTE,IBLK)
C*******************************************************************************
C  COMPANION ROUTINE OF 'RWLINE' FOR READING AND WRITING
C  A RECORD OF DATA
C*******************************************************************************
      INTEGER*2 I2BYTE(1),IBLK(NBLK)
      LOGICAL*1 I1BYTE(2),IBYTE(NBLK)
      EQUIVALENCE (I1BYTE(1),I2BYTE(1))
C
      IF(IRW.EQ.1) GOTO 100
C
C  ....GET A BLOCK AND DECOMPRESS
      READ(IUNIT'IREC) IBYTE
      DO 20 IP=1,NBLK
        I1BYTE(2)=IBYTE(IP)
   20   IBLK(IP)=I2BYTE(1)
      RETURN
C
C  ....COMPRESS A BLOCK AND WRITE OUT
  100 DO 50 IP=1,NBLK
        I2BYTE(1)=IBLK(IP)
   50   IBYTE(IP)=I1BYTE(2)
      WRITE(IUNIT'IREC) IBYTE
      RETURN
      END
//GO.FT05F001 DD *
OTHER  198  232    7
    1 5940    1 6960    2   30
ENTIRE TM SCENE (SAMPLED EVERY 30)
//GO.FT08F001 DD UNIT=T6250,VOL=SER=010501,DISP=OLD,LABEL=(1,BLP,,IN),
//   DCB=(RECFM=F,LRECL=28672,BLKSIZE=28672,DEN=4)
//GO.FT09F001 DD DSN=SBS.MEH1B.CHIM7B.R198C232,DISP=OLD
```



OFEK06984

# STATISTICS

```
//BOOKST2 JOB (D6600217),'VF-BOOKST2',TIME=(,59),MSGLEVEL=(1,0),
// USER=SBS01,PASSWORD=??????
/*ROUTE PRINT MVS.R8
// EXEC FORTGCLG,PARM.LKED='NOXREF,NOMAP'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C  IMAGE STATISTICS PROGRAM
C
C  COMPUTES THE SIMPLE STATISTICS AS WELL AS THE VARIANCE-COVARIANCE
C  AND CORRELATION MATRICES FOR ALL OF THE BANDS IN AN IMAGE FILE.
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NCMAX - MAXIMUM # OF COLUMNS OR PIXELS IN A LINE
C          THIS VALUE MUST NOT BE LARGER THAN THE 'LINE.' ARRAY SIZES.
C  NBMAX - THE MAXIMUM NUMBER OF BANDS OF DATA WHICH COULD BE USED.
C          (THIS NUMBER MAY BE LARGER THAN THE ACTUAL NUMBER OF BANDS
C          IN THE IMAGE FILE, HOWEVER, IT MUST NOT BE LARGER THAN THE
C          STATISTICS ARRAY SIZES IN MAIN.
C  NBLK  - # OF PIXELS PER LOGICAL RECORD IN IMAGE FILE.
C           (MUST BE CONSISTENT WITH OTHER PROGRAMS AND IMPORTANTLY
C            WITH THE ARRAY 'IBYTE' SIZE IN 'RWLINE' SUBROUTINE.
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS IN THE IMAGE
C FILE MUST BE SET IN THE 'DEFINE FILE 9(#' STATEMENT.
C
C-----> ESSENTIALLY THE ONLY PARAMETER TO RESET INSIDE THE SOURCE CODE
C       FOR RUNNING WITH DIFFERENT FILES ARE THE # OF RECORDS IN THE
C       THE DEFINE FILE STATMENT.
C
C LOGICAL UNITS USED:
C  6 - LINE PRINTER OR TERMINAL
C  9 - IMAGE FILE
C
C*************************************************************************
      DIMENSION MINC(6), MAXC(6), RMEANC(6), VARC(6), STDC(6),
     &          COVC(6,6), CORRC(6,6)
      DIMENSION FILTIT(20)
      INTEGER*2 LINE1(2100), LINE2(2100)
      DATA IERR/0/
      DATA NBLK/100/, NCMAX/2100/, NBMAX/6/
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4323,100,L,IREC)
C
C......READ HEADER RECORD
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
      IF(NBAND.GT.NBMAX) GOTO 9000
C
C  ....COMPUTE # OF RECORDS PER ROW
      NACC=(NCOLS/NBLK) + 1
      IREM=MOD(NCOLS,NBLK)
      IF(IREM.EQ.0) NACC=NACC-1
```





OFEK06985

```
C......PRINT CREDITS AND HEADER INFORMATION
C
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C......GENERATE STATISTICS
      CALL STAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NROWS,NCOLS,
     &          NCMAX,LINE1,LINE2,NBAND,NBLK,NACC,IERR)
      IF(IERR.EQ.1) GOTO 9999
C.......PRINT OUT STATISTICS
      CALL PUSTAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NBAND)
      GOTO 9999
C
C.......ERROR MESSAGES ..............
 9000 WRITE(6,9001) NBANDS,NBMAX
 9001 FORMAT(1X,'*** SORRY, CANNOT COMPUTE STATISTICS BECAUSE '/
     &       1X,'    THERE ARE ',I2,' BANDS IN THE IMAGE FILE AND'/
     &       1X,'    YOU HAVE THE PROGRAM LIMITS AT ',I2,'.'/
     &       1X,'    INCREASE THE STATS ARRAYS AND NBMAX IN MAIN.')
 9999 STOP
      END
C*********************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*********************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*********************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          STATISTICS PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=   ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*********************************************************************
      SUBROUTINE STAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NROWS,
     &                NCOLS,NCMAX,LINE1,LINE2,NBAND,NBLK,NACC,IERR)
C*********************************************************************
C  GENERATES THE STATISTICS FOR ALL BANDS FOR ENTIRE
C  THE IMAGE.
C
C MAJOR ARRAYS:
C  LINE1  - TEMPORARY STORAGE FOR A LINE OF A BAND OF DATA
C  LINE2  - TEMPORARY STORAGE FOR A LINE OF A BAND OF DATA
C  RMEANC - VECTOR OF MEAN STATS
C  MINC   - VECTOR OF MINIMUM BRIGHTNESS VALUES
C  MAXC   - VECTOR OF MAXIMUM BRIGHTNESS VALUES
C  VARC   - VECTOR OF VARIANCES
C  STDC   - VECTOR OF STANDARD DEVIATIONS
C  COVC   - MATRIX OF COVARIANCE VALUES
C  CORRC  - MATRIX OF CORRELATION VALUES
C*********************************************************************
```



OFEK06986

```
      INTEGER*2 LINE1(NCMAX), LINE2(NCMAX)
C.....NOTE: COVC AND CORRC ARRAYS ARE STORED AS COVC(ROW,COLUMN)
      DIMENSION MINC(NBAND), MAXC(NBAND), RMEANC(NBAND),
     &          VARC(NBAND),STDC(NBAND),COVC(NBAND,NBAND),
     &          CORRC(NBAND,NBAND)
      DOUBLE PRECISION SUMX, SS, SP
C
C......COMPUTE SIMPLE STATISTICS FOR EACH BAND
C    ...SET UP DEGREES OF FREEDOM; USE N DEGREES OF FREEDOM FOR MEAN,
C    ...USE N-1 DEGREES OF FREEDOM FOR ALL OTHER STATS.
C
      TOTPIX=FLOAT(NROWS*NCOLS)
      TOTPI2=TOTPIX-1.
C
C    ...FOR EACH BAND, CALCULATE MEAN
C
      DO 100 I=1,NBAND
        MINC(I)=99999
        MAXC(I)=-99999
        SUMX=0.
C
        DO 50 J=1,NROWS
          CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,J,I,NBLK,0,IERR)
          IF(IERR.EQ.1) RETURN
          DO 20 IPIX=1,NCOLS
            SUMX=SUMX + LINE1(IPIX)
            IF(LINE1(IPIX).LT.MINC(I)) MINC(I)=LINE1(IPIX)
            IF(LINE1(IPIX).GT.MAXC(I)) MAXC(I)=LINE1(IPIX)
   20       CONTINUE
   50     CONTINUE
C
        RMEANC(I)=SUMX/TOTPIX
  100   CONTINUE
C
C......CALCULATE VARIANCE-COVARIANCE AND MATRIX
C
      DO 300 I=1,NBAND
        SS=0.
        DO 200 J=1,I
          SP=0.
C
          DO 150 K=1,NROWS
          CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,K,I,NBLK,0,IERR)
          CALL RWLINE(9,LINE2,NCMAX,NROWS,NACC,K,J,NBLK,0,IERR)
            DO 130 IP=1,NCOLS
              SP= SP + (LINE1(IP)-RMEANC(I)) * (LINE2(IP)-RMEANC(J))
  130         CONTINUE
            IF(I.NE.J) GOTO 150
            DO 140 IP=1,NCOLS
              SS= SS + (LINE1(IP)-RMEANC(I))**2
  140         CONTINUE
  150       CONTINUE
C
          IF(I.EQ.J) VARC(I)= SS / TOTPI2
          IF(I.EQ.J) STDC(I)= SQRT(VARC(I))
```



OFEK06987

```
            COVC(I,J)= SP / TOTPI2
  200       CONTINUE
C
C       ....FILL UPPER RIGHT OF MATRIX
            IF(I.EQ.1) GOTO 300
            IE=I-1
            DO 250 JC=1,IE
  250         COVC(JC,I)=COVC(I,JC)
  300     CONTINUE
C
C......CALCULATE CORRELATION STATISTICS
        DO 400 I=1,NBAND
          DO 350 J=1,I
            CORRC(I,J)= COVC(I,J) / (STDC(I)*STDC(J))
  350       CONTINUE
  400     CONTINUE
C
        RETURN
        END
C*************************************************************************
        SUBROUTINE RWHEAD(IUNIT,NROWS,NCOLS,NBAND,FILTIT,IRW)
C*************************************************************************
C  READS OR WRITES A HEADER RECORD FOR THE DIRECT ACCESS FILE
C  IUNIT - SPECIFIES THE FILE TO ACCESS (9-IMAGE FILE, 11-CLASSIFIED
C          FILE)
C  NROWS - # OF ROWS IN IMAGE FILE
C  NCOLS - # OF COLUMNS IN IMAGE FILE
C  NBAND - # OF BANDS IN THE IMAGE FILE
C  IRW   - SPECIFIES WHETHER TO READ(0) OR WRITE(1) A LINE
C*************************************************************************
        DIMENSION FILTIT(20)
        IREC=1
        IF(IRW.EQ.1) GOTO 10
        READ(IUNIT'IREC) FILTIT
        IREC=2
        READ(IUNIT'IREC) NROWS,NCOLS,NBAND
        RETURN
     10 WRITE(IUNIT'IREC) FILTIT
        IREC=2
        WRITE(IUNIT'IREC) NROWS,NCOLS,NBAND
        RETURN
        END
C*************************************************************************
        SUBROUTINE RWLINE(IUNIT,LINE,NCMAX,NROWS,NACC,IROW,IBAND,NBLK,
       &                  IRW,IERR)
C*************************************************************************
C  READS OR WRITES A ROW OF DATA OF A SPECIFIED CHANNEL FROM
C  A DIRECT ACCESS FILE.
C
C  MAJOR VARIABLES:
C   IUNIT - SPECIFIES THE FILE TO ACCESS (9-IMAGE FILE, 11-CLASSIFIED
C           FILE)
C   LINE  - ARRAY OF DN VALUES FOR THE ROW
C           (NOTE: THIS IS PRESENTLY SET FOR 1 BYTE PER PIXEL)
C   NCMAX - MAXIMUM # OF COLUMNS OR PIXELS POSSIBLE IN A LINE
```



OEEK06988

```
C   IBYTE - A RECORD OF DATA DIMENSIONED TO 'NBLK'
C   NROWS - TOTAL NUMBER OF ROWS FOR A CHANNEL
C   NACC  - THE NUMBER OF BLOCKS PER ROW OF DATA
C   IROW  - THE REQUESTED ROW
C   IBAND - THE CHANNEL NUMBER
C   NBLK  - NUMBER OF PIXELS PER BLOCK
C   IRW   - SPECIFIES WHETHER TO READ(0) OR WRITE(1) A LINE
C   IERR  - RETURNS A 1 IF THERE IS A CONFLICT IN THE # BYTES PER
C           RECORD
C
C NOTE: THIS IO ROUTINE USES A RECORD LENGTH OF 100 BYTES.
C       IF A DIFFERENT RECORD LENGTH IS DESIRABLE ONE MUST REDIMENSION
C       THE ARRAY 'IBYTE' AND RESET THE DATA STATEMENT FOR NB.
C*********************************************************************
      INTEGER*2 LINE(NCMAX)
      LOGICAL*1 IBYTE(100)
      DATA NB/100/
C
      IF(NBLK.NE.NB) GOTO 9000

      IREC2=((IBAND-1)*NROWS*NACC) + ((IROW-1)*NACC) + 2
      IBA=1
      DO 50 I=1,NACC
        IREC=IREC2+I
        CALL READLI(IUNIT,NBLK,IREC,IRW,IBYTE,LINE(IBA))
        IBA=IBA+NBLK
   50   CONTINUE
      RETURN
 9000 WRITE(6,9001) NBLK,NB
 9001 FORMAT(1X,'*** SORRY, CANNOT USE THE IMAGE FILE AS YOU ARE'/
     &       1X,'    SUGGESTING A BLOCK SIZE OF ',I5,' BUT THE '/
     &       1X,'    RWLINE SUBROUTINE IS PRESET TO ',I5,' BYTES')
      IERR=1
      RETURN
      END
C*********************************************************************
      SUBROUTINE READLI(IUNIT,NBLK,IREC,IRW,IBYTE,IBLK)
C*********************************************************************
C  COMPANION ROUTINE OF 'RWLINE' FOR READING AND WRITING
C  A RECORD OF DATA
C*********************************************************************
      INTEGER*2 I2BYTE(1),IBLK(NBLK)
      LOGICAL*1 I1BYTE(2),IBYTE(NBLK)
      EQUIVALENCE (I1BYTE(1),I2BYTE(1))
C
      IF(IRW.EQ.1) GOTO 100
C
C  ....GET A BLOCK AND DECOMPRESS
      READ(IUNIT'IREC) IBYTE
      DO 20 IP=1,NBLK
        I1BYTE(2)=IBYTE(IP)
   20   IBLK(IP)=I2BYTE(1)
      RETURN
C
C  ....COMPRESS A BLOCK AND WRITE OUT
```



OFEK06989

```
  100 DO 50 IP=1,NBLK
        I2BYTE(1)=IBLK(IP)
   50   IBYTE(IP)=I1BYTE(2)
       WRITE(IUNIT'IREC) IBYTE
       RETURN
       END
C*************************************************************************
      SUBROUTINE PUSTAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NBAND)
C*************************************************************************
C PRINTS THE SUMMARY STATISTICS FOR ALL BANDS OF THE IMAGE FILE.
C
C LIMITS: THE WRITE STATMENTS ARE SET UP FOR NO MORE THAN 8 BANDS OF
C         DATA PER IMAGE FILE. IF YOU WANT MORE, CHANGE ARRAY 'ICOUN',
C         ITS DATA STATEMENT, THE 'NOMAX' DATA STATEMENT, AND THE 8I'S
C         AND 8F'S IN THE OUTPUT STATEMENTS.
C*************************************************************************
       DIMENSION RMEANC(NBAND), STDC(NBAND), VARC(NBAND),
     &           MINC(NBAND), MAXC(NBAND), COVC(NBAND,NBAND),
     &           CORRC(NBAND,NBAND)
       DIMENSION ICOUN(8)
       DATA ICOUN/1,2,3,4,5,6,7,8/, NOMAX/8/
C
       IF(NBAND.GT.NOMAX) GOTO 9000
C
C.......PRINT OUT BANNER AND SIMPLE STATS
       WRITE(6,101) (ICOUN(J),J=1,NBAND)
  101  FORMAT(1X,'    UNIVARIATE STATISTICS'/
     &       1X,70('_')/
     &       1X,' BAND  ',8I10)
       WRITE(6,111)
  111  FORMAT(1X,70('_'))
C
       WRITE(6,102) (RMEANC(J),J=1,NBAND)
  102  FORMAT(1X,'MEAN      ',8F10.1)
       WRITE(6,103) (STDC(J),J=1,NBAND)
  103  FORMAT(1X,'STD. DEV.',1X,8F10.2)
       WRITE(6,104) (VARC(J),J=1,NBAND)
  104  FORMAT(1X,'VARIANCE ',8F10.1)
       WRITE(6,105) (MINC(J),J=1,NBAND)
  105  FORMAT(1X,'MINIMUM  ',8I10)
       WRITE(6,106) (MAXC(J),J=1,NBAND)
  106  FORMAT(1X,'MAXIMUM  ',8I10)
       WRITE(6,111)
C
C......PRINT OUT COVARIANCE MATRIX
       WRITE(6,121)
  121  FORMAT(1X,////'    VARIANCE-COVARIANCE MATRIX'/1X,70('_'))
       WRITE(6,122) (ICOUN(J),J=1,NBAND)
  122  FORMAT(1X,'BAND',8I10)
       WRITE(6,111)
       DO 150 I=1,NBAND
         WRITE(6,123) I,(COVC(I,J),J=1,I)
  123    FORMAT(1X,I2,4X,8F10.2)
  150    CONTINUE
       WRITE(6,111)
```



OFEK06990

```
C
C......PRINT OUT CORRELATION MATRIX
      WRITE(6,221)
  221 FORMAT(1X,////'     CORRELATION MATRIX'/1X,70('_'))
      WRITE(6,222) (ICOUN(J),J=1,NBAND)
  222 FORMAT(1X,'BAND',8I10)
      WRITE(6,111)
      DO 250 I=1,NBAND
        WRITE(6,223) I,(CORRC(I,J),J=1,I)
  223   FORMAT(1X,I2,4X,8F10.2)
  250   CONTINUE
      WRITE(6,111)
      RETURN
C
C......ERROR MESSAGES
 9000 WRITE(6,9001) NBAND,NOMAX
 9001 FORMAT(1X,'*** SORRY, YOU ARE TRANSGRESSING THE LIMITS OF THE'/
     &       1X,'    PUSTAT SUBROUTINE BY USING ',I2,' BANDS.  LIMITS'/
     &       1X,'    OF ',I2,' BANDS IN EFFECT.  MODIFY PUSTAT '/
     &       1X,'    SUBROUTINE.')
      RETURN
      END
/*
//GO.FT09F001 DD DSN=SBS.MEH1B.BOOK6C.R240C256,DISP=SHR
```



OFEK06991

# HISTOGRAM

```
//BOOKHI JOB (D6600217,99,99),'VF-BOOKHI',TIME=(0,9),MSGLEVEL=(1,0),
//  MSGCLASS=A,PASSWORD=??????,USER=SBS01
/*JOBPARM CARDS=9999
/*ROUTE PRINT MVS.R8
/*ROUTE PUNCH RMT8
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C HISTOGRAM PROGRAM
C
C PRODUCES A HISTOGRAM OF THE SUBMATRIX SPECIFIED ON A VECTOR PLOTTER.
C
C
C  THE PARAMETER LINES TO INPUT ARE INDICATED BELOW:
C
C COLUMNS  VARIABLE              DESCRIPTION
C..... LINE # 1
C  1- 5     IBR      - BEGINNING ROW OF SUBIMAGE TO BE HISTOGRAMMED.
C  6-10     IER      - ENDING     "   "     "      "   "       "
C 11-15     IBC      - BEGINNING COLUMN    "      "   "       "
C 16-20     IEC      - ENDING     "        "      "   "       "
C 21-25     IBAND    - BAND # TO HISTOGRAM
C 26-30     ITYPE    - THE TYPE OF HISTOGRAM TO BE PLOTTED
C                      (1=NORMAL HISTOGRAM OF DATA, 2=MIN-MAX STRETCH
C                       OF DATA, 3=PERCENTAGE STRETCH OF DATA)
C 31-35     PERCEN   - PERCENTAGE STRETCH ON BOTH SIDES
C 36-40     IYAX     - SPECIFIES Y-AXIS CONTROL (0=PROGRAM SET,1=USER SET)
C 41-45     IZYMAX   - MAXIMUM Z VALUE LABEL ON Y-AXIS (USED IF IYAX=1)
C
C..... LINE # 2
C  1- 5     XO       - X ORIGIN FOR THE X-Y PLOT AXIS
C  6-10     YO       - Y ORIGIN FOR THE X-Y PLOT AXIS
C 11-20     XLEN     - LENGTH OF X-AXIS(INCHES)
C 21-30     YLEN     - LENGTH OF Y-AXIS(INCHES)
C 31-40     XLABL    - X-AXIS LABELING INCREMENT(IN UNITS OF INPUT DATA)
C 41-50     YLABL    - Y-AXIS LABELING INCREMENT(IN UNITS OF INPUT DATA)
C
C......LINES # 3-5
C  ANNOT    - ANNOTATION TO BE PLOTTED BELOW THE GRAPH
C             (A MAXIMUM OF 80 CHARACTERS PER CARD WILL BE PLOTTED
C             AT RHEIGH SIZE)
C
C IF ANOTHER HISTOGRAM IS DESIRED ON THE SAME PLOT, INPUT THE SAME
C SEQUENCE OF LINES.
C
C ---> IMPORTANT, THE USER MUST SET THE # OF RECORDS IN THE IMAGE FILE
C      INSIDE THE SOURCE CODE UNLESS HIS MACHINE WILL ALLOW OTHER
C      TYPES OF DIRECT ACCESS SPECIFICATIONS.
C
C-------------------------------------------------------------------------
C IF THE USER HAS AN IMAGE WITH LINES LARGER THAN 2100 PIXELS THE LINE1
C ARRAY MUST BE MODIFIED ALONG WITH THE NCMAX VARIABLE.
C
C  MAJOR VARIABLES:
```





OFEK06992

```
C   IHIST1 - VECTOR CONTAINING THE FREQUENCY OF EACH DN VALUE
C            (DIMENSIONED TO THE VALUE OF NHIST)
C   IHIST2 - VECTOR USED IN HISTOGRAM STRETCHING
C            (DIMENSIONED TO THE VALUE OF NHIST)
C   NCMAX  - MAXIMUM NUMBER OF COLUMNS OR PIXELS IN A LINE. THIS VALUE
C            MUST NOT BE LARGER THAN THE 'LINE1' ARRAY SIZE.
C   NBLK   - # OF COLUMNS PER LOGICAL RECORD IN THE IMAGE FILE.
C            (MUST BE CONSISTENT WITH OTHER PROGRAMS AND IMPORTANTLY
C            WITH THE ARRAY 'IBYTE' SIZE IN 'RWLINE'
C   RHEIGH - HEIGHT OF NUMERIC LABELS(INCHES)
C   CSIZE  - HEIGHT OF ALPHANUMERIC LABELS(INCHES)
C
C*************************************************************************
      DIMENSION FILTIT(20), IHIST1(256), IHIST2(256)
      INTEGER*2 LINE1(2100)
      DATA  RHEIGH/.07/, CSIZE/.10/, NHIST/256/, IZDN/255/
      DATA NBLK/100/, NCMAX/2100/
      DATA  IERR/0/
C
C-----> 1ST ARGUMENT SHOULD BE SET THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4161,100,L,IREC)
C
C   ....GET HEADER FROM IMAGE FILE
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
C
C  ....COMPUTE # OF RECORDS PER ROW
      NACC=(NCOLS/NBLK) + 1
      IREM=MOD(NCOLS,NBLK)
      IF(IREM.EQ.0) NACC=NACC-1
C
C   ....PRINT BANNER
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C......READ INPUT PARAMETERS
  100 CALL READIN(IBR,IER,IBC,IEC,IYAX,IZYMAX,IBAND,NBAND,NROWS,
     &            NCOLS,ITYPE,PERCEN,XO,YO,XLEN,YLEN,IXLABL,
     &            IYLABL,IERR)
      IF(IERR.EQ.1) GOTO 9999
      NPIX=(IER-IBR+1) * (IEC-IBC+1)
C
C.......COMPUTE FREQUENCY VECTOR
      CALL FREQ(IHIST1,NHIST,IBAND,NBLK,IBR,IER,IBC,IEC,IZDN,
     &          NROWS,NACC,IZMIN,IZMAX,LINE1,NCMAX,IERR)
      IF(IERR.EQ.1) GOTO 9999
C
C.......WATCH OUT FOR STRETCHED HISTOGRAMS
  270 IF(ITYPE.EQ.3) CALL PSTRET(IHIST1,NHIST,IZMIN,IZMAX,NPIX,
     &   PERCEN)
      IF(ITYPE.EQ.2 .OR. ITYPE.EQ.3) CALL STRET(IHIST1,IHIST2,NHIST,
     &   IZMIN,IZMAX,IZDN)
      IF(ITYPE.EQ.2) WRITE(6,201) IZMIN,IZMAX
  201 FORMAT(1X,/' MIN-MAX STRETCH REQUESTED    MIN=',I5,'  MAX=',I5)
      IF(ITYPE.EQ.3) WRITE(6,202) IZMIN,IZMAX
```



OFEK06993

