USCIPI REPORT #640

Final Research Report on
Computer Processing of ERTS Images

by

Gunar S. Robinson and Werner Frei

September 1975

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

TABLE OF CONTENTS

1. INTRODUCTION

Geologic and geomorphic studies have greatly benefited in many areas from the computer-aided analysis of LANDSAT (formerly Earth Resources Technology Satellite) images. For example, spectral information from ERTS-1 has been shown to be compatible with ground spectral reflectance measurements [1], and a close correlation has been discovered between the surface cover of vegetation and the geomorphologic characteristics of the area [2]. One reason for the attractiveness of computer processing is that subtle ground features present in the multispectral images can be enhanced by digital image processing in order to facilitate the task of manual interpretation of satellite and other imagery by qualified geomorphologists. Furthermore, concepts borrowed from the field of pattern recognition are now being applied increasingly to help the human interpretative task. This has led in certain cases to a completely automatic classification of certain terrain regions [3]. The potential benefits of computer image processing for oil exploration are thus very clear.

The present study has been made to explore and identify digital image processing techniques which are best suited for this particular application. The principal goal was the extraction and enhancement of terrain features of interest in this context and the suppression of artifacts resulting from imperfect satellite sensor characteristics, transmission errors, variable illumination, etc. Significant advances have been made in this domain, as described in the next sections of this report. For example, economical image enhancement algorithms have been developed, in which properties of the human visual system were integrated into the numerical manipulations to produce significantly better visual results. This approach has given images that convey more information to the observer in charge of geomorphological studies. Other results include the successful inhibition of parasitic "band" structure caused by imperfectly matched satellite sensors, the implementation of new ways to extract spectral ground reflection factors by use of logarithmic band ratios, principal components, etc. Line terrain defaults have been enhanced by spatial filtering, crispening, homomorphic and Wiener filter. Color has been used to display the information contained in several spectral bands in a more dramatic fashion in order to reveal possibly hidden geomorphic evidence.

In addition to these direct results, new avenues for further research have emerged from the interaction between the fields of image processing and geology. For example, new uses of pseudo-color have produced images with a number of previously invisible patterns. The correlation of such patterns with ground truth information will reveal which technique is most appropriate for the application considered. On the other hand, classification studies made with the General Electric Image-100 system have shown that spectral classifications often correlate with underlying edge structures. Such structures are amenable to image analysis and it is envisioned that a combination of "intelligent" automatic structure search and spectral classification will provide a significant step towards the automated geomorphic classification of ERTS imagery.

References

[1] Hoffer, W. and LARS Staff, "Techniques for Computer-Aided Analysis of ERTS-1 Data, Useful in Geologic, Forest and Water Resource Surveys," Proceedings of the Third Earth Resources Technology Satellite-1 Symposium, held in Washington, D. C., Dec. 10-14, 1973, Vol. I, Section B, pp. 1687-1708.

[2] Goetz, A. F. H. et al., "Application of ERTS Images and Image Processing to Regional Geologic Problems and Geologic Mapping in Northern Arizona," JPL Technical Report 32-1597, May, 15, 1975.

[3] Idelsohn, J. M., "A Learning System for Terrain Recognition," Pattern Recognition, Pergamon Press, 1970, Vol. 2, pp. 293-301.

2. SUMMARY OF ACCOMPLISHMENTS

This summary is structured according to the statement of work of the research proposal submitted to Chevron Oil Field Research. It is a brief review of the achievements obtained under this contract and serves as an abstract for the more detailed descriptions which follow. It is pointed out that several promising projects were pursued in addition to the statement of work. A great deal of effort was also spent on the formatting of appropriate data bases for all subsequent processing. Two subsections of ERTS frames from different overflights were selected which both cover the Bell Creek area in south Montana. Ad hoc programs were written to compensate for the aspect ratio and earth rotation distortions of ERTS images, the reduction and enlargement of images and the correction of photographic display distortions.

Review of Statement of Work and Comments

1) Sensor and Display Correction.

It was anticipated that the four bands of ERTS data would be converted to true surface radiance measures by compensating for sensor/system nonlinearities. It now appears that this task is severely limited by the absence of appropriate satellite calibration data. In addition, it is generally recognized that such corrections may be of questionable value considering the effect of varying atmospheric absorptions.

A related type of distortion has been combatted successfully however. A correction technique has been implemented, which inhibits the artificial image banding structure produced by a mismatch of the six satellite sensors in each spectral band. This banding suppression algorithm has been used systematically after it was developed.

The correction of display distortions has also been advanced substantially and is described in the Appendix.

2) Histogram manipulation was expected to produce images of each spectral bands with substantially better contrast. The power of this essentially known technique has been significantly increased by use of a visual model to match the processing operation with human visual factors. This new technique, called histogram hyperbolization, has been used systematically thereafter.

3) The finite aperture and noise inherent to ERTS sensors limit the amount of information which can be extracted from each spectral image. Spatial high-pass filtering, homomorphic filtering and edge crispening has been implemented to increase the visibility of fine detail and structures. Wiener filtering has also been implemented in order to compensate for the aperture and noise limitations of the satellite.

4) The extraction of relevant terrain features for human and further machine classification was followed under two different approaches; the Karhunen-Loève expansion or principle component analysis and the band ratios respectively. Both techniques have produced highly differentiated pictures suitable for further analysis, and the ratio approach has been refined by use of logarithmic ratios.

The use of color for the display of multivariate information has been explored, giving a number of possible color combinations with far more information than classical color composites. It is expected that a geomorphic correlation study with ground truth information will indicate which color combination is most appropriate for the present purpose.

5) The estimation of natural terrain color is possible despite the absence of a blue sensor aboard the ERTS satellite. Such estimates rely on certain assumptions on the spectral energy distribution of the terrain which will have to be further justified.

6) It is believed that the "principle component","ratio" and "edge" images provide a wealth of information which can be combined for a more automatic classification approach. In this context, it will be essential to draw on ground truth information to define appropriate training sets for this particular application. For example, it appears that important lineaments can be detected by a combined search of spectral signatures and terrain structures outlined by edges.

3. PROCESSING AND ENHANCEMENT TECHNIQUES

This section is devoted to a detailed description of the image processing principles and theory developed for this project. Each subsection describes one particular operation and the results obtained. In some cases, several distinct operations have been cascaded for more effective results. All the respective software is documented in section four.

3.1 Aspect Ratio and Earth Rotation Compensation

The rectangular scanning aperture of the ERTS satellite and the earth rotation while scanning cause a combined geometric distortion as depicted in Fig. 3.1-1. To compensate for these distortions, a linear pixel interpolation algorithm was used to map arrays of 752 x 512 pixels into 512 x 512 pictures which correspond to a square terrain area of approximately 41 x 41 kilometers (Fig. 3.1-2). (See program ERTSIS, Section 4.) Using this algorithm, to 512 x 512 images were created from the raw ERTS tapes to serve as a data base for further processing. Both images cover the Bell Creek area in south Montana which contains known oil reserves and actual exploitation sites.

3.2 Image Reduction and Enlargement

Some processing operations such as Wiener filtering and homomorphic filtering are performed more economically on smaller size images. Therefore, averaging and interpolation programs have been designed to reduce and enlarge pictures before and after such sophisticated processing techniques.

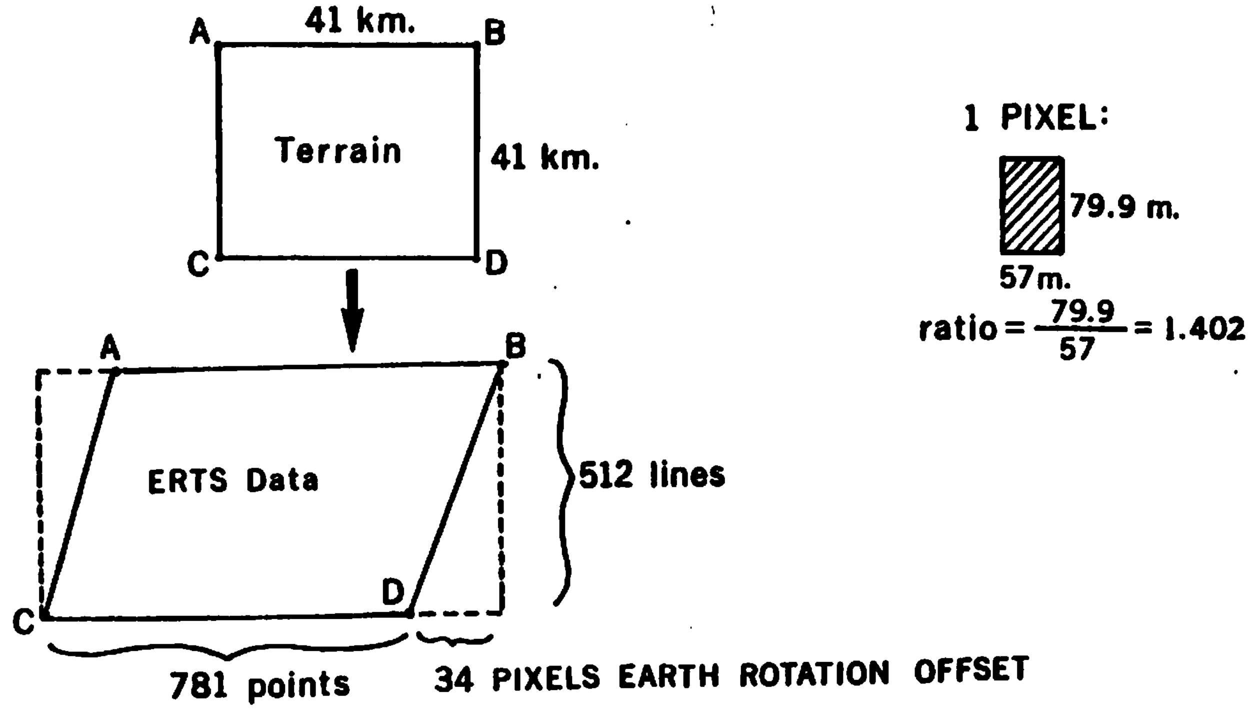


Figure 3.1-1. Aspect ratio and earth rotation compensation

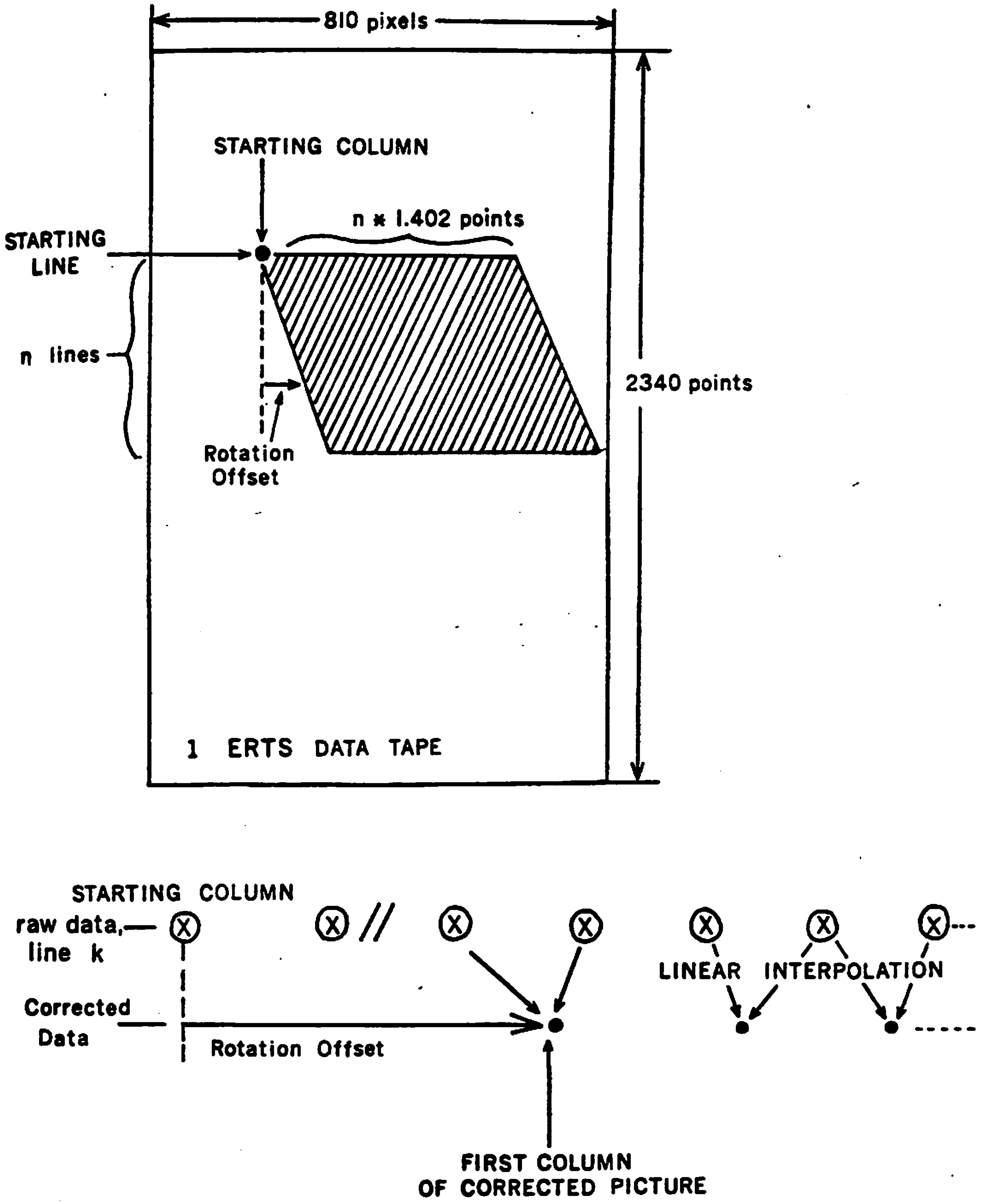
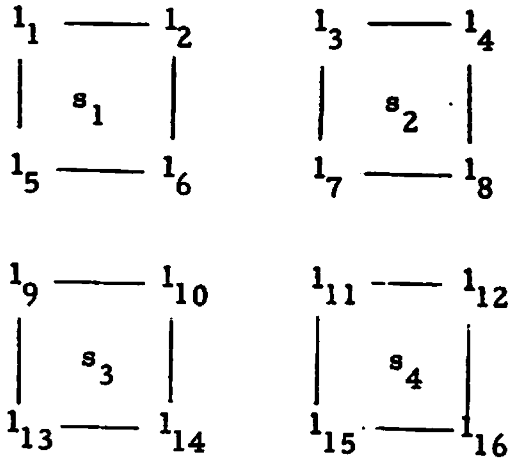


Figure 3.1-2. Geometric correctionร

Image size reduction is accomplished by averaging intensities of four nearest pixels on two scan lines as shown in Fig. 3.2-1, where



$$s_1 = \tfrac{1}{4}(l_1+l_2+l_6+l_6)$$

$$s_2 = \tfrac{1}{4}(l_3+l_4+l_5+l_8)$$

etc.

Figure 3.2-1. Image Size Reduction.

l_i and s_i are the intensity values of the large and resulting small pictures, respectively.

Figure 3.2-2 shows the result of size reduction from 512 x 512 pixels to 256 x 256 pixels on Band 4 (Green).

Image enlargement is accomplished by linear interpolation in two dimensions. For computational simplicity two dimensional interpolation is implemented as two consecutive one-dimensional interpolations, i.e. first, every scan line is interpolated such that the number of samples is twice or four times the original. This one dimensional interpolation is shown in Fig. 3.2-3 in the case of enlargement by two. Result of

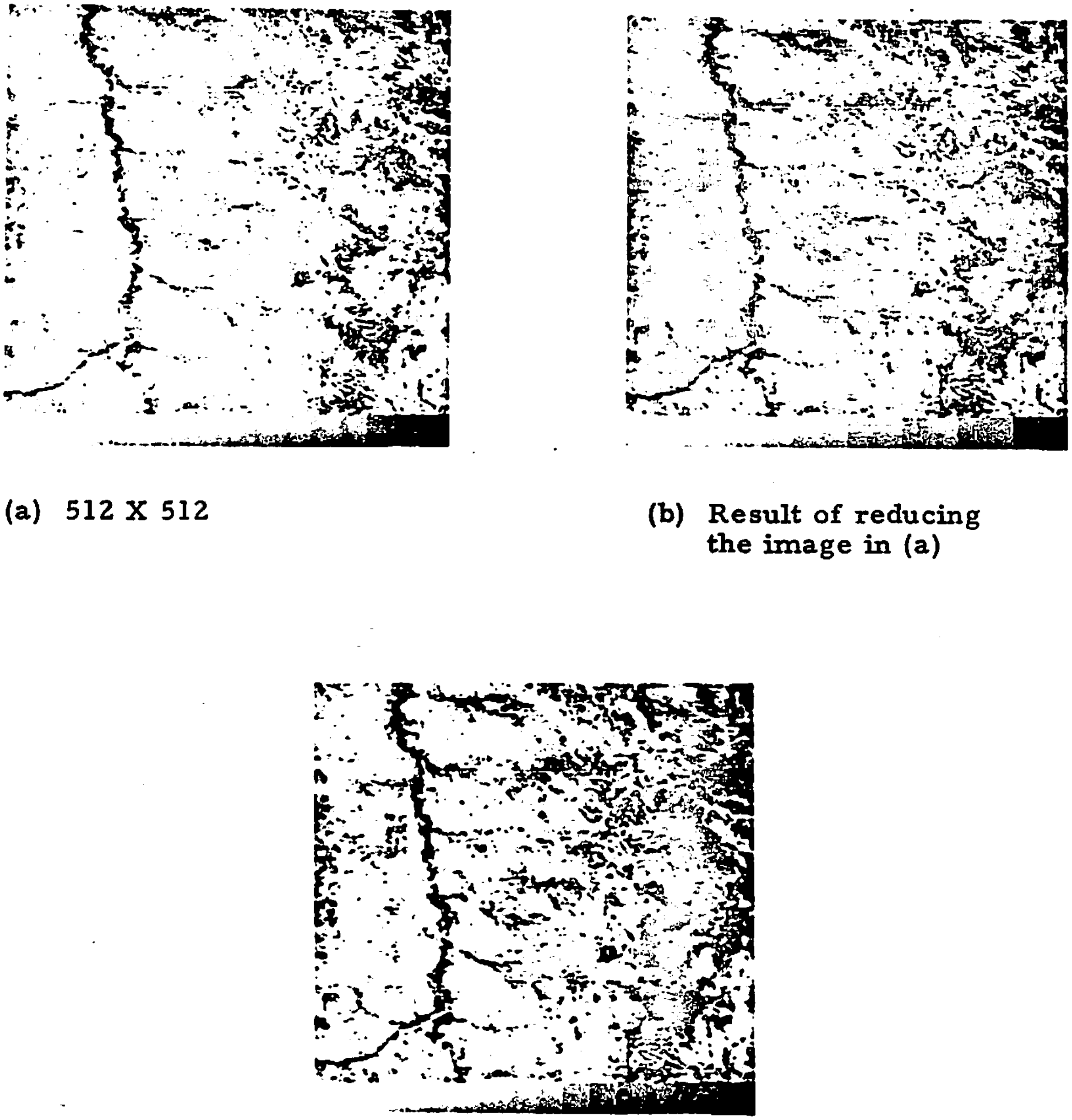
$$\begin{tikzcd} s_1 & l_1 \arrow[l] \arrow[r] & s_2 & l_2 \arrow[l] \arrow[r] & s_3 \end{tikzcd}$$

$$l_1 = \frac{s_1+s_2}{2}$$

$$l_2 = \frac{s_2+s_3}{2}, \text{ etc.}$$

Figure 3.2-3. One-Dimensional Linear Interpolation.

enlargement of reduced picture is shown in Fig. 3.2-2c.



(a) 512 X 512

(b) Result of reducing the image in (a)

(c) Result of enlarging the image in (b)

Figure 3.2-2. Effect of size reduction and enlargement on Band 4 (Green) of ERTS data.

3.3 Histogram Stretching, Equalization and Hyperbolization

ERTS images have typically poor contrast as is revealed by amplitude histograms. (Figs. 3.3-1, 2, 3 and 4.) Various contrast enhancement techniques are available, such as stretching, histogram equalization, etc. They make use of a linear or nonlinear transfer function which may be stored in a look-up table for fast processing. If a transfer function y = f(x) is used, the histogram of the processed picture is modified as follows (Fig. 3.3-5)

$$p(y) = \frac{d}{dy}\left(\int_0^{x(y)} p(x)dx\right) \qquad (1)$$

Where p(x), p(y) are the amplitude probability functions of the original and processed images respectively, and x(y) the inverse of the transfer function y(x). A successful technique is histogram equalization, which tends to make all amplitude levels of the processed picture equi-probable. Let

$$y(x) = \int_0^{x} p(x)dx = F(x) \qquad (2)$$

e.g. the cumulative amplitude distribution of x, then the histogram of the processed picture becomes

$$p(y) = \frac{d}{dy}\left(\int_0^{x=F^{-1}(y)} p(x)dx\right) = \frac{d}{dy}\left[F(F^{-1}(y))\right] = \text{constant} \qquad (3)$$

The implementation of this scheme is done as follows:

a) Estimate the amplitude distribution p(x) of the original image (count the number of pixels of amplitude x ($0 \leq x \leq 255$) and divide by the total number of samples.

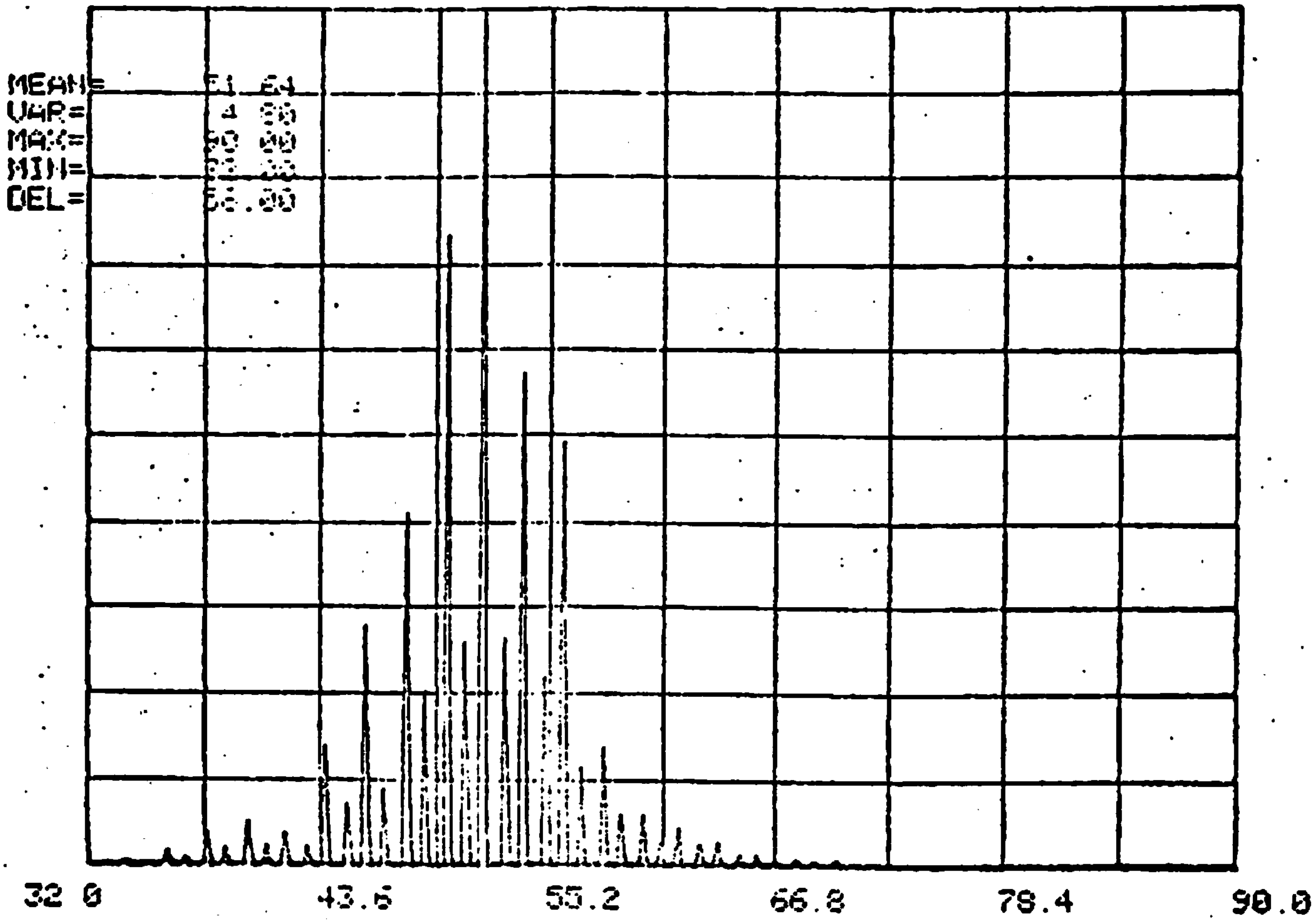


Figure 3.3-1. Original histogram, Band 4.