```
  202 FORMAT(1X,/' PERCENT STRETCH REQUESTED    MIN=',I5,'  MAX=',I5)
C
C  ....LIST HISTOGRAM FREQUENCIES
      CALL LSTHIS(IHIST1,NHIST)
C
C.......PLOT THE HISTOGRAM
      CALL PLOTS(-30.,-30.,0)
      CALL PLTHIS(IHIST1,NHIST,IXLABL,IYLABL,IZYMAX,IZDN,XO,YO,
     &                  XLEN,YLEN,RHEIGH,CSIZE,XINC,YINC,IYAX)
C
C  ....PLOT TITLES
      CALL PLTITL(XO,YO,RHEIGH,CSIZE,IERR)
      IF(IERR.EQ.1) GOTO 9000
C......GO SEE IF THERE IS ANOTHER HISTOGRAM TO PLOT
      GOTO 100
C
 9000 CALL PLOT(0.,0.,3)
 9999 STOP
      END
C*************************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*************************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*************************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          HISTOGRAM PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=   ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*************************************************************************
      SUBROUTINE READIN(IBR,IER,IBC,IEC,IYAX,IZYMAX,IBAND,NBAND,
     &                  NROWS,NCOLS,ITYPE,PERCEN,XO,YO,XLEN,YLEN,
     &                  IXLABL,IYLABL,IERR)
C*************************************************************************
C  READS THE PARAMETERS SPECIFIED BY THE USER AND ECHOS THEM OUT.
C  IF A CONFLICT IN THE IMAGE FILE AND THE SUBMATRIX TO BE SPECIFIED
C  IS IDENTIFIED, THE PROGRAM TERMINATES BY SETTING IERR=1.
C*************************************************************************
C
      READ(5,11,END=9500) IBR,IER,IBC,IEC,IBAND,ITYPE,PERCEN,
     &                    IYAX,IZYMAX
   11 FORMAT(6I5,F5.0,2I5)
C
      READ(5,21,END=9500) XO,YO,XLEN,YLEN,IXLABL,IYLABL
   21 FORMAT(4F5.0,2I5)
C
C  ....ECHO OUT PARAMETERS READ IN
```



OFEK06994

```
      WRITE(6,31)IBR,IER,IBC,IEC,IBAND,XO,YO,XLEN,YLEN,
     &           IXLABL,IYLABL
   31 FORMAT(1X//1X,'    SUBMATRIX SPECIFIED'/
     &           1X,'  - ROWS -    - COLUMNS -'/
     &           1X,' BEGIN   END  BEGIN  END     BAND'/
     &           1X,4I6,5X,I2//
     &    1X,'  -ORIGIN-  -AXIS LENGTHS-   -AXIS LABEL INCS.-'/
     &    1X,'   X    Y       X     Y            X    Y'/
     &    1X,2F5.1,3X,F5.1,1X,F5.1,7X,2I5)
      IF(IYAX.EQ.1)  WRITE(6,81) IZYMAX
   81 FORMAT(1X,//' MAXIMUM Y COLUMN VALUE=',I5)
C
      IF(IBAND.LT.1 .OR. IBAND.GT.NBAND) GOTO 9000
      IF(IBR.LT.1 .OR. IBC.LT.1 .OR. IBR.GT.NROWS .OR. IBC.GT.NCOLS)
     &     GOTO 9100
      IF(IER.LT.1 .OR. IEC.LT.1 .OR. IER.GT.NROWS .OR. IEC.GT.NCOLS)
     &     GOTO 9100
      RETURN
C
C .........ERROR MESSAGES.................
 9000 WRITE(6,9001) IBAND,NBAND
 9001 FORMAT(1X,//1X,'*** YOU SPECIFIED AN ILLEGAL BAND # OF ',I2,';'/
     &           1X,'    ONLY ',I2,' BANDS IN IMAGE FILE.')
      IERR=1
      RETURN
 9100 WRITE(6,9101)
 9101 FORMAT(1X,//1X,'*** SUBMATRIX REQUESTED OUTSIDE OF IMAGE '/
     &           1X,'    FILE LIMITS')
 9500 IERR=1
      RETURN
      END
C*************************************************************************
      SUBROUTINE PSTRET(IHIST1,NHIST,IZMIN,IZMAX,NPIX,PERCEN)
C*************************************************************************
C  COMPUTES THE MINIMUM AND MAXIMUM DN VALUES FOR A PERCENTAGE STRETCH
C*************************************************************************
      DIMENSION IHIST1(NHIST)
C
      NIN=((PERCEN/2.)/100.) * FLOAT(NPIX)
      WRITE(6,9) NIN
    9 FORMAT(1X,' HISTOGRAM STRETCHED ',I5,' PIXELS ON SIDES')
C.....GET MAXIMUM VALUE
      ITOTIN=0
   10 ITOTIN=ITOTIN + IHIST1(IZMAX+1)
      IF(ITOTIN.GE.NIN) GOTO 20
      IZMAX=IZMAX-1
      GOTO 10
C
C.....GET MINIMUM VALUE
   20 ITOTIN=0
   30 ITOTIN=ITOTIN + IHIST1(IZMIN+1)
      IF(ITOTIN.GE.NIN) GOTO 50
      IZMIN=IZMIN+1
      GOTO 30
   50 RETURN
```



OFEK06995

```
      END
C*********************************************************************
      SUBROUTINE FREQ(IHIST1,NHIST,IBAND,NBLK,IBR,IER,IBC,IEC,IZDN,
     &                NROWS,NACC,IZMIN,IZMAX,LINE1,NCMAX,IERR)
C*********************************************************************
C COMPUTES THE FREQUENCY OF EACH DN VALUE IN THE SUBMATRIX.
C*********************************************************************
      DIMENSION IHIST1(NHIST)
      INTEGER*2 LINE1(NCMAX)
C
      NHIST1=NHIST+1
      DO 10 I=1,NHIST
   10   IHIST1(I)=0
C
      DO 200 IROW=IBR,IER
        CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,IROW,IBAND,NBLK,0,IERR)
        IF(IERR.EQ.1) RETURN
C.....PUT SUBIMAGE IN VECTOR
        DO 50 JPIX=IBC,IEC
          IDN=LINE1(JPIX)
          IF(IDN.LT.0 .OR. IDN.GT.IZDN) GO TO 50
C.....COMPUTE STATISTICS FOR HISTOGRAM
          IHIST1(IDN+1)=IHIST1(IDN+1)+1
   50     CONTINUE
  200   CONTINUE
C
C.....COMPUTE MAXIMUM AND MINIMUM
      DO 250 I=1,NHIST
        IZMIN=I-1
        IF(IHIST1(I).EQ.0) GOTO 250
        GOTO 260
  250   CONTINUE
      DO 260 I=1,NHIST
        INDX=NHIST1-I
        IZMAX=INDX-1
        IF(IHIST1(INDX).EQ.0) GOTO 260
        GOTO 9000
  260   CONTINUE
 9000 RETURN
      END
C*********************************************************************
      SUBROUTINE LSTHIS(IHIST1,NHIST)
C*********************************************************************
C LISTS THE FREQUENCY FOR THE HISTOGRAM
C
C NOTE: IF THE USER WANTS TO ALTER THE DN VALUE RANGE TO SOMETHING OTHER
C       THAN 255 HE MUST CHANGE THE INCREMENTS WITHIN THE DO LOOP AND
C       THE 'IDNMAX' IN THE DATA STATEMENT.
C*********************************************************************
      DIMENSION IHIST1(NHIST)
      DATA IDNMAX/256/
C
      IF(NHIST.GT.IDNMAX) GOTO 9000
      WRITE(6,11)
   11 FORMAT(1X,//1X,'------ HISTOGRAM TO BE PLOTTED ------'//
```



OFEK06996

```
     & 1X,4(' DN      # ',7X)/)
C
      DO 150 I=1,64
        I1=I-1
        I2=I+63
        I3=I+127
        I4=I+191
        WRITE(6,141) I1,IHIST1(I1+1),I2,IHIST1(I2+1),
     &               I3,IHIST1(I3+1),I4,IHIST1(I4+1)
  141   FORMAT(1X,I3,I7,8X,I3,I7,8X,I3,I7,8X,I3,I7)
  150   CONTINUE
      RETURN
C
C.........ERROR MESSAGE.........
 9000 WRITE(6,9001) NHIST,IDNMAX
 9001 FORMAT(1X,'*** SORRY, YOU ARE REQUESTING A DN VALUE RANGE OF'/
     &       1X,'    UP TO ',I8,'.  LIMITS OF ',I8,' IN EFFECT.'/
     &       1X,'    MODIFY LSTHIS SUBROUTINE FOR LISTING')
      RETURN
      END
C*************************************************************************
      SUBROUTINE STRET(IHIST1,IHIST2,NHIST,IZMIN,IZMAX,IZDN)
C*************************************************************************
C  STRETCHES THE HISTOGRAM
C*************************************************************************
      DIMENSION IHIST1(NHIST), IHIST2(NHIST)
C
      DO 10 I=1,NHIST
   10   IHIST2(I)=0
C
      IB=IZMIN+1
      IE=IZMAX+1
      RANGE=IZMAX-IZMIN+1
C
      DO 50 I=IB,IE
        IDN=I-1
        NUM=IHIST1(I)
        RINDX= ((IDN-IZMIN) / RANGE) * FLOAT(NHIST-1)
        INDX=INT(RINDX)
        IF(INDX.LT.0) INDX=0
        IF(INDX.GT.IZDN) INDX=IZDN
        IHIST2(INDX+1)= IHIST2(INDX+1) + NUM
   50   CONTINUE
C
C......RELOAD ORIGINAL HISTOGRAM
      DO 100 I=1,NHIST
  100   IHIST1(I)=IHIST2(I)
C
      RETURN
      END
C*************************************************************************
      SUBROUTINE PLTHIS(IHIST1,NHIST,IXLABL,IYLABL,IZYMAX,IZDN,XO,YO,
     &                  XLEN,YLEN,RHEIGH,CSIZE,XINC,YINC,IYAX)
C*************************************************************************
C CALLS THE PLOTTER ROUTINES FOR CONSTRUCTING THE HISTOGRAM.
```



OFEK06997

```
C*********************************************************************
      DIMENSION IHIST1(NHIST)
C
C.....CALCULATE MAXIMUM PIXELS IN ANY DN VALUE
      IDNUM=0
      IZDN2=IZDN+1
      DO 400 I=1,IZDN2
        IF(IHIST1(I).GT.IDNUM) IDNUM=IHIST1(I)
  400   CONTINUE
C
      IMAX=(IDNUM/IYLABL)
      YMAX=FLOAT((IMAX*IYLABL)+IYLABL)
      IYMAX=IFIX(YMAX)
C
C.....HAS USER CHOSEN TO SPECIFY MAXIMUM Z VALUE ON Y AXIS?
      IF(IYAX.EQ.1) YMAX=FLOAT(IZYMAX)
      IF(IYAX.EQ.1) IYMAX=IZYMAX
      YINC=YLEN/YMAX
      XINC=XLEN/IZDN
C-------------------------------------------------------------
C
C.....DRAW AND LABEL EACH AXIS
C.....PLOT EACH FREQUENCY COLUMN
      X1=XO
      YINMA=YO+YLEN
C
      DO 500 I=1,IZDN
        X1=X1+XINC
        Y2=YO + (FLOAT(IHIST1(I))*YINC)
        IF(IYAX.NE.1) GO TO 480
        IF(IHIST1(I).GT.IZYMAX) CALL YERR(Y2,YINMA,IHIST1(I),IERR)
  480   CALL PLT(X1,YO,3)
        CALL PLT(X1,Y2,2)
  500   CONTINUE
C
C.....PLOT X-AXIS
      CALL PLT(XO+XINC,YO,3)
      CALL PLT(X1,YO,2)
C
C.....PLOT X-AXIS LABEL
      X9=((X1-XO)/2.) + XO - ((CSIZE*16.)/2.)
      Y9=YO-(RHEIGH*4.)
      CALL SYMBOL(X9,Y9,CSIZE,'BRIGHTNESS VALUE',0.,16)
C
C.....PLOT BOTH Y-AXES
      X1=X1+XINC
      XMAX=X1
      Y2=YO + (YMAX*YINC)
      CALL PLT(XO,YO,3)
      CALL PLT(XO,Y2,2)
      CALL PLT(XMAX,YO,3)
      CALL PLT(XMAX,Y2,2)
C
C.....PLOT Y-AXES LABELS
      Y9=((Y2-YO)/2.) + YO - ((CSIZE*9.)/2.)
```



OFEK06998

```
      IZLONG=NPLEN(YMAX,-1)
      X8=XO-(RHEIGH*(FLOAT(IZLONG)+2.5))
      X9=XMAX+(RHEIGH*(FLOAT(IZLONG)+3.5))
      CALL SYMBOL(X8,Y9,CSIZE,'FREQUENCY',90.,9)
      CALL SYMBOL(X9,Y9,CSIZE,'FREQUENCY',90.,9)
C
C.....LABEL X-AXIS
      IXL=0
      X=XO+(XINC*.5)
      Y=YO-(RHEIGH*1.5)
  515 XL=FLOAT(IXL)
      XOFF=(NPLEN(XL,-1)*RHEIGH)*.5
      X=XO+(XL*XINC)-XOFF
      CALL NUMBER(X,Y,RHEIGH,XL,0.,-1)
      IXL=IXL+IXLABL
      IF(IXL.GT.IZDN) GOTO 600
      GO TO 515
C
C.....LABEL Y-AXES
  600 IYL=0
  615 YL=FLOAT(IYL)
      XOFF=NPLEN(YL,-1)*RHEIGH
      X1=XO-(XOFF+RHEIGH)
      X2=XMAX+RHEIGH+((IZLONG*RHEIGH)-XOFF)
      Y1=YO+(YL*YINC)-(RHEIGH*.5)
      CALL NUMBER(X1,Y1,RHEIGH,YL,0.,-1)
      CALL NUMBER(X2,Y1,RHEIGH,YL,0.,-1)
      IYL=IYL+IYLABL
      IF(IYL.GT.IYMAX) RETURN
      GOTO 615
      END
C*************************************************************************
      SUBROUTINE YERR(Y2,YINMA,IZ,IERR)
C*************************************************************************
C  PLOTS A WARNING FOR A DN VALUE WITH MORE PIXELS THAN USER ALLOWS
C  TO BE PLOTTED AS A HISTOGRAM COLUMN
C*************************************************************************
      IERR=IERR+1
      IF(IERR.GT.10) GO TO 10
      WRITE(6,9) IZ
   9  FORMAT(1X,'** WARNING  FREQUENCY COLUMN TOO LARGE   FREQ=',
     &       I10)
  10  Y2=YINMA
      RETURN
      END
C*************************************************************************
      SUBROUTINE PLTITL(XO,YO,RHEIGH,CSIZE,IQUIT)
C*************************************************************************
C PLOTS THE TITLE BELOW THE HISTOGRAM
C*************************************************************************
      DIMENSION ANNOT(20)
C
C.....EXPECT TO PLOT 3 LINES OF TITLE MAXIMUM
      INC=0
      Y1=YO- ((RHEIGH*4.) + (CSIZE*2.5))
```



OFEK06999

```
  701 READ(5,710,END=9000) ANNOT
  710 FORMAT(20A4)
      INC=INC+1
      CALL SYMBOL(XO,Y1,CSIZE,ANNOT,0.,80)
      Y1=Y1-(CSIZE*1.7)
      IF(INC.LT.3) GO TO 701
      RETURN
 9000 IQUIT=1
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      FUNCTION NPLEN(RNUM,NDEC)
C  THIS ROUTINE CALCULATES THE NUMBER OF DIGITS IN A POSITIVE NUMBER
C  RNUM   - THE POSITIVE REAL NUMBER
C  NDEC   - NUMBER OF DECIMAL PLACES TO CONSIDER IN THE COUNTING OF
C           DIGITS.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      LEN=0
      RVAL=RNUM
C
  100 IF(RVAL.LT.1.00) GO TO 200
       LEN=LEN+1
       RVAL=RVAL/10.
      GO TO 100
  200 NPLEN=LEN+NDEC+1
      IF(RVAL.EQ.0.00) NPLEN=LEN+NDEC+2
      RETURN
      END
C**************************************************************************
      SUBROUTINE PLT(X,Y,IP)
C**************************************************************************
C  CALLS THE APPROPRIATE PEN UP-DOWN MOVES FROM THE PLOTTING SOFTWARE
C  INCORPORATED. THIS ROUTINE MAY EASILY BE REPLACED FOR OTHER
C  SOFTWARE/HARDWARE CONFIGURATIONS.
C
C  IP -CONTROLS THE PEN-UP DOWN MOVEMENTS.
C       IF IP=3, MOVE TO POINT X,Y WITH PEN UP.
C       IF IP=2, MOVE TO POINT X,Y WITH PEN DOWN.
C**************************************************************************
      CALL PLOT(X,Y,IP)
      RETURN
      END
/*
//LKED.SYSLIB DD DSN=SBS.GOULDLIB,DISP=SHR,VOL=SER=,UNIT=
//    DD DSN=FORT.FORTLIB,DISP=SHR
//    DD DSN=ACAD.SUBLIB,DISP=SHR
//GO.FT05F001 DD *
    1  198    1  232    4    1   2.    010000
   1.  1.5  7.0  4.0   64 1000
CHARLESTON TM SCENE   BAND 4
TM READ TEST

/*
//GO.FT09F001 DD DSN=SBS.MEH1B.CHTM7B.R198C232,DISP=SHR,LABEL=(,,,IN)
//GO.SYSVECTR DD DSN=&&VECTORS,UNIT=SYSDA,DISP=(MOD,PASS),
```



OFEK07000

```
//      SPACE=(1320,(1000,400)),DCB=BLKSIZE=1320
//GO.SYSPOUT  DD SYSOUT=*
//BITSET EXEC PGM=BITSN50
//STEPLIB  DD DSN=SBS.GOULDLIB,DISP=SHR
//SYSVECTR DD DSN=&&VECTORS,DISP=(OLD,DELETE)
//SYSUT1   DD DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(18200,200)
//SYSPLOT  DD SYSOUT=(B,,PLOT),DEST=N1R8
//SYSPOUT  DD SYSOUT=*
//
```



OEEK07001

# BRIGHTNESS MAP

```
//BOOKBM JOB (D6600217,99,99),'VF-BOOKBM',TIME=(,19),MSGLEVEL=(1,0),
//  MSGCLASS=A,PASSWORD=??????,USER=SBS01,REGION=500K
/*JOBPARM CARDS=9999
/*ROUTE PRINT MVS.R8
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C BRIGHTNESS MAP
C      CONTINUOUS DENSITY SHADING TECHNIQUE EMPLOYED
C
C
C COLUMNS                DESCRIPTION
C..... LINE # 1:
C  1- 5     BEGINNING ROW OF INPUT MATRIX TO BE PLOTTED
C  6-10     ENDING ROW    OF   "     "     "  "     "
C 11-15     BEGINNING COLUMN   "     "     "  "     "
C 16-20     ENDING COLUMN      "     "     "  "     "
C 21-25     BAND NUMBER TO MAP
C 26-30     CONTROL FOR CONTRAST STRETCH
C       OPTIONS 1-3 INCORPORATE TOBLER'S NON-LINEAR RESPONSE CONSTANT.
C            1=NO STRETCH
C            2=LINEAR STRETCH WITH MINIMUM AND MAXIMUM
C            3=PERCENTAGE LINEAR STRETCH
C       OPTIONS 4-6 ARE ORDINARY LINEAR FUNCTIONS.
C           4=NO STRETCH
C           5=LINEAR STRETCH WITH MINIMUM AND MAXIMUM
C           6=PERCENTAGE LINEAR STRETCH
C           7=PLOTS CELL VALUES WITHIN EACH CELL
C           8=DISCRETE CLASSED MAP
C           (IF YOUR ADVENTURESOME, CARD 3 WILL CONTAIN # OF CLASSES,
C            CARD 4 WILL CONTAIN THE CLASS VALUES;
C            CARD 5 WILL CONTAIN SHADING PATTERNS FOR EACH CLASS;
C            THE TITLE CARDS WILL THEN FOLLOW.)
C
C 31-35     LOWEST POSSIBLE Z VALUE.
C 36-40     HIGHEST POSSIBLE Z VALUE.
C 41-45     TOTAL PERCENTAGE LINEAR STRETCH (BOTH SIDES)
C
C
C..... CARD # 2
C  1- 5     ORIGIN OF GRID MATRIX (LEFTMOST COLUMNS)
C  6-10     ORIGIN OF GRID MATRIX (BOTTOMMOST ROW)
C 11-15     CELL SIZE(INCHES)
C 16-20     MINIMUM DISTANCE BETWEEN SHADING LINES(INCHES)
C                (FOR 100-DOT PER INCH PLOTTER .017 IS A GOOD CHOICE.)
C 21-25     CONTROLS PLOTTING OF CELL BOUNDARIES(0=PLOT,
C 26-30     TSIZE   - SIZE OF TITLE CHARACTERS(OPTIONAL, DEFAULT IS
C                     CHARACTERS EQUAL TO THE CELL SIZE.
C
C..... CARD # 3 AND FOLLOWING
C
C  1-80     TITLES - ANY SORT OF ALPHANUMERIC INFORMATION.
C
C*************************************************************************
```

```
      DIMENSION ICLASS(25), SHD(25)
      DIMENSION TITL1(20), FILTIT(20)
      DIMENSION IHIST1(256), IHIST2(256)
      INTEGER*2 LINE1(2100)
      DATA NBLK/100/, NCMAX/2100/, NBMAX/6/, NCLMAX/25/
      DATA NHIST/256/, IZDN/256/
      IERR=0
C
      INC=1
C
C
C-----------------------------------------------------------------------
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(1195,100,L,IREC)
C
C......READ HEADER RECORD FROM FILE
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
C
C  ....COMPUTE # OF RECORDS PER ROW
      NACC=(NCOLS/NBLK) + 1
      IREM=MOD(NCOLS,NBLK)
      IF(IREM.EQ.0) NACC=NACC+1
C
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C......GET USER SUPPLIED PARAMETERS
      CALL READIN(NROWS,NCOLS,IBR,IER,IBC,IEC,IBAND,NBAND,IMM,CSIZE,
     &            TSIZE,IZMIN,IZMAX,PERCEN,XMIN,YMIN,RLWID,RLABL,
     &            ICPLOT,NCLASS,ICLASS,SHD,INCR,NCLMAX,IERR)
      IF(IERR.EQ.1) GOTO 9999
C
C
C......COMPUTE TOTAL NUMBER OF CELLS IN MATRIX.
      N=((IER-IBR)+1)*((IEC-IBC)+1)
      CHALF=CSIZE*.5
      RNSIZ=CSIZE*.3
C
C......IF CONTRAST STRETCH SPECIFIED COMPUTE MINIMUM AND MAXIMUM VALUES.
      IF(IMM.EQ.1 .OR. IMM.EQ.4 .OR. IMM.EQ.7) GO TO 100
C
C......COMPUTE HISTOGRAM OF STATS
      CALL FREQ(IBAND,NBLK,IBR,IER,IBC,IEC,IZDN,NROWS,NACC,
     &          IZMIN,IZMAX,IHIST1,NHIST,LINE1,NCMAX,IERR)
      IF(IERR.EQ.1) GO TO 9999
C
C......DO WE WANT TO STRETCH THE BRIGHTNESS VALUES?
      IF(IMM.EQ.3 .OR. IMM.EQ.6)
     &     CALL PSTRET(IHIST1,NHIST,IZMIN,IZMAX,N,NPIX,PERCEN)
C.....STRETCH THE HISTOGRAM
      CALL STRET(IHIST1,IHIST2,NHIST,IZMIN,IZMAX,IZDN)
C
C
      IF(IMM.EQ.2 .OR. IMM.EQ.5) WRITE(6,54) IZMIN,IZMAX
```



OFEK07003

```
   54 FORMAT(1X,//1X,'--- Z VALUES USED FOR CONTRAST ENHANCEMENT ---'/
     &       1X,' MINIMUM Z VALUE       MAXIMUM Z VALUE'/
     &       1X,2X,I8,13X,I8)
C
C-------------------------------------------------------------------
C......INITIALIZE PLOTTING DEVICE
  100 CALL PLOTS(-60.,-60.,0)
C
C-------------------------------------------------------------------
C..... SECTION FOR PLOTTING MAP
C.......PLOT CELLS FROM TOP TO BOTTOM, LEFT TO RIGHT.
C......CALCULATE STARTING POSITION FOR ROW 'I'
C
C......CALCULATE SEPARATION LIMITS
       SMAX= CSIZE*1.41421356
       ZBOT= ( (2*SMAX*RLWID) - (RLWID*RLWID) ) / (SMAX*SMAX)
       ZMIN=FLOAT(IZMIN)
       ZMAX=FLOAT(IZMAX)
       NR=IER-IBR+1
       CHEIGH=CSIZE
       IF(CHEIGH.GT..10) CHEIGH=.10
       IF(TSIZE.LE.0.) TSIZE=CHEIGH
       YC2=YMIN+(FLOAT(NR)*CSIZE) + CSIZE + (CHEIGH*6.) + (TSIZE*3.)
       YMAX=YC2
       XMAX=XMIN+ (FLOAT(IEC-IBC+1)*CSIZE)
       YC1=YC2-CSIZE
       XC1=XMIN
       CALL ROWLAB(XC1,YC1,XMAX,CSIZE,CHEIGH,IBR)
       CALL COLLAB(XC1,YC1,CSIZE,CHEIGH,IBC,IEC,INCR)
       DO 900 I=IBR,IER
         XC1=XMIN
         XC2=XMIN+CSIZE
         IF(MOD(I,INCR).EQ.0) CALL ROWLAB(XC1,YC1,XMAX,CSIZE,CHEIGH,I)
         CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,I,IBAND,NBLK,
     &               IRW,IERR)
         IF(IERR.EQ.1) GOTO 9999
         DO 800 J=IBC,IEC
C
           IVAL=LINE1(J)
           REFL=FLOAT(IVAL)
C.........PLOT GRID
           IF(ICPLOT.EQ.0) CALL CELPLT(XC1,YC1,XC2,YC2)
C.........SHADE CELL
           GO TO(600,600,600,600,600,600,700,720), IMM
C
C.........FLIP Z VALUES TO EMULATE AIR PHOTO POSITIVE.
  600      Z=ZMM(ZMIN,ZMAX,REFL)
           ZEYE= (Z* (1.0-ZBOT) ) + ZBOT
           IF(IMM.LE.3) ZEYE=Z**1.4
           SPAC=(RLWID/ZEYE)* (1.+(1.-ZEYE)**0.5)
           CALL OCELS(XC1,YC1,XC2,YC2,SPAC)
           GO TO 750
C.....      PLOT Z-VALUES INSTEAD OF SHADING
  700      CALL CELLAB(XC1,YC1,CHALF,RNSIZ,REFL)
           GO TO 750
```



OFEK07004

```
C.....     DISCRETE CLASSES
  720      CALL DISCRE(XC1,YC1,XC2,YC2,ICLASS,RLWID,SHD,REFL,ZBOT,
     &                 NCLASS,NCLMAX)
C.........COMPUTE COORDINATES OF NEXT CELL
  750      XC1=XC1+CSIZE
           XC2=XC2+CSIZE
  800      CONTINUE
         YC1=YC1-CSIZE
         YC2=YC2-CSIZE
  900    CONTINUE
       CALL ROWLAB(XMIN,YC2,XMAX,CSIZE,CHEIGH,IER)
       XMAX=XC1
C......DRAW BORDER AROUND ENTIRE MAP IF NECESSARY
       IF(ICPLOT.EQ.1) CALL CELPLT(XMIN,YC2,XMAX,YMAX)
C
       YP=YC2
C
C......PLOT ANNOTATION ON BOTTOM OF MAP
       CALL ANNOT(XMIN,XMAX,YP,CHEIGH,IMM)
C
C
C......READ AND PLOT TITLE INFORMATION.
 1991 XP=XMIN
C-------------------------------------------------------------------------
C......CREATE LEGUND FOR PLOT
 2000 IF(IMM.LT.7) CALL LEGUND(ZMIN,ZMAX,XMIN,YMIN,XP,YP,RLABL,
     &              CSIZE,CHEIGH,RLWID,ZBOT,IBC,IEC,IBR,IER,IROT,IMM)
 2100 IF(IMM.EQ.8) CALL DISLEG(XMIN,YP,CSIZE,RLWID,ZBOT,ICLASS,SHD,
     &                         NCLMAX,NCLASS)
C
 2200  YP=YP-TSIZE
C
       CALL TITLE(XP,YP,TSIZE)
C
       CALL PLOT(0.,0.,999)
 9999 STOP
       END
C*************************************************************************
       SUBROUTINE READIN(NROWS,NCOLS,IBR,IER,IBC,IEC,IBAND,NBAND,IMM,
      &                  CSIZE,TSIZE,IZMIN,IZMAX,PERCEN,XMIN,YMIN,RLWID,
      &                  RLABL,ICPLOT,NCLASS,ICLASS,SHD,INCR,NCLMAX,
      &                  IERR)
C*************************************************************************
C  READS THE PARAMETERS SPECIFIED BY THE USER AND ECHOS THEM OUT.
C  IF A CONFLICT IN THE IMAGE FILE AND THE SUBMATRIX TO BE SPECIFIED
C  IS IDENTIFIED, THE PROGRAM TERMINATES BY SETTING IERR=1.
C*************************************************************************
       DIMENSION ICLASS(NCLMAX), SHD(NCLMAX)
       INCR=10
       RLABL=32.
C
C......INPUT CONTROL PARAMETERS.
       READ(5,1,END=9500)IBR,IER,IBC,IEC,IBAND,IMM,IZMIN,IZMAX,PERCEN
     1 FORMAT(8I5,F5.0)
       READ(5,2,END=9500) XMIN,YMIN,CSIZE,RLWID,ICPLOT,TSIZE
```



OFEK07005

```
    2 FORMAT(4F5.0,I5,F5.0)
      IF(IZMAX-IZMIN.LT.20) RLABL=1.
      IF(IMM.NE.8) GO TO 50
      READ(5,5,END=9500) NCLASS,(ICLASS(J),J=1,NCLASS)
    5 FORMAT(I5/(25I2))
      READ(5,6,END=9500) (SHD(J),J=1,NCLASS)
    6 FORMAT((15F5.0))
C
   50 WRITE(6,52)
   52 FORMAT(1X,'---- CONTROL PARAMETERS ----')
C  ....ECHO OUT PARAMETERS READ IN
      WRITE(6,31)IBR,IER,IBC,IEC,IBAND,XMIN,YMIN,IMM
   31 FORMAT(1X//1X,'     SUBMATRIX SPECIFIED'/
     &           1X,'  - ROWS -    - COLUMNS -'/
     &           1X,' BEGIN   END  BEGIN  END     BAND'/
     &           1X,4I6,5X,I2//
     &    1X,'  -ORIGIN- '/
     &    1X,'   X     Y        IMM'/
     &    1X,2F5.1,6X,I2)
C
      IF(IMM.EQ.8) WRITE(6,21) NCLASS,(ICLASS(J),J=1,NCLASS)
   21 FORMAT(1X,'-- NUMBER OF CLASSES=',I3/
     &       1X,'CLASSES TO USE= ',12I6)
      IF(IMM.EQ.8) WRITE(6,22) (SHD(J),J=1,NCLASS)
   22 FORMAT(1X,'--SHADING PATTERNS=',12F5.1)
C
      IF(IBAND.LT.1 .OR. IBAND.GT.NBAND) GOTO 9000
      IF(IBR.LT.1 .OR. IBC.LT.1 .OR. IBR.GT.NROWS .OR. IBC.GT.NCOLS)
     &     GOTO 9100
      IF(IER.LT.1 .OR. IEC.LT.1 .OR. IER.GT.NROWS .OR. IEC.GT.NCOLS)
     &     GOTO 9100
      RETURN
C
C .........ERROR MESSAGES.................
 9000 WRITE(6,9001) IBAND,NBAND
 9001 FORMAT(1X,//1X,'*** YOU SPECIFIED AN ILLEGAL BAND # OF ',I2,';'/
     &           1X,'    ONLY ',I2,' BANDS IN IMAGE FILE.')
      IERR=1
      RETURN
 9100 WRITE(6,9101)
 9101 FORMAT(1X,//1X,'*** SUBMATRIX REQUESTED OUTSIDE OF IMAGE '/
     &           1X,'    FILE LIMITS')
 9500 IERR=1
      RETURN
      END
C*********************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*********************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*********************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          BRIGHTNESS MAPPING PROGRAM'//
```



OFEK07006

```
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=    ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*************************************************************************
      SUBROUTINE PSTRET(IHIST1,NHIST,IZMIN,IZMAX,N,NIN,PERCEN)
C*************************************************************************
C  COMPUTES THE MINIMUM AND MAXIMUM DN VALUES FOR A PERCENTAGE STRETCH
C*************************************************************************
      DIMENSION IHIST1(NHIST)
C
      NIN=((PERCEN/2.)/100.) * FLOAT(N)
      WRITE(6,9) NIN
    9 FORMAT(1X,' HISTOGRAM STRETCHED ',I5,' PIXELS ON SIDES')
C.....GET MAXIMUM VALUE
      ITOTIN=0
   10 ITOTIN=ITOTIN + IHIST1(IZMAX+1)
      IF(ITOTIN.GE.NIN) GOTO 20
      IZMAX=IZMAX-1
      GOTO 10
C
C.....GET MINIMUM VALUE
   20 ITOTIN=0
   30 ITOTIN=ITOTIN + IHIST1(IZMIN+1)
      IF(ITOTIN.GE.NIN) GOTO 50
      IZMIN=IZMIN+1
      GOTO 30
C
   50 WRITE(6,51) IZMIN,IZMAX,PERCEN
   51 FORMAT(1X,//1X,'--- Z VALUES USED FOR CONTRAST ENHANCEMENT ---'/
     &      1X,' MINIMUM Z VALUE      MAXIMUM Z VALUE      PERCENTAGE'/
     &      1X,2X,I8,13X,I8,12X,F6.3)
      RETURN
      END
C*************************************************************************
      SUBROUTINE FREQ(IBAND,NBLK,IBR,IER,IBC,IEC,IZDN,NROWS,NACC,
     &                IZMIN,IZMAX,IHIST1,NHIST,LINE1,NCMAX,IERR)
C*************************************************************************
C COMPUTES THE FREQUENCY OF EACH DN VALUE IN THE SUBMATRIX.
C*************************************************************************
      DIMENSION IHIST1(NHIST)
      INTEGER*2 LINE1(NCMAX)
C
      NHIST1=NHIST+1
      DO 10 I=1,NHIST
   10   IHIST1(I)=0
C
      DO 200 IROW=IBR,IER
        CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,IROW,IBAND,NBLK,0,IERR)
        IF(IERR.EQ.1) RETURN
C.....PUT SUBIMAGE IN VECTOR
```



OEEK07007

```
        DO 50 JPIX=IBC,IEC
          IDN=LINE1(JPIX)
          IF(IDN.LT.0 .OR. IDN.GT.IZDN) GO TO 50
C.....COMPUTE STATISTICS FOR HISTOGRAM
          IHIST1(IDN+1)=IHIST1(IDN+1)+1
   50     CONTINUE
  200   CONTINUE
C
C.....COMPUTE MAXIMUM AND MINIMUM
      DO 250 I=1,NHIST
        IZMIN=I-1
        IF(IHIST1(I).EQ.0) GOTO 250
        GOTO 260
  250   CONTINUE
      DO 260 I=1,NHIST
        INDX=NHIST1-I
        IZMAX=INDX-1
        IF(IHIST1(INDX).EQ.0) GOTO 260
        GOTO 9000
  260   CONTINUE
 9000 RETURN
      END
C*************************************************************************
      SUBROUTINE STRET(IHIST1,IHIST2,NHIST,IZMIN,IZMAX,IZDN)
C*************************************************************************
C  STRETCHES THE HISTOGRAM
C*************************************************************************
      DIMENSION IHIST1(NHIST), IHIST2(NHIST)
C
      DO 10 I=1,NHIST
   10   IHIST2(I)=0
C
      IB=IZMIN+1
      IE=IZMAX+1
      RANGE=IZMAX-IZMIN+1
C
      DO 50 I=IB,IE
        IDN=I-1
        NUM=IHIST1(I)
        RINDX= ((IDN-IZMIN) / RANGE) * FLOAT(NHIST-1)
        INDX=INT(RINDX)
        IF(INDX.LT.0) INDX=0
        IF(INDX.GT.IZDN) INDX=IZDN
        IHIST2(INDX+1)= IHIST2(INDX+1) + NUM
   50   CONTINUE
C
C.....RELOAD ORIGINAL HISTOGRAM
      DO 100 I=1,NHIST
  100   IHIST1(I)=IHIST2(I)
C
      RETURN
      END
C*************************************************************************
      SUBROUTINE OCELS(XC1,YC1,XC2,YC2,SPAC)
C*************************************************************************
```



OFEK07008

```
C  FILLS A SQUARE CELL WITH A PERPENDICULAR CROSS-HATCH LINE MESH.
C     XC1,YC1 - COORDINATES OF LOWER LEFT HAND CORNER OF CELL.
C     XC2,YC2 - COORDINATES OF UPPER RIGHT HAND CORNER OF CELL.
C     SPAC    - DISTANCE BETWEEN SHADING LINES(IN INCHES).
C*************************************************************************
      CHD=SPAC*1.4142136
C
C.....PLOT 45 DEGREE LINE
      Y1=YC2
      X2=XC1
  100 X2=X2+CHD
      Y1=Y1-CHD
      IF(Y1.LT.YC1) GO TO 110
       CALL PLT(XC1,Y1,3)
       CALL PLT(X2,YC2,2)
      GO TO 100
  110 X1=XC1+(YC1-Y1)
      Y2=YC2-(X2-XC2)
  120 IF(Y2.LT.YC1) GO TO 200
       CALL PLT(X1,YC1,3)
       CALL PLT(XC2,Y2,2)
      X1=X1+CHD
      Y2=Y2-CHD
      GO TO 120
C
C.....PLOT 135 DEGREE LINE
  200 X2=XC2
      Y1=YC2
  205 X2=X2-CHD
      Y1=Y1-CHD
      IF(Y1.LT.YC1) GO TO 210
       CALL PLT(XC2,Y1,3)
       CALL PLT(X2,YC2,2)
      GO TO 205
  210 X1=XC2-(YC1-Y1)
      Y2=YC2-(XC1-X2)
  220 IF(Y2.LT.YC1) RETURN
       CALL PLT(X1,YC1,3)
       CALL PLT(XC1,Y2,2)
      X1=X1-CHD
      Y2=Y2-CHD
      GO TO 220
      END
C*************************************************************************
      SUBROUTINE CELPLT(XC1,YC1,XC2,YC2)
C*************************************************************************
C  DRAWS A RECTANGULAR CELL.
C  XC1,YC1 - COORDINATES OF LOWER LEFT HAND CORNER OF CELL
C  XC2,YC2 - COORDINATES OF UPPER RIGHT HAND CORNER OF CELL
C*************************************************************************
      CALL PLT(XC1,YC1,3)
      CALL PLT(XC1,YC2,2)
      CALL PLT(XC2,YC2,2)
      CALL PLT(XC2,YC1,2)
      CALL PLT(XC1,YC1,2)
```



OFEK07009

```
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL FUNCTION ZMM(ZMIN,ZMAX,R)
C  CALCULATE Z VALUE FROM EITHER NO CONTRAST STRETCH OR SIMPLE LINEAR
C  STRETCH
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      Z=ZMIN+(ZMAX-R)
C.....NORMALIZE REFLECTANCE VALUE(AND/OR STRETCH)
      ZMM=(Z-ZMIN)/(ZMAX-ZMIN)
      IF(ZMM.LE.0.) ZMM=.001
      IF(ZMM.GE.1.) ZMM=.999
      RETURN
      END
C*****************************************************************************
      SUBROUTINE CELLAB(XC1,YC1,CHALF,RNSIZ,REFL)
C*****************************************************************************
C  PLOTS THE VALUE OF EACH GRID CELL IN THE CELL
C
C  XC1, YC1 - BOTTOM LEFT COORDINATES OF CELL.
C*****************************************************************************
      LEN=NPLEN(REFL,-1)
C...  ..PLOT VALUE FOR UNROTATED VERSION
      XL=XC1+CHALF-(FLOAT(LEN) *.5 *RNSIZ)
      YL=YC1+CHALF-(RNSIZ *.5)
      CALL NUMBER(XL,YL,RNSIZ,REFL,0.,-1)
      RETURN
      END
C*****************************************************************************
      SUBROUTINE DISCRE(XC1,YC1,XC2,YC2,ICLASS,RLWID,SHD,REFL,ZBOT,
     &                  NCLASS,NCLMAX)
C*****************************************************************************
C SHADES CELLS BASED ON USER DEFINED CLASS INTERVALS AND PERCENTAGE
C  OF AREA INKED.
C*****************************************************************************
      DIMENSION ICLASS(NCLMAX), SHD(NCLMAX)
C
C.....SEARCH FOR CORRECT CLASS, FIND PERCENTAGE TO INK
      DO 100 I=1,NCLASS
        PER=SHD(I)
        IF(REFL.EQ.ICLASS(I)) GO TO 101
  100   CONTINUE
      RETURN
  101 IF(PER.LE.0.) PER=.01
      ZEYE=PER*.01
      ZEYE= (ZEYE* (1.0-ZBOT) ) + ZBOT
      SPAC=(RLWID/ZEYE)*(1.+(1.-ZEYE)**0.5)
      CALL OCELS(XC1,YC1,XC2,YC2,SPAC)
      RETURN
      END
C*****************************************************************************
      SUBROUTINE ROWLAB(XC1,YC1,XMAX,CSIZE,CHEIGH,INUM)
C*****************************************************************************
C   LABELS ROWS WITH THE ROW NUMBER
C*****************************************************************************
```



OFEK07010

```
      ILAB=INUM
      LEN=NPLEN(FLOAT(ILAB),-1)
      XL=XC1 - CHEIGH - (FLOAT(LEN)*CHEIGH)
      YL=YC1+((CSIZE-CHEIGH)/2.)
      CALL NUMBER(XL,YL,CHEIGH,FLOAT(INUM),0.,-1)
      XL=XMAX +CHEIGH
      CALL NUMBER(XL,YL,CHEIGH,FLOAT(INUM),0.,-1)
      RETURN
      END
C*************************************************************************
      SUBROUTINE COLLAB(XC1,YC1,CSIZE,CHEIGH,IBC,IEC,INCR)
C*************************************************************************
C  LABELS ALL OF THE COLUMNS ACROSS THE TOP
C*************************************************************************
      IBC2=IBC+1
      IEC2=IEC-1
      LEN=NPLEN(FLOAT(IBC),-1)
      XL=(XC1+(CSIZE*.5)) - ((FLOAT(LEN)*CHEIGH)*.5)
      YL=YC1+CSIZE+CHEIGH
      CALL NUMBER(XL,YL,CHEIGH,FLOAT(IBC),0.,-1)
C.....UNROTATED VERSION -FOR EACH COLUMN LABEL
      DO 100 I=IBC2,IEC2
        IF(MOD(I,INCR).NE.0) GO TO 100
        LEN=NPLEN(FLOAT(I),-1)
        XLM=(XC1+(CSIZE*FLOAT(I-IBC))) + (CSIZE*.5)
        XL=XLM - ((FLOAT(LEN)*CHEIGH)*.5)
        CALL NUMBER(XL,YL,CHEIGH,FLOAT(I),0.,-1)
  100   CONTINUE
        XLM=XC1+(FLOAT(IEC-IBC+1)*CSIZE) - (CSIZE*.7)
        XL=XLM-(FLOAT(NPLEN(FLOAT(IEC),-1))*CHEIGH*.5)
       CALL NUMBER(XL,YL,CHEIGH,FLOAT(IEC),0.,-1)
       RETURN
       END
C*************************************************************************
      SUBROUTINE ANNOT(XMIN,XMAX,YP,CHEIGH,IMM)
C*************************************************************************
C  PLOTS ANNOTATION CONCERNING THE MAP ON THE BOTTOM
C*************************************************************************
C
      SIZE=CHEIGH
      SIZE2=(XMAX-XMIN)/42.
      IF(SIZE.GT.SIZE2) SIZE=SIZE2
      YP=YP - (SIZE*1.4)
C
      GO TO (1,2,3,1,2,3,7,8),IMM
    1 CALL SYMBOL(XMIN,YP,SIZE,'RAW DATA',0.,8)
      GO TO 10
    2 CALL SYMBOL(XMIN,YP,SIZE,'MIN-MAX STRETCH',0.,15)
      GO TO 10
    3 CALL SYMBOL(XMIN,YP,SIZE,'PERCENTAGE STRETCH',0.,18)
      GO TO 10
    7 CALL SYMBOL(XMIN,YP,SIZE,'CELL VALUES',0.,11)
      GO TO 10
    8 CALL SYMBOL(XMIN,YP,SIZE,'DISCRETE CLASSES',0.,16)
C.....PLOT CREDITS
```



OFEK07011

```
   10 XP=XMAX-(SIZE*22.)
      CALL SYMBOL(XP,YP,SIZE,'BRIGHTNESS MAP PACKAGE',0.,22)
      YP=YP-(SIZE*1.4)
      CALL SYMBOL(XP,YP,SIZE,'  GEOGRAPHY DEPARTMENT',0.,22)
      YP=YP-(SIZE*1.4)
      CALL SYMBOL(XP,YP,SIZE,'  U. OF SOUTH CAROLINA',0.,22)
      IF(IMM.EQ.8) YP=YP-SIZE
      RETURN
      END
C*********************************************************************
      SUBROUTINE LEGUND(ZMIN,ZMAX,XMIN,YMIN,XPT,YPT,RLABL,CSIZE,CHEIGH,
     &                  RLWID,ZBOT,IBC,IEC,IBR,IER,IROT,IMM)
C*********************************************************************
C  THIS ROUTINE PLOTS A CONTINUOUS DENSITY LEGUND AND ALSO ANNOTATES
C   THE LEGUND ACCORDING TO THE SPECIFIED CONTRAST ENHANCEMENT.
C
C  CHEIGH - HEIGHT OF LEGUND LABELS(IN INCHES)
C  ZSHIF  - CHANGE IN Z VALUE FOR NEIGHBORING LEGUND CELLS
C  XSHIF  - DISTANCE(IN INCHES) BETWEEN EACH LABEL OF LEGUND.
C
C*********************************************************************
      NCOL=IEC-IBC+1
      GAPLEG=CHEIGH*4. + CSIZE
      ZSHIF=((ZMAX-ZMIN)+1.0)/FLOAT(NCOL)
      XSHIF=(RLABL/ZSHIF)*CSIZE
       XC1=XMIN
       YC1=YPT-GAPLEG
       YPT=YC1
    7 XC2=XC1+CSIZE
      YC2=YC1+CSIZE
      Z=ZMIN
C
C-------------------------------------------------------------------
C......PLOT CELLS AND SHADE
      DO 100 I=1,NCOL
C.......NORMALIZE DATA
        IF(I.EQ.NCOL) Z=ZMAX
        ZN=ZMM(ZMIN,ZMAX,Z)
        IF(IMM.LE.3) ZN=ZN**1.4
        ZEYE=(ZN* (1.0-ZBOT) ) + ZBOT
        SPAC=(RLWID/ZEYE) * (1.+(1.-ZEYE)**0.5)
C
        CALL CELPLT(XC1,YC1,XC2,YC2)
        CALL OCELS(XC1,YC1,XC2,YC2,SPAC)
C
        Z=Z+ZSHIF
        XC1=XC1+CSIZE
        XC2=XC2+CSIZE
  100   CONTINUE
      X=XPT+(CSIZE/2.)
      Y=YPT+(CSIZE/2.)
C
C-------------------------------------------------------------------
C......ANNOTATE LEGUND
C
```



OFEK07012

```
C......PLOT NUMBERS FOR UNROTATED VERSION
C......PLOT FIRST NUMBER
      LEN=NPLEN(ZMIN,-1)
      YP=YC2+(CHEIGH*1.3)
      XP=X-((FLOAT(LEN)*CHEIGH)/2.)
      CALL NUMBER(XP,YP,CHEIGH,ZMIN,0.,-1)
      IF(IMM.EQ.1 .OR. IMM.EQ.4) GO TO 205
      YP2=YP+(CHEIGH*1.8)
      XP=X-(CHEIGH*2.5)
      CALL SYMBOL(XP,YP2,CHEIGH,5H(MIN),0.,5)
  205 IZ=ZMIN/RLABL
      Z=(FLOAT(IZ)*RLABL)+RLABL
      X=X+((Z-ZMIN)/ZSHIF)*CSIZE
  210 IF(Z.GE.ZMAX) GO TO 300
C......PLOT REMAINING NUMBERS
      LEN=NPLEN(Z,-1)
      XP=X-((FLOAT(LEN)*CHEIGH)/2.)
      CALL NUMBER(XP,YP,CHEIGH,Z,0.,-1)
      X=X+XSHIF
      IF(Z.EQ.ZMAX) GO TO 999
      Z=Z+RLABL
      GO TO 210
  300 LEN=NPLEN(ZMAX,-1)
      X=(NCOL*CSIZE)-(CSIZE/2.) + XMIN
      XP=X-((FLOAT(LEN)*CHEIGH)/2.)
      CALL NUMBER(XP,YP,CHEIGH,ZMAX,0.,-1)
      IF(IMM.EQ.1 .OR. IMM.EQ.4) GO TO 999
      XP=X-(CHEIGH*2.5)
      CALL SYMBOL(XP,YP2,CHEIGH,5H(MAX),0.,5)
  999 RETURN
      END
C*************************************************************************
      SUBROUTINE DISLEG(XMIN,YP,CSIZE,RLWID,ZBOT,ICLASS,SHD,NCLMAX,
     &                  NCLASS)
C*************************************************************************
C  CREATES A LEGEND FOR DISCRETE CLASS MAPS
C*************************************************************************
      DIMENSION ICLASS(NCLMAX), SHD(NCLMAX)
C
      CPART=CSIZE*.75
      CLIT=CSIZE*.13
      YP=YP-CSIZE
      XC2=XMIN+CSIZE
C
C .....PLOT ALL CLASSES
      DO 100 I=1,NCLASS
        YP=YP-CSIZE-CLIT
        YC1=YP-CSIZE
        CALL CELPLT(XMIN,YC1,XC2,YP)
C  .....COMPUTE SHADE PATTERN AND SHADE
        IF(SHD(I).LE.0.) SHD(I)=.01
        ZEYE=SHD(I)*.01
        ZEYE= (ZEYE* (1.0-ZBOT) ) + ZBOT
        SPAC=(RLWID/ZEYE)*(1.+(1.-ZEYE)**0.5)
        CALL OCELS(XMIN,YC1,XC2,YP,SPAC)
```



OFEK07013