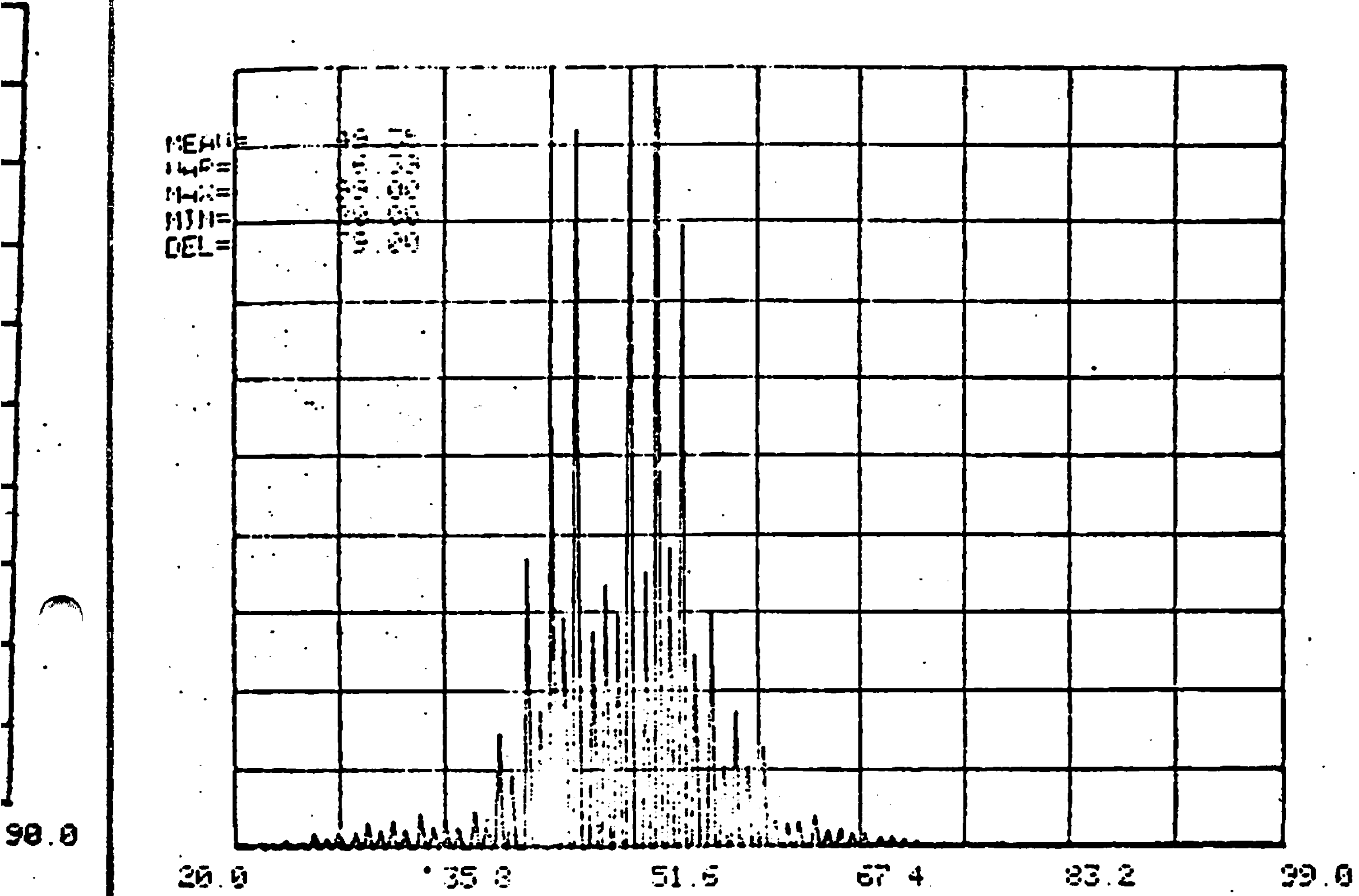


Figure 3.3-2. Original histogram, Band 5.

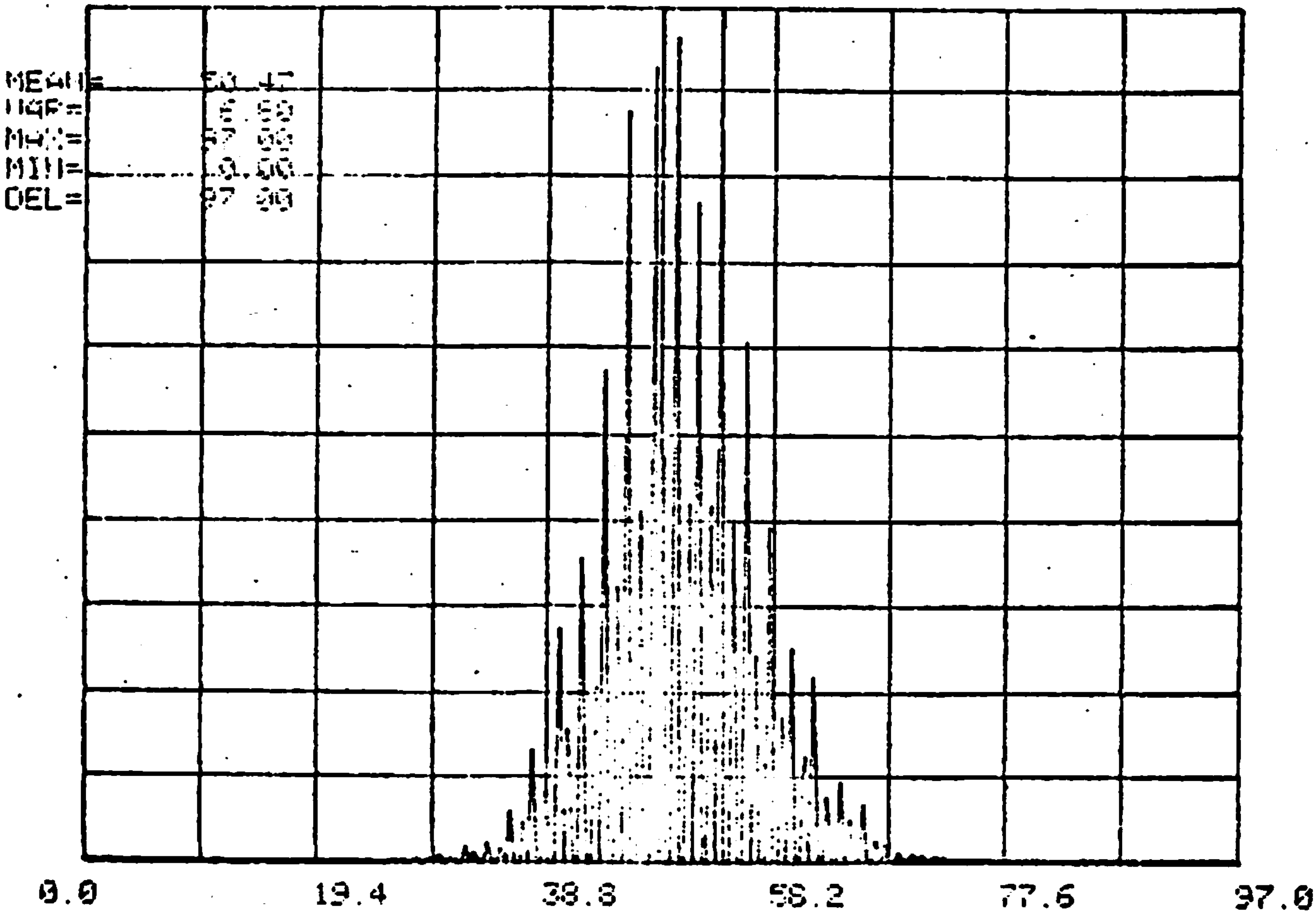


Figure 3.3-3. Original histogram, Band 6.

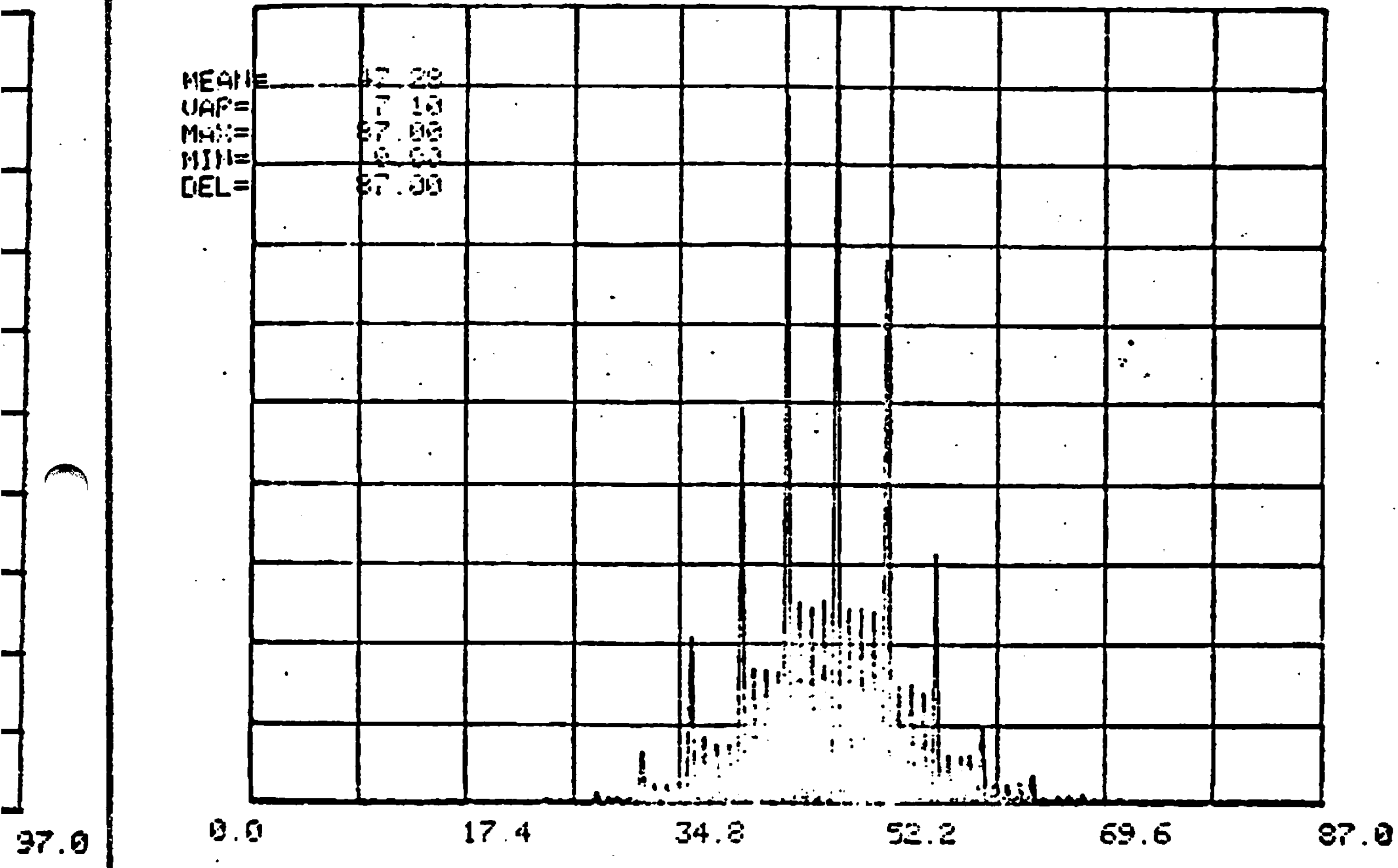


Figure 3.3-4. Original histogram, Band 7.

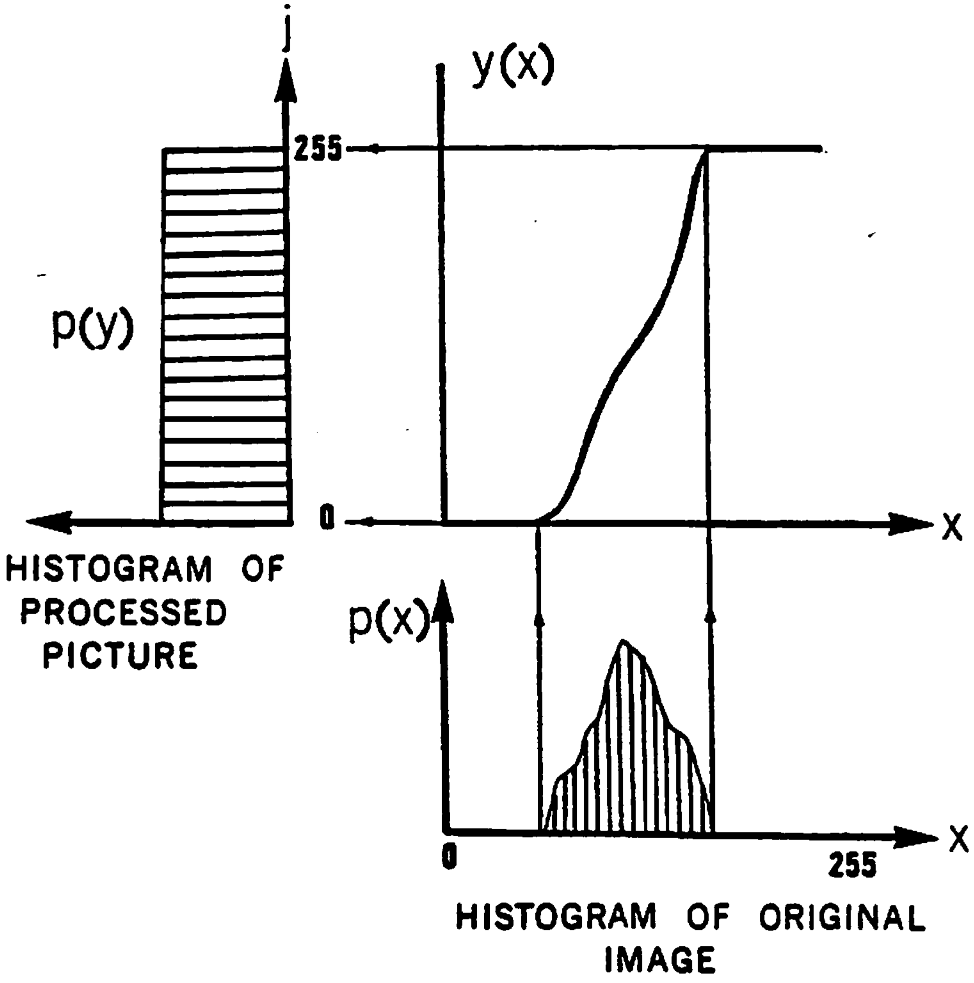


Figure 3.3-5. Contrast enhancement by histogram equalization

b) Accumulate the histogram to create the transfer function

y = F(x)

$$y(x_n) = \sum_{i=0}^{n} p(x_i)$$

c) Normalize y(x) such that y(255) = 255.

d) Store the normalized values of y(x) and use as a look-up table. Because the data is discrete rather than continuous, perfect histogram equalization cannot be accomplished. What happens is that the levels of the original picture are redistributed over the whole range (0...255) available. The histograms of processed pictures are therefore not perfectly uniform but have levels with zero occurrence (Fig. 3.3-6). One possible drawback of this operation is contouring, a phenomenon known in the context of coarse image quantization. This problem has however not been noticeable at all in every single picture processed here.

Besides these classical techniques, an improved histogram manipulation scheme has been designed for images that have to be analyzed by a human observer. Noting that the human visual sensitivity is proportional to the logarithm of image intensities, a histogram modification algorithm has been developed which matches the image amplitude distribution with the visual sensitivity function (Fig. 3.3-7). It is shown that this is accomplished by the creation of a hyperbolic amplitude distribution.

Consider the logarithmic-like response of the visual photo-receptors which can be approximated by:

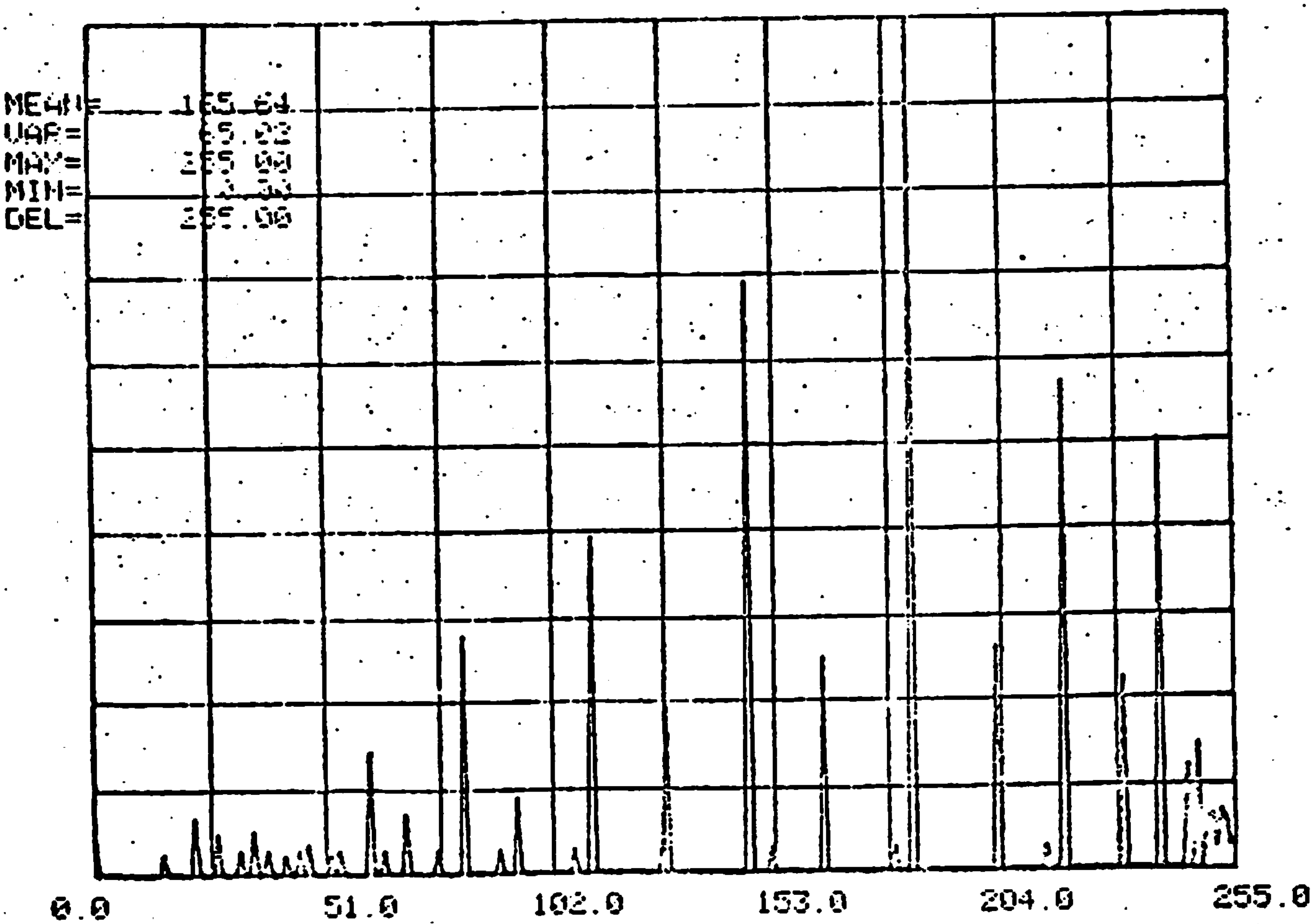


Figure 3.3-6. Equalized histogram, Band 4.

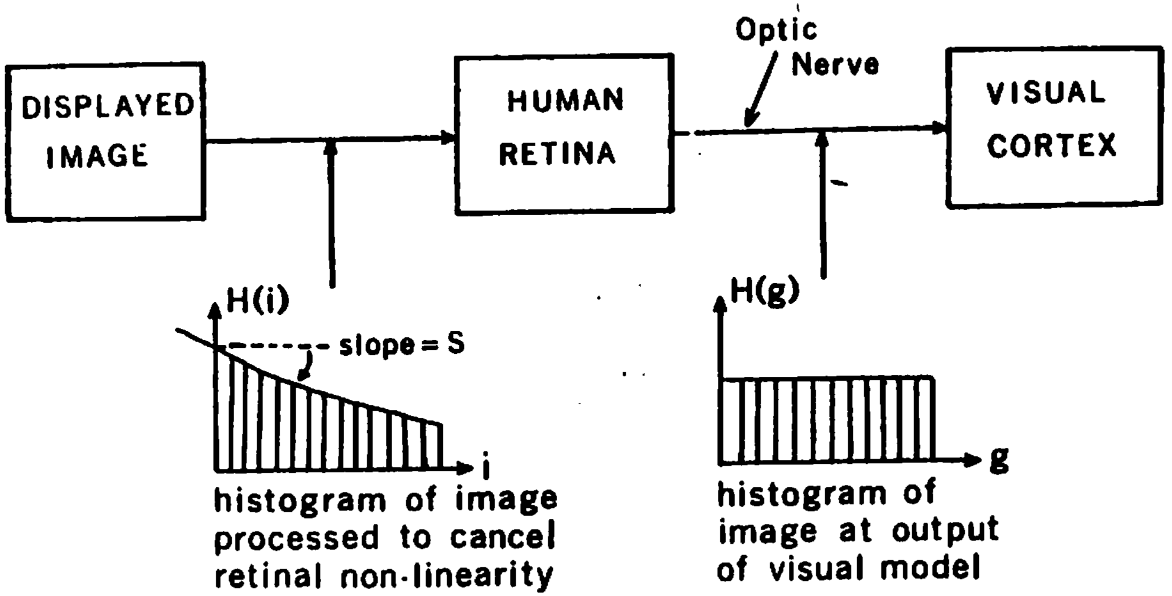

Optic
Nerve
DISPLAYED IMAGE
HUMAN RETINA
VISUAL CORTEX
H(i)
slope = S
i
histogram of image processed to cancel retinal non-linearity
H(g)
g
histogram of image at output of visual model

Figure 3.3-7. Visual system model enhancement

$$g(x,y) = c_1 \log(I(x,y) + c_2) \tag{4}$$

and assume for convenience that I is normalized to $(0 < I < 1)$; a monotonically increasing function J(I) is sought such that the probability p(g) is constant, e.g.

$$p(g) = \frac{1}{g_{max} - g_{min}} \tag{5}$$

with $g_{max} = c_1 \log(1+c_2)$ and $g_{min} = c_1 \log(c_2)$. The cumulative distribution F(g) is

$$F(g) = \int_{g_{min}}^{g} p(g)dg = \frac{g - g_{min}}{g_{max} - g_{min}} \tag{6}$$

and the unknown distribution F(J) is

$$F(J) = \int_{0}^{J} p(J)dg = \int_{g_{min}}^{g(J)} p(g)dg \tag{7}$$

Therefore

$$F(J) = \frac{g(J) - g_{min}}{g_{max} - g_{min}} = \frac{\log\left(\frac{J + C_2}{C_2}\right)}{\log\left(\frac{1 + C_2}{C_2}\right)} \tag{8}$$

and the desired density function p(J) is

$$p(J) = \frac{d}{dJ} F(J) = \frac{1}{(J + C_2)\log\left(\frac{1 + C_2}{C_2}\right)} \tag{9}$$

Equation (9) represents a hyperbolic amplitude density function. The unknown parameter c reflects the neural activity g(0) evoked by the minimum displayed intensity and depends upon ambient lighting in the case of a television display.

A more convenient parameterization of p(J) may be obtained by considering the slope s of p(J) $(0 < J < 1)$ at the origin $J = 0$

$$S = \frac{d}{dJ} p(J)\bigg|_{J=0} = \frac{-1}{C_2^2 \log\left(\frac{1 + C_2}{C_2}\right)} \quad , \quad (10)$$

Figure 3.3-8 demonstrates the effect of histogram hyperbolization. Figure 3.3-9a is the original 512 x 512 section of the original ERTS image (band IV), after the geometric corrections. Figure 3.3-9b has been histogram equalized according to the classical procedure and Figure 3-3.9c and 3.3-9d show the effect of histogram hyperbolization, with s = -0.5 and s = -1.0, respectively. Computationally, histogram manipulation algorithms typically make use of a look-up table. Histogram hyperbolization is therefore as fast as equalization, a very economical scalar enhancement technique. Figure 3.3-10 shows the other three bands of the same data base processed in this fashion. Because this technique appears to produce the best visual display of the present type imagery, histogram hyperbolization has been used as a display preprocessing step in most operations described thereafter.