```
      X2=XC2+CSIZE
      Y2=YC1+CLIT
      RCLASS=FLOAT(ICLASS(I))
      NDIG=NPLEN(RCLASS,-1)
      CALL NUMBER(X2,Y2,CPART,RCLASS,0.,-1)
  100 CONTINUE
      YP=YP-CSIZE
      RETURN
      END
C*************************************************************************
      SUBROUTINE TITLE(XP,YP,TSIZE)
C*************************************************************************
      DIMENSION TITL(20)
    1 READ(5,11,END=9999) (TITL(J),J=1,20)
   11 FORMAT(20A4)
C
C.....CALCULATE TITLE PLOTTING POSITION.
      YP=YP-(1.5*TSIZE)
      CALL SYMBOL(XP,YP,TSIZE,TITL,0.,80)
      GO TO 1
C
 9999 RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      FUNCTION NPLEN(RNUM,NDEC)
C  THIS ROUTINE CALCULATES THE NUMBER OF DIGITS IN A POSITIVE NUMBER
C  RNUM   - THE POSITIVE REAL NUMBER
C  NDEC   - NUMBER OF DECIMAL PLACES TO CONSIDER IN THE COUNTING OF
C           DIGITS.
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      LEN=0
      RVAL=RNUM
C
  100 IF(RVAL.LT.1.00) GO TO 200
      LEN=LEN+1
      RVAL=RVAL/10.
      GO TO 100
  200 NPLEN=LEN+NDEC+1
      IF(RVAL.EQ.0.00) NPLEN=LEN+NDEC+2
      RETURN
      END
C*************************************************************************
      SUBROUTINE PLT(X,Y,IP)
C*************************************************************************
C  CALLS THE APPROPRIATE PEN UP-DOWN MOVES FROM THE PLOTTING SOFTWARE
C  INCORPORATED. THIS ROUTINE MAY EASILY BE REPLACED FOR OTHER
C  SOFTWARE/HARDWARE CONFIGURATIONS.
C
C  IP -CONTROLS THE PEN-UP DOWN MOVEMENTS.
C       IF IP=3, MOVE TO POINT X,Y WITH PEN UP.
C       IF IP=2, MOVE TO POINT X,Y WITH PEN DOWN.
C*************************************************************************
      CALL PLOT(X,Y,IP)
      RETURN
      END
```



OFEK07014

```
/*
//LKED.SYSLIB DD DSN=SBS.GOULDLIB,DISP=SHR,VOL=SER=,UNIT=
//    DD DSN=FORT.FORTLIB,DISP=SHR
//    DD DSN=ACAD.SUBLIB,DISP=SHR
//GO.FT05F001 DD *
    1  149     1  177     3     5     0  255    6.
  2.0  2.0   .10 .015     0   .12
LOS ANGELES, CA.  BAND 6
/*
//GO.FT09F001 DD DSN=SBS.MEH1B.LAMS4C.R149C177,DISP=SHR
//GO.SYSVECTR DD DSN=&&VECTORS,UNIT=SYSDA,DISP=(MOD,PASS),
//      SPACE=(1320,(1000,400)),DCB=BLKSIZE=1320
//GO.SYSPOUT  DD SYSOUT=*
//BITSET EXEC PGM=BTSIN50
//STEPLIB  DD DSN=SBS.GOULDLIB,DISP=SHR
//SYSVECTR DD DSN=&&VECTORS,DISP=(OLD,DELETE)
//SYSUT1   DD DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(18200,200)
//SYSPLOT  DD SYSOUT=(B,,PLOT),DEST=N1R8
//SYSPOUT  DD SYSOUT=*
//
```



OFEK07015

## CREATE STAT

```
//BOOKCRS JOB (D6600217),'VF-BOOKCRS',TIME=(0,9),MSGLEVEL=(1,0),
// USER=SBS01,PASSWORD=??????
/*ROUTE PRINT MVS.R8
// EXEC DSUTIL
//SYSIN DD *
 DELETE DSN=SBS.MEH.BOOK6C.STATS
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C  CREATESTAT PROGRAM
C
C  THIS PROGRAM MUST BE RUN INITIALLY BEFORE ANY TRAINING SITES
C  ARE SELECTED FOR AN AREA.  THE PROGRAM WRITE 0'S IN ALL SPACES
C  FOR THE STATS.
C
C  THINGS TO SET:
C  COMPUTE THE BLOCKSIZE FROM THE USER'S DOCUMENT.
C  COMPUTE 1/4 THE BLOCKSIZE (I.E., BLOCKSIZE/4)
C  PUT THIS VALUE AS THE DIMENSION OF ILINE ARRAY AND THE DATA STATEMENT
C     FOR ILINE.
C  PUT THE APPROPRIATE # OF RECORDS IN THE VARIABLE NREC AND IN THE
C     DEFINE FILE 10(# STATEMENT.
C*************************************************************************
      DIMENSION ILINE(46)
      DATA NCLASS/25/
      DATA ILINE/46*0/
C
C    ....NUMBER OF RECORDS IS CALCULATED BY:
C        # = (NUMCLS*2) + 2
      DEFINE FILE 10(52,184,L,IREC)
      NREC=52
      DO 100 I=1,NREC
        IREC=I
        WRITE(10'IREC) ILINE
  100   CONTINUE
  900 STOP
      END
//GO.FT10F001 DD DISP=(NEW,CATLG,DELETE),UNIT=3330V,
//     DSN=SBS.MEH.BOOK6C.STATS,SPACE=(184,(52),RLSE),
//     VOL=SER=SBS102,DCB=(RECFM=F,LRECL=184,BLKSIZE=184)
```



OFEK07016

# TRAINING

```
//BOOKTR2 JOB (D6600217),'VF-BOOKTR2',TIME=(,19),MSGLEVEL=(1,0),
// USER=SBS01,PASSWORD=??????
/*ROUTE PRINT MVS.R8
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C  TRAINING PROGRAM
C    COMPUTES STATISTICS FOR TRAINING CLASSES
C
C  EACH TRAINING SITE IS SPECIFIED ON A SEPARATE LINE.
C COLUMNS       DESCRIPTION
C  1- 5     TRAINING CLASS #
C  6-10     BEGINNING ROW OF SITE
C 11-15     ENDING ROW OF SITE
C 16-20     BEGINNING COLUMN OF SITE
C 21-25     ENDING COLUMN OF SITE
C 26-32     CLASS DESCRIPTION
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NCMAX - MAXIMUM # OF COLUMNS OR PIXELS IN A LINE
C          THIS VALUE MUST NOT BE LARGER THAN THE 'LINE.' ARRAY SIZES.
C  NBMAX - THE MAXIMUM NUMBER OF BANDS OF DATA WHICH COULD BE USED.
C          (THIS NUMBER MAY BE LARGER THAN THE ACTUAL NUMBER OF BANDS
C          IN THE IMAGE FILE, HOWEVER, IT MUST NOT BE LARGER THAN THE
C          STATISTICS ARRAY SIZES IN MAIN.
C  NBLK  - # OF PIXELS PER LOGICAL RECORD IN IMAGE FILE.
C           (MUST BE CONSISTENT WITH OTHER PROGRAMS AND IMPORTANTLY
C            WITH THE ARRAY 'IBYTE' SIZE IN 'RWLINE' SUBROUTINE.
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS IN THE IMAGE
C FILE MUST BE SET IN THE 'DEFINE FILE 9(#' STATEMENT; ALSO, THE #
C OF RECORDS AND BLOCKSIZE OF THE TRAINING STATISTICS FILE MUST BE
C SET IN THE 'DEFINE FILE 10(#,##'.
C
C-----> ESSENTIALLY THE ONLY PARAMETER TO RESET INSIDE THE SOURCE CODE
C       FOR RUNNING WITH DIFFERENT FILES ARE THE # OF RECORDS IN THE
C       DEFINE FILE STATMENT.
C
C LOGICAL UNITS USED:
C  5 - USER INPUT ON LINES
C  6 - LINE PRINTER OR TERMINAL
C  9 - IMAGE FILE
C 10 - TRAINING STATISTICS FILE
C*************************************************************************
      DIMENSION ICLASS(25), IROCOL(25,4), NSAMP(25), RNAMEC(25,4)
     &          FILTIT(20)
      DIMENSION MINC(6), MAXC(6), RMEANC(6), VARC(6), STDC(6),
     &          COVC(6,6), CORRC(6,6)
      INTEGER*2 LINE1(2100), LINE2(2100), ISTATU(25)
      DATA NBLK/100/, NCMAX/2100/, NBMAX/6/, NCLMAX/25/, NUMCLS/0/
      DATA ICLASS/25*0/, IROCOL/100*0/
      IERR=0
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
```





OFEK07017

```
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4323,100,L,IREC)
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C----->   # RECS = (NCLMAX * 2) + 2
C-----> 2ND ARGUMENT IS CALCULATED BY:
      DEFINE FILE 10(52,184,L,IRECC)
C
C......READ HEADER RECORD FROM FILE
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
      IF(NBAND.GT.NBMAX) GOTO 9000
C
C  ....COMPUTE # OF RECORDS PER ROW
      NACC=(NCOLS/NBLK) + 1
      IREM=MOD(NCOLS,NBLK)
      IF(IREM.EQ.0) NACC=NACC+1
C
C......PRINT PROGRAM BANNER
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C......READ TRAINING CLASS SPECIFICATIONS
      CALL READIN(NROWS,NCOLS,ICLASS,IROCOL,NSAMP,RNAMEC,
     &            NUMCLS,NCLMAX,IERR)
      IF(IERR.EQ.1) GOTO 9100
C
C......PERFORM STATISTICS COLLECTION FOR ALL CLASSES
      CALL TRAIN(NROWS,NCOLS,NBAND,NACC,NBLK,LINE1,LINE2,
     &           NCMAX,ICLASS,IROCOL,NSAMP,RNAMEC,NUMCLS,
     &           NCLMAX,ISTATU,MINC,MAXC,RMEANC,VARC,STDC,
     &           COVC,CORRC,IERR)
C
      GOTO 9999
C..... ERROR MESSAGES ..........................
 9000 WRITE(6,9001) NBANDS,NBMAX
 9001 FORMAT(1X,'*** SORRY, CANNOT COMPUTE STATISTICS BECAUSE '/
     &       1X,'    THERE ARE ',I2,' BANDS IN THE IMAGE FILE AND'/
     &       1X,'    YOU HAVE THE PROGRAM LIMITS AT ',I2,'.'/
     &       1X,'    INCREASE THE STATS ARRAYS AND NBMAX IN MAIN.')
 9100 WRITE(6,9101)
 9101 FORMAT(1X,//'**** PROCESSING STOPPED BECAUSE OF PREVIOUS ERRORS')
 9999 STOP
      END
C*********************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*********************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*********************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          TRAINING PROGRAM'//
     &  1X,60('*')//
```



OFEK07018

```
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=    ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*******************************************************************************
      SUBROUTINE READIN(NROWS,NCOLS,ICLASS,IROCOL,NSAMP,RNAMEC,
     &                  NUMCLS,NCLMAX,IERR)
C*******************************************************************************
C  READS THE PARAMETERS SPECIFIED BY THE USER AND ECHOS THEM OUT.
C  IF A CONFLICT IN THE IMAGE FILE AND THE TRAINING SITE IS IDENTIFIED,
C  THE PROGRAM TERMINATES BY SETTING IERR=1.
C
C MAJOR VARIABLES:
C   ICLASS - ARRAY OF CLASS NUMBERS FOR ALL SAMPLES.
C   IROCOL - ARRAY OF BEGINNING ROW, ENDING ROW, BEGINNING
C            COLUMN, AND ENDING COLUMN FOR EACH TRAINING
C            SITE OF EACH CLASS.
C   NSAMP  - NUMBER OF SAMPLE SITES FOR EACH CLASS.
C*******************************************************************************
      DIMENSION ICLASS(NCLMAX), IROCOL(NCLMAX,4), NSAMP(NCLMAX),
     *          RNAMEC(NCLMAX,4)
      DIMENSION RNAME(4)
C
      WRITE(6,11)
   11 FORMAT(1X,'----> READING SAMPLE SITE SPECIFICATIONS '//
     &       1X,'          BEGIN  END     BEGIN      END'/
     &       1X,' CLASS    ROW    ROW     COLUMN     COLUMN     NAME'/)
      NUM=1
    1 READ(5,21,ERR=9000,END=100) ICLASS(NUM),IROCOL(NUM,1),
     &                  IROCOL(NUM,2),IROCOL(NUM,3),IROCOL(NUM,4),
     &                  RNAME
   21 FORMAT(5I5,4A4)
      DO 22 ICHAR=1,4
   22   RNAMEC(ICLASS(NUM),ICHAR)= RNAME(ICHAR)
      DO 25 I=1,2
        I2=I+2
        IF(IROCOL(NUM,I).LT.1 .OR. IROCOL(NUM,I).GT.NROWS) GOTO 9100
        IF(IROCOL(NUM,I2).LT.1 .OR. IROCOL(NUM,I2).GT.NCOLS) GOTO 9100
   25   CONTINUE
        IF(IROCOL(NUM,1) .GT. IROCOL(NUM,2)) GOTO 9100
        IF(IROCOL(NUM,3) .GT. IROCOL(NUM,4)) GOTO 9100
      WRITE(6,26) ICLASS(NUM),IROCOL(NUM,1),IROCOL(NUM,2),
     &            IROCOL(NUM,3),IROCOL(NUM,4),RNAME
   26 FORMAT(1X,I5,2I6,4X,I6,3X,I6,5X,4A4)
      NUM=NUM+1
      GOTO 1
  100 IF(IERR.EQ.1) RETURN
      NUM=NUM-1
      NUM2=NUM-1
C.....SORT SAMPLES BY CLASS
      DO 150 I=1,NUM2
        I2=I+1
```



OFEK07019

```
        DO 140 J=I2,NUM
          IF(ICLASS(J).GT.ICLASS(I)) GOTO 140
                CALL SHUFFL(ICLASS(I),ICLASS(J))
                CALL SHUFFL(IROCOL(I,1),IROCOL(J,1))
                CALL SHUFFL(IROCOL(I,2),IROCOL(J,2))
                CALL SHUFFL(IROCOL(I,3),IROCOL(J,3))
                CALL SHUFFL(IROCOL(I,4),IROCOL(J,4))
  140     CONTINUE
  150   CONTINUE
C
C......COMPUTE NUMBER OF SAMPLES PER CLASS
      NUMCLS=0
      IB=1
      IE=2
C  .....INCREMENT TO FIND RANGE OF SAMPLES
  210 IF(IE.GT.NUM) GOTO 220
      IF(ICLASS(IE).NE.ICLASS(IB)) GOTO 220
      IE=IE+1
      GOTO 210
C   ....# SAMPLES FOR A CLASS FOUND
  220 NUMCLS=NUMCLS+1
      NSAMP(NUMCLS)=IE-IB
      IF(NSAMP(NUMCLS).GT.4) GOTO 9200
      IF(IE.GT.NUM) RETURN
      IB=IE
      IE=IB+1
      GOTO 210
C........ ERROR MESSAGES ..............................
 9000 WRITE(6,9001)
 9001 FORMAT(1X,/' **** ERROR - PROBLEM READING A SAMPLE; CHECK INPUT')
 9002 IERR=1
      GOTO 1
 9100 WRITE(6,9101) ICLASS(NUM),IROCOL(NUM,1),IROCOL(NUM,2),
     &             IROCOL(NUM,3),IROCOL(NUM,4)
 9101 FORMAT(1X,I5,2I6,4X,I6,3X,I6/
     &       1X,'**** ERROR - LAST SAMPLE OUTSIDE OF IMAGE')
      IERR=1
      GOTO 1
 9200 WRITE(6,9001) NSAMP(NUMCLS), NUMCLS
 9201 FORMAT(1X,/' **** SORRY - NO MORE THAN 4 SAMPLES PER CLASS ARE'/
     &           '              ALLOWED. YOU REQUESTED ',I2,' CLASSES'/
     &           '              SAMPLES FOR CLASS # ',I2)
      IERR=1
      GOTO 210
      END
C*************************************************************************
      SUBROUTINE SHUFFL(IVAR1,IVAR2)
C*************************************************************************
C  SWITCHES TWO ELEMNTS
C*************************************************************************
      IVARH=IVAR1
      IVAR1=IVAR2
      IVAR2=IVARH
      RETURN
      END
```



OFEK07020

```
C*********************************************************************
      SUBROUTINE TRAIN(NROWS,NCOLS,NBAND,NACC,NBLK,LINE1,LINE2,
     &           NCMAX,ICLASS,IROCOL,NSAMP,RNAMEC,NUMCLS,
     &           NCLMAX,ISTATU,MINC,MAXC,RMEANC,VARC,STDC,
     &           COVC,CORRC,IERR)
C*********************************************************************
C  COLLECTS THE STATISTICS FOR EACH CLASS AND WRITES THEM
C  INTO A DIRECT ACCESS FILE
C
C IROCOL(1,1) = BEGIN ROW          IROCOL(1,2) = END ROW
C IROCOL(1,3) = BEGIN COLUMN       IROCOL(1,4) = END COLUMN
C*********************************************************************
      DOUBLE PRECISION SUMX, SS, SP
      INTEGER*2 LINE1(NCMAX), LINE2(NCMAX)
C ... NOTE: COVC AND CORRC ARRAYS ARE STORED AS COVC(ROW,COLUMN)
      DIMENSION MINC(NBAND), MAXC(NBAND), RMEANC(NBAND),
     &          VARC(NBAND), STDC(NBAND), COVC(NBAND,NBAND),
     &          CORRC(NBAND,NBAND)
      DIMENSION ICLASS(NCLMAX), IROCOL(NCLMAX,4), NSAMP(NCLMAX),
     &          RNAMEC(NCLMAX,4)
      INTEGER*2 ISTATU(NCLMAX)
      DIMENSION RNAME(4), ICOORD(4,4)
C
C.......READ STATISTICS FILE HEADER
       CALL RWSTHE(ISTATU,NCLMAX,0)
C
C......COMPUTE SIMPLE STATISTICS FOR EACH CHANNEL
      INDX1=1
      DO 500 ICLS=1,NUMCLS
        INDX2=INDX1 + NSAMP(ICLS)-1
C
        DO 100 I=1,NBAND
          MINC(I)=99999
          MAXC(I)=-99999
          SUMX=0.
          ITOTPX=0
C
          DO 60 ISAMP=INDX1,INDX2
            IS=ISAMP-INDX1+1
            IBR=IROCOL(ISAMP,1)
            IER=IROCOL(ISAMP,2)
            IBC=IROCOL(ISAMP,3)
            IEC=IROCOL(ISAMP,4)
            ICOORD(IS,1)=IBR
            ICOORD(IS,2)=IER
            ICOORD(IS,3)=IBC
            ICOORD(IS,4)=IEC
            ITOTPX=ITOTPX + ((IER-IBR)+1)*((IEC-IBC)+1)
              DO 50 J=IBR,IER
                CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,J,I,NBLK,0,IERR)
                IF(IERR.EQ.1) RETURN
                DO 20 IPIX=IBC,IEC
                SUMX=SUMX + LINE1(IPIX)
                IF(LINE1(IPIX).LT.MINC(I)) MINC(I)=LINE1(IPIX)
                IF(LINE1(IPIX).GT.MAXC(I)) MAXC(I)=LINE1(IPIX)
```



OFEK07021

```
   20          CONTINUE
   50        CONTINUE
   60      CONTINUE
C
         TOTPI2=ITOTPX-1
         RMEANC(I)=SUMX/FLOAT(ITOTPX)
  100    CONTINUE
C
C......CALCULATE 1ST AND 2ND MOMENTS AND VARIANCE-COVARIANCE
       DO 300 I=1,NBAND
          SS=0.
         DO 200 J=1,I
           SP=0.
           DO 160 ISAMP=INDX1,INDX2
             IBR=IROCOL(ISAMP,1)
             IER=IROCOL(ISAMP,2)
             IBC=IROCOL(ISAMP,3)
             IEC=IROCOL(ISAMP,4)
C          ....FOR EACH ROW OF THE IMAGE
             DO 150 K=IBR,IER
             CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,K,I,NBLK,0,IERR)
             IF(IERR.EQ.1) RETURN
             CALL RWLINE(9,LINE2,NCMAX,NROWS,NACC,K,J,NBLK,0,IERR)
             IF(IERR.EQ.1) RETURN
               DO 130 IP=IBC,IEC
                 SP=SP+(LINE1(IP)-RMEANC(I)) * (LINE2(IP)-RMEANC(J))
  130            CONTINUE
               IF(I.NE.J) GOTO 150
               DO 140 IP=IBC,IEC
                 SS= SS + (LINE1(IP)-RMEANC(I))**2
  140            CONTINUE
  150          CONTINUE
  160        CONTINUE
           IF(I.EQ.J) VARC(I)= SS / TOTPI2
           IF(I.EQ.J) STDC(I)= SQRT(VARC(I))
           COVC(I,J)= SP / TOTPI2
C
  200      CONTINUE
C
C     ......FILL UPPER RIGHT OF MATRIX
         IF(I.EQ.1) GOTO 300
         IE=I-1
         DO 280 JC=1,IE
  280      COVC(JC,I)=COVC(I,JC)
  300    CONTINUE
C
C.....CALCULATE CORRELATION STATISTICS
       DO 400 I=1,NBAND
         DO 350 J=1,I
           CORRC(I,J)= COVC(I,J) / (STDC(I)*STDC(J))
  350      CONTINUE
  400    CONTINUE
C
C.....PRINT OUT STATISTICS
       DO 410 ICHAR=1,4
```



OFEK07022

```
  410    RNAME(ICHAR)= RNAMEC(ICLS,ICHAR)
         WRITE(6,411) RNAME,ICLS,NSAMP(ICLS),ITOTPX
  411    FORMAT('1',//1X,5('-'),'> STATS FOR ',4A4,'  CLASS #',I3,
     &          I2,' SAMPLES  ',I8,' PIXELS'//)
         CALL PUSTAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NBAND)
C
C.....WRITE RECORD TO STATS FILE
         CALL RWSTAT(RMEANC,VARC,MINC,MAXC,COVC,RNAME,ICLS,NBAND,
     &              NSAMP(ICLS),ICOORD,ITOTPX,1)
         ISTATU(ICLS)=1
         INDX1=INDX2+1
  500    CONTINUE
C
C.....UPDATE STATISTICS FILE HEADER
         CALL RWSTHE(ISTATU,NCLMAX,1)
       RETURN
       END
C*************************************************************************
      SUBROUTINE PUSTAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NBAND)
C*************************************************************************
C PRINT THE SUMMARY STATISTICS FOR ALL BANDS OF A TRAINING CLASS
C
C LIMITS: THE WRITE STATEMENTS ARE SET UP FOR NO MORE THAN 8 BANDS OF
C         DATA PER IMAGE FILE. IF YOU WANT MORE, CHANGE ARRAY 'ICOUN',
C         ITS DATA STATEMENT, THE 'NOMAX' DATA STATEMENT, AND THE 8I'S
C         AND 8F'S IN THE OUTPUT STATEMENTS.
C*************************************************************************
      DIMENSION RMEANC(NBAND), STDC(NBAND), VARC(NBAND),
     &          MINC(NBAND), MAXC(NBAND), COVC(NBAND,NBAND),
     &          CORRC(NBAND,NBAND)
      DIMENSION ICOUN(8)
      DATA ICOUN/1,2,3,4,5,6,7,8/, NOMAX/8/
C
      IF(NBAND.GT.NOMAX) GOTO 9000
C
C.....PRINT OUT BANNER AND SIMPLE STATS
      WRITE(6,101) (ICOUN(J),J=1,NBAND)
  101 FORMAT(1X,'    UNIVARIATE STATISTICS'/
     &       1X,70('_')/
     &       1X,' BAND  ',8I10)
      WRITE(6,111)
  111 FORMAT(1X,70('_'))
C
      WRITE(6,102) (RMEANC(J),J=1,NBAND)
  102 FORMAT(1X,'MEAN      ',8F10.1)
      WRITE(6,103) (STDC(J),J=1,NBAND)
  103 FORMAT(1X,'STD. DEV.',1X,8F10.2)
      WRITE(6,104) (VARC(J),J=1,NBAND)
  104 FORMAT(1X,'VARIANCE ',8F10.1)
      WRITE(6,105) (MINC(J),J=1,NBAND)
  105 FORMAT(1X,'MINIMUM  ',8I10)
      WRITE(6,106) (MAXC(J),J=1,NBAND)
  106 FORMAT(1X,'MAXIMUM  ',8I10)
      WRITE(6,111)
C
```



OFEK07023

```
C......PRINT OUT COVARIANCE MATRIX
      WRITE(6,121)
  121 FORMAT(1X,////'    VARIANCE-COVARIANCE MATRIX'/1X,70('_'))
      WRITE(6,122) (ICOUN(J),J=1,NBAND)
  122 FORMAT(1X,'BAND',8I10)
      WRITE(6,111)
      DO 150 I=1,NBAND
        WRITE(6,123) I,(COVC(I,J),J=1,I)
  123   FORMAT(1X,I2,4X,8F10.2)
  150   CONTINUE
      WRITE(6,111)
C
C......PRINT OUT CORRELATION MATRIX
      WRITE(6,221)
  221 FORMAT(1X,////'    CORRELATION MATRIX'/1X,70('_'))
      WRITE(6,222) (ICOUN(J),J=1,NBAND)
  222 FORMAT(1X,'BAND',8I10)
      WRITE(6,111)
      DO 250 I=1,NBAND
        WRITE(6,223) I,(CORRC(I,J),J=1,I)
  223   FORMAT(1X,I2,4X,8F10.2)
  250   CONTINUE
      WRITE(6,111)
      RETURN
C
C......ERROR MESSAGES.............
 9000 WRITE(6,9001) NBAND,NOMAX
 9001 FORMAT(1X,'*** SORRY, YOU ARE TRANSGRESSING THE LIMITS OF THE'/
     &       1X,'    PUSTAT SUBROUTINE BY USING ',I2,' BANDS. LIMITS'/
     &       1X,'    OF ',I2,' BANDS IN EFFECT.  MODIFY PUSTAT '/
     &       1X,'    SUBROUTINE.')
      RETURN
      END
C*****************************************************************
      SUBROUTINE RWSTHE(ISTATU,IRW)
C*****************************************************************
C  READS OR WRITES THE HEADER FROM THE STATISTICS FILE
C*****************************************************************
      INTEGER*2 ISTATU(50)
      IRECC=1
      IF(IRW.EQ.0) READ(10'IRECC) ISTATU
      IF(IRW.EQ.1) WRITE(10'IRECC) ISTATU
      RETURN
      END
C*****************************************************************
      SUBROUTINE RWSTAT(RMEANC,VARC,MINC,MAXC,COVC,RNAME,
     &                  ICLS,NSITE,ITOTPX,IRW)
C*****************************************************************
C  READS OR WRITES THE STATISTICS FOR A CLASS TO DISK
C*****************************************************************
      DIMENSION MINC(6), MAXC(6), RMEANC(6), VARC(6),
     &          COVC(6,6), RNAME(6)
C
C.....COMPUTE BEGINNING RECORD
  100 IRECC=(ICLS-1)*2 + 2
      IF(IRW.EQ.1) GOTO 110
      READ(10'IRECC) RNAME,NSITE,ITOTPX,RMEANC,VARC,
     &               MINC,MAXC
      IRECC=(ICLS-1)*2 + 3
      READ(10'IRECC) COVC
      RETURN
  110 WRITE(10'IRECC) RNAME,NSITE,ITOTPX,RMEANC,VARC,
     &               MINC,MAXC
      IRECC=(ICLS-1)*2 + 3
      WRITE(10'IRECC) COVC
      RETURN
      END
```



OFEK07024

```
/*
//GO.FT05F001 DD *
    1   70   80  125  132RESIDENTIAL
    2   38   41  159  162COMMERCIAL
    2   60   65   84   90COMMERCIAL
    3  152  160   98  108WETLAND
    4  179  185   34   39FOREST
    5  160  170  180  190WATER
//GO.FT09F001 DD DSN=SBS.MEH1B.BOOK6C.R240C256,DISP=SHR,LABEL=(,,,IN)
//GO.FT10F001 DD DSN=SBS.MEH.BOOK6C.STATS,DISP=OLD
```



OFEK07025

# PRINTSTATS

```
//BOOKPS JOB (D6600217),'VF-BOOKPS',TIME=(,19),MSGLEVEL=(1,0),
// USER=SBS01,PASSWORD=??????
/*ROUTE PRINT MVS.R8
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C  PRINTSTATS PROGRAM
C    PRINTS THE STATISTICS FOR TRAINING CLASSES
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NBMAX - THE MAXIMUM NUMBER OF BANDS OF DATA WHICH COULD BE USED.
C          (THIS NUMBER MAY BE LARGER THAN THE ACTUAL NUMBER OF BANDS
C          IN THE IMAGE FILE, HOWEVER, IT MUST NOT BE LARGER THAN THE
C          STATISTICS ARRAY SIZES IN MAIN.
C  NBLK  - # OF PIXELS PER LOGICAL RECORD IN IMAGE FILE.
C           (MUST BE CONSISTENT WITH OTHER PROGRAMS AND IMPORTANTLY
C            WITH THE ARRAY 'IBYTE' SIZE IN 'RWLINE' SUBROUTINE.
C  NCLMAX- MAXIMUM # OF TRAINING CLASSES WHICH MAY BE SELECTED
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS IN THE IMAGE
C FILE MUST BE SET IN THE 'DEFINE FILE 9(#' STATEMENT; ALSO, THE #
C OF RECORDS AND BLOCKSIZE OF THE TRAINING STATISTICS FILE MUST BE
C SET IN THE 'DEFINE FILE 10(#,##'.
C
C LOGICAL UNITS USED:
C  6 - LINE PRINTER OR TERMINAL
C  9 - IMAGE FILE
C 10 - TRAINING STATISTICS FILE
C*************************************************************************
      DIMENSION RNAMEC(25,4), FILTIT(20)
      DIMENSION MINC(6), MAXC(6), RMEANC(6), VARC(6), STDC(6),
     &          COVC(6,6), CORRC(6,6)
      INTEGER*2 ISTATU(25)
      DATA NBLK/100/, NBMAX/6/, NCLMAX/25/
      IERR=0
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4323,100,L,IREC)
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C----->   # RECS = (NCLMAX * 2) + 2
      DEFINE FILE 10(52,184,L,IRECC)
C
C......READ HEADER RECORD FROM FILE
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
      IF(NBAND.GT.NBMAX) GOTO 9000
C
C.........READ HEADER FROM STATS FILE
        CALL RWSTHE(ISTATU,NCLMAX,0)
C
C......PRINT PROGRAM BANNER
```

```
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C.....PRINT STATISTICS FOR ALL CLASSES
      CALL PSTAT(NBAND,NCLMAX,MINC,MAXC,RMEANC,VARC,STDC,COVC,CORRC,
     &           ISTATU,IERR)
      GOTO 9999
C
C.......ERROR MESSAGES .............
 9000 WRITE(6,9001) NBANDS,NBMAX
 9001 FORMAT(1X,'*** SORRY, CANNOT LIST STATISTICS BECAUSE '/
     &       1X,'    THERE ARE ',I2,' BANDS IN THE IMAGE FILE AND'/
     &       1X,'    YOU HAVE THE PROGRAM LIMITS AT ',I2,'.'/
     &       1X,'    INCREASE THE STATS ARRAYS AND NBMAX IN MAIN.')
 9999 STOP
      END
C*************************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*************************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*************************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          PRINT STATISTICS PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=    ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*************************************************************************
      SUBROUTINE PSTAT(NBAND,NCLMAX,MINC,MAXC,RMEANC,VARC,STDC,COVC,
     &           CORRC,ISTATU,IERR)
C*************************************************************************
C  LOOPS THROUGH ALL THE STATISTICS COLLECTED AND PRINTS THEM
C
C*************************************************************************
C ... NOTE: COVC AND CORRC ARRAYS ARE STORED AS COVC(ROW,COLUMN)
      DIMENSION MINC(NBAND), MAXC(NBAND), RMEANC(NBAND), VARC(NBAND),
     &          STDC(NBAND),COVC(NBAND,NBAND), CORRC(NBAND,NBAND)
      DIMENSION RNAME(4), ICOORD(4,4)
      INTEGER*2 ISTATU(NCLMAX)
C
C.........PRINT OUT STATISTICS
      DO 100 IPOS=1,NCLMAX
        IF(ISTATU(IPOS).EQ.0) GOTO 100
        CALL RWSTAT(RMEANC,VARC,MINC,MAXC,COVC,RNAME,IPOS,NBAND,
     &              NSITE,ICOORD,ITOTPX,IRW)
        WRITE(6,11) RNAME,IPOS,NSITE,ITOTPX
   11   FORMAT('1',//1X,5('-'),'> STATS FOR ',4A4,'  CLASS #',I3,
     &         I2,' SAMPLES  ',I8,' PIXELS'//)
        DO 20 I=1,NBAND
```



OFEK07027

```
            STDC(I)= SQRT(VARC(I))
   20       CONTINUE
         DO 60 I=1,NBAND
           DO 50 J=1,I
            CORRC(I,J)= COVC(I,J) / (STDC(I)*STDC(J))
   50       CONTINUE
   60      CONTINUE
         CALL PUSTAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NBAND)
  100    CONTINUE
       RETURN
       END
C*******************************************************************************
      SUBROUTINE PUSTAT(RMEANC,STDC,VARC,MINC,MAXC,COVC,CORRC,NBAND)
C*******************************************************************************
C PRINT THE SUMMARY STATISTICS FOR ALL BANDS OF A TRAINING CLASS
C
C LIMITS: THE WRITE STATEMENTS ARE SET UP FOR NO MORE THAN 8 BANDS OF
C         DATA PER IMAGE FILE. IF YOU WANT MORE, CHANGE ARRAY 'ICOUN',
C         ITS DATA STATEMENT, THE 'NOMAX' DATA STATEMENT, AND THE 8I'S
C         AND 8F'S IN THE OUTPUT STATEMENTS.
C*******************************************************************************
      DIMENSION RMEANC(NBAND), STDC(NBAND), VARC(NBAND),
     &          MINC(NBAND), MAXC(NBAND), COVC(NBAND,NBAND),
     &          CORRC(NBAND,NBAND)
      DIMENSION ICOUN(8)
      DATA ICOUN/1,2,3,4,5,6,7,8/, NOMAX/8/
C
      IF(NBAND.GT.NOMAX) GOTO 9000
C
C.....PRINT OUT BANNER AND SIMPLE STATS
      WRITE(6,101) (ICOUN(J),J=1,NBAND)
  101 FORMAT(1X,'    UNIVARIATE STATISTICS'/
     &       1X,70(' ')/
     &       1X,' BAND  ',8I10)
      WRITE(6,111)
  111 FORMAT(1X,70('_'))
C
      WRITE(6,102) (RMEANC(J),J=1,NBAND)
  102 FORMAT(1X,'MEAN      ',8F10.1)
      WRITE(6,103) (STDC(J),J=1,NBAND)
  103 FORMAT(1X,'STD. DEV.',1X,8F10.2)
      WRITE(6,104) (VARC(J),J=1,NBAND)
  104 FORMAT(1X,'VARIANCE ',8F10.1)
      WRITE(6,105) (MINC(J),J=1,NBAND)
  105 FORMAT(1X,'MINIMUM  ',8I10)
      WRITE(6,106) (MAXC(J),J=1,NBAND)
  106 FORMAT(1X,'MAXIMUM  ',8I10)
      WRITE(6,111)
C
C......PRINT OUT COVARIANCE MATRIX
      WRITE(6,121)
  121 FORMAT(1X,////'    VARIANCE-COVARIANCE MATRIX'/1X,70('_'))
      WRITE(6,122) (ICOUN(J),J=1,NBAND)
  122 FORMAT(1X,'BAND',8I10)
      WRITE(6,111)
```



OFEK07028

```
      DO 150 I=1,NBAND
        WRITE(6,123) I,(COVC(I,J),J=1,I)
  123   FORMAT(1X,I2,4X,8F10.2)
  150   CONTINUE
      WRITE(6,111)
C
C.....PRINT OUT CORRELATION MATRIX
      WRITE(6,221)
  221 FORMAT(1X,////'     CORRELATION MATRIX'/1X,70('_'))
      WRITE(6,222) (ICOUN(J),J=1,NBAND)
  222 FORMAT(1X,'BAND',8I10)
      WRITE(6,111)
      DO 250 I=1,NBAND
        WRITE(6,223) I,(CORRC(I,J),J=1,I)
  223   FORMAT(1X,I2,4X,8F10.2)
  250   CONTINUE
      WRITE(6,111)
      RETURN
C
C.....ERROR MESSAGES.................
 9000 WRITE(6,9001) NBAND,NOMAX
 9001 FORMAT(1X,'*** SORRY, YOU ARE TRANSGRESSING THE LIMITS OF THE'/
     &       1X,'    PUSTAT SUBROUTINE BY USING ',I2,' BANDS. LIMITS'/
     &       1X,'    OF ',I2,' BANDS IN EFFECT.  MODIFY PUSTAT '/
     &       1X,'    SUBROUTINE.')
      RETURN
      END
/*
//GO.FT09F001 DD DSN=SBS.MEH1B.BOOK6C.R240C256,DISP=SHR,LABEL=(,,,IN)
//GO.FT10F001 DD DSN=SBS.MEH.BOOK6C.STATS,DISP=SHR,LABEL=(,,,IN)
```



OFEK07029

# PIPED

```
//BOOKPP JOB (D6600217,99,99),'VF-BOOKPP',MSGLEVEL=(1,0),MSGCLASS=A,
//  TIME=(,9),PASSWORD=??????,USER=SBS01
/*ROUTE  PRINT MVS.R8
/*ROUTE  PUNCH RMT8
/*JOBPARM CARDS=9999
//  EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C  BOOKPP
C   PLOTS TRAINING CLASSES AS PARALLELEPIPEDS
C
C
C INPUT DATA LINES:
C COLUMNS  VARIABLE                  DESCRIPTION
C
C..LINE # 1
C  1- 4     TYPE    - CONTROL OF THE NUMBER OF CLASSES TO BE PLOTTED
C                      'ALL ' - FOR ALL CLASSES IN THE FILE
C                      'USER' - FOR JUST THOSE THE USER SPECIFIES
C  6-10     DEV     - THE NUMBER OF STANDARD DEVIATIONS FOR PIPEDS
C 16-25     XZROT   - ROTATION AROUND THE X-AXIS(DEGREES)
C 26-35     YZROT   - ROTATION AROUND THE Y-AXIS(DEGREES)
C 36-37     IC1     - CHANNEL TO USE AS X-AXIS
C 38-39     IC2     - CHANNEL TO USE AS Y-AXIS
C 40-41     IC3     - CHANNEL TO USE AS Z-AXIS
C
C..LINE # 2
C  1- 5     NCLASS  - THE NUMBER OF CLASSES TO PLOT
C  6- 8             - 1ST CLASS TO PLOT
C  9-11             - 2ND CLASS TO PLOT
C 12-14             - AND SO ON IN I3 FORMAT
C
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NBMAX - MAXIMUM # OF BANDS USED BY THE PROGRAM; SHOULD BE >= THE
C          NUMBER OF BANDS IN THE IMAGE FILE AND EQUAL TO THE ARRAY
C          SIZES FOR THE STATISTICS IN MAIN.
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS IN THE IMAGE
C FILE MUST BE SET IN THE 'DEFINE FILE 9(#' STATEMENT; ALSO, THE #
C OF RECORDS AND BLOCKSIZE OF THE TRAINING STATISTICS FILE MUST BE
C SET IN THE 'DEFINE FILE 10(#,##'.
C
C-----> ESSENTIALLY THE ONLY PARAMETER TO RESET INSIDE THE SOURCE CODE
C       FOR RUNNING WITH DIFFERENT FILES ARE THE # OF RECORDS IN THE
C       DEFINE FILE STATMENT.
C
C LOGICAL UNITS USED:
C  6 - LINE PRINTER OR TERMINAL
C  9 - IMAGE FILE
C 10 - TRAINING STATISTICS FILE
C*************************************************************************
      DIMENSION MINC(6), MAXC(6), RMEANC(6), VARC(6), STDC(6),
     &          COVC(6,6), CORRC(6,6)
```

```
      DIMENSION RMINPP(25,3),RMAXPP(25,3), IREQUS(25)
      DIMENSION FILTIT(20)
      INTEGER*2 ISTATU(25), ICLUSE(25)
      DATA NBMAX/6/, NCLMAX/25/, ICLUSE/25*0/
      DATA IREQUS/25*0/, IERR/0/
C
C.....READ HEADER FROM FILES
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4323,100,L,IREC)
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C----->   # RECS = (NCLMAX * 2) + 2
C-----> 2ND ARGUMENT IS CALCULATED BY:
C       BLOCKSIZE =*************
      DEFINE FILE 10(52,184,L,IRECC)
C
C.....READ HEADER RECORD FROM FILE
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
      IF(NBAND.GT.NBMAX) GOTO 9000
C
C  ....READ HEADER FROM STATS FILE
      CALL RWSTHE(ISTATU,NCLMAX,0)
C
C.....PRINT CREDITS
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C.....READ IN PARAMETERS AND CLASS STATS
      CALL READIN(ISTATU,ICLUSE,RMINPP,RMAXPP,NCLMAX,NBAND,IREQUS,
     &            RMAX,PSIZE,MINC,MAXC,RMEANC,VARC,STDC,COVC,CORRC,
     &            XZROT,YZROT,IC1,IC2,IC3,IERR)
      IF(IERR.EQ.1) GOTO 9999
      NTICKS=(RMAX/20.0)+1
      RMAX=FLOAT(NTICKS)*20.0
C
C
      CALL PLOTS(-60.,-60.,0)
      CALL GRAPH(NCLMAX,NBAND,XZROT,YZROT,NTICKS,RMAX,RMINPP,RMAXPP,
     &           PSIZE,ICLUSE,IC1,IC2,IC3)
      CALL PLOT(0.,0.,999)
      GOTO 9999
C
C.....ERROR MESSAGES .............
 9000 WRITE(6,9001) NBAND,NBMAX
 9001 FORMAT(1X,'*** SORRY, CANNOT COMPUTE PIPEDS BECAUSE THERE'/
     &       1X,'    ARE ',I2,' BANDS IN THE IMAGE FILE AND YOU '/
     &       1X,'    HAVE THE PROGRAM LIMITS AT ',I2,'.  INCREASE'/
     &       1X,'    THE STATS ARRAYS AND NBMAX IN MAIN.')
 9999 STOP
      END
C*********************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*********************************************************************
```



OFEK07031

```
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*********************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          PARALLELEPIPED PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=    ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*********************************************************************
      SUBROUTINE READIN(ISTATU,ICLUSE,RMINPP,RMAXPP,NCLMAX,NBAND,
     &                  IREQUS,RMAX,PSIZE,MINC,MAXC,RMEANC,VARC,STDC,
     &                  COVC,CORRC,XZROT,YZROT,IC1,IC2,IC3,IERR)
C*********************************************************************
      DIMENSION MINC(NBAND), MAXC(NBAND), RMEANC(NBAND),
     &          VARC(NBAND), COVC(NBAND,NBAND), CORRC(NBAND,NBAND),
     &          RNAME(4)
      DIMENSION RMINPP(NCLMAX,3), RMAXPP(NCLMAX,3)
      DIMENSION RID(20), ICOORD(4,4)
      INTEGER*2 ISTATU(NCLMAX), ICLUSE(NCLMAX), IREQUS(NCLMAX)
      DATA RALL/'ALL '/, PIRAD/.01745329/, DEVIAT/1./
C
      READ(5,10) TYPE,DEV,XZROT,YZROT,IC1,IC2,IC3,PSIZE
   10 FORMAT (A4,1X,3F5.0,3I2,F5.0)
      IF(PSIZE.LE.0.) PSIZE=6.0
C
      WRITE(6,1) TYPE,DEV,XZROT,YZROT,IC1,IC2,IC3
    1 FORMAT(1X,'---> INPUT PARAMETERS:'/
     &       1X,'SELECTION TYPE= ',A4/1X,'STD. DEV=',F5.1/
     &       1X,'X ROTATION=',F5.1/1X,'Y ROTATION=',F5.1/
     &       1X,'BANDS TO USE =',3I3)
C
C.....CONVERT TO RADIANS
      XZROT=-(XZROT*PIRAD)
      YZROT=-(YZROT*PIRAD)
      IF(DEV.GT.0.) DEVIAT=DEV
      IF(TYPE.EQ.RALL) GOTO 70
C.....IF USER DOESN'T WANT ALL CLASSES, READ SUBSET
C
       READ(5,11) NCLASS,(IREQUS(J),J=1,NCLASS)
   11  FORMAT(I2,20I2)
       IF(NCLASS.LT.1 .OR. NCLASS.GT.NCLMAX) GOTO 9000
       WRITE(6,12) NCLASS,(IREQUS(J),J=1,NCLASS)
   12  FORMAT(1X,'# OF CLASSES TO PLOT=',I2/
     &        1X,'CLASSES = ',20I3)
C.....CREATE ARRAY CONTAINING PIPEDS TO PLOT
      DO 50 I=1,NCLMAX
        ICLS=IREQUS(I)
        IF(ICLS.LT.1 .OR. ICLS.GT.NCLMAX) GOTO 9100
```



OFEK07032

```
      IF(ISTATU(ICLS).EQ.0) GOTO 9200
      ICLUSE(ICLS)=1
   50 CONTINUE
      GOTO 100
C
   70 DO 80 I=1,NCLMAX
      ICLUSE(I)=ISTATU(I)
   80 CONTINUE
C
C.....FOR EACH CLASS TO USE, GET THE STATISTICS
  100 RMAX=-999
C
      WRITE(6,51)
   51 FORMAT(1X,//1X,8X,'   - X AXIS -',3X,'   - Y AXIS -',
     &       1X,'   - Z AXIS -'/
     &       1X,'CLASS',3X,3('   MIN    MAX',3X))
C
      DO 200 I=1,NCLMAX
      ICLS=I
      IF(ICLUSE(ICLS).EQ.0) GOTO 200
      CALL RWSTAT(RMEANC,VARC,MINC,MAXC,COVC,RNAME,ICLS,NBAND,
     &            NSAMP,ICOORD,ITOTPX,0)
      RMINPP(I,1)= RMEANC(IC1) - (SQRT(VARC(IC1))*DEVIAT)
      RMINPP(I,2)= RMEANC(IC2) - (SQRT(VARC(IC2))*DEVIAT)
      RMINPP(I,3)= RMEANC(IC3) - (SQRT(VARC(IC3))*DEVIAT)
      RMAXPP(I,1)= RMEANC(IC1) + (SQRT(VARC(IC1))*DEVIAT)
      RMAXPP(I,2)= RMEANC(IC2) + (SQRT(VARC(IC2))*DEVIAT)
      RMAXPP(I,3)= RMEANC(IC3) + (SQRT(VARC(IC3))*DEVIAT)
C
C    ....CHECK TO BE SURE STATS ARE IN BOUNDS; ALSO FIND THE MAXIMUM
      DO 150 J=1,3
        IF(RMINPP(I,J).LT.0) RMINPP(I,J)=0.
        IF(RMAXPP(I,J).GT.255.) RMAXPP(I,J)=255.
        IF(RMAXPP(I,J).GT.RMAX) RMAX=RMAXPP(I,J)
  150   CONTINUE
      WRITE(6,151) I,(RMINPP(I,J),RMAXPP(I,J),J=1,3)
  151 FORMAT(1X,I3,4X,3(2F7.2,2X))
  200 CONTINUE
C
      RETURN
C
C.....ERROR MESSAGES.....
 9000 WRITE(6,9001) NCLASS
 9001 FORMAT(1X,'*** SORRY  YOU REQUESTED ',I3,' CLASSES TO BE '/
     &       1X,'           PLOTTED.  THE MAXIMUM # OF CLASSES WHICH'/
     &       1X,'           MAY BE PLOTTED IS ',I2,'  INCREASE STATS'/
     &       1X,'           ARRAY SIZES.')
      IERR=1
      RETURN
 9100 WRITE(6,9101) ICLS
 9101 FORMAT(1X,'*** ERROR  YOU SPECIFIED AND ILLEGAL CLASS #.'/
     &       1X,'           THE ILLEGAL CLASS WAS ',I2)
      IERR=1
      RETURN
 9200 WRITE(6,9201) ICLS
```



OFEK07033

```
 9201 FORMAT(1X,'*** ERROR  YOU SPECIFIED A CLASS WHICH DID NOT'/
     &       1X,'           HAVE STATISTICS GENERATED FOR IT.'/
     &       1X,'           THE NON EXISTING CLASS IS ',I2)
       IERR=1
       RETURN
       END
C*************************************************************************
      SUBROUTINE GRAPH(NCLMAX,NBAND,XZROT,YZROT,NTICKS,RMAX,RMINPP,
     &                 RMAXPP,PSIZE,ICLUSE,IC1,IC2,IC3)
C*************************************************************************
C  ACTUALLY DIRECTS THE PLOTTER TO CONSTRUCT THE INDIVIDUAL PIPEDS
C  AND DRAW AND LABEL THE AXES.
C*************************************************************************
      DIMENSION RMINPP(NCLMAX,3), RMAXPP(NCLMAX,3)
      DIMENSION CM(3,3)
      DIMENSION X(8),Y(8),Z(8),XPP(8),YPP(8)
      DIMENSION XT1(56),YT1(56),ZT1(56), XT2(56),YT2(56),ZT2(56)
      DIMENSION XTICK1(56), YTICK1(56), XTICK2(56), YTICK2(56)
C.....A,B,C HOLD ORIGINAL COORD. OF LEFT HANDED COORD. SYSTEM
C.....A2,B2,C2 HOLD TRANSFORMED COORDINATES.
      DIMENSION A(4),B(4),C(4),A2(4),B2(4),C2(4)
      INTEGER*2 ICLUSE(NCLMAX)
      DATA A/4*0./, B/4*0./, C/4*0./
C
      CALL PLOT(3.,2.,-3)
      FACT=PSIZE/RMAX
      CALL FACTOR(FACT)
      CSIZE=.1/FACT
      CSIZE2=.15/FACT
C
C.....W IS 2 PI RADIANS OR 360 DEGREES.
C.....P IS A COUNTER USED FOR LABELING PARALLELEPIPEDS 1,2,ETC.
      RLDASH=.09/FACT
      RLSPAC=.07/FACT
      W=6.2832
      P=1.0
C
C......CM(3,3) HOLDS THE CONCATENATION OF MATRIX TRANSFORMATIONS
C......A AND B (REFER TO TEXT). ELEMENTS OF MATRIX CM ARE THEN
C......APPLIED TO TRANSFORM THE ORIGINAL COODINATES X,Y,Z OF
C......EACH PARALLELEPIPED.
      CM(1,1)=COS(YZROT)
      CM(1,2)=SIN(W-YZROT)*SIN(XZROT)
      CM(1,3)=SIN(W-YZROT)*COS(XZROT)
      CM(2,1)=0.0
      CM(2,2)=COS(XZROT)
      CM(2,3)=SIN(W-XZROT)
      CM(3,1)=SIN(YZROT)
      CM(3,2)=COS(YZROT)*SIN(XZROT)
      CM(3,3)=COS(YZROT)*COS(XZROT)
C
C......USING UPPER AND LOWER THRESHOLD COORD. FOR EACH CHANNEL,
C......COMPUTE ORIGINAL X,Y,Z COORD. FOR THE 8 CORNERS
C......THIS IS DONE FOR EACH CLASS.
      DO 900 J=1,NCLMAX
```



OFEK07034