3.4 Banding Elimination

One unique feature of the ERTS satellites is the scanning arrangement, which consists of four banks of six sensors each. An oscillating mirror images a strip of land on each bank of sensors, such that six lines of data are effectively scanned simultaneously during each sweep of the mirror. The rotation axis of the mirror is parallel to the line of flight so that the next sweep provides the next six lines of each spectral image

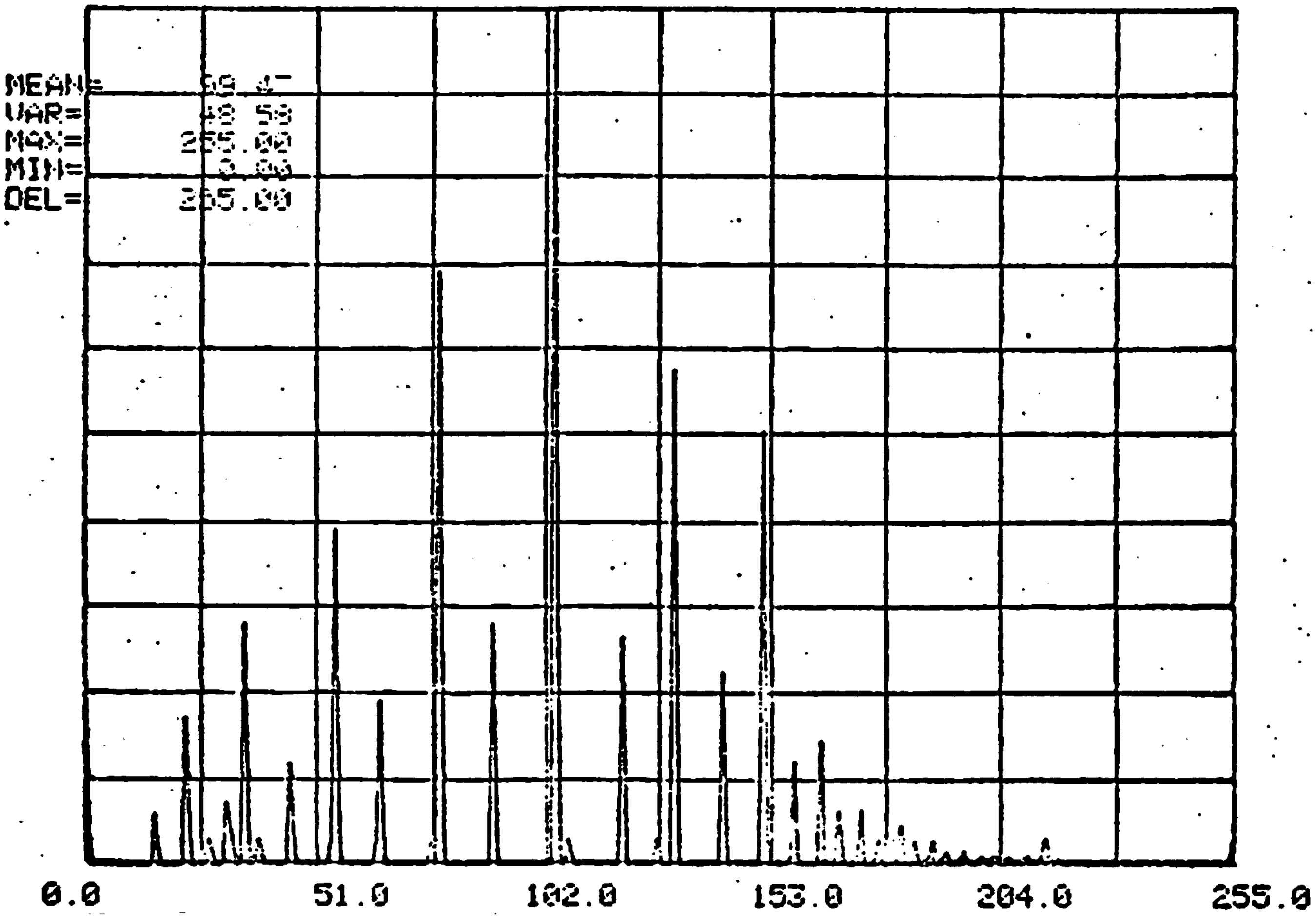
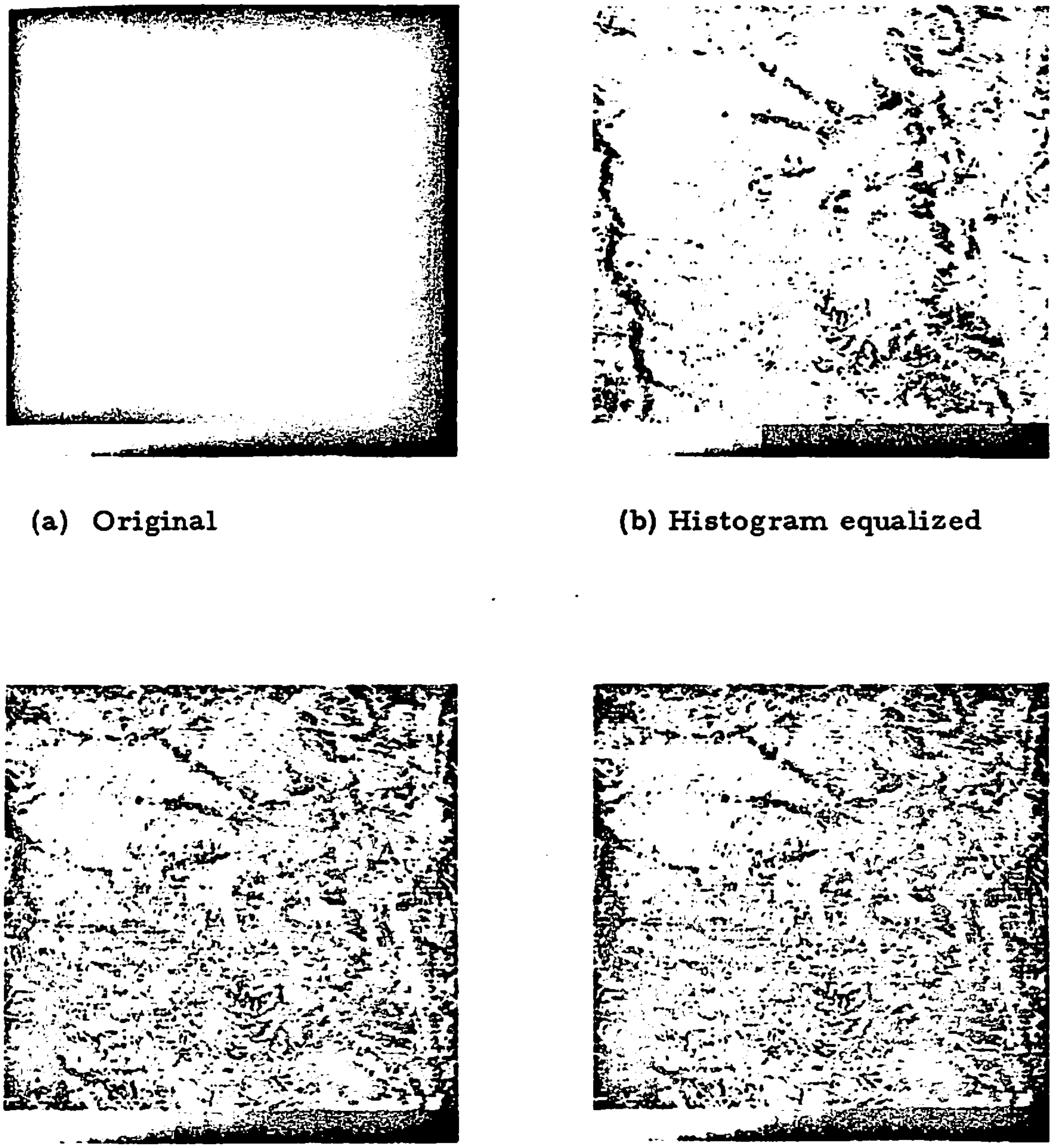


Figure 3.3-8. Hyperbolized histogram (S=-.5), Band 4.



(a) Original

(b) Histogram equalized

(c) Histogram hyperbolized S=-0.3

(d) Histogram hyperbolized S= -0.5

Figure 3.3-9. Histogram modification.

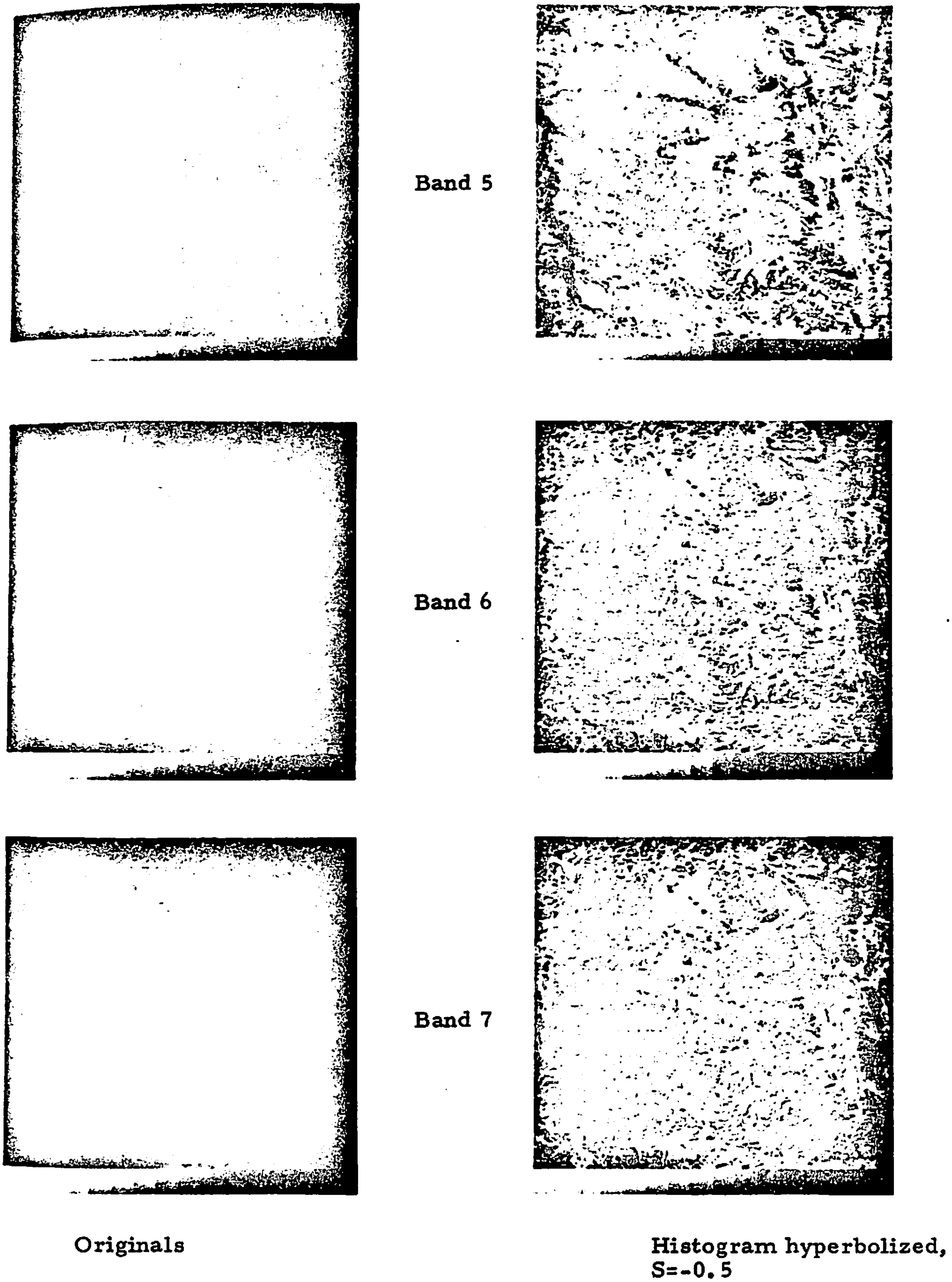
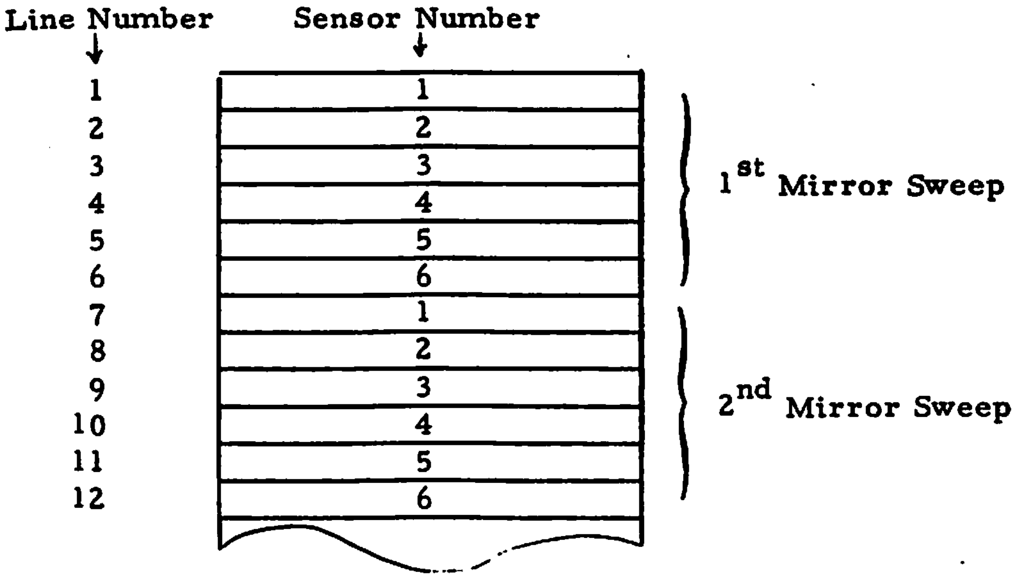


Figure 3.3-10. Histogram hyperbolization.



Line Number	Sensor Number
1	1
2	2
3	3
4	4
5	5
6	6
7	1
8	2
9	3
10	4
11	5
12	6

Figure 3.4-1.

(Fig. 3.4-1), etc. Unfortunately, the sensors and amplifiers are not perfectly matched, so that an artificial "band" structure is apparent in typical ERTS images, especially after image enhancement. For example, suppose that the sensor #2 has a smaller gain than the other sensors; as a consequence, the lines #2, 8, 14, etc. will appear darker throughout the picture and thus a "banding" structure is formed. One approach to cope with this artifact has been to take the Fourier transform of the image in a direction perpendicular to the scanning path (e.g. columnwise). Since the banding is strictly periodic, its energy is concentrated at harmonics of 6 lines. Spectrum interpolation can then be used to suppress excess energy at those frequencies. An alternative solution has been developed to combat the banding artifact, which is computationally much faster. It is based upon the zero-order amplitude distribution of the image data corresponding to each individual sensor. Let p(x) be the probability that the ground radiance has a value x and $y_i(x)$ $(i = 1, \ldots, 6)$ the transfer function of each sensor and amplifiers. The probability of an amplitude y in the data is then

$$p(y) = \frac{d}{dy}\left(\int_0^{x_i(y)} p(x)dx\right)$$

where $x_i(y)$ is the inverse of $y_i(x)$. Note that we have to identify each line of data with the appropriate sensor. That is easily done by letting the index i be equal to the line index, modulo 6

$$i = (\text{line index})_{\text{mod } 6}$$

Let us call the sets of lines counted modulo six "subimages" (e.g. the lines 1.7, 13, etc. would form the subimage no.1, lines 2, 8, 14, etc. subimage no. 2, etc.). The differences between sensor non-linearities, amplifier gains, offsets, etc. can now be theoretically eliminated by transforming each subimage separately such that the resulting histograms are all identical. For example one can equalize or hyperbolize the histograms of all subimages (see previous section). Figure 3.8-1 is an example of this banding suppression technique. Note that the correction cannot be absolutely perfect, because the data is discrete. Histogram manipulation does not give perfectly smooth distributions as discussed in the previous section, and therefore the sensor inequalities cannot be completely eliminated.

3.5 Spatial Filtering and Homomorphic Filtering

Spatial high-pass filtering is an appropriate tool to enhance fine image detail at the expense of smooth, large area variations such as shading due to illumination variations. Because the energy of such features is mainly concentrated in the higher spatial frequencies, spatial high-pass filtering tends to emphasize relevant image information. In addition, power spectrum estimates of images reveal that most of the energy is typically concentrated at the lower end of the spectrum. A much more efficient use of the dynamic range of the display can thus be obtained if low spatial frequencies are de-emphasized.

Considering that image intensities are a product of illumination and reflectance, homomorphic filtering [1] can be used to separate the typical low-frequency illumination components from the high-frequency reflectance components (Fig. 3.5-1). The cascade of logarithmic receptors and linear inhibitions in the visual system actually perform a very similar operation [2], Fig. 3.5-2.

The advantages of high-pass homomorphic filtering can be summarized as follows:

1) Enhancement of relevant object reflectance features.

2) De-emphasis of less important illumination components.

3) Better utilization of the display dynamic range.

Examination of the histograms of high-passed images reveals, however, a Gaussian shaped distribution of amplitude levels. A modification of homomorphic filtering is therefore proposed, in which the filtered image is subjected to histogram hyperbolization before it is displayed.

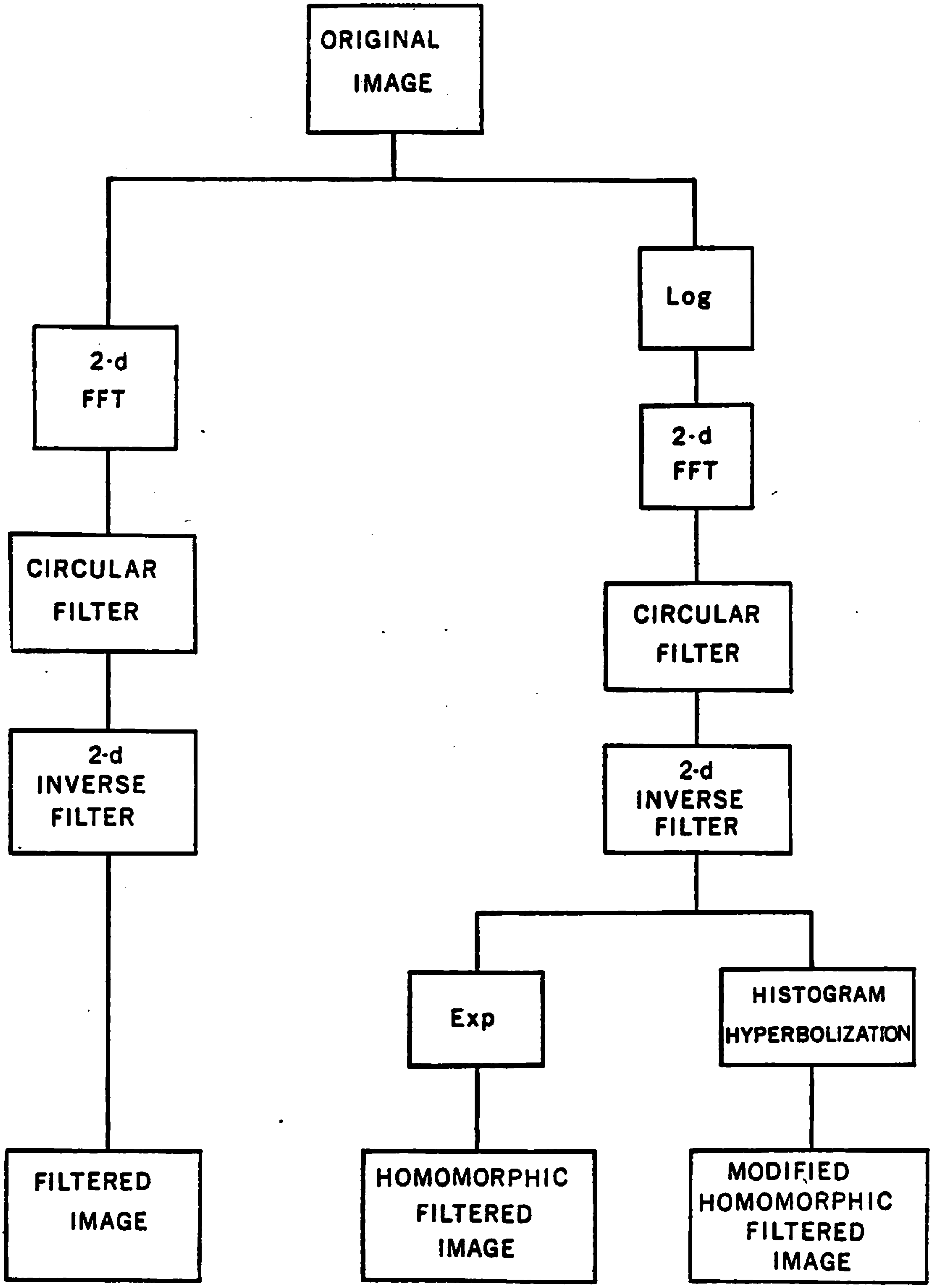


Figure 3.4-5. Spatial filtering

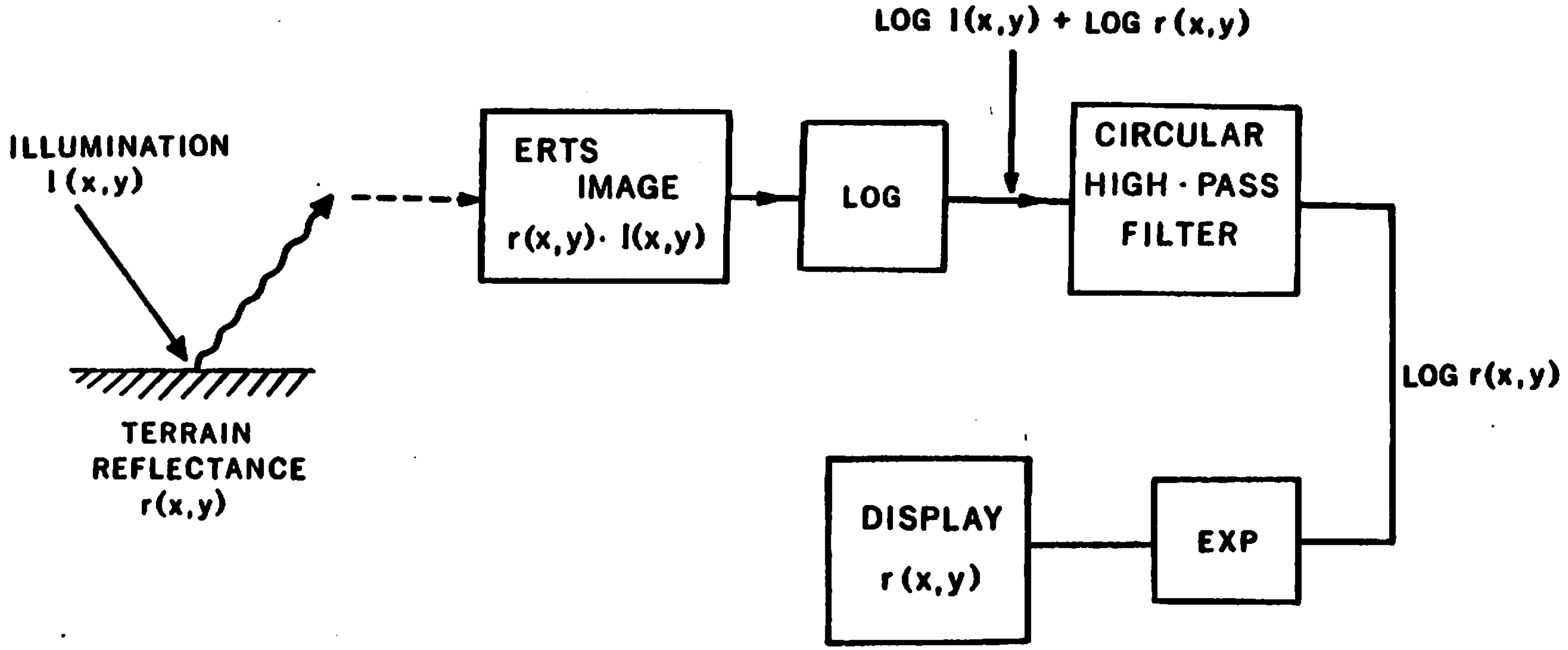


Figure 3.5-1. Homomorphic filtering

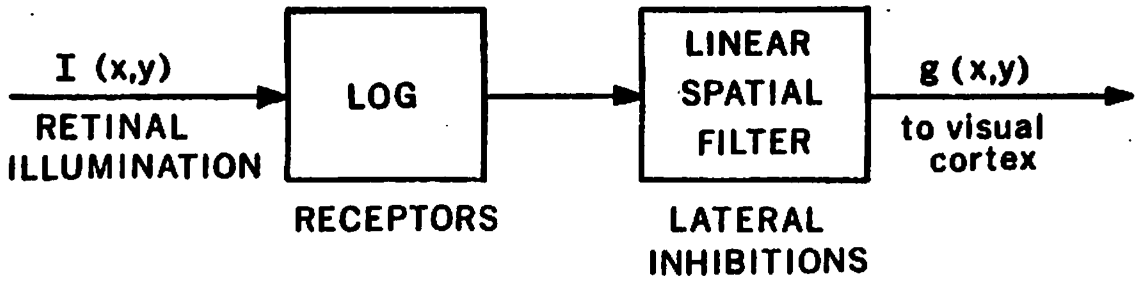
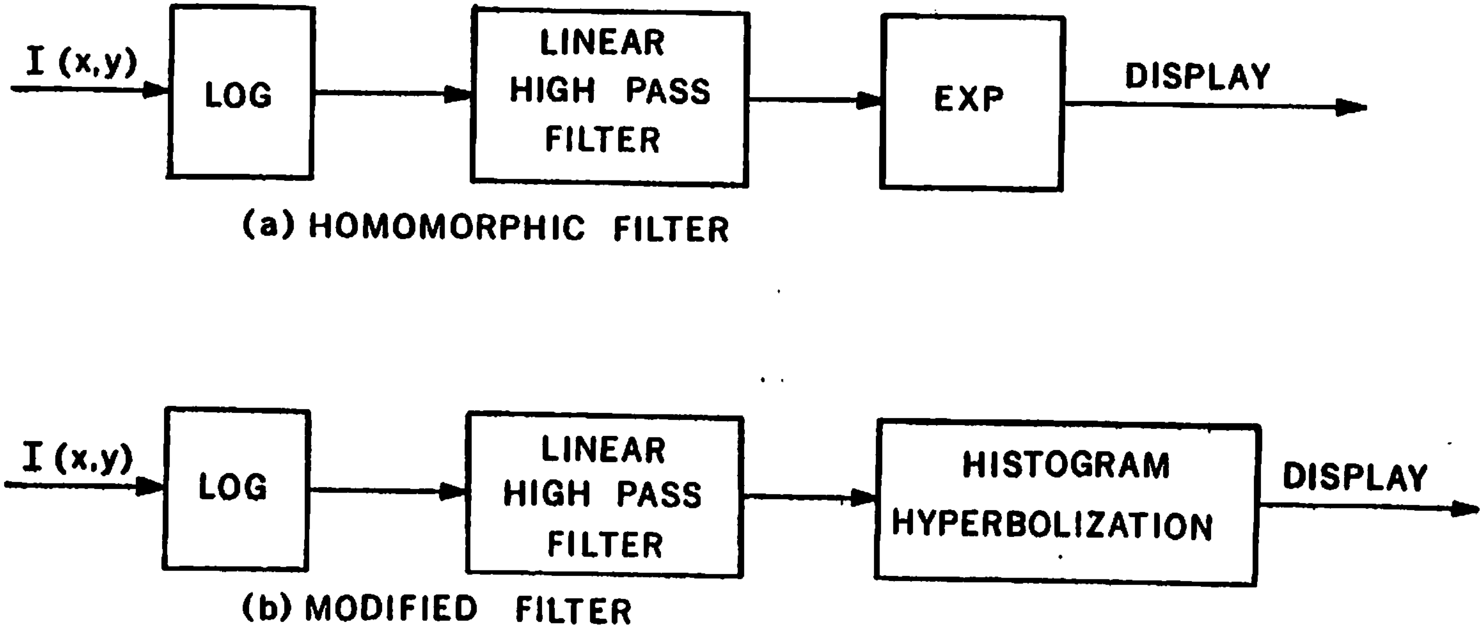