```
      IF(ICLUSE(J).EQ.0) GOTO 900
      X(1)=RMINPP(J,1)
      Y(1)=RMINPP(J,2)
      Z(1)=RMINPP(J,3)
      X(2)=RMINPP(J,1)
      Y(2)=RMAXPP(J,2)
      Z(2)=RMINPP(J,3)
      X(3)=RMAXPP(J,1)
      Y(3)=RMAXPP(J,2)
      Z(3)=RMINPP(J,3)
      X(4)=RMAXPP(J,1)
      Y(4)=RMINPP(J,2)
      Z(4)=RMINPP(J,3)
      X(5)=RMINPP(J,1)
      Y(5)=RMINPP(J,2)
      Z(5)=RMAXPP(J,3)
      X(6)=RMINPP(J,1)
      Y(6)=RMAXPP(J,2)
      Z(6)=RMAXPP(J,3)
      X(7)=RMAXPP(J,1)
      Y(7)=RMAXPP(J,2)
      Z(7)=RMAXPP(J,3)
      X(8)=RMAXPP(J,1)
      Y(8)=RMINPP(J,2)
      Z(8)=RMAXPP(J,3)
C
C.....TAKE ORIGINAL X,Y,Z COORDINATES, APPLY CM TRANSFORMATION,
C.....STORE NEW VALUES IN XPP,YPP
      DO 120 I=1,8
        XPP(I)=X(I)*CM(1,1)+Y(I)*CM(1,2)+Z(I)*CM(1,3)
        YPP(I)=X(I)*CM(2,1)+Y(I)*CM(2,2)+Z(I)*CM(2,3)
 120    CONTINUE
C
C.....FIND CENTER OF PARALLELEPIPED AND LABEL IT .. 1,2,ETC.
      RJ=FLOAT(J)
      X2=XPP(1) + (ABS(XPP(1)-XPP(4))/2.0)
      Y2=YPP(1) + (ABS(YPP(1)-YPP(2))/2.0)
      CALL NUMBER(X2,Y2,CSIZE,RJ,0.0,-1)
C
C-----------------------------------------------------------------------
C.....DRAW FRONT OF PARALLELEPIPED USING SOLID LINE.
      CALL PLOT(XPP(1),YPP(1),3)
      DO 150 I=2,4
        CALL PLOT(XPP(I),YPP(I),2)
 150    CONTINUE
      CALL PLOT(XPP(1),YPP(1),2)
C
C.....DRAW BACK OF PARALLELEPIPED USING DASHED LINE.
      DO 160 I=1,3
        IND=4+I
        IND2=IND+1
        CALL DASHM(XPP(IND),YPP(IND),XPP(IND2),YPP(IND2),
     &             RLDASH,RLSPAC)
 160    CONTINUE
      CALL DASHM(XPP(8),YPP(8),XPP(5),YPP(5),RLDASH,RLSPAC)
```



OFEK07035

```
C
C......CONNECT REMAINDER OF PARALLELEPIPED CORNERS.
        DO 168 I=1,4
          INDX=4+I
          CALL DASHM(XPP(I),YPP(I),XPP(INDX),YPP(INDX),RLDASH,RLSPAC)
  168     CONTINUE
  900   CONTINUE
C
C ---------------------------------------------------------------
C......SET LENGTH OF ALL AXES TO MAXIMUM VALUE OF PIPEDS
      A(2)=RMAX
      B(3)=RMAX
      C(4)=RMAX
C......SET TICK MARK LOCATIONS
      RVAL=0.
      DO 20 I=1,NTICKS
        RVAL=RVAL+20.
        XT1(I)=RVAL
        YT1(I+NTICKS)=RVAL
        ZT1(I+2*NTICKS)=RVAL
        XT2(I)=RVAL
        XT2(I+NTICKS)=2.
        YT2(I+NTICKS)=RVAL
        YT2(I)=2.
        YT2(I+2*NTICKS)=2.
   20   ZT2(I+2*NTICKS)=RVAL
C
C......COMPUTE TRANSFORM COORDINATES OF TICK MARKS.
      NUM=NTICKS*3
      DO 130 I=1,NUM
        XTICK1(I)=XT1(I)*CM(1,1)+YT1(I)*CM(1,2)+ZT1(I)*CM(1,3)
        YTICK1(I)=XT1(I)*CM(2,1)+YT1(I)*CM(2,2)+ZT1(I)*CM(2,3)
        XTICK2(I)=XT2(I)*CM(1,1)+YT2(I)*CM(1,2)+ZT2(I)*CM(1,3)
        YTICK2(I)=XT2(I)*CM(2,1)+YT2(I)*CM(2,2)+ZT2(I)*CM(2,3)
  130    CONTINUE
C
C......COMPUTE TRANSFORM COORDINATES OF AXES.
      DO 140 I=1,4
        A2(I)=A(I)*CM(1,1)+B(I)*CM(1,2)+C(I)*CM(1,3)
        B2(I)=A(I)*CM(2,1)+B(I)*CM(2,2)+C(I)*CM(2,3)
        C2(I)=A(I)*CM(3,1)+B(I)*CM(3,2)+C(I)*CM(3,3)
 140    CONTINUE
C
C......DRAW AXES
      CALL DASHM(A2(1),B2(1),A2(4),B2(4),RLDASH,RLSPAC)
      CALL PLOT(A2(1),B2(1),3)
      CALL PLOT(A2(2),B2(2),2)
      CALL PLOT(A2(1),B2(1),3)
      CALL PLOT(A2(3),B2(3),2)
C
C......DRAW TICK MARKS
      NUM=NTICKS*3
      DO 170 I=1,NUM
        CALL PLOT(XTICK1(I),YTICK1(I),3)
        CALL PLOT(XTICK2(I),YTICK2(I),2)
```



OFEK07036

```
 170    CONTINUE
      Q=0.0
C
C.....ANNOTATE AXES TICK MARKS WITH REFLECTANCE VALUES. .2,4,ETC.
      DO 180 I=1,NTICKS
        Q=Q+20.0
        IF(Q.LT.10.) XOFFX=CSIZE/2.
        IF(Q.LT.10.) XOFFY=CSIZE + (CSIZE*.5)
        IF(Q.GT.10. .AND. Q.LT.100.) XOFFX=CSIZE
        IF(Q.GT.10. .AND. Q.LT.100.) XOFFY=2.*CSIZE + (CSIZE*.5)
        IF(Q.GE.100.) XOFFX=CSIZE*1.5
        IF(Q.GE.100.) XOFFY=3.*CSIZE + (CSIZE*.5)
        CALL NUMBER(XTICK1(I)-XOFFX,YTICK1(I)-(CSIZE*2.),CSIZE,
     &              Q,0.0,-1)
        CALL NUMBER(XTICK1(I+NTICKS)-XOFFY,YTICK1(I+NTICKS)-(CSIZE*.5),
     &              CSIZE,Q,0.0,-1)
 180    CONTINUE
C
C......COMPUTE X-AXIS LABEL LOCATION
      AA=((A2(2)-A2(1))/2.) - (CSIZE2*3.)
      BB=((B2(2)-B2(1))/2.) - (CSIZE2*4.)
      CALL SYMBOL(AA,BB,CSIZE2,'BAND ',0.0,5)
      CALL NUMBER(AA+(CSIZE2*5.),BB,CSIZE2,FLOAT(IC1),0.,-1)
C......COMPUTE Y-AXIS LABEL LOCATION
      AA=((A2(3)-A2(1))/2.) -(CSIZE2*6.)
      BB=((B2(3)-B2(1))/2.)
      CALL SYMBOL(AA,BB,CSIZE2,'BAND ',90.0,5)
      CALL NUMBER(AA,BB+(CSIZE2*5.),CSIZE2,FLOAT(IC2),90.,-1)
C
C......PLOT Z-AXIS LABEL
      XOFF=CSIZE2*8.
      AA=A2(1)
      BB=B2(1)-CSIZE2*6.
      CALL SYMBOL(AA,BB,CSIZE2,'ZAXIS: BAND ',0.,12)
      CALL NUMBER(AA+(CSIZE2*12.),BB,CSIZE2,FLOAT(IC3),0.,-1)
C
      RETURN
      END
C*************************************************************************
      SUBROUTINE DASHM(X1,Y1,X2,Y2,RLDASH,RLSPAC)
C
C*************************************************************************
C  DRAWS A DASHED LINE CONNECTING TWO POINTS. THE ALGORITHM INVOLVES
C  THE COMPUTATION OF EACH DASH AND SPACE LOCATION VIA SIMILAR
C  TRIANGLES.
C
C  MAJOR VARIABLES:
C  X1,Y1  - BEGINNING POINT OF LINE
C  X2,Y2  - ENDING POINT OF LINE
C  RLDASH - LENGTH OF EACH DASH(IN INCHES)
C  RLSPAC - LENGTH OF EACH SPACE(IN INCHES)
C*************************************************************************
      DIST=SQRT((X1-X2)**2 + (Y1-Y2)**2)
      IF(DIST.LE.0.) RETURN
      AC=(Y2-Y1)/DIST
```



OFEK07037

```
      BC=(X2-X1)/DIST
C
      X3=X1
      Y3=Y1
      TLEN=0.0
      CALL PLOT(X3,Y3,3)
    1 TLEN=TLEN+RLDASH
      IF(TLEN.GE.DIST) GO TO 200
C.....COMPUTE 'DASH' LOCATION
      Y4=(TLEN*AC)+Y1
      X4=(TLEN*BC)+X1
      CALL PLOT(X4,Y4,2)
      TLEN=TLEN+RLSPAC
      IF(TLEN.GE.DIST) GO TO 300
C.....COMPUTE 'SPACE' LOCATON
      Y4=(TLEN*AC)+Y1
      X4=(TLEN*BC)+X1
      CALL PLOT(X4,Y4,3)
      GO TO 1
C
  200 CALL PLOT(X2,Y2,2)
  300 RETURN
      END
//LKED.SYSLIB DD DSN=SBS.GOULDLIB,DISP=SHR,VOL=SER=,UNIT=
//    DD DSN=FORT.FORTLIB,DISP=SHR
//    DD DSN=ACAD.SUBLIB,DISP=SHR
//GO.FT05F001 DD *
ALL    1.0 30.0 60.0 4 5 6   6.
//GO.FT09F001 DD DSN=SBS.MEH1B.BOOK6C.R240C256,DISP=SHR,LABEL=(,,,IN)
//GO.FT10F001 DD DSN=SBS.MEH.BOOK6C.STATS,DISP=SHR,LABEL=(,,,IN)
//GO.SYSVECTR DD DSN=&&VECTORS,UNIT=SYSDA,DISP=(MOD,PASS),
//      SPACE=(1320,(500,200)),DCB=BLKSIZE=1320
//GO.SYSPOUT  DD SYSOUT=*
//BITSET EXEC PGM=BTSTN50
//STEPLIB  DD DSN=SBS.GOULDLIB,DISP=SHR
//SYSVECTR DD DSN=&&VECTORS,DISP=(OLD,DELETE)
//SYSUT1   DD DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(18200,200)
//SYSPLOT  DD SYSOUT=(B,,PLOT),DEST=N1R8
//SYSPOUT  DD SYSOUT=*
//
```



OFEK07038

# DIVERGENCE

```
//BOOKDI JOB (D6600217),'VF-BOOKDI',TIME=(,9),MSGLEVEL=(1,0),
// USER=SBS01,PASSWORD=??????
/*ROUTE PRINT MVS.B8
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*********************************************************************
C  DIVERGENCE PROGRAM
C    COMPUTES DIVERGENCE STATISTICS FOR CLASS AND CHANNEL COMBINATIONS.
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NBMAX - MAXIMUM # OF BANDS USED BY THE PROGRAM; SHOULD BE >= THE
C          NUMBER OF BANDS IN THE IMAGE FILE AND EQUAL TO THE ARRAY
C          SIZES FOR THE STATISTICS IN MAIN.
C
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS IN THE IMAGE
C FILE MUST BE SET IN THE 'DEFINE FILE 9(#' STATEMENT; ALSO, THE #
C OF RECORDS AND BLOCKSIZE OF THE TRAINING STATISTICS FILE MUST BE
C SET IN THE 'DEFINE FILE 10(#,##'.
C
C LOGICAL UNITS USED:
C  5 - USER INPUT PARAMETERS
C  6 - LINE PRINTER OR TERMINAL
C  9 - IMAGE FILE
C 10 - TRAINING STATISTICS FILE
C*********************************************************************
      DIMENSION RNAMEC(25,4), RMNALL(25,6), RCOVC(25,6,6)
      DIMENSION MINC(6), MAXC(6), RMEANC(6), VARC(6), STDC(6),
     &          COVC(6,6), CORRC(6,6)
      DIMENSION IBACOM(25,5), ICLCOM(25,2), LBAND(6)
      DIMENSION U1(6,6),U2(6,6),UH1(6,6),UH2(6,6),UH3(6,6),UH4(6,6),
     &          UH(6,6)
      DIMENSION RM1(6),RM2(6),RMH1(6),RMH2(6)
      DIMENSION ICLASS(25),IBAND(6), FILTIT(20)
      DIMENSION RDIV(25), RIDIV(25)
      INTEGER*2 ISTATU(25)
      DATA NBLK/100/, NBMAX/6/, NCLMAX/25/
      IERR=0
C
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4323,100,L,IREC)
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C------>   # RECS = (NCLMAX * 2) + 2
C-----> LOKK UP 2ND ARGUMENT IN USER'S MANUAL.
      DEFINE FILE 10(52,184,L,IRECC)
C
C......READ HEADER RECORD FROM FILE
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
      IF(NBAND.GT.NBMAX) GOTO 9000
```





OFEK07039

```
C
C.......READ STATISTICS FILE HEADER
        CALL RWSTHE(ISTATU,NCLMAX,0)
C
C......PRINT PROGRAM BANNER
       CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C......READ TYPE OF CLASSIFICATION TO PERFORM
       CALL READIN(ICLASS,IBAND,NCLASS,NCLMAX,NBMAX,NBAND,NBUSE,
      &                    NBPC,IERR)
       IF(IERR.EQ.1) GOTO 9999
C
C......READ TRAINING CLASS STATISTICS
       CALL ALLSTA(MINC,MAXC,RMEANC,VARC,STDC,COVC,CORRC,
      &                    RMNALL,RCOVC,RNAMEC,ISTATU,NCLMAX,NBAND,NBMAX)
       WRITE(6,2)
     2 FORMAT(1X,//'--> STATISTICS READ')
C......PERFORM CLASSIFICATON FOR SUBIMAGE
       CALL DIVSET(RMNALL,RCOVC,RNAMEC,IBAND,ICLASS,NCLASS,NBAND,
      &                    U1,U2,UH1,UH2,UH3,UH4,UH,RM1,RM2,RMH1,RMH2,
      &                    RDIV,RTDIV,IBACOM,ICLCOM,LBAND,NCLMAX,
      &                    NBUSE,NBPC,CTYPE)
       GOTO 9999
C
C.......ERROR MESSAGES ..............
 9000 WRITE(6,9001) NBANDS,NBMAX
 9001 FORMAT(1X,'*** SORRY, CANNOT COMPUTE DIVERGENCES BECAUSE '/
     &       1X,'    THERE ARE ',I2,' BANDS IN THE IMAGE FILE AND'/
     &       1X,'    YOU HAVE THE PROGRAM LIMITS AT ',I2,'.'/
     &       1X,'    INCREASE THE STATS ARRAYS AND NBMAX IN MAIN.')
 9999 STOP
      END
C*************************************************************************
      SUBROUTINE READIN(ICLASS,IBAND,NCLASS,NCLMAX,NBMAX,NBAND,NBUSE,
     &                    NBPC,IERR)
C*************************************************************************
C INPUTS THE PARAMETERS TO CONTROL THE CLASSIFICATION
C*************************************************************************
      DIMENSION ICLASS(NCLMAX),IBAND(NBMAX)
C
C .....INPUT # OF BANDS
      READ(5,12) NBUSE,NBPC,(IBAND(J),J=1,NBUSE)
   12 FORMAT(2I2,6I2)
      IF(NBUSE.GT.NBAND) GOTO 9000
      IF(NBPC.GT.NBUSE) GOTO 9010
C .....INPUT CLASSES TO USE
      NCLASS=1
   20 READ(5,21,END=100) ICLASS(NCLASS)
   21 FORMAT(I2)
      NCLASS=NCLASS+1
      GOTO 20
C
  100 NCLASS=NCLASS-1
      WRITE(6,101) NBUSE,NBPC,(IBAND(J),J=1,NBUSE)
  101 FORMAT(1X,'# OF BANDS=',I2,'  # BANDS PER COMBINATION=',I2/
```



OFEK07040

```
     &       1X,'   BANDS=  ',8I2)
        WRITE(6,102) NCLASS,(ICLASS(J),J=1,NCLASS)
  102 FORMAT(1X,'# OF CLASSES=',I3/
     &       1X,'   CLASSES=',20I3/1X,20I3/1X,10I3)
        RETURN
C
C.....ERROR MESSAGES........
 9000 WRITE(6,9001) NBUSE,NBAND
 9001 FORMAT(1X,'*** ERROR **  ',I3,' BANDS REQUESTED; ONLY ',I3,
     &       'BANDS AVAILABLE'/)
        IERR=1
        RETURN
 9010 WRITE(6,9011) NBPC,NBUSE
 9011 FORMAT(1X,'*** ERROR **  ',I3,' BANDS PER COMBINATION.',I3,
     &       'BANDS AVAILABLE'/)
        IERR=1
        RETURN
        END
C*************************************************************************
        SUBROUTINE ALLSTA(MINC,MAXC,RMEANC,VARC,STDC,COVC,CORRC,
     &                    RMNALL,RCOVC,RNAMEC,ISTATU,NCLMAX,NBAND,NBMAX)
C*************************************************************************
C ... NOTE: COVC AND CORRC ARRAYS ARE STORED AS COVC(ROW,COLUMN)
        DIMENSION MINC(NBAND), MAXC(NBAND), RMEANC(NBAND), VARC(NBAND),
     &            STDC(NBAND), COVC(NBAND,NBAND), CORRC(NBAND,NBAND)
        DIMENSION RNAMEC(NCLMAX,4), RMNALL(NCLMAX,NBMAX),
     &            RCOVC(NCLMAX,NBMAX,NBMAX)
        DIMENSION RNAME(4), ICOORD(4,4)
        INTEGER*2 ISTATU(NCLMAX)
C
C.....RETRIEVE STATS FOR EACH CLASS
C.....COMPUTE SIMPLE STATISTICS FOR EACH CHANNEL
        DO 100 ICLS=1,NCLMAX
          IF(ISTATU(ICLS).EQ.0) GOTO 100
C........READ RECORD TO STATS FILE
          CALL RWSTAT(RMEANC,VARC,MINC,MAXC,COVC,RNAME,ICLS,NBAND,NSITE,
     &                ICOORD,ITOTPX,IRW)
C
C .......SAVE STATS IN ARRAYS
          DO 20 IBA=1,NBAND
            RMNALL(ICLS,IBA)= RMEANC(IBA)
            DO 10 JBA=1,NBAND
   10         RCOVC(ICLS,IBA,JBA)= COVC(IBA,JBA)
   20       CONTINUE
C ........SAVE CLASS NAMES
          DO 50 ICR=1,4
   50       RNAMEC(ICLS,ICR)= RNAME(ICR)
  100     CONTINUE
C
        RETURN
        END
C*************************************************************************
        SUBROUTINE DIVSET(RMNALL,RCOVC,RNAMEC,IBAND,ICLASS,NCLASS,NBAND,
     &                    U1,U2,UH1,UH2,UH3,UH4,UH,RM1,RM2,RMH1,RMH2,
     &                    RDIV,RTDIV,IBACOM,ICLCOM,LBAND,NCLMAX,
```



OFEK07041

```
     &                    NBUSE,NBPC,CTYPE)
C*************************************************************************
C  MINIMUM DISTANCE CLASSIFIER
C*************************************************************************
      DIMENSION RNAMEC(NCLMAX,NBAND), RMNALL(NCLMAX,NBAND),
     &          RCOVC(NCLMAX,NBAND,NBAND)
      DIMENSION ICLASS(NCLMAX), IBAND(NBAND), IBACOM(NCLMAX,5),
     &          ICLCOM(NCLMAX,2), LBAND(NBPC)
      DIMENSION U1(NBPC,NBPC),U2(NBPC,NBPC),UH1(NBPC,NBPC),
     &          UH2(NBPC,NBPC),UH3(NBPC,NBPC),UH4(NBPC,NBPC),
     &          UH(NBPC,NBPC)
      DIMENSION RM1(NBPC),RM2(NBPC),RMH1(NBPC),RMH2(NBPC)
      DIMENSION RDIV(NCLMAX), RTDIV(NCLMAX)
C
C......COMPUTE POSSIBLE COMBINATIONS OF BANDS
      CALL COMBO(NBUSE,NBPC,NCLMAX,NBCOM,IBAND,IBACOM)
C......COMPUTE POSSIBLE COMBINATIONS OF CLASSES
      CALL COMBO(NCLASS,2,NCLMAX,NCCOM,ICLASS,ICLCOM)
C
C .....NOW, LET'S COMPUTE THE DIVERGENCE STATS.
      CALL DIVERG(RMNALL,RCOVC,RNAMEC,IBAND,LBAND,ICLASS,NCLASS,
     &            NCLMAX,NBAND,NBUSE,NBPC,CTYPE,U1,U2,UH1,UH2,
     &            UH3,UH4,UH,RM1,RM2,RMH1,RMH2,RDIV,RTDIV,IBACOM,
     &            ICLCOM,NBCOM,NCCOM)
C
      RETURN
      END
C*************************************************************************
      SUBROUTINE DIVERG(RMNALL,RCOVC,RNAMEC,IBAND,LBAND,ICLASS,NCLASS,
     &                  NCLMAX,NBAND,NBUSE,NBPC,CTYPE,U1,U2,UH1,UH2,
     &                  UH3,UH4,UH,RM1,RM2,RMH1,RMH2,RDIV,RTDIV,IBACOM,
     &                  ICLCOM,NBCOM,NCCOM)
C*************************************************************************
C  COMPUTES THE  DIVERGENCE STATISTICS
C*************************************************************************
      DIMENSION RNAMEC(NCLMAX,NBAND), RMNALL(NCLMAX,NBAND),
     &          RCOVC(NCLMAX,NBAND,NBAND)
      DIMENSION ICLASS(NCLASS),IBAND(NBUSE), LBAND(NBPC),
     &          IBACOM(NCLMAX,5), ICLCOM(NCLMAX,2)
      DIMENSION U1(NBPC,NBPC),U2(NBPC,NBPC),UH1(NBPC,NBPC),
     &          UH2(NBPC,NBPC),UH3(NBPC,NBPC),UH4(NBPC,NBPC),
     &          UH(NBPC,NBPC)
      DIMENSION RM1(NBPC),RM2(NBPC),RMH1(NBPC),RMH2(NBPC)
      DIMENSION RDIV(NCLMAX), RTDIV(NCLMAX)
C
      WRITE(6,51)
   51 FORMAT('1',30('-'),' DIVERGENCE TABLE ',30('-')//
     &       1X,20X,'( DIVERGENCE / TRANSFORMED DIVERGENCE )'//
     &       1X,'BAND COMBINATIONS   AVG.DIV.     CLASS COMBINATIONS'/)
        WRITE(6,61) (ICLCOM(J,1),J=1,NCCOM)
        WRITE(6,61) (ICLCOM(J,2),J=1,NCCOM)
   61   FORMAT(1X,27X,1516)
C......FOR EACH BAND COMBINATION,
      DO 500 IB=1,NBCOM
        TOTDIV=0.
```



OFEK07042

```
      DO 120 IB2=1,NBPC
  120   LBAND(IB2)=IBACOM(IB,IB2)
        WRITE(6,121) (LBAND(J),J=1,NBPC)
  121   FORMAT(1X,/6I3)
C     ....FOR EACH CLASS COMBINATION
      DO 400 IC=1,NCCOM
        ICLS1=ICLCOM(IC,1)
        ICLS2=ICLCOM(IC,2)
        CALL MGET(RCOVC,RMNALL,LBAND,U1,RM1,NBAND,NCLMAX,
     &            NBPC,ICLS1,2)
        CALL MGET(RCOVC,RMNALL,LBAND,U2,RM1,NBAND,NCLMAX,
     &            NBPC,ICLS2,2)
        CALL MSUBT(U1,U2,UH1,NBPC,NBPC)
        CALL MGET(RCOVC,RMNALL,LBAND,UH,RM1,NBAND,NCLMAX,
     &            NBPC,ICLS1,2)
        CALL MINV(UH,UH2,NBPC,DET)
        CALL MGET(RCOVC,RMNALL,LBAND,UH,RM1,NBAND,NCLMAX,
     &            NBPC,ICLS2,2)
        CALL MINV(UH,UH3,NBPC,DET)
        CALL MSUBT(UH3,UH2,UH4,NBPC,NBPC)
        CALL MMULT(UH1,UH4,U1,NBPC,NBPC,NBPC,NBPC)
        CALL MTRACE(U1,NBPC,TRACE1)
        CALL MGET(RCOVC,RMNALL,LBAND,U1,RM1,NBAND,NCLMAX,
     &            NBPC,ICLS1,1)
        CALL MGET(RCOVC,RMNALL,LBAND,U1,RM2,NBAND,NCLMAX,
     &            NBPC,ICLS2,1)
        CALL MADD(UH2,UH3,UH4,NBPC,NBPC)
        CALL MSUBT(RM1,RM2,RMH1,NBPC,1)
        CALL MMULT(UH4,RMH1,U1,NBPC,NBPC,NBPC,1)
        CALL MMULT(U1,RMH1,UH1,NBPC,1,1,NBPC)
        CALL MTRACE(UH1,NBPC,TRACE2)
        DIV=(.5*TRACE1) + (.5*TRACE2)
        DIV2=DIV
        IF(DIV.GT.900.) DIV2=900.
        TRDIV=2000. * (1.-EXP(-DIV2/8.))
        RDIV(IC)=DIV
        RTDIV(IC)=TRDIV
        TOTDIV=TOTDIV+TRDIV
  400   CONTINUE
C      ....COMPUTE AVERAGE DIVERGENCE
       AVGTDI= TOTDIV/FLOAT(NCCOM)
C     ...PRINT OUT RESULTS
      WRITE(6,401) (RDIV(J),J=1,NCCOM)
  401 FORMAT('+',29X,15F6.0)
      WRITE(6,402) AVGTDI,(RTDIV(J),J=1,NCCOM)
  402 FORMAT(1X,22X,F6.0,1X,15F6.0)
C
  500 CONTINUE
C
      WRITE(6,501)
  501 FORMAT(1X,///' CLASS #      NAME')
      DO 600 IC=1,NCLASS
  600   WRITE(6,601) ICLASS(IC), (RNAMEC(IC,J),J=1,4)
  601   FORMAT(1X,I4,4X,4A4)
      RETURN
```



OFEK07043

```
      END
C*************************************************************************
      SUBROUTINE MSUBT(RMAT1,RMAT2,RMAT3,NR,NC)
C*************************************************************************
C SUBTRACTS MATRIX RMAT2 FROM RMAT1 TO PRODUCE RMAT3.
C  ALL MATRICES ARE NR ROWS BY NC COLUMNS.
C*************************************************************************
      DIMENSION RMAT1(NR,NC), RMAT2(NR,NC), RMAT3(NR,NC)
      DO 20 I=1,NR
        DO 10 J=1,NC
          RMAT3(I,J)=RMAT1(I,J)-RMAT2(I,J)
   10     CONTINUE
   20   CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE MADD(RMAT1,RMAT2,RMAT3,NR,NC)
C*************************************************************************
C SUBTRACTS MATRIX RMAT2 FROM RMAT1 TO PRODUCE RMAT3.
C  ALL MATRICES ARE NR ROWS BY NC COLUMNS.
C*************************************************************************
      DIMENSION RMAT1(NR,NC), RMAT2(NR,NC), RMAT3(NR,NC)
      DO 20 I=1,NR
        DO 10 J=1,NC
          RMAT3(I,J)=RMAT1(I,J)+RMAT2(I,J)
   10     CONTINUE
   20   CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE MTRACE(RMAT1,NRC,TRACE)
C*************************************************************************
C COMPUTES THE TRACE OF A MATRIX RMAT1.
C  A IS NRC ROWS BY NRC COLUMNS.
C*************************************************************************
      DIMENSION RMAT1(NRC,NRC)
      TRACE=0.0
      DO 10 I=1,NRC
        TRACE=TRACE+RMAT1(I,I)
   10   CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE MMULT(RMAT1,RMAT2,RMAT3,NAR,NAC,NBR,NBC)
C*************************************************************************
C  MULTIPLIES MATRIX RMAT1 BY MATRIX RMAT2;
C    THE RESULT IS PLACED IN RMAT3.
C  RMAT1 IS NAR BY NAC COLUMNS
C  RMAT2 IS NBR ROWS BY NBC COLUMNS
C  RMAT3 IS NAR ROWS BY NBC COLUMNS
C*************************************************************************
      DIMENSION RMAT1(NAR,NAC), RMAT2(NBR,NBC), RMAT3(NAR,NBC)
      DO 30 I=1,NAR
        DO 20 J=1,NBC
          RMAT3(I,J)=0.0
```



OFEK07044

```
          DO 10 K=1,NAC
            RMAT3(I,J)=RMAT3(I,J) + (RMAT1(I,K) * RMAT2(K,J))
   10       CONTINUE
   20     CONTINUE
   30   CONTINUE
        RETURN
        END
C*************************************************************************
      SUBROUTINE MINV(RMAT1,RMAT2,NRC,DET)
C*************************************************************************
C FINDS THE INVERSE OF MATRIX RMAT1.  RMAT2 IS THE INVERSE OF RMAT1.
C  MATRICES RMAT1 & RMAT2 ARE NRC ROWS BY NRC COLUMNS
C  'DET' IS THE DETERMINANT OF RMAT1.
C*************************************************************************
      DIMENSION RMAT1(NRC,NRC), RMAT2(NRC,NRC)
C
      DO 20 I=1,NRC
        DO 10 J=1,NRC
          RMAT2(I,J)=0.0
   10     CONTINUE
        RMAT2(I,I)=1.0
   20   CONTINUE
        DET=1.0
C
C......CALCULATE THE INVERSE OF RMAT1
      DO 100 I=1,NRC
C     ...DIVIDE THE ITH ROW OF RMAT1 AND RMAT2 BY RMAT1(I,I)
        DIV=RMAT1(I,I)
        DET=DET*DIV
        DO 30 J=1,NRC
          RMAT1(I,J)=RMAT1(I,J)/DIV
          RMAT2(I,J)=RMAT2(I,J)/DIV
   30     CONTINUE
C     ....REDUCE THE ITH COLUMN OF RMAT1 TO ZERO
        DO 50 J=1,NRC
          IF(I-J) 35,50,35
   35     RATIO=RMAT1(J,I)
          DO 40 K=1,NRC
            RMAT1(J,K)=RMAT1(J,K) - RATIO*RMAT1(I,K)
            RMAT2(J,K)=RMAT2(J,K) - RATIO*RMAT2(I,K)
   40       CONTINUE
   50     CONTINUE
  100   CONTINUE
        RETURN
        END
C*************************************************************************
      SUBROUTINE MGET(RCOVC,RMNALL,LBAND,U,RM,NBAND,NCLMAX,
     &                NBPC,ICLS,ITYP)
C*************************************************************************
C RETRIEVES A MEAN, COVARIANCE, OR INVERSE OF COVARIANCE
C   MATRIX.  ONLY THOSE BANDS USED FOR THE DIVERGENCE ARE
C   RETURNED.
C
C   ICLS - CLASS IDENTIFIER
C . ITYP - '1' FOR MEAN VECTOR
```



OFEK07045

```
C          '2' FOR COVARIANCE MATRIX
C
C RETURNED:
C   U  - COVARIANCE MATRIX
C    OR
C   RM - MEAN VECTOR
C*************************************************************************
      DIMENSION RCOVC(NCLMAX,NBAND,NBAND),
     &          RMNALL(NCLMAX,NBAND), LBAND(NBPC)
      DIMENSION U(NBPC,NBPC), RM(NBPC)
C
      GOTO (100,200), ITYP
C......RETRIEVE MEAN VECTOR
  100 DO 110 I=1,NBPC
  110   RM(I)=RMNALL(ICLS,LBAND(I))
        RETURN
C......RETRIEVE COVARIANCE MATRIX
  200 DO 210 I=1,NBPC
        DO 210 J=1,NBPC
  210     U(I,J)=RCOVC(ICLS,LBAND(I),LBAND(J))
        RETURN
      END
C*************************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*************************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*************************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          DIVERGENCE PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=   ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*************************************************************************
      SUBROUTINE COMBO(NC,NPC,NCLMAX,NCOMB,ID,ICOMBO)
C*************************************************************************
C  COMPUTES THE CORRECT CHANNEL COMBINATIONS
C   LIMITED TO 5 CHANNELS PER COMBINAITION
C
C  NC  - NUMBER OF VALUES
C  NPC - NUMBER OF VALUES PER COMBINATION
C*************************************************************************
      DIMENSION ID(NC), ICOMBO(NCLMAX,NPC)
      NCOMB=0
C
      IE1=NC-NPC+1
      DO 50 I=1,IE1
        IF(NPC.EQ.1) GOTO 6
```



OFEK07046

```
      IB2=I+1
      IE2=NC-NPC+2
      DO 40 II=IB2,IE2
        IF(NPC.EQ.2) GOTO 6
        IB3=II+1
        IE3=NC-NPC+3
        DO 30 III=IB3,IE3
          IF(NPC.EQ.3) GOTO 6
          IB4=III+1
          IE4=NC-NPC+4
          DO 20 IIII=IB4,IE4
            IF(NPC.EQ.4) GOTO 6
            IB5=IIII+1
            IE5=NC-NPC+5
            DO 10 IIIII=IB5,IE5
  6           NCOMB=NCOMB+1
              GOTO (1,2,3,4,5), NPC
  5           ICOMBO(NCOMB,5)=ID(IIIII)
  4           ICOMBO(NCOMB,4)=ID(IIII)
  3           ICOMBO(NCOMB,3)=ID(III)
  2           ICOMBO(NCOMB,2)=ID(II)
  1           ICOMBO(NCOMB,1)=ID(I)
 10           CONTINUE
 20         CONTINUE
 30       CONTINUE
 40     CONTINUE
 50   CONTINUE
      RETURN
      END
/*
//GO.FT05F001 DD *
 6 4 1 2 3 4 5 6
 1
 2
 3
 4
 5
//GO.FT09F001 DD DSN=SBS.MEH1B.BOOK6C.R240C256,DISP=SHR,LABEL=(,,,IN)
//GO.FT10F001 DD DSN=SBS.MEH.BOOK6C.STATS,DISP=SHR,LABEL=(,,,IN)
```



OFEK07047

# CLASSIFY

```
//BOOKCL JOB (D6600217),'VF-BOOKCL',TIME=(,19),MSGLEVEL=(1,0),
// USER=SBS01,PASSWORD=??????
/*ROUTE PRINT MVS.R8
// EXEC DSUTIL
 DELETE DSN=SBS.MEH1B.CLASSIF.R240C256
// EXEC FORTGCLG,PARM.LKED='NOMAP,NOXREF'
//FORT.SYSPRINT DD DUMMY
//FORT.SYSIN DD *
C*************************************************************************
C  CLASSIFICATION PROGRAM
C    CLASSIFIES AN IMAGE OR CREATES A CONTINGENCY TABLE
C
C INPUT PARAMETER LINES ARE:
C LINE 1:
C  COLUMNS    DESCRIPTION
C   1-4       PUT A 'CONT' FOR COMPUTING A CONTINGENCY TABLE ON
C             TRAINING SITES ONLY, OR 'MAP ' FOR A CLASSIFIED MAP
C             TO BE CREATED ON DISK.
C   5-8       CLASSIFIER TYPE: 'MIND' FOR MINIMUM DISTANCE CLASSIFIER
C
C LINE 2:
C  COLUMNS    DESCRIPTION
C   1-2       # OF BANDS TO USE IN CLASSIFIER
C   3-4       BAND # TO USE
C   5-6       BAND # TO USE
C  ETC.        IN I2 FORMAT
C
C LINE 3:
C  COLUMNS    DESCRIPTION
C   1-2       # OF CLASSES TO USE IN CLASSIFIER
C   3-4       CLASS # TO USE
C   5-6       CLASS # TO USE
C  ETC.        IN I2 FORMAT
C
C MAIN VARIABLES AND CORRESPONDING ARRAY SIZE LIMITS:
C  NCLMAX- MAXIMUM # OF CLASSES USED IN THE IMAGE FILE
C
C  NCMAX - MAXIMUM # OF COLUMNS OR PIXELS IN A LINE
C          THIS VALUE MUST NOT BE LARGER THAN THE 'LINE.' ARRAY SIZES.
C  NBMAX - THE MAXIMUM NUMBER OF BANDS OF DATA WHICH COULD BE USED.
C          (THIS NUMBER MAY BE LARGER THAN THE ACTUAL NUMBER OF BANDS
C          IN THE IMAGE FILE, HOWEVER, IT MUST NOT BE LARGER THAN THE
C          STATISTICS ARRAY SIZES IN MAIN.
C  NBLK  - # OF PIXELS PER LOGICAL RECORD IN IMAGE FILE.
C           (MUST BE CONSISTENT WITH OTHER PROGRAMS AND IMPORTANTLY
C            WITH THE ARRAY 'IBYTE' SIZE IN 'RWLINE' SUBROUTINE.
C
C IN ADDITION TO THE VARIABLES ABOVE THE # OF RECORDS MUST BE SET
C IN THE 'DEFINE FILE 9(#' STATEMENT.
C ALSO, THE # OF RECORDS AND BLOCKSIZE  MUST BESET IN THE:
C                'DEFINE FILE 10(#,##' (TRAINING FILE)
C                'DEFINE FILE 11(#,##' (CLASSIFIED FILE)
C
C LOGICAL UNITS USED:
C  5 - USER INPUT PARAMETERS
```



OFEK07048

```
C  6 - LINE PRINTER OR TERMINAL
C  9 - IMAGE FILE
C 10 - TRAINING STATISTICS FILE
C 11 - CLASSIFIED FILE (ONLY IF 'MAP ' IS REQUESTED)
C*****************************************************************
      DIMENSION RNAMEC(25,6), RMNALL(25,6), RSTALL(25,6),
     &          IALCOR(25,4,4), NSAMP(25), ITOTP(25), FILTIT(20)
      DIMENSION ICLASS(25), IBAND(6), ITOTC(25), DIST(25),
     &          ITABLE(25,25), ITCOM(25)
      DIMENSION MINC(6), MAXC(6), RMEANC(6), VARC(6), STDC(6),
     &          COVC(6,6), CORRC(6,6)
      INTEGER*2 LINE6(2100,6), LINCLS(2100), LINE1(2100)
      INTEGER*2 ISTATU(25)
      DATA ITOTC/25*0/, ITCOM/25*0/
      DATA RMIND/'MIND'/, RMAXL/'MAXL'/, RMAP/'MAP '/, RCONT/'CONT'/
      DATA NBLK/100/, NCMAX/2100/, NBMAX/6/, NCLMAX/25/
      DATA IERR/0/
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C-----> (INTEGER((NCOLS/NBLK)+1)*NROWS)*NBAND + 3
      DEFINE FILE 9(4323,100,L,IREC)
C
C-----> 1ST ARGUMENT SHOULD BE SET TO THE # OF RECORDS IN YOUR
C-----> PARTICULAR FILE. # OF RECORDS IS CALCULATED BY:
C----->   # RECS = (NCLMAX * 2) + 2
      DEFINE FILE 10(52,184,L,IRECC)
C
C   ...SET UP FOR CLASSIFIED DATA IF REQUESTED
      DEFINE FILE 11(723,100,L,IREC2)
C
C......READ HEADER RECORD FROM FILE
      CALL RWHEAD(9,NROWS,NCOLS,NBAND,FILTIT,0)
      IF(NBAND.GT.NBMAX) GOTO 9000
C
C  ....COMPUTE # OF RECORDS PER ROW
      NACC=(NCOLS/NBLK) + 1
      IREM=MOD(NCOLS,NBLK)
      IF(IREM.EQ.0) NACC=NACC-1
C
C.......READ STATISTICS FILE HEADER
      CALL RWSTHE(ISTATU,NCLMAX,0)
C
C......PRINT PROGRAM BANNER
      CALL CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C
C......READ TYPE OF CLASSIFICATION TO PERFORM & OTHER USER REQUESTS
      CALL READIN(IBAND,ICLASS,NCLMAX,NBMAX,NCLASS,NBAND,NBUSE,
     &            ISTATU,CTYPE,RCLASS,IERR)
      IF(IERR.EQ.1) GOTO 9999
C
C......READ TRAINING CLASS STATISTICS
      CALL ALLSTA(RMNALL,RSTALL,RNAMEC,ICLASS,NCLASS,NBAND,
     &            NCLMAX,MINC,MAXC,RMEANC,VARC,STDC,COVC,CORRC,
     &            IALCOR,NSAMP,ITOTP,ISTATU)
```



OFEK07049

```
      WRITE(6,2)
    2 FORMAT(1X,'--> STATISTICS READ')
C
C......PERFORM CLASSIFICATON FOR IMAGE
C  ....WRITE HEADER FOR CLASSIFIED IMAGE
      IF(CTYPE.EQ.RMAP) CALL RWHEAD(11,NROWS,NCOLS,1,FILTIT,1)
      IF(CTYPE.EQ.RMAP) CALL MIND(RMNALL,RSTALL,RNAMEC,IBAND,ICLASS,
     &                NCLASS,NBUSE,NBMAX,LINE6,LINCLS,LINE1,ITOTC,DIST,
     &                NCMAX,NCLMAX,NACC,RCLASS,NROWS,NCOLS,NBLK)
C
C......GENERATE CONTINGENCY TABLE FOR TRAINING STATS
      IF(CTYPE.EQ.RCONT) CALL CONTIN(RMNALL,RSTALL,RNAMEC,IBAND,ICLASS,
     &                NCLASS,NSAMP,ITOTP,NBUSE,NBMAX,LINE6,LINE1,
     &                IALCOR,ITOTC,DIST,ITABLE,ITCOM,NCMAX,NCLMAX,NACC,
     &                RCLASS,NROWS,NCOLS,NBLK)
      GOTO 9999
C
C.......ERROR MESSAGES ..............
 9000 WRITE(6,9001) NBANDS,NBMAX
 9001 FORMAT(1X,'*** SORRY, CANNOT COMPUTE STATISTICS BECAUSE '/
     &       1X,'    THERE ARE ',I2,' BANDS IN THE IMAGE FILE AND'/
     &       1X,'    YOU HAVE THE PROGRAM LIMITS AT ',I2,'.'/
     &       1X,'    INCREASE THE STATS ARRAYS AND NBMAX IN MAIN.')
      GOTO 9999
 9999 STOP
      END
C*********************************************************************
      SUBROUTINE CREDIT(NROWS,NCOLS,NBAND,FILTIT)
C*********************************************************************
C SIMPLY PRINTS A BANNER ON THE OUTPUT WITH LIMITS OF THE
C IMAGE FILE.
C*********************************************************************
      DIMENSION FILTIT(20)
      WRITE(6,11) FILTIT,NROWS,NCOLS,NBAND
   11 FORMAT(1X,60('*')/
     &  '          CLASSIFICATION PROGRAM'//
     &  1X,60('*')//
     &  '---> PARAMETERS FOR IMAGE FILE:'/
     &  1X,20A4//
     &  ' NUMBER OF ROWS=    ',I5,/
     &  ' NUMBER OF COLUMNS=',I5,/
     &  ' NUMBER OF BANDS=',I5//)
      RETURN
      END
C*********************************************************************
      SUBROUTINE READIN(IBAND,ICLASS,NCLMAX,NBMAX,NCLASS,NBAND,NBUSE,
     &                  ISTATU,CTYPE,RCLASS,IERR)
C*********************************************************************
C INPUTS THE PARAMETERS TO CONTROL THE CLASSIFICATION
C*********************************************************************
      DIMENSION ICLASS(NCLMAX), IBAND(NBMAX)
      INTEGER*2 ISTATU(NCLMAX)
      DATA RMAP/'MAP '/, RCONT/'CONT'/, RMIND/'MIND'/, RMAXL/'MAXL'/
C
    1 READ(5,11) CTYPE,RCLASS
```



OFEK07050

```
   11 FORMAT(2A4)
C  ....CHECK CLASSIFIER REQUESTED
      IF(CTYPE.NE.RMAP .AND. CTYPE.NE.RCONT) GOTO 9000
C .....INPUT # OF BANDS AND BAND #'S TO USE
      READ(5,12,END=9900) NBUSE,(IBAND(J),J=1,NBUSE)
   12 FORMAT(I2,6I2)
C .....INPUT CLASSES TO USE
      NCLASS=1
   20 READ(5,21,END=9900) NCLASS,(ICLASS(J),J=1,NCLASS)
   21 FORMAT(I2,39I2)
C
      IF(NBUSE.LT.1 .OR. NBUSE.GT.NBAND) GOTO 9100
      DO 10 I=1,NBUSE
        IBA=IBAND(I)
        IF(IBA.LT.1 .OR. IBA.GT.NBAND) GOTO 9200
   10   CONTINUE
      IF(NCLASS.LT.1) GOTO 9300
      DO 30 I=1,NCLASS
        ICLS=ICLASS(I)
        IF(ISTATU(ICLS).EQ.0) GOTO 9400
   30   CONTINUE
      RETURN
 9000 WRITE(6,9001) CTYPE
 9001 FORMAT(1X,'*** ERROR ',A4,'IS NOT A VALID CHOICE')
      IERR=1
      RETURN
 9100 WRITE(6,9101) NBAND,NBUSE
 9101 FORMAT(1X,'*** ERROR, THERE ARE ',I2,' BANDS IN THE IMAGE'/
     &       1X,'           FILE AND YOU ARE REQUESTING ',I2,' BANDS')
      IERR=1
      RETURN
 9200 WRITE(6,9201) IBA
 9201 FORMAT(1X,'*** ERROR, YOU REQUESTED BAND',I2,' WHICH DOES NOT'/
     &       1X,'           EXIST IN THE IMAGE FILE')
      IERR=1
      RETURN
 9300 WRITE(6,9301) NCLASS
 9301 FORMAT(1X,'*** ERROR, YOU HAVE REQUESTED ',I2,' CLASSES')
      IERR=1
      RETURN
 9400 WRITE(6,9401) ICLS
 9401 FORMAT(1X,'*** ERROR, CLASS # ',I2,' HAS NOT BEEN TRAINED ON ')
 9900 IERR=1
      RETURN
      END
C*****************************************************************************
      SUBROUTINE ALLSTA(RMALL,RSTALL,RNAMEC,ICLASS,NCLASS,NBAND,
     &                  NCLMAX,MINC,MAXC,RMEANC,VARC,STDC,COVC,CORRC,
     &                  IALCOR,NSAMP,ITOTP,ISTATU)
C*****************************************************************************
C READS THE STATISTICS FOR ALL CLASSES FROM THE STATS FILE
C*****************************************************************************
C ... NOTE: COVC AND CORRC ARRAYS ARE STORED AS COVC(ROW,COLUMN)
      DIMENSION MINC(NBAND), MAXC(NBAND), RMEANC(NBAND), VARC(NBAND),
     &          STDC(NBAND),COVC(NBAND,NBAND), CORRC(NBAND,NBAND)
```



OFEK07051

```
      DIMENSION RNAMEC(NCLMAX,NBAND), RMNALL(NCLMAX,NBAND),
     &          RSTALL(NCLMAX,NBAND)
      DIMENSION ICLASS(NCLMAX), RNAME(4), ICOORD(4,4),
     &          IALCOR(NCLMAX,4,4), NSAMP(NCLMAX),ITOTP(NCLMAX)
      INTEGER*2 ISTATU(NCLMAX)
C
C
C.....RETRIEVE STATS FOR EACH CLASS
C.....COMPUTE SIMPLE STATISTICS FOR EACH CHANNEL
      DO 100 ICLS=1,NCLMAX
        IF(ISTATU(ICLS).EQ.0) GOTO 100
C.......READ RECORD TO STATS FILE
        CALL RWSTAT(RMEANC,VARC,MINC,MAXC,COVC,RNAME,ICLS,NBAND,
     &              NS,ICOORD,ITOTPX,0)
        DO 20 IBA=1,NBAND
          RMNALL(ICLS,IBA)= RMEANC(IBA)
          RSTALL(ICLS,IBA)= SQRT(VARC(IBA))
   20     CONTINUE
        DO 50 ICR=1,4
          DO 40 IC=1,4
   40       IALCOR(ICLS,ICR,IC)=ICOORD(ICR,IC)
   50     RNAMEC(ICLS,ICR)= RNAME(ICR)
        NSAMP(ICLS)=NS
        ITOTP(ICLS)=ITOTPX
  100   CONTINUE
C
      RETURN
      END
C*******************************************************************************
      SUBROUTINE MIND(RMNALL,RSTALL,RNAMEC,IBAND,ICLASS,NCLASS,NBUSE,
     &                NBMAX,LINE6,LINCLS,LINE1,ITOTC,DIST,
     &                NCMAX,NCLMAX,NACC,RCLASS,NROWS,NCOLS,NBLK)
C*******************************************************************************
C  MINIMUM DISTANCE CLASSIFIER FOR ENTIRE IMAGE
C*******************************************************************************
      DIMENSION RNAMEC(NCLMAX,4), RMNALL(NCLMAX,NBMAX),
     &          RSTALL(NCLMAX,NBMAX)
      DIMENSION ICLASS(NCLMAX), IBAND(NBUSE)
      DIMENSION DIST(NCLMAX), ITOTC(NCLMAX)
      INTEGER*2 LINE6(NCMAX,NBUSE), LINCLS(NCMAX), LINE1(NCMAX)
C
C .....WRITE HEADER RECORD FOR OUTPUT FILE
      WRITE(6,1)
    1 FORMAT(1X,'--> MINIMUM DISTANCE CLASSIFIER CHOSEN'/)
C
C.....PRINT OUT MINIMUM DISTANCES FOR CLASSES
      WRITE(6,2) (IBAND(J),J=1,NBUSE)
    2 FORMAT('1','CLASS',32X,'BANDS'/
     &       1X,27X,10(2X,I2,3X))
      DO 5 I=1,NCLASS
        ICLS=ICLASS(I)
        WRITE(6,3) ICLS,(RNAMEC(ICLS,J),J=1,4),
     &             (RMNALL(ICLS,IBAND(J)),J=1,NBUSE)
    3   FORMAT(1X,I3,2X,4A4,'MEAN',2X,10(F5.1,2X))
        WRITE(6,4) (RSTALL(ICLS,IBAND(J)),J=1,NBUSE)
```



OFEK07052