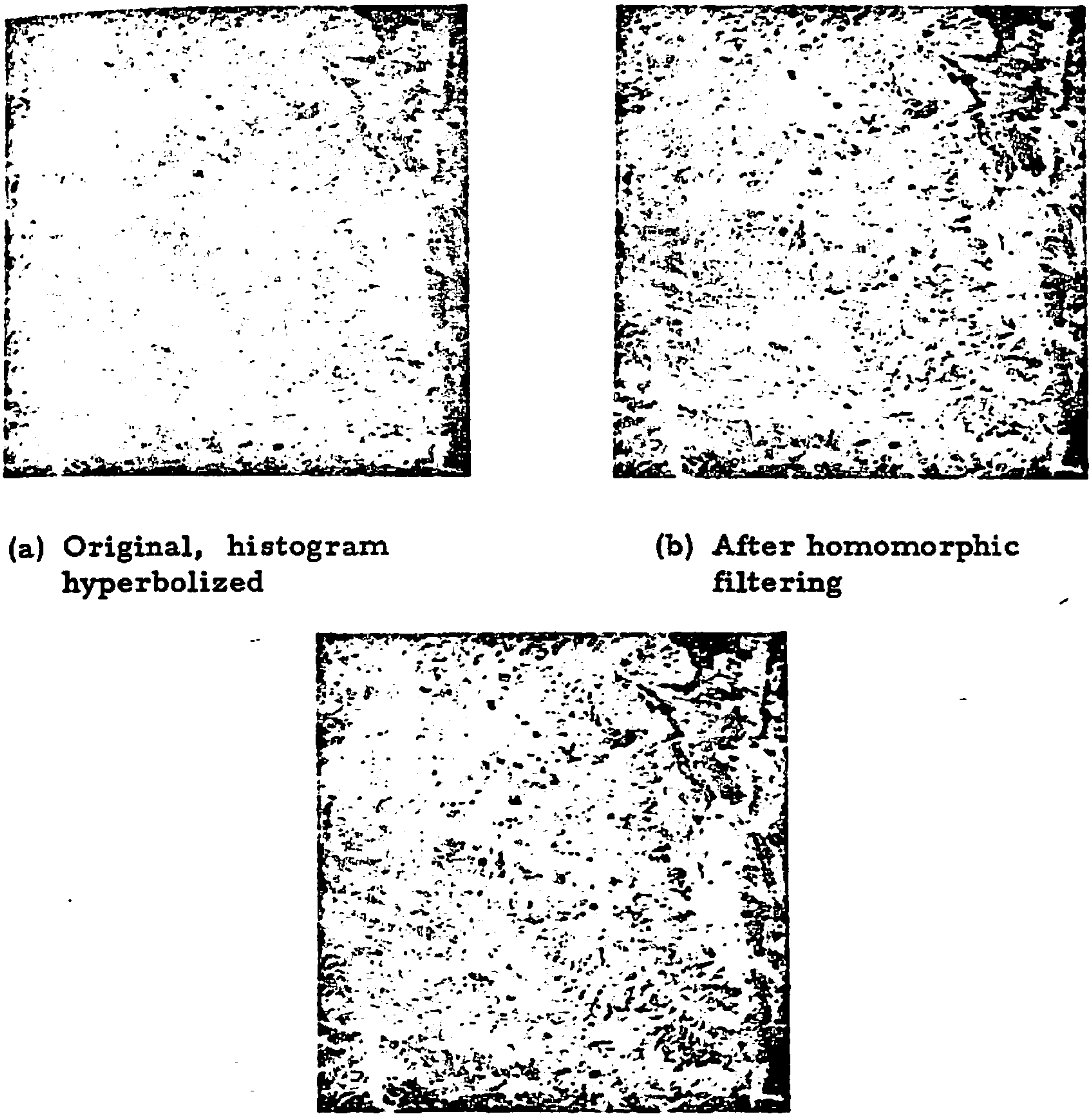
Figure 3. 5-2. Simplified visual model



(a) HOMOMORPHIC FILTER

(b) MODIFIED FILTER

Figure 3.5-3. Non-linear high-pass filtering.



(a) Original, histogram hyperbolized

(b) After homomorphic filtering

(c) S = -0.5

(c) After homomorphic filtering and histogram hyperbolization.

Figure 3.5-4. Spatial filtering

This innovation is depicted in Fig. 3.5-3. This procedure combines the advantages of homomorphic filtering and histogram hyperbolization as discussed in section 3.3. Figure 3.5-4b is a homomorphic highpass version of band 7, shown histogram enhanced in Figure 3.5-4a. Figure 3.5-4c was processed according to the diagram of Fig. 3-5-2b, with $s = -0.5$. Note that no extra computations are required since the hyperbolic transformation replaces the exponentiation of the homomorphic filter.

Computationally, all circular filtering operations are relatively expensive however. It involves taking the two-dimensional Fourier transform of the image and its inverse as depicted in Fig. 3.5-5. Typical execution times for a transform are approximately 1.5 and 5 minutes for 256 x 256, respectively 512 x 512 pixel images on the DEC-PDP KI10 of the image processing institute. Faster edge enhancement is achieved by crispening and finite size convolutions described in the next section.

References

[1] T. G. Stockham, Jr., "Image Processing in the Context of a Visual Model," Proc. of the IEEE, Vol. 60, July, 1972, pp. 828-1309.

[2] T. N. Cornsweet, Visual Perception, Academic Press, N.Y., 1970.

3.6 Crispening

Crispening is a term borrowed from the field of television. In brief, this technique is used in cameras and flying spot scanners to give television pictures a sharper appearance. This is done by enhancing abrupt variations of the video signal, and hence edges of the picture.

One implementation of crispening is depicted in Fig. 3.6-1, where the second derivative of the video signal is added to the signal itself, accentuating edges. This operation can be approximated with a great deal of flexibility by digital image processing. The numerical implementation of crispening is done by finite convolution, schematized in Fig. 3.6-2a. In one dimension, the output picture $I'(x_i)$ is related to the input picture $I(x_i)$ by

$$I'(x_i) = \sum_{j=-J}^{+J} a_j I(x_{i-j}) \qquad (1)$$

Figure 3.6-2b shows a typical one-dimensional weighting function. The terms a_j determine the amount of edge crispening which can be measured by the percentage "overshoot" for a step function (Fig. 3.6-2c). This operation can be easily generalized to two dimensions in the digital case and equation (1) becomes

$$I'(x_i, y_i) = \sum_{k=-K}^{+K} \sum_{l=-L}^{+L} a_{ij} \cdot I(x_{i-k}, y_{j-l})$$

The two-dimensional finite aperture a_{ij} used here is depicted in Fig. 3.6-3, with a % "overshoot" defined as in Fig. 3-6-2c. Figure 3.6-4

$$\begin{array}{cccccc}
 & & j & & & \\
 & a_{-1,2} & a_{0,2} & a_{1,2} & & \\
a_{-2,1} & a_{-1,1} & a_{0,1} & a_{1,1} & a_{2,1} & \\
a_{-2,0} & a_{-1,0} & a_{0,0} & a_{1,0} & a_{2,0} & i \\
a_{-2,-1} & a_{-1,-1} & a_{0,-1} & a_{1,-1} & a_{2,-1} & \\
 & a_{-1,-2} & a_{0'-2} & a_{1,-2} & &
\end{array}$$

Figure 3.6-3

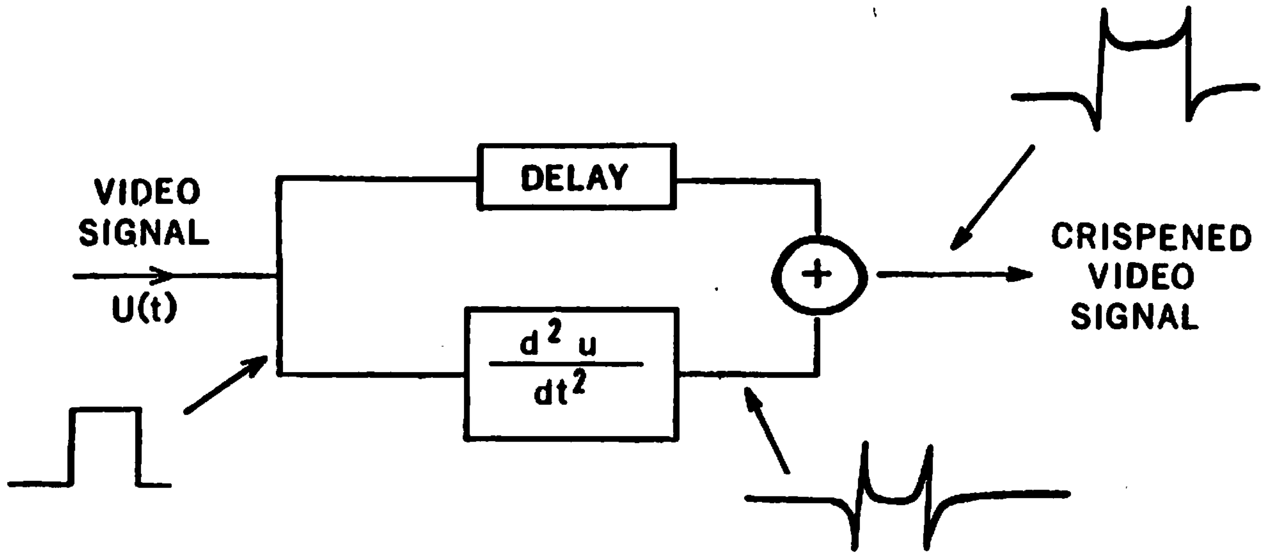


Figure 3.6-1.

Figure 3.6-2a

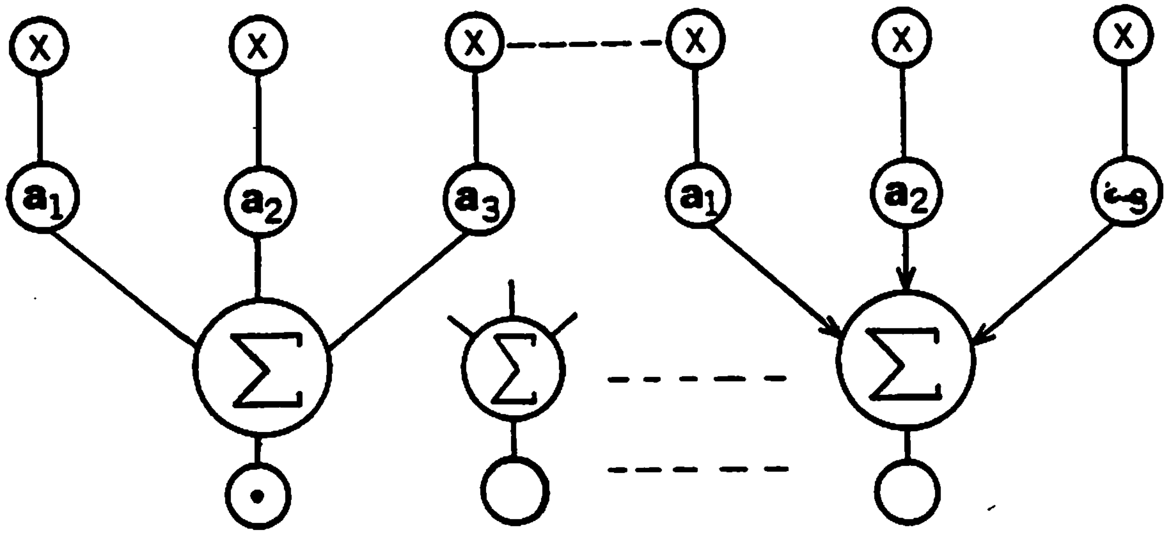


Figure 3.6-2b

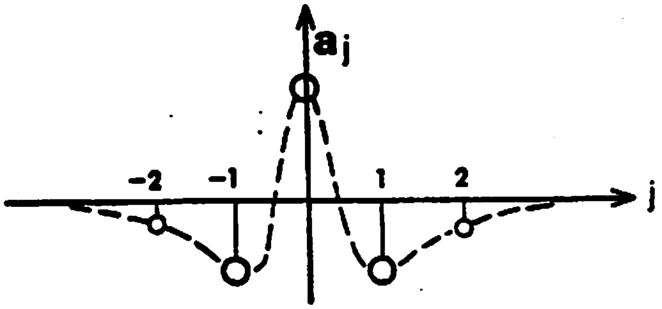
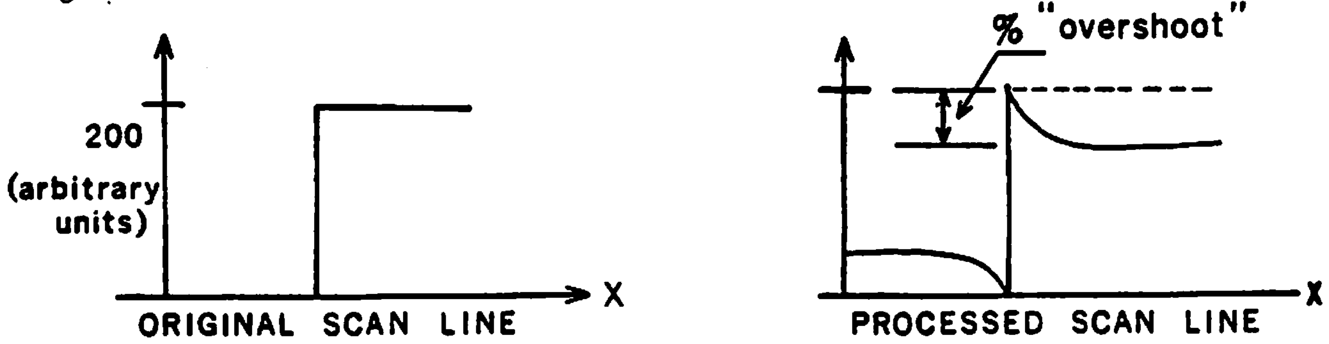
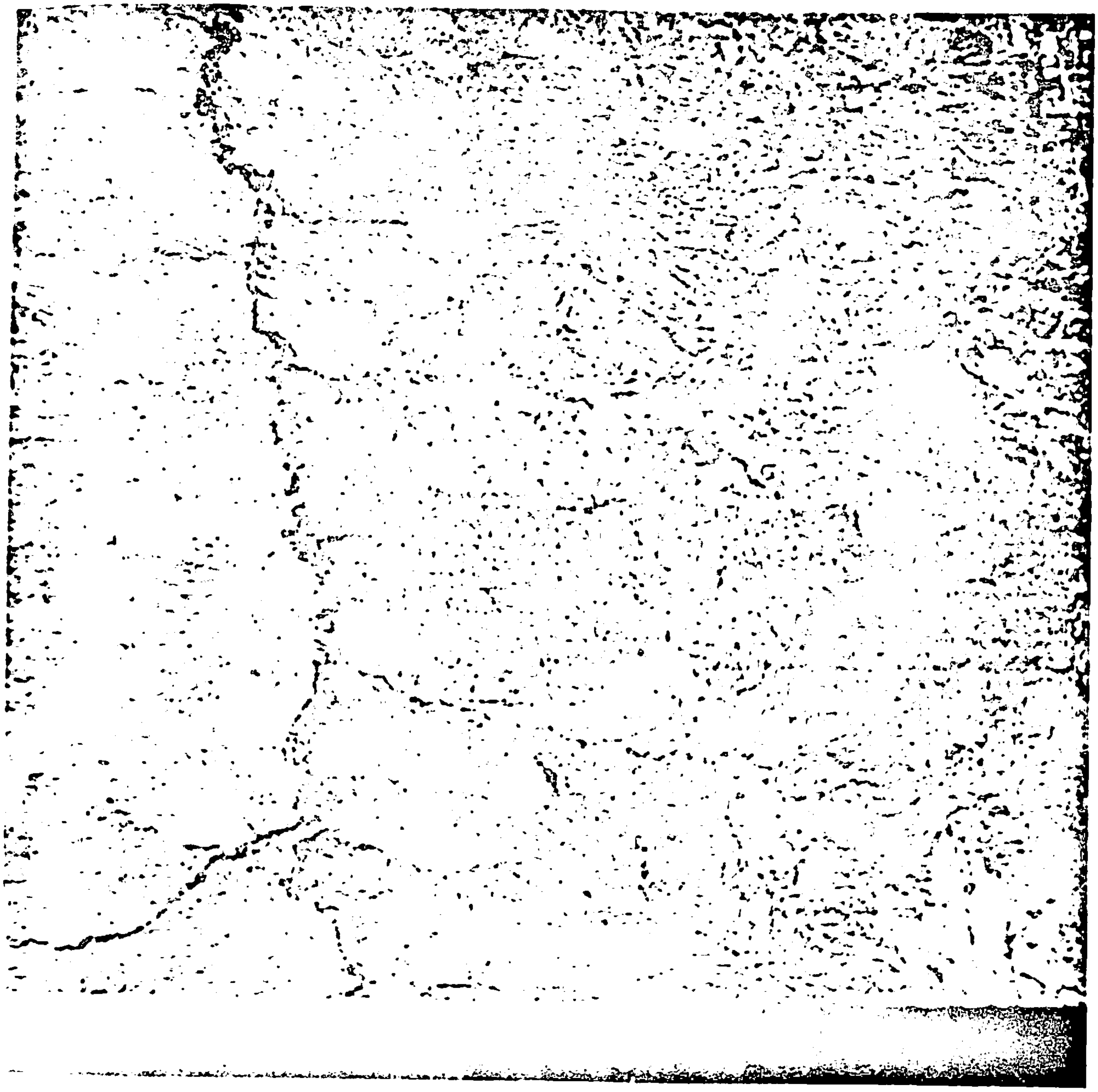


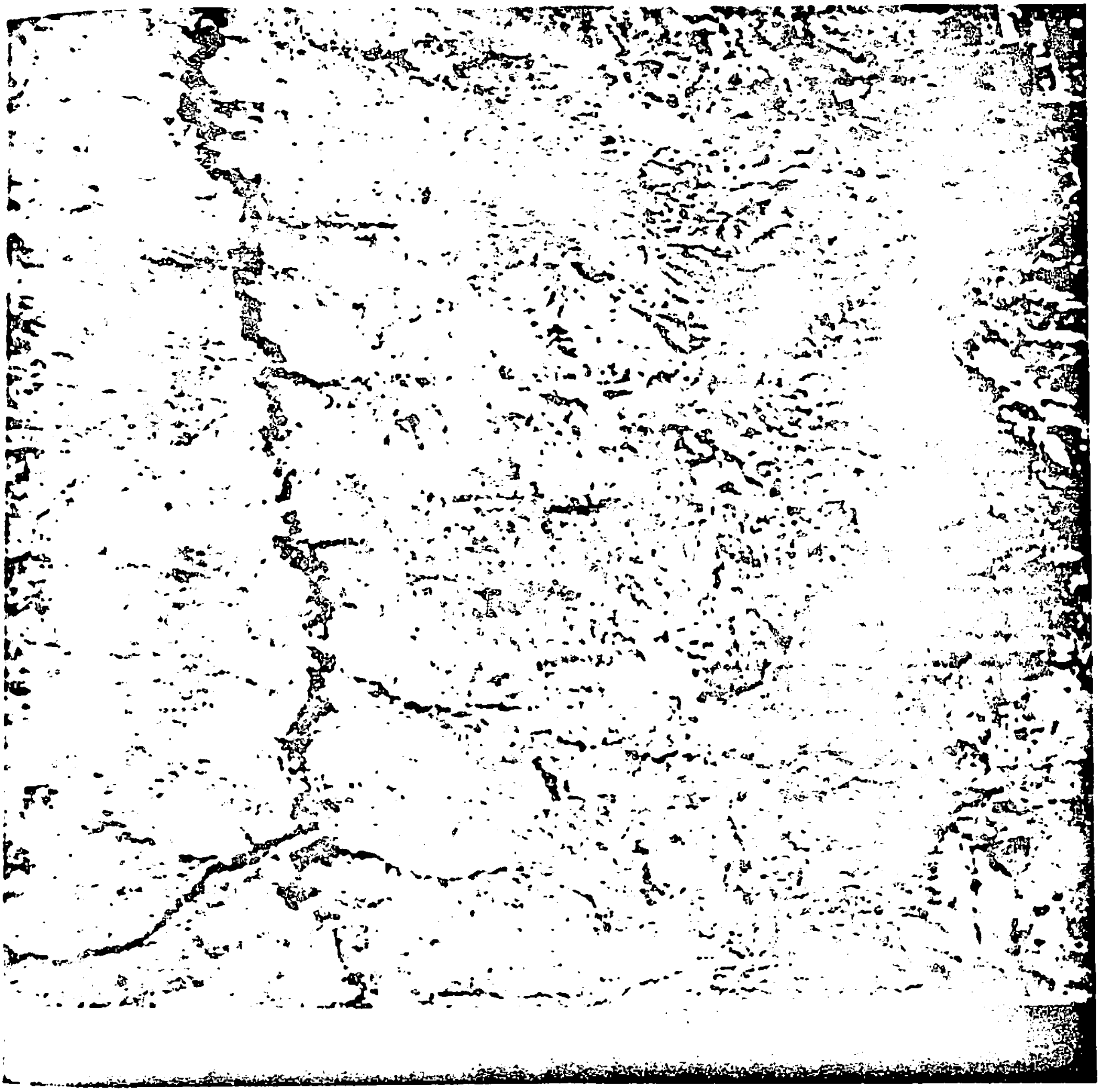
Figure 3.6-2c





80% Overshoot

Figure 3.6-4a. Example of edge crispening.



20% Overshoot

Figure 3.6-4b. Example of edge crispening.

shows two examples of two-dimensional crispening or finite convolution applied to ERTS images. In the first case an "overshoot" factor of 20% was used and in the second case 80%. Note that this latter example tends to suppress large area contrast in favor of edges. This technique can be modified somewhat to provide an edge extraction algorithm by introducing appropriate threshold decisions.

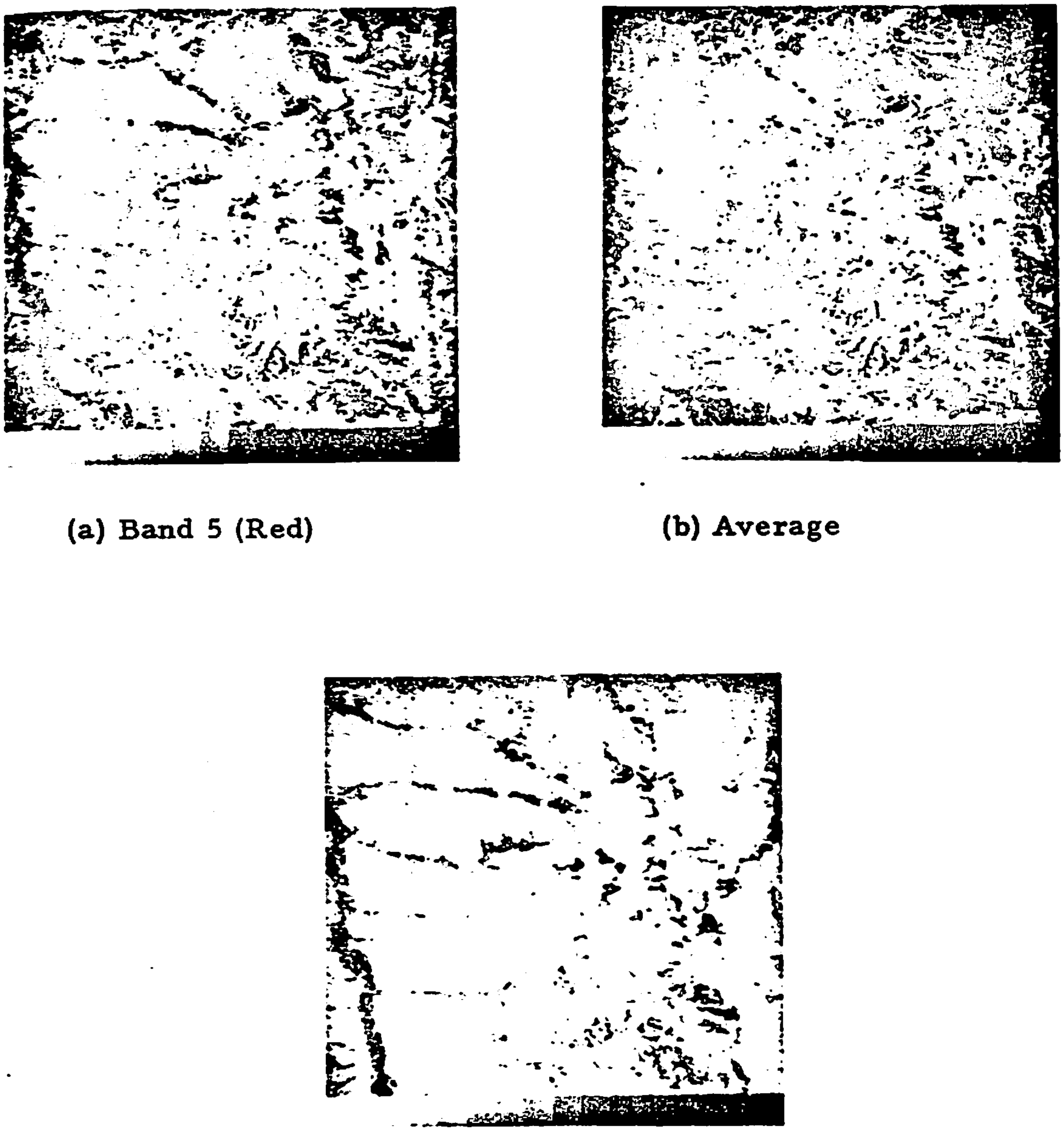
3.7 Band Ratios and Log Band Ratios

3.7.1 Introduction

Ratioing of ERTS pictures is a useful preprocessing technique for multispectral recognition and classification. Signatures obtained from a training sample under one set of conditions may not have a good discrimination capability for a given classification scheme if the same area is observed under a different set of conditions. If the changes result from simple multiplicative factors such as the brightness level, then the ratio of the bands would be invariant.

Taking various ratios of the green, red and the two infrared bands (bands 4, 5, 6 and 7, respectively) of the ERTS data results in eliminatim of brightness variations due to topographic relief. Such ratio images have been shown to be more useful for determining boundaries between lithologic units and vegetation groups [1].

Ratios may be taken to emphasize variations due to color also. Figure 3.7-1 and 3.7-2 show the ratios red and the far infrared bands to the average of all four bands. Such ratioing process produces a color

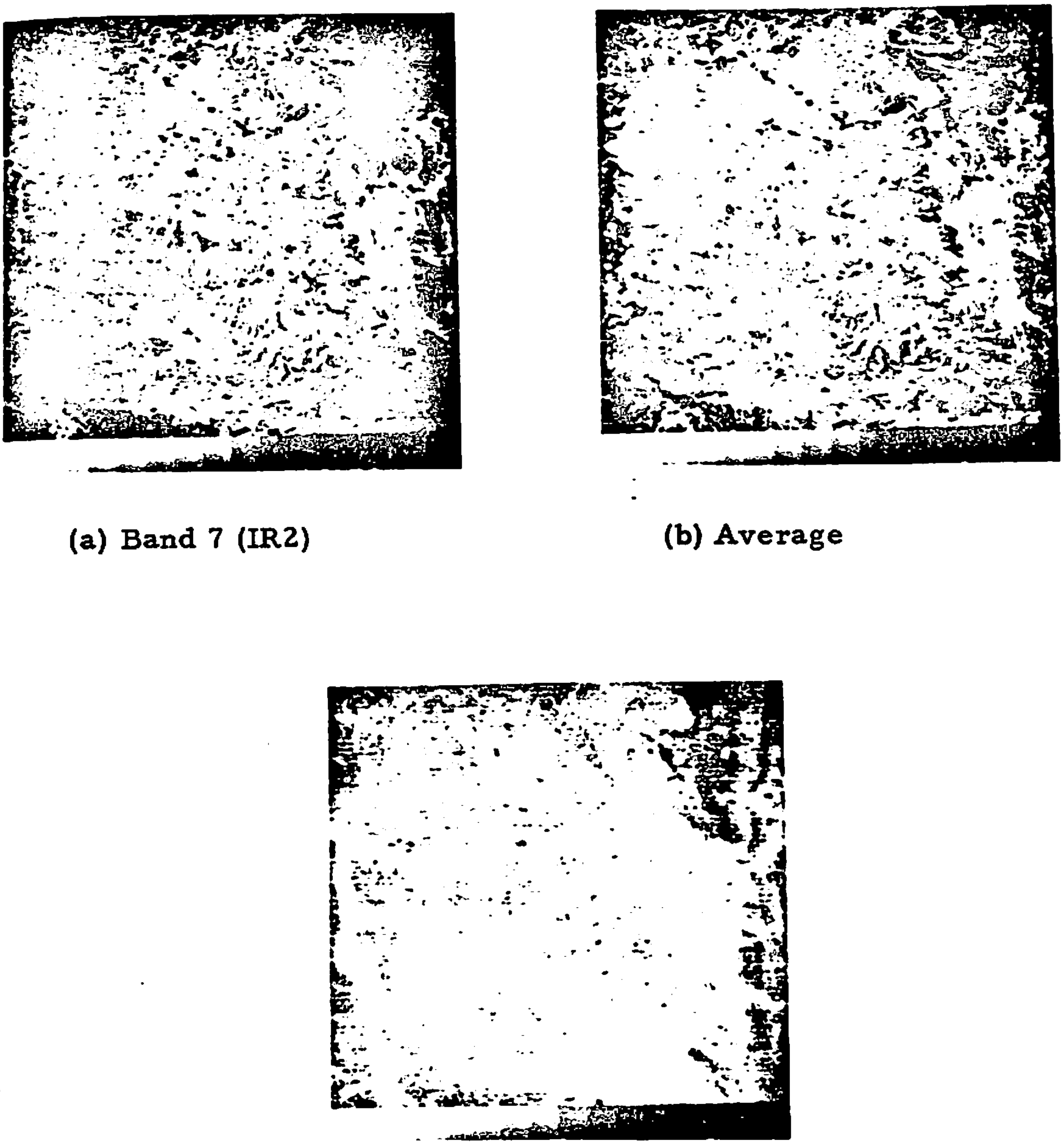


(a) Band 5 (Red)

(b) Average

(c) Ratio of Band 5/Average

Figure 3.7-1. Example of Band Ratios



(a) Band 7 (IR2)

(b) Average

(c) Ratio of Band 7/Average

Figure 3.7-2. Example of Band Ratios

display whose color variations are more indicative of material variations than the simple pseudocolor displays.

3.7.2 Direct Ratios and Logarithmic Ratios

The ratios shown in Figures 3.7-1 and 3.7-2 are direct ratios, obtained by forming a scaled ratio of two bands. Figures 3.7-3 shows direct ratios of ERTS bands 4, 5, 6 and 7. This figure shows the effect of banding present in ERTS images reflects in ratio images.

Figures 3.7-4 and 3.7-5 show the direct and logarithmic ratios of a different ERTS picture after banding correction. The advantage of logarithmic ratios becomes apparent if we consider that the original spectral images and the computed ratio image are quantized, to eight bits in the present case. Let a and b be the pixel values of bands A and B, ranging from 1 to 256 (to avoid division by zero) and the value of the ratio pixels. The maximum quantization error of a and b is ±0.5. Taking extreme values, the quantization error of the bands A and B is reflected in the output as:

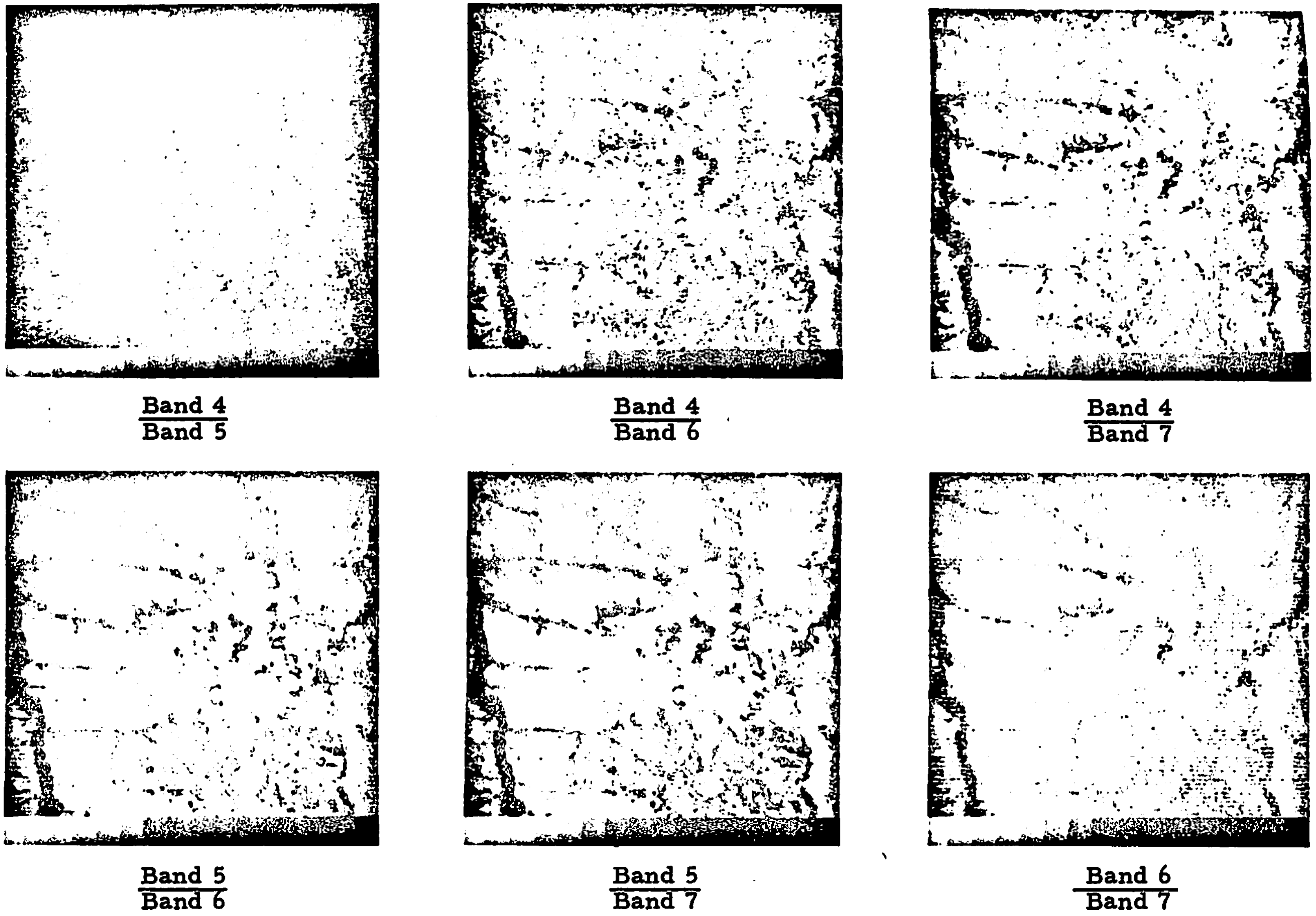
$$r_{max} = \frac{256 \pm .5}{1 \pm .5}$$

$$r_{min} = \frac{1 \pm .5}{256 \pm .5}$$

$$|\Delta r_{max}| = \max(r_{max}) - \min(r_{max}) = 342.67$$

$$|\Delta r_{min}| = \max(r_{min}) - \min(r_{min}) = 3.92 \times 10^{-3}$$

The quantization error of a and b therefore affects the ratio image greatly if r is large and to a negligible extent if r is small. Taking



$\frac{\text{Band 4}}{\text{Band 5}}$

$\frac{\text{Band 4}}{\text{Band 6}}$

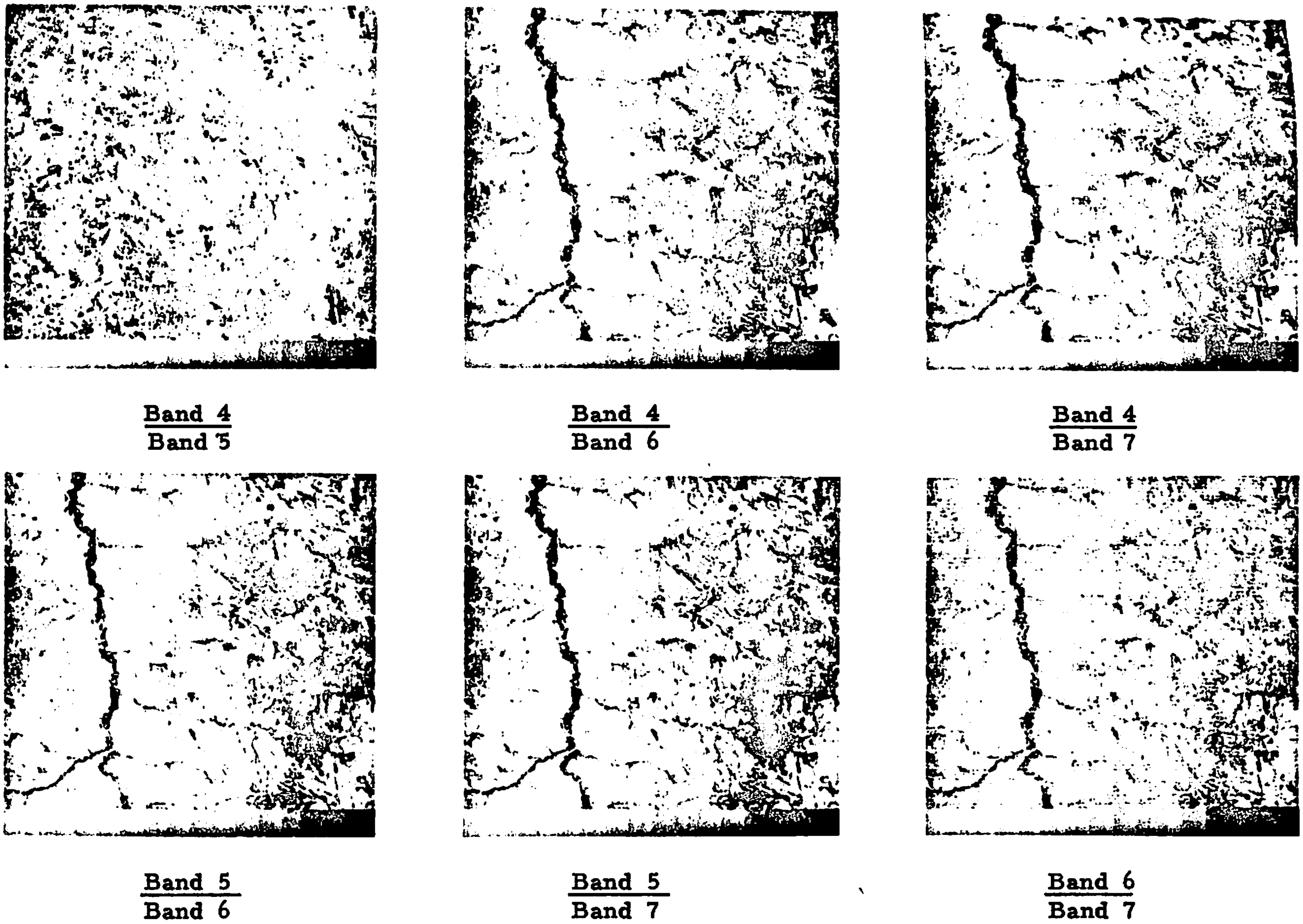
$\frac{\text{Band 4}}{\text{Band 7}}$

$\frac{\text{Band 5}}{\text{Band 6}}$

$\frac{\text{Band 5}}{\text{Band 7}}$

$\frac{\text{Band 6}}{\text{Band 7}}$

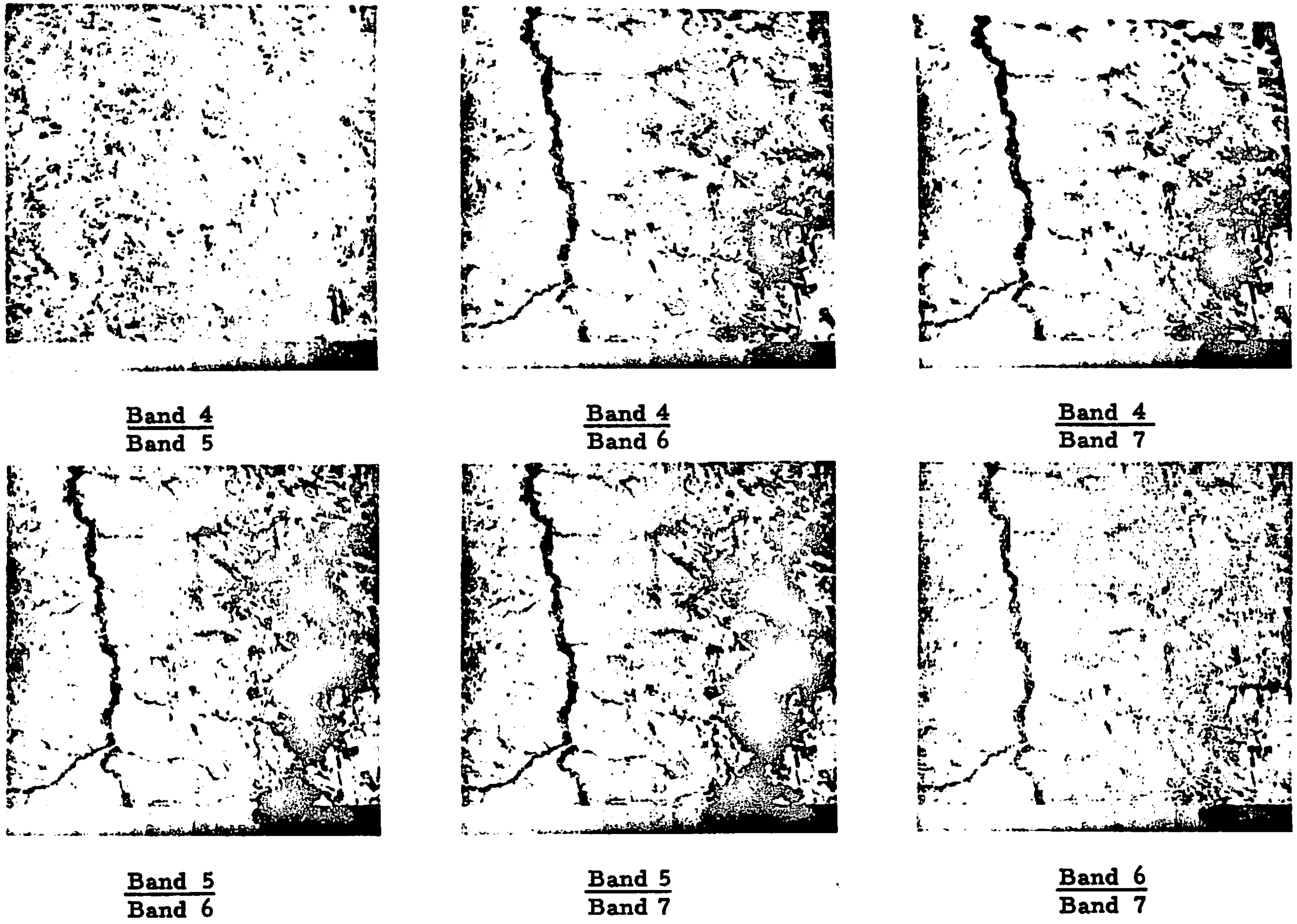
Figure 3.7-3. Direct Ratios



$\frac{\text{Band 4}}{\text{Band 5}}$ $\frac{\text{Band 4}}{\text{Band 6}}$ $\frac{\text{Band 4}}{\text{Band 7}}$

$\frac{\text{Band 5}}{\text{Band 6}}$ $\frac{\text{Band 5}}{\text{Band 7}}$ $\frac{\text{Band 6}}{\text{Band 7}}$

Figure 3.7-4. Direct Ratios



$\frac{\text{Band 4}}{\text{Band 5}}$ $\frac{\text{Band 4}}{\text{Band 6}}$ $\frac{\text{Band 4}}{\text{Band 7}}$

$\frac{\text{Band 5}}{\text{Band 6}}$ $\frac{\text{Band 5}}{\text{Band 7}}$ $\frac{\text{Band 6}}{\text{Band 7}}$

Figure 3.7-5: Log Ratios

the logarithm of the ratio and rescaling the log ratio image to the range 0...255 provides a uniform distribution of quantization error throughout the range of r. It is indeed easy to verify that $\Delta r_{max} = \Delta r_{min}$ if r is defined as $r = \log a/b$.

In the cases studied, it has appeared that logarithmic ratio images contain more visual information than direct ratio images. It is felt that experiments with more images are necessary to confirm the above conclusion.

3.7.3 Principal Components of Ratios

The covariance matrix of various ratios could give some insight in choosing a set of ratios for a classification scheme: Ratios that are uncorrelated are likely to produce better results than those that are highly correlated. This idea suggests the use of the principal components of the ratios instead of ratios themselves. Figures 3.7-7 and 3.7-6 show the principal components obtained obtained from logarithmic ratios and direct ratios, respectively.