```
    4   FORMAT(1X,21X,'STD.',2X,10(F5.1,2X))
    5   CONTINUE

      DO 200 IROW=1,NROWS
C     ...GET ALL BANDS FOR THE CURRENT ROW
        DO 50 IB=1,NBUSE
          IBA=IBAND(IB)
          CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,IROW,IBA,NBLK,0,IERR)
          DO 20 ICOL=1,NCOLS
   20       LINE6(ICOL,IBA)=LINE1(ICOL)
   50     CONTINUE
C
C  ....CLASSIFY THE ENTIRE ROW
      DO 100 ICOL=1,NCOLS
C
C    ....CALCULATE EUCLIDEAN DISTANCES FOR EACH OF THE POSSIBLE CLASSES
        RMIND=99999999.
        DO 80 IC=1,NCLASS
          ICLS=ICLASS(IC)
          DIST(ICLS)=0.
          DO 60 IBA2=1,NBUSE
            IBA=IBAND(IBA2)
            VAL=LINE6(ICOL,IBA)
            DIST(ICLS)=DIST(ICLS)+ (VAL-RMNALL(ICLS,IBA))**2
   60       CONTINUE
C
          DIST(ICLS)=SQRT(DIST(ICLS))
C       ....IF CURRENT DIST. IS LESS THAN OLD DIST. RESET CLASS
          IF(RMIND.LT.DIST(ICLS)) GOTO 80
          MINCLS=ICLS
          RMIND=DIST(ICLS)
   80     CONTINUE
C
C       ...KEEP TRACK OF TOTAL STATS & KEEP BUILDING CLASSIFIED LINE
        ITOTC(MINCLS)=ITOTC(MINCLS)+1
        LINCLS(ICOL)=MINCLS
  100   CONTINUE
C.........OUTPUT DATA LINE
        CALL RWLINE(11,LINCLS,NCMAX,NROWS,NACC,IROW,1,NBLK,1,IERR)
  200   CONTINUE
C
C......PRINT ALL STATISTICS
      CALL STOUT(RNAMEC,ICLASS,ITOTC,NCLMAX,NCLASS)
      RETURN
      END
C*******************************************************************************
      SUBROUTINE CONTIN(RMNALL,RSTALL,RNAMEC,IBAND,ICLASS,NCLASS,NSAMP,
     &                  ITOTP,NBUSE,NBMAX,LINE6,LINE1,IALCOR,ITOTC,DIST,
     &                  ITABLE,ITOOM,NCMAX,NCLMAX,NACC,RCLASS,NROWS,
     &                  NCOLS,NBLK)
C*******************************************************************************
C  MINIMUM DISTANCE CLASSIFIER; CONTINGENCY TABLE OUTPUT
C*******************************************************************************
      DIMENSION RNAMEC(NCLMAX,4), RMNALL(NCLMAX,NBMAX),
     &          RSTALL(NCLMAX,NBMAX), IALCOR(NCLMAX,4,4)
```



OFEK07053

```
      DIMENSION ICLASS(NCLMAX), IBAND(NBUSE), NSAMP(NCLMAX),
     &          ITOTP(NCLMAX), ITCOM(NCLMAX)
      DIMENSION DIST(NCLMAX), ITOTC(NCLMAX), ITABLE(NCLMAX,NCLMAX)
      INTEGER*2 LINE6(NCMAX,NBUSE), LINE1(NCMAX)
C
C .....WRITE HEADER RECORD FOR OUTPUT FILE
      WRITE(6,1)
    1 FORMAT(1X,'--> CONTINGENCY TABLE REQUESTED'/)
C
C......PRINT OUT MINIMUM DISTANCES FOR CLASSES
      WRITE(6,2) (IBAND(J),J=1,NBUSE)
    2 FORMAT('1','CLASS',32X,'BANDS'/
     &       1X,27X,10(2X,I2,3X))
      DO 5 I=1,NCLASS
        ICLS=ICLASS(I)
        WRITE(6,3) ICLS,(RNAMEC(ICLS,J),J=1,4),
     &             (RMNALL(ICLS,IBAND(J)),J=1,NBUSE)
    3   FORMAT(1X,I3,2X,4A4,'MEAN',2X,10(F5.1,2X))
        WRITE(6,4) (RSTALL(ICLS,IBAND(J)),J=1,NBUSE)
    4   FORMAT(1X,21X,'STD.',2X,10(F5.1,2X))
    5   CONTINUE

C
      DO 500 I=1,NCLASS
        ICLS1=ICLASS(I)
        NUMSAM=NSAMP(ICLS1)
C
        DO 400 ISAM=1,NUMSAM
          IBR=IALCOR(ICLS1,ISAM,1)
          IER=IALCOR(ICLS1,ISAM,2)
          IBC=IALCOR(ICLS1,ISAM,3)
          IEC=IALCOR(ICLS1,ISAM,4)
C
          DO 200 IROW=IBR,IER
C         ...GET ALL BANDS FOR THE CURRENT ROW
            DO 150 IB=1,NBUSE
              IBA=IBAND(IB)
              CALL RWLINE(9,LINE1,NCMAX,NROWS,NACC,IROW,IBA,NBLK,
     &                    0,IERR)
              DO 120 ICOL=IBC,IEC
                LINE6(ICOL,IBA)=LINE1(ICOL)
  120           CONTINUE
  150         CONTINUE
C
C  ....CLASSIFY THE PART OF ROW CORRESPONDING TO CLASS
            DO 100 ICOL=IBC,IEC
C
C         ....CALCULATE EUCLIDEAN DISTANCES FOR EACH OF THE CLASS
              RMIND=99999999.
              DO 80 IC=1,NCLASS
                ICLS=ICLASS(IC)
                DIST(ICLS)=0.
                DO 60 IBA2=1,NBUSE
                  IBA=IBAND(IBA2)
                  VAL=LINE6(ICOL,IBA)
```



OFEK07054

```
                DIST(ICLS)=DIST(ICLS)+ (VAL-RMNALL(ICLS,IBA))**2
  60            CONTINUE
C
              DIST(ICLS)=SQRT(DIST(ICLS))
C           ....IF CURRENT DIST. IS LESS THAN OLD DIST. RESET CLASS
              IF(RMIND.LT.DIST(ICLS)) GOTO 80
              MINCLS=ICLS
              RMIND=DIST(ICLS)
  80          CONTINUE
C
C         ...KEEP TRACK OF TOTAL STATS & KEEP BUILDING CONTINGENCY TABLE
            ITABLE(ICLS1,MINCLS)=ITABLE(ICLS1,MINCLS) + 1
 100          CONTINUE
 200        CONTINUE
 400      CONTINUE
 500    CONTINUE
      ITOTAC=0
      ITOTPX=0
C
      DO 600 ICLS=1,NCLASS
        ITOTPX=ITOTPX + ITOTP(ICLS)
 600    ITOTAC=ITOTAC + ITABLE(ICLS,ICLS)
      PERAC=FLOAT(ITOTAC) / FLOAT(ITOTPX)
C
C.....PRINT TABLE
      CALL TABOUT(RNAMEC,ICLASS,ITOTP,ITABLE,ITCOM,NCLMAX,NCLASS,
     &            ITOTPX,PERAC)
      RETURN
      END
C*******************************************************************************
      SUBROUTINE TABOUT(RNAMEC,ICLASS,ITOTP,ITABLE,ITCOM,NCLMAX,
     &                  NCLASS,ITOTPX,PERAC)
C*******************************************************************************
C  PRINT THE CONTINGENCY TABLE
C*******************************************************************************
      DIMENSION RNAMEC(NCLMAX,4)
      DIMENSION ICLASS(NCLMAX), ITABLE(NCLMAX,NCLMAX)
      DIMENSION ITOTP(NCLMAX), ITCOM(NCLMAX)
      DIMENSION RTOT(2), ROMIT(2), RCOMIT(2), RACCUR(2), BLANK(25),
     &          RTOP(4),RPER(2)
      DATA RTOT/'TOTA','L   '/, ROMIT/'OMIS','SION'/,
     &     RCOMIT/'COMI','SION'/, RACCUR/'ACCU','RACY'/,
     &     RTOP/'PRED','ICTE','D CL','ASS '/,
     &     RPER/'   %','    '/, BLANK/25*'    '/
C
      NCL2=(NCLASS-1)*2
      WRITE(6,1) (RTOP(J),J=1,4), (BLANK(J),J=1,NCL2),
     &           ROMIT,RCOMIT,RACCUR
    1 FORMAT(1X,///1X,22X,4A4,25(2X,2A4))
      WRITE(6,2) ((RNAMEC(ICLASS(J),K),K=1,2),J=1,NCLASS),RTOT,RPER,
     &           RPER,RPER
    2 FORMAT(1X,18X,25(2X,2A4))
      WRITE(6,3)
    3 FORMAT(1X,'# ACTUAL CLASS')
      DO 50 I=1,NCLASS
```



OFEK07055

```
        ICLS=ICLASS(I)
       ITOMIT=0
       ITCOMT=0
        DO 20 J=1,NCLASS
          JCLS=ICLASS(J)
          IF(ICLS.NE.JCLS) ITOMIT= ITOMIT + ITABLE(ICLS,JCLS)
          IF(ICLS.NE.JCLS) ITCOMT= ITCOMT + ITABLE(JCLS,ICLS)
          ITCOM(J)= ITCOM(J) + ITABLE(ICLS,JCLS)
   20     CONTINUE
        ITOMIT= (FLOAT(ITOMIT) / FLOAT(ITOTP(ICLS))) * 100.
        ITCOMT= (FLOAT(ITCOMT) / FLOAT(ITOTP(ICLS))) * 100.
        ITACC= (FLOAT(ITABLE(ICLS,ICLS)) / FLOAT(ITOTP(ICLS))) * 100.
C
        WRITE(6,4) ICLS,(RNAMEC(ICLS,J),J=1,4),
     &             (ITABLE(ICLS,ICLASS(J)),J=1,NCLASS), ITOTP(ICLS),
     &             ITOMIT,ITCOMT,ITACC
    4   FORMAT(1X,I2,1X,4A4,25(I5,5X))
   50   CONTINUE
       WRITE(6,11) (ITCOM(J),J=1,NCLASS), ITOTPX
   11  FORMAT(1X/1X,'TOTAL',14X,25(I5,5X))
       WRITE(6,12) PERAC
   12  FORMAT(1X,' OVERALL ACCURACY=',F6.2,'%')
       RETURN
       END
C*************************************************************************
      SUBROUTINE STOUT(RNAMEC,ICLASS,ITOTC,NCLMAX,NCLASS)
C*************************************************************************
C  PRINT THE SUMMARY STATISTICS FOR EACH CLASS
C*************************************************************************
      DIMENSION RNAMEC(NCLMAX,4)
      DIMENSION ICLASS(NCLMAX)
      DIMENSION ITOTC(NCLMAX)
      WRITE(6,1)
    1 FORMAT(1X,///'                                        TOTAL '/
     &       1X,'CLASS       NAME                    PIXELS'/)
      DO 50 I=1,NCLASS
        ICLS=ICLASS(I)
        WRITE(6,2) ICLS,(RNAMEC(ICLS,J),J=1,4),ITOTC(ICLS)
    2   FORMAT(1X,I5,5X,4A4,3X,I8)
   50   CONTINUE
       RETURN
       END
/*
//GO.FT05F001 DD *
MAP MIND
 3 4 5 6
 5 1 2 3 4 5
//GO.FT09F001 DD DSN=SBS.MEH1B.BOOK6C.R240C256,DISP=SHR,LABEL=(,,,IN)
//GO.FT10F001 DD DSN=SBS.MEH.BOOK6C.STATS,DISP=SHR,LABEL=(,,,IN)
//GO.FT11F001 DD DISP=(NEW,CATLG,UNCATLG),UNIT=3330V,
//     DSN=SBS.MEH1B.CLASSIF.R240C256,SPACE=(100,(723),RLSE),
//     VOL=SER=SBS102,DCB=(RECFM=F,LRECL=100,BLKSIZE=100)
```



OFEK07056

# APPENDIX B

# Addresses of Public and Commercial Suppliers of Digital Image Processing Hardware and/or Software

AIPE: Fred Gunther, 9464 Wandering Way, Columbia, MD 21045.
APPLEPIPS: The Telesys Group, 5455 Wingborne Court, Columbia, MD 21045.
COMTAL: Comtal Corporation, 505 West Woodbury Road, Altadena, CA 91001.
COSMIC: 112 Barrow Hall, The University of Georgia, Athens, GA 30602. COSMIC provides the following image processing software systems: AOIPS, ELAS, ERIPS, LARSYS-TAMU, ORSER-OCCULT, SMIPS, VICAR-IBIS.
DECISION IMAGES: 228 Alexander Road, Princeton, NJ 08540.
ERDAS: Advanced Technology Development Center, Suite N206, 430 Tenth St. NW, Atlanta, GA. 30318.
EYTON: Ron Eyton, Department of Geography, University of Alberta, Edmonton, Alberta, Canada T6G-2H4.
GRIDAPPLE and ARCINFO-GRID-GRIDTOPO: ESRI, 380 New York Street, Redlands, CA 92373.
GYPSEY: Robert Haralick, Department of Electrical Engineering, Virginia Polytechnic, Blacksburg, VA.
HOTLIPS: Nebraska Remote Sensing Center, 113 Nebraska Hall, 901 N. 17th Street, Lincoln, NE 68588–0517.
IDIMS: ESL Inc., 495 Java Drive, Sunnyvale, CA 94086.





OFEK07057

I²S: International Imaging Systems, 1500 Buckeye Drive, Milpitas, CA 95035.
IMAGES: John R. Jensen, Department of Geography, University of South Carolina, Columbia, SC 29208.
KARS: T. H. L. Williams, Department of Geography-Meteorology, University of Kansas, Lawrence, KS 66045.
LogE-EYECOM and ISI: Spatial Data Inc., 420 S. Fairview Avenue, Goleta, CA 93116.
MICROPIPS: The Telesys Group, 5455 Wingborne Court, Columbia, Md 21045.
Mini-LARSYS: Terry Philips, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, IN 47907.
MYERS: H. Joseph Myers, IBM Inc., 1530 Page Mill Road, Palo Alto, CA 94304.
RIPS: EROS Data Center, Applications Branch, Sioux Falls, SD 57198.
SPECDAT: Spectral Data Corporation, 112 Parkway Drive South, Hauppauge, NY 11787.
TERRA-MAR: 2113 Landings Drive, Mountain View, CA 94043.



OFEK07058

# Glossary

**Aerial photograph, vertical:** An aerial photograph made with the optical axis of the camera approximately perpendicular to the earth's datum plane.

**Aerial reconnaissance:** The process of obtaining information from an aerial perspective.

**Algorithm:** A computer-oriented step-by-step procedure for solving a problem.

**Alphanumeric:** A character set composed of letters, integers, punctuation marks, and special symbols.

**Analog:** A form of data recording that works on the principle of continuous measurement, rather than discrete counting.

**Angle of reflection:** The angle that electromagnetic radiation (EMR) reflected from a surface makes with the perpendicular (normal) to the surface.

**Anniversary dates:** Those dates in a remote sensing analysis that occur approximately on the same day of the year in multiple years. Such dates tend to minimize differences in reflectance or emittance caused by seasonal vegetation changes, soil moisture variation, or sun angle difference.

**Atmospheric windows:** Wavelength regions in which radiation can pass through the atmosphere with relatively little attenuation; e.g., 0.3 to 2.5 $\mu$m, 3.0 to 4.0 $\mu$m, 4.2 to 5.0 $\mu$m, and 7.0 to 15.0 $\mu$m.

**Attitude:** The angular orientation of a remote sensing system with respect to a geographical reference system.

**Bilinear interpolation:** A resampling technique used for the geometric correction of images.

**Bit:** A binary digit (0 or 1).

**Bit-mapped:** A type of computer graphic display where every picture element's (pixel) exact red, green, and blue (RGB) characteristics are stored in computer memory and can be displayed on a pixel-by-pixel basis.





OFEK07059

**Brightness map:** A computer graphic display of the brightness values found in digital remote sensor data.

**Brightness value:** Remote sensing systems record the amount of reflected or emitted energy exiting from the earth's surface. These data are stored as digital values on computer-compatible tapes for digital image processing purposes. The greater the brightness of the scene (or return from the scene), the higher the digital value. Hence the values stored on the CCT are often referred to as brightness values. It is not proper to call them reflectance values unless the brightness values have been scaled to truly represent reflectance values.

**Byte:** 8 bits of digital data.

**Camera:** A lightproof chamber in which the image of an exterior object is projected on a sensitized plate or film.

**Change-detection:** The comparison of two images acquired at different times.

**Class:** A surface characteristic type that is of interest to the investigator such as forest or water.

**Classification:** The process of assigning individual pixels of a multispectral image to discrete categories.

**Clustering:** The statistical analysis of a set of pixels to detect their inherent tendency to form clusters in multidimensional measurement space.

**Collateral data:** Secondary data pertaining to an area of interest. Also called ancillary data.

**Color:** That property of an object which is dependent on the wavelength of the light it reflects or, in the case of a luminescent body, the wavelength of light that it emits. If this light is of a single wavelength, the color seen is a pure spectral color; but if light of two or more wavelengths is emitted, the color will be mixed.

**Color composite:** A color image produced by assigning a color to each of a number of images of a scene and optically or digitally superimposing the result.

**Color infrared film:** Photographic film sensitive to energy in the visible and near-infrared wavelengths, generally from 0.4 to 0.9 $\mu$m; usually used with a minus-blue (yellow) filter, which results in an effective film sensitivity of 0.5 to 0.9 $\mu$m. Color infrared film is especially useful for detecting changes in the condition of the vegetative canopy which are often manifested in the near-infrared region of the spectrum. Color infrared film is not sensitive in the thermal infrared region and therefore cannot be used as a heat-sensitive detector.

**Computer-compatible tapes:** Tapes containing digital data. These tapes are usually standard 19-cm (7½-in.)-wide magnetic tapes in nine-track format.

**Context:** The relationship between a single pixel (or group of pixels) and other pixels within the scene.

**Contrast stretching:** Improving the contrast of images by digital processing. The original range of digital values is expanded to utilize the full contrast range of the recording film or display device.

**Convolution:** Digital image filtering usually performed using an $n \times n$ mask (e.g., a $3 \times 3$, $5 \times 5$ mask). The weights of the mask may be specified by the analyst.

**Correlation:** The degree of interrelation between variables in a manner not influenced by measurement units.

**Covariance:** The measure of how two variables change in relation to each other. If larger values of $Y$ tend to be associated with larger values of $X$, the covariance will be positive. If



OFEK07060

larger values of $Y$ are associated with smaller values of $X$, the covariance will be negative. When there is no particular association between $X$ and $Y$, the covariance will approach zero.

**Cubic convolution:** A resampling technique used for the geometric correction of images.

**Cursor:** Aiming device, such as a lens with cross hairs on a digitizer or an interactive computer display.

**Data acquisition system:** Those devices that measure physical variables and record them.

**Data dimensionality:** The number of variables (e.g., bands) present in the data set.

**Decision rule:** The criterion used to establish discriminant functions for image classification.

**Densitometer:** A device used to measure the average density of a small area of specified size on a photographic transparency or print. Microdensitometers can measure the average density in areas a few hundred micrometers square.

**Density:** A measure of the light-absorbing capability of the silver or dye deposited at an $x, y$ location in an image.

**Density slicing:** The process of converting the continuous gray tone of an image into a series of density intervals, or slices, each corresponding to a specific digital range.

**Dichroic:** A beam-splitting mirror that efficiently reflects certain wavelengths while efficiently transmitting others.

**Digital data:** Data displayed, recorded, or stored in binary notation.

**Digitization:** The process of converting material from a continuous into a discrete format.

**Discriminate function:** Mathematical functions which are commonly derived from training samples and a decision rule and are used to divide the measurement space into decision regions as part of image classification.

**Display:** An output device that produces a visible representation of a data set for visual analysis.

**Divergence:** A measure of statistical separability used to select the optimum subset of bands for discriminating among several land-use or land-cover classes of interest.

**Edge:** The boundary of an object in a photograph or image, usually characterized by a drastic change in the brightness value from the immediate interior of the boundary to the immediate exterior of the boundary.

**Edge enhancement:** The use of analytical techniques to emphasize edges in imagery.

**Electromagnetic spectrum:** The ordered array of electromagnetic radiation extending from short cosmic waves to long radio waves.

**Environment:** An external condition, or the sum of such conditions, in which a piece of equipment or a system operates, as in temperature environment, vibration environment, or space environment.

**External errors:** Due to platform perturbations and the modulation of the scene characteristics, which are variable in nature.

**Feature:** A vector with components which are functions of the initial measurement pattern variables or some subsequence of the measurement $n$-tuples. Feature vectors frequently have fewer components than the initial measurement vectors and are designed to contain a high amount of information about the discrimination between units of the types of categories in the given category set. Features often contain information about gray shade, texture, shape, or context.

**Feature selection:** The process by which the best subset of $n$ bands is selected for extracting useful thematic information from the remotely sensed data. The goal is to reduce the dimension-



OFEK07061

ality of the data set while maintaining the important information content in the bands selected.

**Field of view:** The solid angle through which an instrument is sensitive to radiation.

**Filtering:** The removal of certain spectral or spatial frequencies to enhance features in the remaining image.

**Flight line:** A line drawn on a map or chart to represent the track over which an aircraft or satellite has flown or is to fly.

**Frequency:** Number of oscillations per unit or number of wavelengths that pass a point per unit time.

**GCP (ground control point):** A geographical feature of known location that is recognizable on images and maps and can be used to determine geometrical corrections.

**Geocoding:** Geographical referencing or coding of data.

**Geographic information system:** A data base management system used to store, retrieve, manipulate, analyze, and display spatial information.

**Geometrical transformations:** Adjustments made in image data to change its geometric character.

**Gray scale:** A monochrome strip of shades ranging from white to black with intermediate shades of gray.

**Grid-based GIS:** A geographic information system (GIS) based on a gridded (raster) data structure.

**Ground resolution element:** The area on the terrain that is covered by the instantaneous field of view of a sensor.

**Ground track:** The vertical projection of the actual flight path of an aerial or space vehicle onto the surface of the earth.

**Ground truth:** Term used in a remote sensing investigations to imply that the data collected in the field are without error.

**Hardware:** The physical components of a computer and its peripheral equipment.

**High-pass filter:** A spatial filter that enhances high spatial frequencies.

**Histogram:** The graphic display of a set of data that shows the frequency of occurrence (along the vertical axis) of individual measurements or values (along the horizontal axis); a frequency distribution.

**Hue:** That attribute of a color by virtue of which it differs from gray of the same brilliance, and which allows it to be classed as red, yellow, green, blue, or intermediate shades of these colors.

**Image:** The representation of one object by something else. Usually, the recorded representation of a scene by optical, electro-optical, optical mechanical, or electronic means.

**Image enhancement:** Any one of a group of operations that improve the detectability of targets or categories. These operations include contrast improvement, edge enhancement, spatial filtering, noise suppression, image smoothing, and image sharpening.

**Image processing:** The various operations that can be applied to image data.

**Image restoration:** A process by which a degraded image is restored to its original condition.

**Information class:** Those categories of land use or land cover that we define as being important.

**Internal errors:** Errors created by the sensor system itself.



OFEK07062

**Joystick:** Interactive control mechanism found on most image processing systems which controls a cursor on a display screen.

**Kernel:** A pixel array used for digital image filtering.

**Land cover:** The biophysical materials covering the surface of the land.

**Landsat:** Unmanned, polar orbiting, U.S. earth resources satellite.

**Land use:** The use to which land is put. May be confused with land cover.

**Layered classification:** The hierarchial process whereby multiple decision rules are used in the classification process.

**Local operations:** Operations that modify the value of each pixel in the context of the brightness values surrounding it.

**Map:** A representation in a plane surface, at an established scale, of the physical features (natural, artificial, or both) of a part of the earth's surface.

**Map, thematic:** A map that depicts particular features or concepts. In conventional use, this term excludes topographical maps.

**Maximum likelihood rule:** A statistical decision criterion to assist in the classification of overlapping signatures; pixels are assigned to the class in which they have the highest probability of being a member.

**Micrometer:** A unit of length equal to one millionth of a meter.

**Minimum distance classifier:** A classification technique that assigns raw data to the class whose mean falls the shortest Euclidean distance from it.

**Mosaic:** An assemblage of overlapping aerial or space photographs or images whose edges have been matched to form a continuous pictorial representation of a portion of the earth's surface.

**Mosaic, controlled:** A mosiac that is registered to ground control and uses prints that have been rectified.

**Multispectral:** Refers to remote sensing in two or more spectral bands, such as visible and near-infrared regions of the spectrum.

**Multispectral scanner:** A scanning detector system that records reflected or emitted energy from the earth's surface in discrete wavelength intervals from 0.3 to 14.0 $\mu$m.

**Nadir:** That point on the ground vertically beneath the perspective center of the camera lens.

**Nearest neighbor:** A resampling technique used for the geometric correction of images.

**Orbit:** The path of a satellite around a body under the influence of gravity.

**Overlap:** The area common to two successive photos along the same flight strip. The amount of overlap is expressed as a percentage of photo area.

**Overlay:** A transparent sheet giving information to supplement that shown on maps.

**Parallax:** The apparent displacement of position of a body with respect to a reference point or system of coordinates, caused by moving the point of observation.

**Parametric:** Classification algorithms that assume that the observed measurement vectors obtained for each class in each band are normally distributed.

**Pattern:** The regularity and characteristic placement of tones or textures. Some descriptive adjectives for patterns are regular, irregular, random, concentric, radial, and rectangular.

**Photogrammetry:** The art and science of obtaining reliable measurements by means of photography.



OFEK07063