In the case of direct ratios (scale factor 50.), the normalized covariance matrix has been computed as

$$\begin{array}{cc}
\text{Ratios}\rightarrow & \begin{array}{cccccc} 4{:}5 & 4{:}6 & 4{:}7 & 5{:}6 & 5{:}7 & 6{:}7 \end{array} \\
\downarrow & \\
\begin{array}{c} 4{:}5 \\ 4{:}6 \\ 4{:}7 \\ 5{:}6 \\ 5{:}7 \\ 6{:}7 \end{array} &
\begin{bmatrix}
1.0 & -0.294 & -0.378 & -0.758 & -0.714 & -0.404 \\
-0.294 & 1.0 & 0.905 & 0.824 & 0.786 & 0.475 \\
-0.378 & 0.905 & 1.0 & 0.819 & 0.901 & 0.789 \\
-0.758 & 0.824 & 0.819 & 1.0 & 0.949 & 0.549 \\
-0.714 & 0.786 & 0.901 & 0.949 & 1.0 & 0.770 \\
-0.404 & 0.475 & 0.789 & 0.549 & 0.770 & 1.0
\end{bmatrix}
\end{array}$$

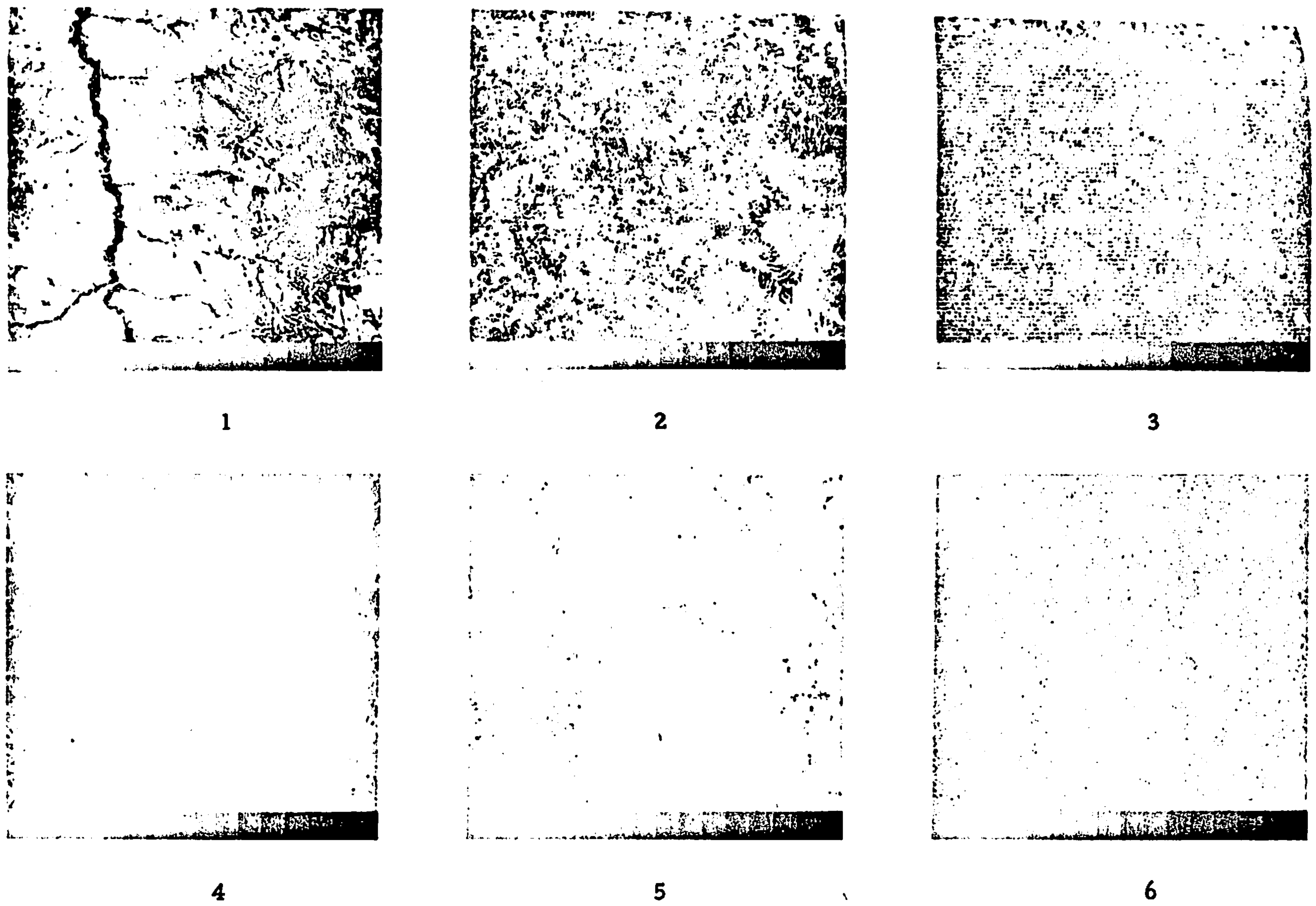


Figure 3.7-6. Principal Components of Direct Ratios

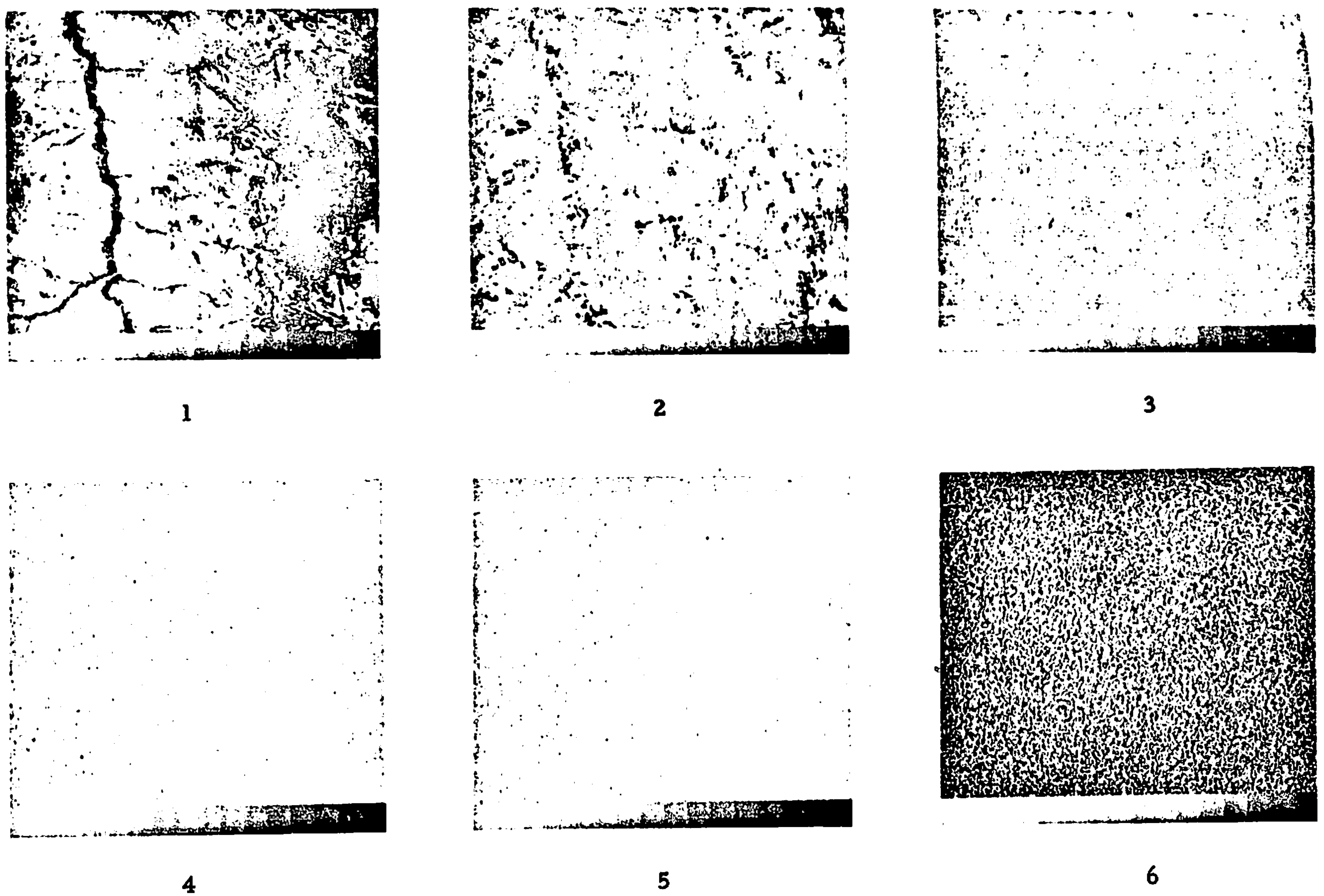


Figure 3.7-7. Principal Components of Log Ratios

The eigenvalues and the percentage of energy they represent is given below:

i	λ_i	$\left(\lambda_i / \sum_{k=1}^{6} \lambda_k\right) \cdot 100(\%)$
1	94.509	82.0
2	14.069	12.0
3	5.904	5.1
4	0.515	0.45
5	0.143	0.12
6	0.077	0.07

The above percentages show that the first two or three principal components contain most of the relevant information in ratio images. This can also be verified by studying the principal components shown in Figure 3.7-6.

In the case of logarithmic ratios, the normalized covariance matrix has been computed as:

$$\begin{array}{c|rrrrrr|}
\text{Ratios} \rightarrow \ \downarrow & 4:5 & 4:6 & 4:7 & 5:6 & 5:7 & 6:7 \\
\hline
4:5 & 1.0 & -0.297 & -0.390 & -0.746 & -0.714 & -0.399 \\
4:6 & -0.297 & 1.0 & 0.910 & 0.837 & 0.812 & 0.486 \\
4:7 & -0.390 & 0.910 & 1.0 & 0.840 & 0.912 & 0.771 \\
5:6 & -0.746 & 0.837 & 0.840 & 1.0 & 0.955 & 0.554 \\
5:7 & -0.714 & 0.812 & 0.912 & 0.955 & 1.0 & 0.751 \\
6:7 & -0.399 & 0.486 & 0.771 & 0.554 & 0.751 & 1.0 \\
\hline
\end{array}$$

The eigenvalues and the percentage of energy they represent is as follows:

i	λ_i	$\left(\lambda_i / \sum_{k=1}^{6} \lambda_k\right) \cdot 100(\%)$
1	35.495	87.0
2	3.270	8.0
3	1.592	3.9
4	0.084	0.2
5	0.082	0.2
6	0.080	0.2

The resulting principal components are shown in Figure 3.7-7.

References

[1] Goetz, A. F. H. et al, "Application of ERTS Images and Image Processing to Regional Geologic Problems and Geologic Mapping in Northern Arizona," JPL Technical Report 32-1597, May 15, 1975.

3.8 Principal Components Analysis

3.8.1 The Motivation for Principal Components Analysis

The ERTS multispectral scanner (MSS) consists of 24 detectors, six in each of four spectral bands: green, red, infrared-1 and infrared-2. These bands are designated by the numbers 4, 5, 6 and 7 and span the approximate spectral ranges of 0.5 to 0.6, 0.6 to 0.7, 0.7 to 0.8 and 0.8 to 1.1 μm.

The four bands of the ERTS data provide correlated images. Standard color composites are usually made by selecting the most relevant three bands from the available bands, and displaying each in a different primary color: red, green and blue. Typically Band 4 (green) is displayed in blue, Band 5 (red) in green, and Band 6 (nearest infrared) in red. This ordinarily does not result in a very satisfactory color display for typical ERTS pictures, because in general bands 4 and 5 and 6 and 7 are highly correlated.

Principal components analysis of ERTS bands is motivated by the desire to extract the most significant three spectral components from the available four. This dimensionality reduction also results in preserving most of the ERTS information in a smaller number of components.

3.8.2 Mathematical Formulation

The principal component analysis of ERTS data involves finding a unitary transformation matrix which, when applied to the four bands, results in a new set of bands (principal components) having several desirable characteristics: the principal components are uncorrelated and each component has variance less than the previous component.

The principal component transformation is also known as the Karhunen-Loève transformation, eigenvector transformation or the Hotelling transformation.

The principal components are obtained from the original four spectral bands by the matrix multiplication

$$Y = AX$$

where X is the vector of spectral intensities on four ERTS bands, Y is the vector of principal components and A is the 4 x 4 Karhunen-Loève transformation matrix. This matrix is derived by diagonalizing the spectral covariance matrix C_X of the spectral bands. The rows of A are the normalized eigenvectors of C_X. The covariance matrix of the principal components is then

$$C_Y = AC_XA^T = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix}$$

where λ_1, λ_2, λ_3 and λ_4 (the variances of the principal components) are the eigenvalues of C_X ordered such that $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4$.

It should be noted that, since A is a unitary transformation, the total data variance is invariant:

$$\sum_{i=4}^{7} \sigma_i^2 = \sum_{i=1}^{4} \lambda_i$$

where the σ_i^2, i = 4, 5, 6, 7 are the variances of the original ERTS bands.

3.8.3 Computational Results and Examples

In this section two examples of principal components analysis will be given.

Example 1. Principal components of the ERTS bands in Figure 8.3-1 are shown in Figure 8.3-2.

The spectral covariance matrix C_X of the four ERTS bands are obtained by computing the spectral covariance matrix on 64 x 64 blocks of ERTS pictures, (each 512 x 512 pixels) and then averaging over all the blocks. For the example shown in Figure 8.3-1, the spectral covariance matrix is

$$\begin{array}{c|cccc|}
\text{Bands} \rightarrow \downarrow & 4 & 5 & 6 & 7 \\
4 & 3988.8 & 3566.6 & 2314.6 & 976.6 \\
5 & 3566.6 & 3841.3 & 2185.7 & 897.2 \\
6 & 2314.6 & 2185.7 & 4845.9 & 3937.7 \\
7 & 975.6 & 897.2 & 3937.7 & 4408.6
\end{array}$$

The normalized covariance matrix is

$$\begin{array}{c|cccc|}
\text{Bands} \rightarrow \downarrow & 4 & 5 & 6 & 7 \\
4 & 1.000 & 0.113 & 0.065 & 0.029 \\
5 & 0.113 & 1.000 & 0.063 & 0.027 \\
6 & 0.065 & 0.063 & 1.000 & 0.106 \\
7 & 0.029 & 0.027 & 0.106 & 1.000
\end{array}$$

Diagonalization of the spectral covariance matrix results in the following Karhunen-Loève transform matrix:

$$A = \begin{bmatrix}
0.46909 & 0.45254 & 0.59965 & 0.46432 \\
-0.51456 & -0.51757 & 0.32918 & 0.59916 \\
-0.01706 & -0.27163 & 0.71608 & -0.64276 \\
-0.71756 & 0.67344 & 0.13892 & -0.11083
\end{bmatrix}$$

The eigenvectors are the rows of the above matrix. It should be observed that eigenvectors are similar to the first four Haar functions, except the sign change. Thus the above Karhunen-Loève transform matrix can be approximated by a 4 x 4 Haar transform for computational purposes.

The eigenvalues are:

i	λ_i	$\left(\lambda_i / \sum_{k=1}^{4} \lambda_k\right) \cdot 100(\%)$
1	11355.2	66.5
2	4953.3	29.0
3	426.8	2.5
4	344.2	2.0

Above percentage values show that the first two principal components represent 95.5% of the total energy. The principal components are obtained in block format. The block size is 64 x 64. The resulting principal components are normalized to have intensities ranging between 0 and 255. It should be noted that the same transform is used for each 64 x 64 block, i.e. instead of computing a different Karhunen-Loève transform for each block, one transform is used for each block. The result is checked by computing the normalized covariance matrix of the principal component:

Principal Components →

$$\begin{array}{cc} \downarrow & \begin{array}{cccc} \quad 1 \quad & \quad 2 \quad & \quad 3 \quad & \quad 4 \quad \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1.00000 & -0.00003 & 0.00005 & 0.00010 \\ -0.00003 & 1.00000 & -0.00013 & -0.00004 \\ 0.00005 & -0.00013 & 1.00000 & 0.00020 \\ 0.00010 & -0.00004 & 0.00020 & 1.00000 \end{bmatrix} \end{array}$$

The near-zero off-diagonal elements in the above matrix show that the principal components are indeed uncorrelated.

Example 2. Principal components analysis has been applied to a different area. In addition, ERTS bands have been corrected, as explained in section 3.2, prior to the principal components analysis.

For this example, the spectral covariance matrix is

$$\begin{array}{cc}
\text{Bands} \rightarrow & \begin{array}{cccc} 4 & 5 & 6 & 7 \end{array} \\
\downarrow & \\
\begin{array}{c} 4 \\ 5 \\ 6 \\ 7 \end{array} & \begin{bmatrix} 57.16 & 75.80 & 39.23 & 18.46 \\ 75.80 & 113.69 & 53.76 & 24.50 \\ 39.23 & 53.76 & 68.97 & 64.78 \\ 18.46 & 24.50 & 54.78 & 85.43 \end{bmatrix}
\end{array}$$

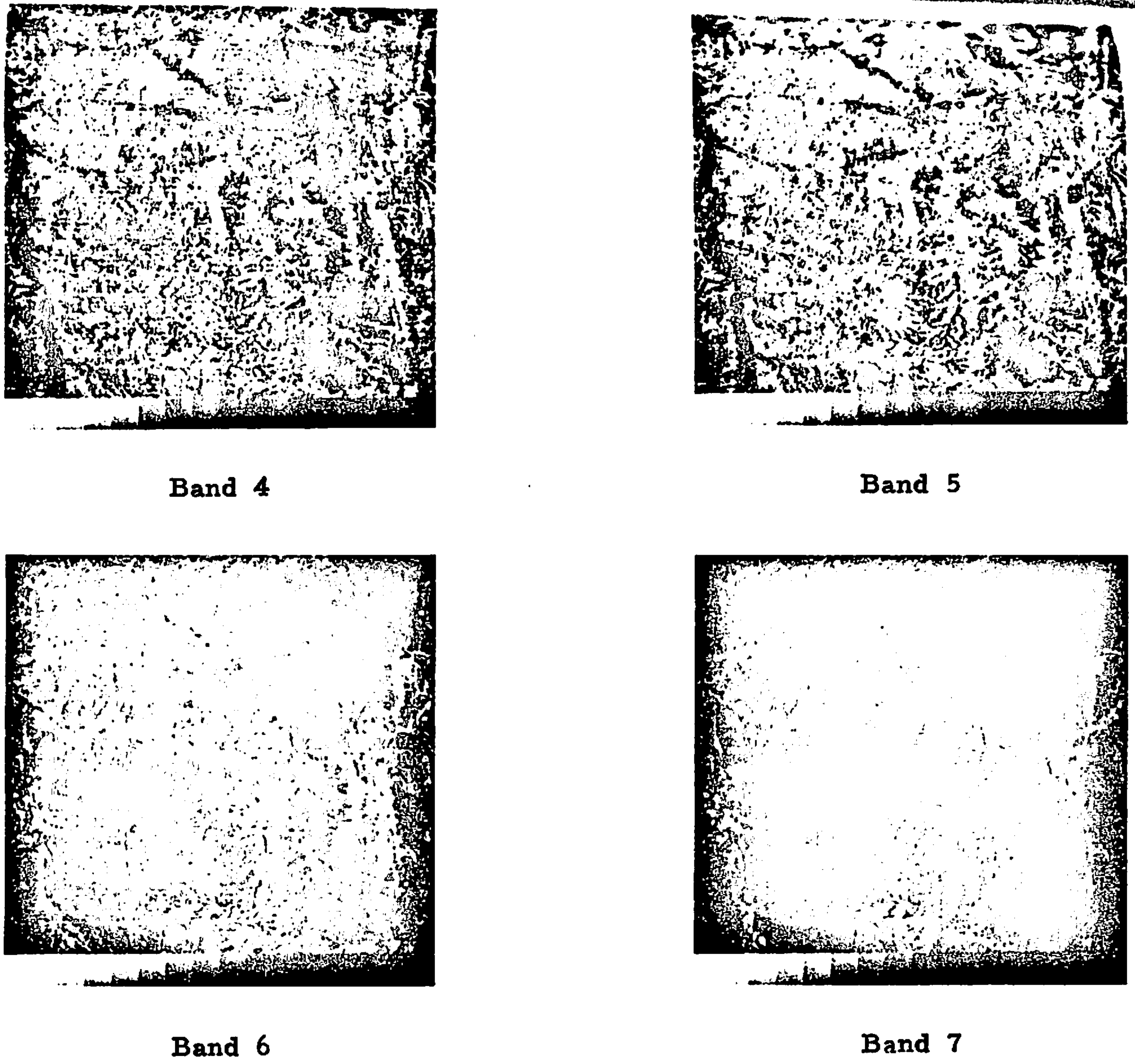
The normalized spectral covariance matrix is

$$\begin{array}{cc}
\text{Bands} \rightarrow & \begin{array}{cccc} 4 & 5 & 6 & 7 \end{array} \\
\downarrow & \\
\begin{array}{c} 4 \\ 5 \\ 6 \\ 7 \end{array} & \begin{bmatrix} 1.000 & 0.117 & 0.078 & 0.033 \\ 0.117 & 1.000 & 0.075 & 0.031 \\ 0.078 & 0.075 & 1.000 & 0.105 \\ 0.033 & 0.031 & 0.105 & 1.000 \end{bmatrix}
\end{array}$$

The Karhunen-Loève transform matrix is

$$A = \begin{bmatrix} 0.44465 & 0.63040 & 0.49520 & 0.39958 \\ -0.32653 & -0.49866 & 0.34168 & 0.72662 \\ 0.32957 & -0.45586 & 0.67249 & -0.48097 \\ 0.76619 & -0.38227 & -0.43103 & 0.28469 \end{bmatrix}$$

The resulting principal components are given in Figure 8.3-3.



Band 4

Band 5

Band 6

Band 7

Figure 3. 8-1. Enhanced ERTS bands

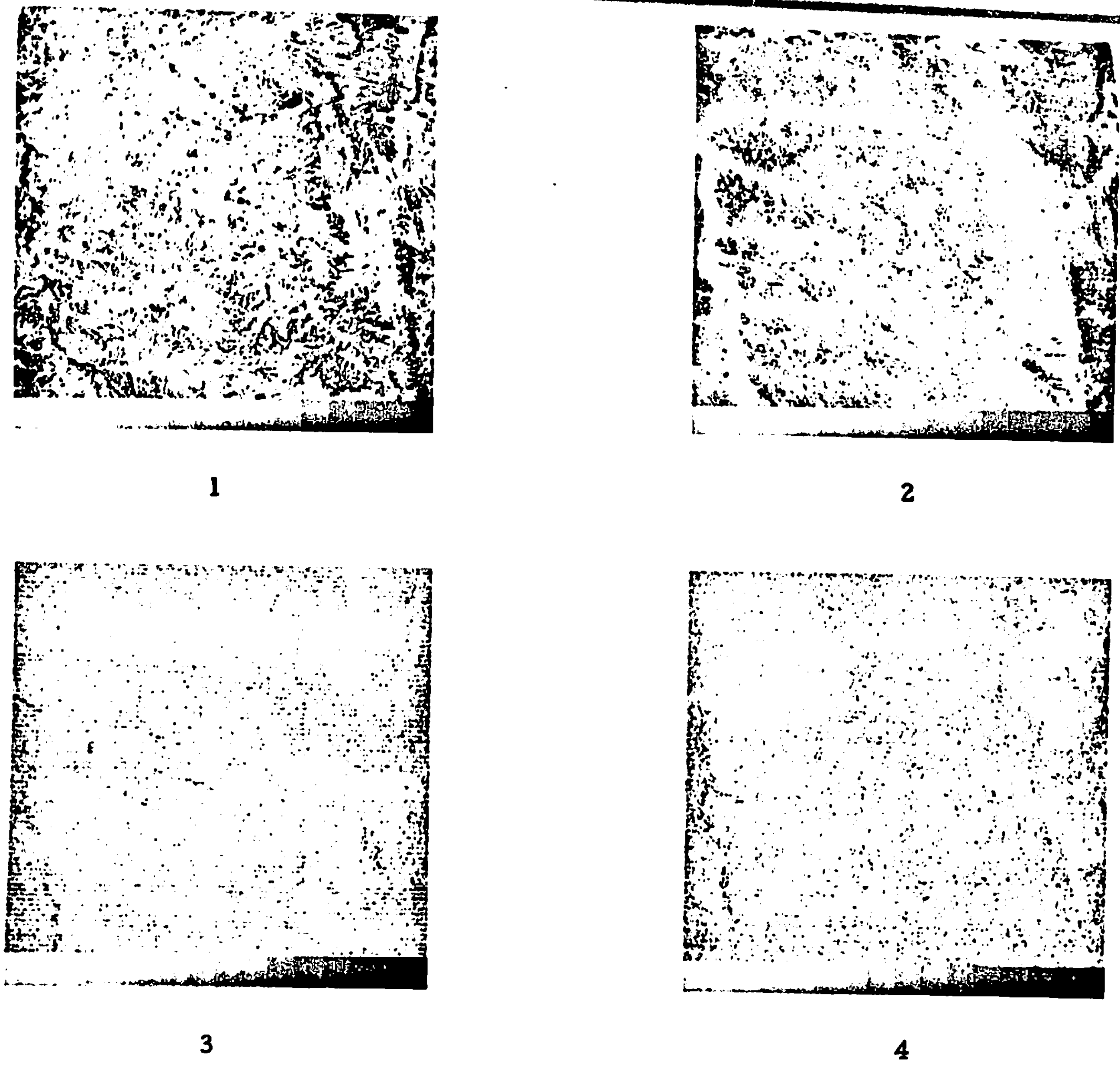


Figure 3.8-2. Principal Components (Example 1)

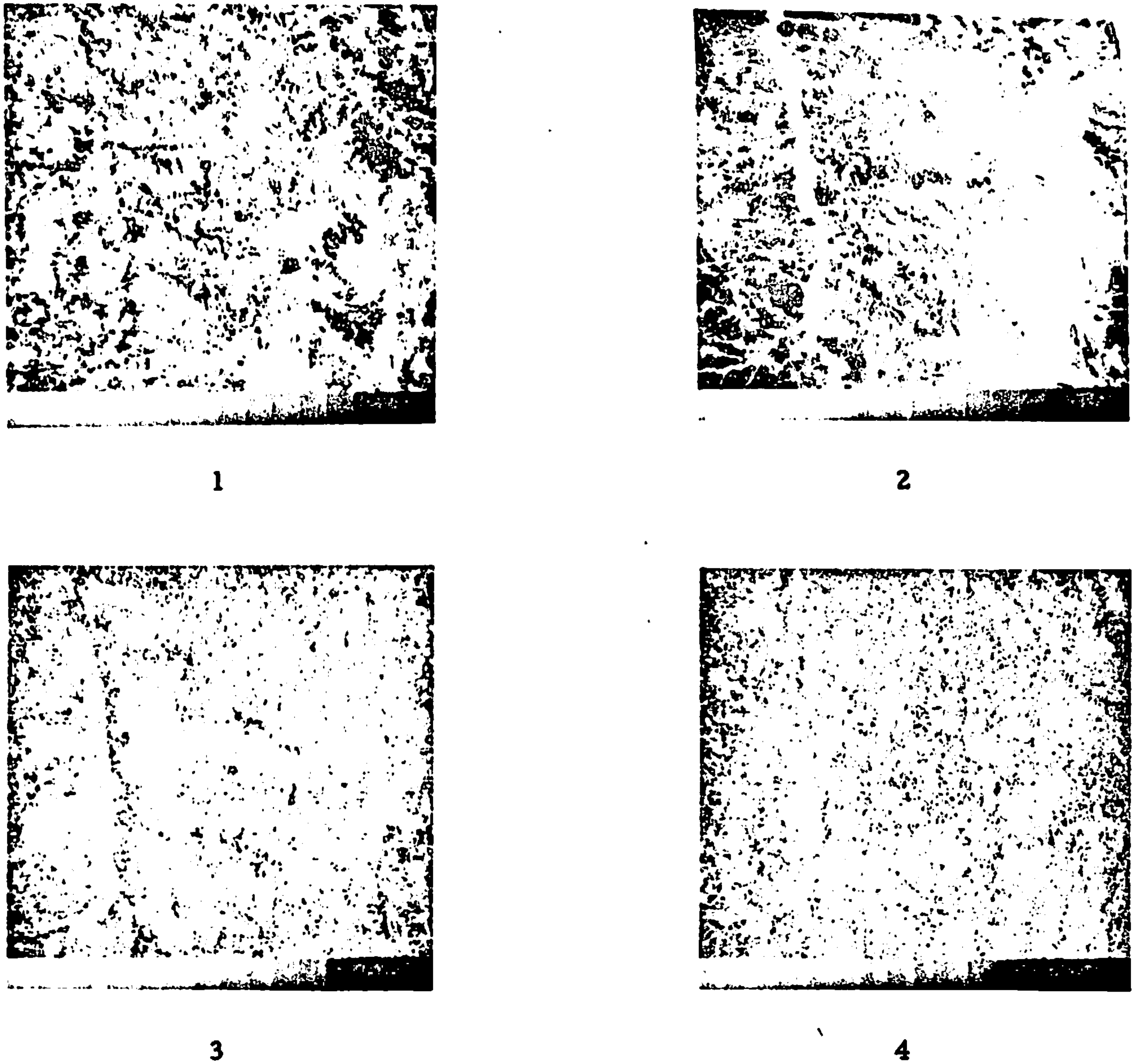


Figure 3. 8-3. Principal Components (Example 2)

The eigenvalues and the percentage of energy they represent are given below:

i	λ_i	$\left(\lambda_i / \sum_{k=1}^{4} \lambda_k\right) \cdot 100(\%)$
1	224.92	69.14
2	90.78	27.91
3	5.42	1.66
4	4.13	1.27

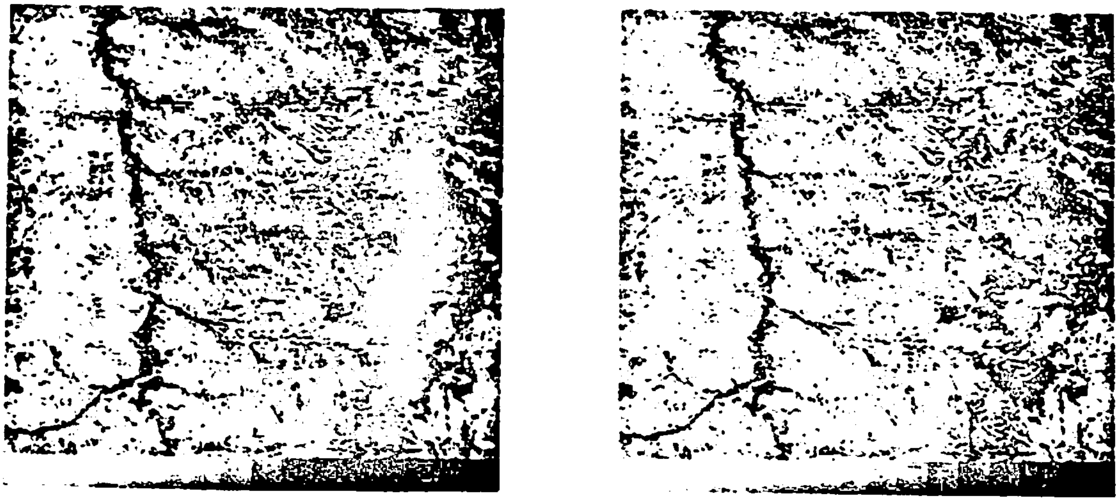
Thus the first two principal components represent 97% to the total energy.

3.9 Wiener Filtering

Wiener filtering is a classical technique of signal estimation that has been applied, primarily, to one-dimensional, continuous signals, with analysis and implementation based upon continuous Fourier signal theory [1].

The classical Wiener filtering technique has been extended to the processing of one- and two-dimensional discrete data by digital operations with emphasis on reduction of computational requirements [2]. This generalized Wiener filtering technique utilizes the transform properties of imaging systems and improves the computational efficiency in enhancing the images corrupted by noise. However, computational requirements are still high and it is necessary to perform the filtering operation in small blocks of the image. Since a filter of this kind is computationally unattractive, a fast Wiener filter has been introduced for noisy image restoration [3]. This filter is obtained by imposing certain modifications on the observed image. It is shown that the observed image can be operated upon to modify its statistical characteristics. Thus, certain operations are introduced which, when applied on pictorial data, allow the modified data to be characterized statistically by a circulant covariance matrix. These operations are further approximated to gain more computational speed at the cost of a slight increase in the mean-square error. Unlike the classical Wiener filter, it has been shown that the fast filter is capable of operating upon large images without the need to break down the observation into small blocks [3].

Figure 3.9-1 shows the result of Wiener filtering on the green band of the ERTS picture of size 512 x 512. The 256 x 256 size center of the picture has been filtered by two-dimensional fast Wiener filtering algorithm. Figure 3.9-1 also shows the result of Wiener filtering on the full picture.



Filtering on 256 x 256 center

Filtering of the full picture

Figure 3.9-1. Wiener Filtering Example on Band 4.

References

1. Davenport, W. B. and Root, W.L., Random Signals and Noise, McGraw-Hill, New York, 1968.

2. Pratt, W. K., "Generalized Wiener Filter Computation Techniques," IEEE Transactions on Computers, Vol. C-21, no. 7, pp. 636-641, July 1972.

3. Davarian, F., "Fast Computational Techniques for Pseudoinverse and Wiener Image Restoration," Ph.D. Thesis, University of Southern California, Electrical Engineering Department, 1975.

3.10 Color as an Aide for Human Terrain Classification

It is well known that color images present more information for the human observer than black and white pictures. This is because color effectively adds two perceptual dimensions, namely hue and saturation to an ordinary brightness image. This property has been exploited to enhance pictures in the variety of fields such as medical imagery [1], air reconaissance and ERTS images to name a few. Color or pseudo-color mappings can be obtained photographically or optically from black and white components or they can be synthesized by computer. For example, one widely used technique is to combine three bands of an ERTS image by projection with red, green and blue filters respectively. Other photographic techniques have been extensively studied elsewhere [2]. The advantage of computer generated color composites is that the color maps can be designed easily in the most arbitrary fashion. However, the almost infinite number of possible combinations makes it difficult to determine the optimum scheme. It appears from previous studies [1,2,3,4] that such an optimum depends very much upon each particular application. One approach has been explored here, in which the color mapping is obtained from pre-processed components which contain more information each than the original spectral band images. Several color mappings have been created from differently pre-processed components. It seems significant that certain ground features have been made apparent in that way, which were previously not visible. It is believed that a closer examination of the present pictures will indicate which approach is most promising in this context, so that the number of alternatives for further studies is reduced to a smaller, more specific set.

The particular approach used here is now discussed in more detail. It was assumed that color composites would carry more information for the human observer if:

a) The components used synthesize the color image are uncorrelated or have little correlation.

b) The histogram of each component is hyperbolized (see section 3.3).

These two conditions ensure that the range of colors available for display is fully used and that information contained in the dissimilarity between the components is displayed as hue and saturation differences. The pre-processed components used here were the principal component images (see section 3.8) and the logarithmic ratio pictures of section 3.7. Table 3.10-1 shows the combinations used for the pictures of Figures 3.10-1, 2, 3, 4 and 6. It is pointed out that the technique described does

Table 3.10-1. Components Used for the Color Enhanced Images of Figures 3.10-1 through 6.

Components	Mapping	Figure
Principal Component #1	Red	3.10-1
Principal Component #2	Green	3.10-1
Principal Component #3	Blue	
Principal Component #2	Red	3.10-2
Principal Component #1	Green	
Principal Component #3	Blue	
Logarithmic Ratio 4/5	Red	3.10-3
Logarithmic Ratio 4/6	Green	
Logarithmic Ratio 4/7	Blue	
Logarithmic Ratio 4/6	Red	3.10-4
Logarithmic Ratio 4/5	Green	
Logarithmic Ratio 5/7	Blue	
Principal Component #1	Red	3.10-5
Principal Component #2	Green	
Logarithmic Ratio 4/5	Blue	
Principal Component #1	Red	3.10-6
Principal Component #2	Green	
Logarithmic Ratio 4/6	Blue	

Note: The color images of figures 3.10-1 thru 3.10-6 have not been included in the printed version of this report, for economic reasons.

Note: The color images of figures 3.10-1 thru 3.10-6 have not been included in the printed version of this report, for economic reasons.

not require highly sophisticated digital color display devices. High resolution black and white transparencies of the components used can be generated on a flying-spot scanner and combined optically with the contractor's color compositor.

One avenue for further studies is the use of a visual model for color perception [3, 5, 6, 7] in order to map the components first into variables that are direct correlates of the human color perception. Inverting the model, red, green and blue components can then be generated for display either with a color television device or a projection color compositor. This approach is schematized in Figure 3.10-7.

References

[1] H. C. Andrews, et al., "Picture Processing by Computer," IEEE Spectrum,

[2] R. H. Stratton and J. J. Sheppard, Jr., "A Photographic Technique for Image Enhancement: Pseudocolor Three-Separation Process," Rand Report R-596-PR, October, 1970.

[3] M. M. Taylor, "Principal Components Colour Display of ERTS Imagery," Proc. of the Third ERTS Satellites Symposium, Washington, D. C., Vol. I, pp. 1877-1897, December 1973.

[4] A. F. H. Goetz et al., "Applications of ERTS Images and Image Processing to Regional Geologic Problems and Geological Mapping in Northern Arizona," JPL Technical Report 32-1597, May, 1975.

[5] Modeling Color Vision for Psycho-Visual Image Processing," USCEE Report 459, September 1973, pp. 112-122.

[6] "A New Model of Color Vision and Some Practical Implications," USCEE Report 530, March 1974, pp. 128-143.

[7] "A Quantitative Model of Color Vision," USCEE Report 540, September 1974, pp. 69-83.

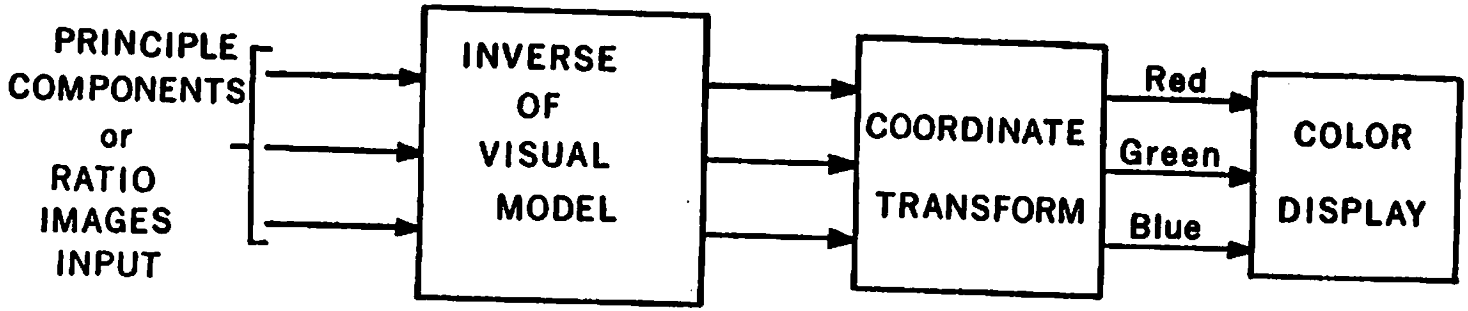


Figure 3.10-7. Use of a visual model to synthesize high information color composites

4. SOFTWARE EFFORT

In the context of this project, almost all programs have been written specifically for study and development. With the exception of the fast two-dimensional image Fourier transform packages, all programs are written in Fortran. The complete software is therefore easily transportable and can be readily modified if necessary. Most programs execute in less than a minute on the PDP-KI10 of the image processing institute. It is felt that optimization and machine language programming would only be warranted in the case of large volume processing. Section 4.1 contains a brief description of most programs developed for this porject and the source listings are contained in section 4.2.

4.1 Software Documentation

The major programs developed for the present project are listed below, accompanied by a brief description. They are ordered as follows:

a) Data base formatting software.

b) Histogram manipulation programs including "anti-banding" algorithms.

c) Spatial filtering and crispening.

d) Band ratioing.

e) Principal components analysis.

f) Wiener filtering.

g) Display pre-distortion programs.

Use has also been made of IPL software whenever possible; for example, the FFT packages "part 1" and "part 2" were used extensively in connection with spatial filtering. Besides I/O statements, however, the only machine specific subroutine is DSKIO by which one line of image data is moved from disk to core and vice versa.

4.2a Data Base Formatting Programs

Program ERTSIS

Compensates for aspect ratio and earth ratio and earth rotation based upon linear interpolation. Input picture is 752 x 512 lines, output is 512 x 512.

Program REDUCE

Reduction of image size from N x N to N/2 x N/2 (typically from 512 x 512 to 256 x 256) by averaging four nearest pixels.

Program ENLARG

Enlargement of image size from N2 x N2 to N x N by linear interpolation.

4.2b Histogram Manipulation Programs

Program STRECH

Linear amplitude strech and shift, based upon amplitude histogram inputs are input mean and stretch factor; program dips values 0 < and > 255.

Program LOGC

Converts input amplitudes to scaled log amplitudes (0 ÷ 255). Input picture may be real because a fast log routine is used (FLOG) instead of a look-up table.

Subroutine FLOG

Fast log routine with limited precision ($\epsilon_r \leq 8 \cdot 10^{-4}$).

Program HISHYP

Histogram equalization and hyperbolization program. Options are: a) the slope SL (must be negative or zero) of the histogram at the origin. sl = 0 means histogram equalization. b) modulo activates the banding suppression algorithm when set equal to 6; modulo = 1 is regular histogram equalization or hyperbolization. c) corr = 1 activates the gamma correction for TV viewing (otherwise zero); corr = 2 activates

the pre-distortion for photographic recording.
d) check = 1 gives print-out of computed histograms. (otherwise zero).

Subroutines:

HISTO
ACCU
HYPER
CST
GAMMA
LOKUP
CHECK

Subroutine HISTO

Computes the histogram of the picture. If the modulo MB is larger than 1, the subroutine returns M histograms corresponding to the M sensor subimages for banding suppression.

Subroutine ACCU

Accumulates histogram(s) for histogram equalization. Returns the appropriate look-up table(s).

Subroutine HYPER

Modifies the above look-up tables for histogram hyperbolization.

Subroutine CST

Solves the transcendental equation

$$S = \frac{-1}{c^2 \cdot \log\left(\frac{c+1}{c}\right)}$$

for c given S; c is used in the subroutine hyper.

Subroutine GAMMA

Look-up table based gamma correction for TV viewing.

Subroutine LOOKUP

Look-up table image transformation program. Tables are supplied by the subroutines ACCU or HYPER, depending upon the input value of the slop SL.

Subroutine CHECK dumps the "before and "after processing histograms.

4.2c <u>Spatial Filtering and Crispening Programs</u>

Program FILTER

Two-dimensional circular filter program. User specifies filter function with an external function. Input has to be the 2-D Fourier transform of the picture, formatted according to the IPL FFT programs "part 1" and "part 2".

Program SHARP

Moving window finite 2-D convolution program. Window is 21 pixels. Masking factor determines amount of DC rejected (for example, DC = 0 means that constant values are rejected completely.

Subroutines:

DSO
DS
CV

Subroutine DSO

Handles the edge lines of the picture (no processing.

Subroutine DS

Reads a line in compact form and expands it to integer representation.

Subroutine CV

Performs the actual finite size convolution. The inputs are six lines of image data and one line of the processed picture is output to disk on each call.

4.2d <u>Band Ratio Programs</u>

RATIO

Linear ratio program. Ratio of two bands is computed, and the output is multiplied by a scale factor which is an input parameter to the program.

LRATIO

Logarithmic ratio program. The input is integer and table look-up is used for the log. Output is scaled to 0 ÷ 255.

4.2e <u>Principal Components Analysis</u>

KL

This program consists of three separate programs: KL1, KL2 and PC. The input to KL is four spectral bands of ERTS, each 512 x 512 pixels. The output is four principal components.

KL1

Computes the spectral covariance matrix of ERTS bands on 64 x 64 blocks and averaging over the entire 512 x 512 images. It then diagonalizes the spectral covariance matrix to obtain the Karhunen-Loève transform matrix.

KL2

Applies to the Karhunen-Loève transform matrix to the four spectral bands of ERTS data and writes the resulting principal components on four temproary data sets.

PC

Finds the range of amplitudes on the temporary data sets obtained in KL2 and then normalizes each temporary data set such that the resulting principal components will have amplitudes that lie between 0 and 255. It then writes out the principal components as four 512 x 512 packed data sets.

4.2f Wiener Filtering

WIENER

Performs one or two dimensional Wiener filtering on an N x N packed or real image. The inputs to the program are the image to be filtered, the blur, the correlation coefficient and the size of the impulse response of the blur. The output is an N x N packed or real filtered image.

4.2g Display Pre-Distortion Programs

Program GAMMA

Look-up table based program to pre-distort the data for the digital TV display. Its effect is to compensate for the CRT non-linearity.

Program PHOTO

Pre-distortion program for the IER flying sopt scanner. Corrections include: a) empirical H-D correction for the Kodak Royal-X film. b) Optical vignetting correction. c) Gamma correction of the CRT tube. The program also superimposes a calibrating grey scale onto the bottom of the picture.

Subroutines:

DENS, GAMMA, SCAL, VIGN

Subroutine DENS

Sets up a look-up table based upon empirically determined photographic density reproduction characteristics of the flying spot scanner.

Subroutine GAMMA

Generates a look-up table for the gamma correction of the TV display.

Subroutine SCAL — Superimposes a gray code onto the bottom of the picture for photographic calibration and checking purposes.

Subroutine VIGN — Pre-distorts the image data to compensate for the vignetting of the photographic gear attached to the flying spot scanner.

5. CONCLUSIONS AND OUTLOOK

Conclusions on this project can be drawn on several levels, for example, image processing, geomorphic studies and the potentials for further developments. Regarding digital image processing, several powerful techniques have been developed which are appropriate for large scale use. Some of the more exotic algorithms may be more appropriate for selected areas only. By and large however, it seems that the potential present in the available ERTS imagery is greatly enhanced by computer image processing. It is somewhat more difficult to assess the impacts of the present developments on oil exploration. It is believed that a careful examination of the present pictures in a correlation study with ground truth information is necessary to reach such conclusions.

Regarding further developments, several new avenues have emerged in the past contract period. One of them is the exploitation of the periodic coverage provided by the ERTS satellites. It is suspected that variations due to climatic or seasonal changes may give useful added knowledge on the underlying geologic formations. Such variations can be detected by registration of two images of the same area from different overflights. After the images are perfectly registered, difference or ratio images can be generated and enhanced, which reveal such temporal changes.

Another suggested topic for further research is the "intelligent" machine search for feature boundaries such as lineaments, outcrops, etc. The use of concepts borrowed from the field of artificial intelligence may provide the tools for the specification of such features, as is suggested, for example, by syntactic image descriptors. It appears that a classification scheme would be much more powerful if its inputs include such feature information in addition to ground spectral reflectances. Finally, it is believed that an interactive approach is necessary to accelerate the design of higher level automatic classifiers. Dedicated hardware and software is being presently developed at the Image Processing Institute to allow highly interactive image processing tasks. As this new resource becomes available, it will be possible to train algorithms interactively without sacrifice of the power and flexibility of the PDP KI10 machine. Since the display to be used in this configuration is a digital color TV, very sophisticated interactive processing tasks will become possible.

6. APPENDIX

The restitution of processed pictures is generally a costly, time-consuming, and yet essential step of digital image processing.

Errors and non-linearities introduced by display equipment or the photographic process can add a surprising amount of unwanted and uncontrolled "image processing". These parasitic effects are by no means always readily visible in the finished product, but they may well invalidate the results of computer image manipulations. A careful control of the electro-optical machinery, the photographic process, as well as an understanding of human visual factors is therefore essential to insure the success and credibility of digital image processing.

1) Visual Factors

Optimum reflection prints, transparencies and television images practically never replicate the brightness distribution of original scenes, just like perfect color images do not reproduce the spectral energy distribution of colored lights. Although comprehensive fidelity criteria for images are yet to be discovered, a few simple rules have been found useful in the optimization of image acquisition and reproduction techniques.

Studies of the reproduction characteristics of optimal images [3] indicate indeed that although absolute brightness influences perceived quality, the quality criterion within the physical limitations of any given reproduction situation is greatly dependent upon its ability to reproduce relative brightness ratios. This fact is intuitively satisfying if we note that pixel brightness ratios are a property of the scene reflectances that is invariant to the absolute intensity of a uniform illumination.

The implications of the above visual phenomenon are that the digital representation of light intensities sensed by a scanning device should ideally be a measure of image brightness ratios rather than arbitrary absolute intensity values. This is easily implemented in practice by recording the logarithm of the measured image intensities. On the reproduction side, care has then to be taken to preserve the recorded brightness ratios, a process that is facilitated by the inherent characteristics of the photographic process to be discussed below.

6.2 The Photographic Process

Exposure of a black and white emulsion to light and subsequent development produces a light absorbing layer characterized by its optical density D which is defined as follows:

$$D = \log_{10} (\text{transmitted light/incident light})$$

With all other parameters fixed, the optical density is ideally related to the intensity of the exposing light I by the function [2]

$$D = \gamma * \log(I * t)$$

where t is the duration of the exposure. This function, well known in photography, is called Hurter-Diffield (HD) or D-log E curve. Actual photographic materials depart from this idealized law at both ends of their useful dynamic range. The factor γ describes the "contrast" of the emulsion and is positive for an ordinary negative material and negative for a reversal process. Because the unexposed emulsion and its substrate are not perfectly transparent, an additional "fog" level D_0 is incorporated into the above equation

$$D = D_0 + \gamma \log_{10} (I*t) \qquad (1)$$

The light reflected from a print or transmitted through a slide is related to the incident light I_0 by [2]

$$I = I_0 * 10^{-D} \qquad (2)$$

The reproduced light intensity I' is given by substitution of eq. (1) into (2)

$$I' = I_0 * 10^{-D_0} * (I*t) \qquad (3)$$

Note that if $\gamma = -1$, the conditions for an optimum reproduction as discussed in the previous section are met.

It is not straightforward to meet the relationship of eq. (3) with actual image processing equipment. The film is typically exposed by a CRT, LED or laser as a series of discrete dots which partly overlap; the exposure may not be uniform over the area of the image, etc. It is possible though to correct for such defects with a numerical pre-distortion of the digital image data. A simple model, appropriate for the correction a CRT scanner, is discussed in the next section.

6.3 The Calibration of I/O Devices

Actual image acquisition and reproduction devices have a number of inherent imperfections which distort the final product. First, the measured or reproduced light intensities are generally not an exponential, resp. a logarithmic function of the associated electrical quantities; the light sensitive or light emitting surfaces of an electron beam device are not perfectly homogeneous; optical systems may introduce significant vignetting, etc. A number of procedures have been devised to cope with such imperfections [3, 4]. For example, table look-up or polynomial approximations may be used to correct for the average deviations of the electro-optical transfer function from the desired behavior. A more refined (and expensive) solution is to vary the coefficients of the correction as a function of the geometric image coordinates. A true assessment of I/O device performance and the gathering of physical data for the design of correction schemes is best done by producing test patterns such as step tablets and measuring the optical density functions obtained on hardcopy or transparency.

To illustrate the above, a new software correction technique for CRT scanners is presented. It is of medium complexity, but computationally very fast and has given excellent results with an IER CRT scanner. The major sources of distortions in this case are schematized in Figure 6-1. From left to right, we have:

a) The CRT light emission I as a function of the drive and bias voltages U and U_0 resp. [5], which can be approximated by:

$$I = (U + U_0 + U_1)^{\gamma_{CRT}}$$

b) The optical vignetting which produces a darkening towards the image corners (Figure 6-2), (particularly annoying if one attempts to produce a mosaique, see Figure 6-3).

c) The D-log E function of the photographic material used.

Assuming that the vignetting is the only space-variant distortion (e.g. assuming that the CRT surface and film material are uniform), a fast look-up table based algorithm has been implemented, such that each source of distortion mentioned above is corrected for the appropriate order. The γ_{CRT}^{-1} and D-log E corrections of Figure 6-4 are straightforward look-up tables based upon measured data. Perhaps the most interesting pre-distortion step is the vignetting correction. Assuming circular symmetry, a second order polynomial of the form

$$I' = I(A + B(x^2+y^2))$$

has been used to boost the light intensities towards the image corners. x and y are the image coordinates referred to the center of the CRT screen. The values $A/2+Bx^2$ are stored in a one-dimensional array and the correction is made by looking up this array twice given the pixel line and column indices x_i and y_i

$$I' = I_0(C(x_i) + C(y_i))$$

The results from this fast correction technique are shown in Figure 6-5. The variations in density across a uniform surface are less than 0.1 density units, whereas the uncorrelated image had corners darkened by as much as 0.35 d-units.

In conclusion, it is the author's experience that certain distortions are often overlooked in practice; artifacts thus introduced have sometimes produced more "image processing" than intricate numerical algorithms. Numerical correction techniques have been implemented to combat such distortions and artifacts. The consistency of the results obtained allow a very good evaluation of the processing techniques studied here.

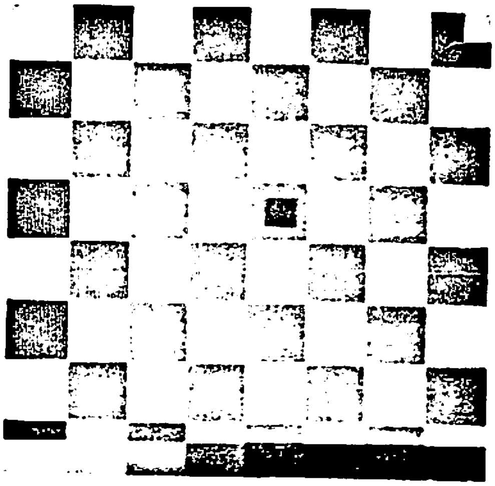


Figure 6.2. Illustration of the vignetting effect.

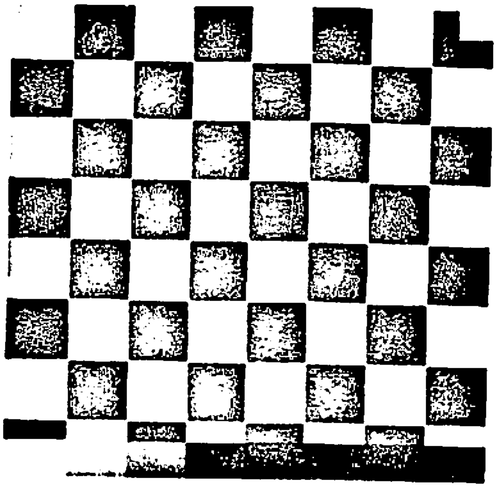


Figure 6.5. Vignetting correction by numerical predistortion.

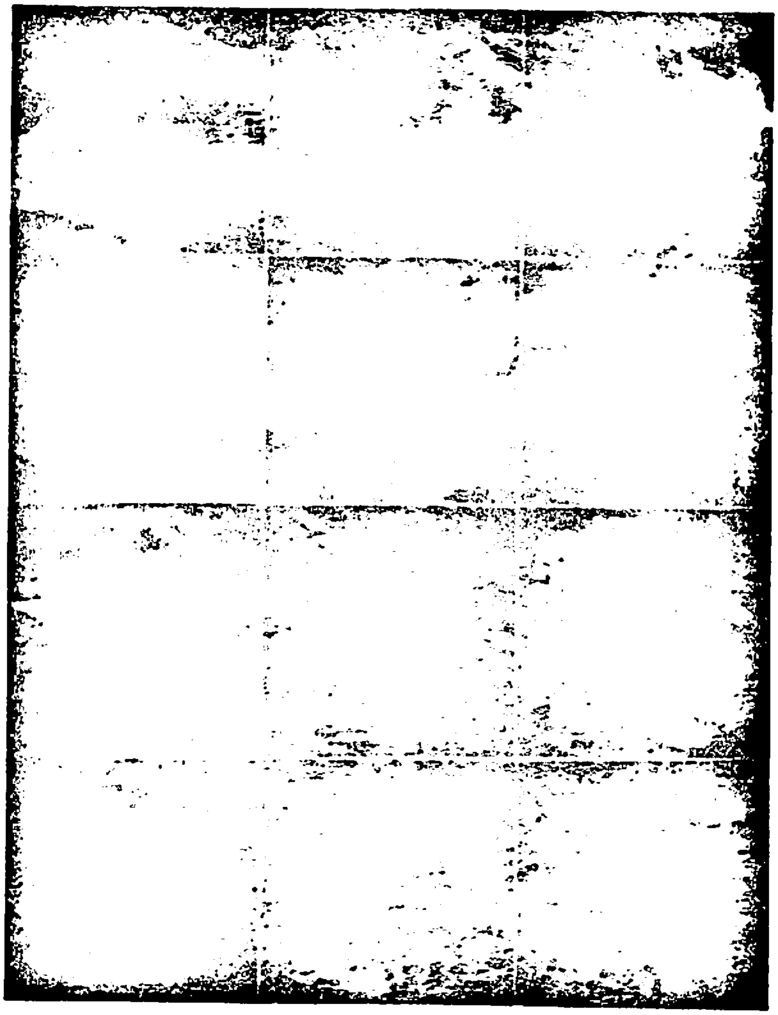
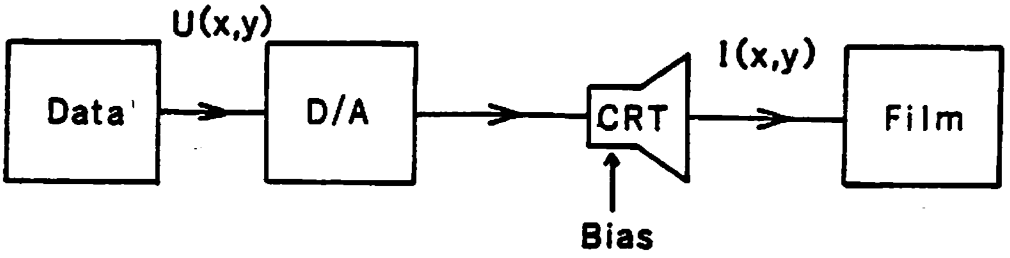


Figure 6.3. The effect of vignetting on photomosaics.



$I(x,y) = \text{aperture}(x,y) \,.\, (U(x,y) + U_0 + \text{Bias})^{\gamma}$

pre distortion:

$$U'(x,y) = \sqrt{U(x,y)} \,.\, \sqrt{U_{max}}$$

Figure 6-4. Display correction

References

[1] T. H. Cornsweet, Visual Perception, Academic Press, New York, 1970.

[2] T. G. Stockham, "Image Processing in the Context of a Visual Model," Proceedings of the IEEE, Vol. 60, July 1972, pp. 828-842.

[3] C. J. Bartleson and E. J. Breneman, "Brightness Perception in Complex Fields," JOSA, pp. 953-957, July 1967.

[4] R. M. Evans, W. T. Hanson, W. L. Brewer, Principles of Color Photography, Wiley and Sons, New York, 1953.

[5] R. Nathan, "Digital Video Data Handling," Technical Report 32-877, Jet Propulsion Laboratory, Pasadena, California, 1966.

[6] F. C. Billingsley, "Applications of Digital Image Processing," Applied Optics, Vol. 9, pp. 289-299, 1970.

[7] F. Kretz and W. Frei, "Optimal Logarithmic Quantization for Picture Processing, USCEE Report No. 530, Naval 1974, pp. 11-19.

```
        DIMENSION A(512),B(752)
        OFSET=34./511.
        ASPR=1.402
        IBIT=8
*
        DO 10 LIN=1,512
        CALL DSKIO(B,752,LIN,1,1,IBIT)
        CALL EXPAND(B,752,1)
*
        LIN1=LIN-1
        DO 20 IA=1,512
*       INDEX IN B-ARRAY
        AIN=LIN1*OFSET+IA*ASPR
        ILO=AIN
        IUP=ILO+1
*       WEIGHTS FOR LIN INTERPOLATION
        WLO=IUP-AIN
        WHI=1.-WLO
20      A(IA)=WLO*B(ILO)+WHI*B(IUP)
        CALL PRESS(A,512,1)
        TYPE 100,LIN
100     FORMAT(' LINE #  DONE  ',I4)
        CALL DSKIO(A,512,LIN,0,2,IBIT)
10      CONTINUE
        CALL FINI
        END
```

```
C   <ROBINSON>REDUCE.FOR;3    WED 27-AJG-75 12:40PM                 PAGE 1

C
C      REDUCTION OF IMAGE SIZE FROM N X N TO
C      N/2 X N/2 BY AVERAGING FOUR NEAREST PIXEL VALUES.
C
      DIMENSION X(1024),Y(1024),Z(512)
      TYPE 200
  200 FORMAT( ' ENTER  THE SIZE OF THE PICTURE TO BE REDUCED   ')
      ACCEPT *,N
      NHALF=N/2
      DO 10 J = 2,N,2
      I = J - 1
      CALL DSKIO(X,N,I,1,1)
      CALL EXPAND(X,N,1)
      CALL DSKIO(Y,N,J,1,1)
      CALL EXPAND(Y,N,1)
      DO 20 L = 2,N,2
      LH = L/2
   20 Z(LH) = (X(L) + X(L-1) + Y(L) + Y(L-1))/4.
      K = J/2
      CALL PRESS(Z,NHALF,1)
      CALL DSKIO(Z,NHALF,K,0,2)
   10 CONTINUE
      CALL FINI
      STOP
      END
```

```
C
C     ENLARGEMENT OF IMAGE SIZE FROM N2 X N2
C     TO N X N BY LINEAR INTERPOLATION.
C
      DIMENSION X(512),U(2,512),A(4,512)
      TYPE 200
 200  FORMAT(' ENTER    N2 AND  N')
      ACCEPT *,N2,N
      N4=N/N2
      N2P=N2+1
      NLR=0
      NLW=0
      DO 600  II=1,N2
      DO 604 I=1,2
      NLR=II-1+I
      IF (NLR.EQ.N2P) GO TO 500
      CALL DSKIO(X,N2,NLR,1,1)
      CALL EXPAND(X,N2,1)
      DO 604 J=1,N2
      JA=N4*(J-1)
      S=X(J)
      SS=X(J+1)
      IF(J.EQ.N2)  SS=2.*X(J)-X(J-1)
      DO 605 LL=1,N4
 605  U(I,JA+LL)=(N4-LL+1)*S/N4+(LL-1)*SS/N4
      GO TO 604
 500  DO 505 JJ=1,N
 505  U(2,JJ)=U(1,JJ)
 604  CONTINUE
      DO 606 J=1,N
      S=U(1,J)
      SS=U(2,J)
      DO 607 L=1,N4
 607  A(L,J)=(N4-L+1)*S/N4+(L-1)*SS/N4
 606  CONTINUE
      DO 608 I=1,N4
      NLW=N4*(II-1)+I
      DO 609 J=1,N
 609  X(J)=A(I,J)
      CALL PRESS(X,N,1)
 608  CALL DSKIO(X,N,NLW,0,2)
 600  CONTINUE
      CALL FINI
      STOP
      END
```

```
        DIMENSION A(256)
        TYPE 100
100     FORMAT (' INPUT PACKED OR UNPAKED? TYPE 8 OR 36')
        ACCEPT *,IBIT
        DO 10 LIN=1,256
        CALL DSKIO(A,256,LIN,1,1,IBIT)
        IF (IBIT.EQ.36) GOTO 5
        CALL EXPAND(A,256,1)
5       DO 20 IP=1,256
        A(IP)=(A(IP)-77.5)*1.51+127.
        IF(A(IP).LT.0.) A(IP)=0.
        IF(A(IP).GT.255.) A(IP)=255.
20      CONTINUE
        CALL PRESS(A,256,1)
10      CALL DSKIO(A,256,LIN,0,2,8)
        CALL FINI
        END
```

```
C   <PEI>LOGC.FOR;5    THU 27-MAR-75 12:25PM

          DIMENSION A(512)
          TYPE 100
100       FORMAT(' ENTER PIC SIZE, INPUT BITSIZE')
          ACCEPT *,ISIZ,IBIT
          DO 10 LIN=1,ISIZ
          CALL DSKIO(A,ISIZ,LIN,1,1,IBIT)
          IF (IBIT.EQ.36) GOTO 5
          CALL EXPAND(A,ISIZ,1)
5         DO 20 IP=1,ISIZ
          A(IP)=A(IP)+1.
          IF(A(IP).LT.1.) STOP
*
*         FLOG IS FAST LOG SUBROUTINE
20        A(IP)=21.5*FLOG(A(IP))
10        CALL DSKIO(A,ISIZ,LIN,0,2,36)
          CALL FINI
          END
```

```
C   <FRED>FASTLG.FOR;1    FRI 14-MAR-75 3:46PM                    PAGE 1

        FUNCTION FLOG(X)
*       RETURNS LOGE(X) WITH MAXIMUM ERROR = 8*10-4
*       # OF ITERATIONS IS BETWEEN 2 AND 8
*
        Y=X
        N=0
10      IF(Y.LT.1.) GOTO 20
        N=N+1
        Y=Y/2.
        GOTO 10
*
20      FLOG=FLOAT(N)/1.442695041
        Y=Y-1.
        AN=1.
        D=0.
30      AN=-AN*Y
        D=D+1.
        TERM=AN/D
        FLOG=FLOG-TERM
        IF(TERM.GT.0.001) GOTO 30
        RETURN
        END
```

```
*       SUBROUTINES       FILES
*       HISTO             H1
*       ACCU              H2
*       HYPER             H3
*       CST               H4
*       GAMMA             H5
*       LOKUP             H6
*       PHOTO             H7
*       CHECK             H8
        EXTERNAL CST
        COMMON TAB(0:255,6),MAX(6),MIN(6),JMP
        ACCEPT *,IU
        IF (IU.EQ.5) TYPE 100
100     FORMAT(' PIC SIZE,SLOPE,MODULO,CORR,CHECK?')
        ACCEPT *,ISIZ,SL,JMP,ICOR,ICK
*
        WRITE (IU,200),ISIZ,SL,JMP,ICOR
200     FORMAT(' SIZE=',I4,5X,'SLOPE=',F4.2,5X,
     1'MODULO=',I2,5X,'CORR CODE=',I2)
*
        CALL HISTO(ISIZ)
        IF(ICK.EQ.1) CALL CHECK(IU)
        CALL ACCU
        IF(ICK.EQ.1) CALL CHECK(IU)
        IF(SL.LT.0.) CALL HYPER(SL,CST)
        IF(ICK.EQ.1) CALL CHECK(IU)
        IF(ICOR.EQ.1) CALL GAMMA
        IF(ICOR.EQ.2) CALL PHOTO
        IF(ICK.EQ.1) CALL CHECK(IU)
        CALL LOKUP(ISIZ)
        CALL FINI
        END
```

```
C   <FREI>HI.FOR;4    FRI 16-MAY-75 9:45AM

        SUBROUTINE HISTO (ISIZ)
        COMMON TAB(0:255,6),MAX(6),MIN(6),JMP
        DIMENSION II(512)
*
        DO 10 J=1,JMP
        DO 10 I=0,255
10      TAB(I,J)=0.
*
        JMP1=JMP+1
        IF(JMP.EQ.1) JMP1=7
        J=1
*
        DO 20 LIN=1,ISIZ,JMP1
        CALL DSKIO(II,ISIZ,LIN,1,1,8)
        CALL IXPAND(II,ISIZ,1)
*
        DO 30 KOL=1,ISIZ
        IND=II(KOL)
30      TAB(IND,J)=TAB(IND,J)+1.
        J=J+1
20      IF(J.GT.JMP) J=1
        RETURN
        END
        SUBROUTINE ACCU
        COMMON TAB(0:255,6),MAX(6),MIN(6),JMP
*
        DO 10 J=1,JMP
*       ACCUMULATE HISTOGRAM
        DO 20 I=1,255
20      TAB(I,J)=TAB(I-1,J)+TAB(I,J)
*
        AMAX=TAB(255,J)
        MAX(J)=0
        MIN(J)=0
*
        DO 30 I=255,0,-1
        IF(MAX(J).NE.0) GOTO 35
        IF(TAB(I,J).LT.AMAX) MAX(J)=I+1
35      IF(MIN(J).NE.0) GOTO 40
30      IF(TAB(I,J).EQ.0) MIN(J)=I+1
*
40      AMIN=TAB(MIN(J),J)
        SC=255./(AMAX-AMIN)
*
        DO 50 I=1,255
50      IF(TAB(I,J).NE.0.) TAB(I,J)=(TAB(I,J)-AMIN)*SC
*
        MIN(J)=0
        DO 60 I=255,0,-1
        IF(MIN(J).NE.0) GOTO 10
60      IF(TAB(I,J).EQ.0.) MIN(J)=I+1
10      CONTINUE
        RETURN
        END
```

```
C   <FREI>H3.FOR;2     THU 15-MAY-75 11:20AM

          SUBROUTINE HYPER(SL,CST)
          COMMON TAB(0:255,6),MAX(6),MIN(6),JMP
*         MODIFY LOOK-UP TABLE(S) FOR HYPERBOLIC HISTOGRAM
          C=CST(SL)
*
          DO 10 J=1,JMP
          X=MAX(J)-MIN(J)
          C1=ALOG((C+1)/C)/X
          LL=MIN(J)
          DO 20 I=LL,255
          ARG=I-LL
          Y=C*(EXP(ARG*C1)-1.)
20        IF(I.LE.MAX(J)) TAB(I,J)=TAB(I,J)*Y
10        CONTINUE
          RETURN
          END
```

```
      FUNCTION CST(SL)
      IF(SL.GT.0.) STOP
*     COMPUTE PARAMETER CST
      CST=0.5
      D=0.5
30    CST=CST+D
      SL0=-1./(CST**2*ALOG((CST+1.)/CST))
      ERR=SL-SL0
      IF(ABS(ERR).LT.0.0001) GOTO 40
      IF(D.LT.0.) D=-D
      IF(ERR.GT.0.) GOTO 30
      D=-D/2.
      GOTO 30
40    RETURN
      END
```

```
C   <FREI>H5.FOR;1     THU 15-MAY-75 11:25AM

          SUBROUTINE GALVA
          COMMON TAB(0:255,6),MAX(6),MIN(6),JMP
          CST=SQRT(255.)
*
          DO 10 J=1,JMP
          DO 10 I=MIN(J),MAX(J)
10        TAB(I,J)=CST*SQRT(TAB(I,J))
          RETURN
          END
```

```
        SUBROUTINE LOKUP(ISIZ)
        COMMON TAB(0:255,6),MAX(6),MIN(6),JMP
        DIMENSION II(512)
        J=1
        DO 10 LIN=1,ISIZ
        CALL DSKIO(II,ISIZ,LIN,1,1,8)
        CALL IXPAND(II,ISIZ,1)
*
        DO 20 KOL=1,ISIZ
20      II(KOL)=TAB(II(KOL),J)
        J=J+1
        IF(J.GT.JMP) J=1
        CALL IPRESS(II,ISIZ,1)
10      CALL DSKIO(II,ISIZ,LIN,0,2,8)
        RETURN
        END
```

```
C   <FREI>H8.FOR;8    FRI 16-MAY-75 9:53AM

         SUBROUTINE CHECK(IUNIT)
         COMMON TAB(0:255,6),MAX(6),MIN(6),JMP
         DO 10 J=1,JMP
         WRITE (IUNIT,100),(J,MIN(J),MAX(J))
10       WRITE (IUNIT,200),(TAB(I,J),I=0,255)
100      FORMAT('-',' J=',I2,'  MIN INDEX=',I4,'  MAX INDEX=',I4)
200      FORMAT('-',(16(2X,F6.0)))
         RETURN
         END
```

```
*       FILTER PROGRAM FOR PHASE DATA
*       FOR PICTURES  ISIZ*ISIZ  (256 OR 512)
*       SUBROUTINES : DSKIO (SOSLIB), FILFUN(F,F2)
*       FILFUN IS A USER SPECIFIED EXTERNAL FUNCTION
*       ARGUMENTS  F=SPATIAL FREQUENCY FROM 0. TO ISIZ/2. ;
*       F2=F**2 (USEFULL FOR SOME FILTER FUNCTIONS)
*
*       THE PROGRAM TAKES ADVANTAGE OF THE SYMMETRY IN
*       THE TRANSFORM DOMAIN DUE TO REAL IMAGE DATA
*
*
        DIMENSION A(1024)
        TYPE 100
100     FORMAT(' PICTURE SIZE?  (ENTER 256 OR 512)')
        ACCEPT *,ISIZ
        TYPE 200
200     FORMAT(' ENTER GAIN OF FILTER ')
        ACCEPT *,CST
        KI1=ISIZ/2+1
        KI2=ISIZ*2
        KI3=KI2+2
        KI4=KI3+2
*
*       LINE LOOP 10
        DO 10 LIN=1,KI1
        CALL DSKIO(A,KI2,LIN,1,1,36)
        LIN2=(LIN-1)**2
        DO 20 I=1,KI1
        II=I*2
        IR=II-1
        IIS=KI4-II
        IRS=KI3-IR
        FSQ=LIN2+(I-1)**2
        F=SQRT(FSQ)
        H=FILFUN(F,FSQ,CST)
        A(II)=A(II)*H
        A(IR)=A(IR)*H
        IF(I.EQ.1) GOTO 20
        A(IIS)=A(IIS)*H
        A(IRS)=A(IRS)*H
20      CONTINUE
*       LINE# LIN FILTERED
        CALL DSKIO(A,KI2,LIN,0,2,36)
10      CONTINUE
        CALL FINI
        END
        INTEGER A(512),B(512),C(512),D(512),E(512)
        COMMON ISIZ,C1,C2,C3,DC
*
        TYPE 100
100     FORMAT(' SIZE,MASKING FACTOR (0...1=NO MSK) ?')
        ACCEPT *,ISIZ,DC
*       (ASSUME C2=2*C1 ; C3=1 ) ...  12C1+8C2+C3=DC
        C3=1.
        C2=(DC-C3)/14.
        C1=C2/2.
        TYPE 200,C1,C2,C3
200     FORMAT(' C1,C2,C3=',3F10.4)
*
        DC=(1.-DC)*128.
*
        CALL DS0(1)
        CALL DS0(2)
*
        CALL DS(A,1)
        CALL DS(B,2)
        CALL DS(C,3)
        CALL DS(D,4)
*
        DO 10 L0=5,ISIZ,5
        L1=L0+1
        L2=L0+2
        L3=L0+3
        L4=L0+4
*
        IF(L0.GT.ISIZ) GOTO 20
        CALL DS(E,L0)
        CALL CV(A,B,C,D,E,L0)
*
        IF(L1.GT.ISIZ) GOTO 20
        CALL DS(A,L1)
        CALL CV(B,C,D,E,A,L1)
*
        IF(L2.GT.ISIZ) GOTO 20
        CALL DS(B,L2)
        CALL CV(C,D,E,A,B,L2)
*
        IF(L3.GT.ISIZ) GOTO 20
        CALL DS(C,L3)
        CALL CV(D,E,A,B,C,L3)
*
        IF(L4.GT.ISIZ) GOTO 20
        CALL DS(D,L4)
10      CALL CV(E,A,B,C,D,L4)
*
20      IR=ISIZ-1
        CALL DS0(IR)
        CALL DS0(ISIZ)
```

```
*
        CALL FINI
        END
*
        SUBROUTINE DSO(LIN)
        COMMON ISIZ
        DIMENSION II(512)
        CALL DSKIO(II,ISIZ,LIN,1,1,8)
        CALL DSKIO(II,ISIZ,LIN,0,2,8)
        RETURN
        END
*
        SUBROUTINE DS(A,LIN)
        INTEGER A(512)
        COMMON ISIZ
        CALL DSKIO(A,ISIZ,LIN,1,1,8)
        CALL IXPAND(A,ISIZ,1)
        RETURN
        END
*
        SUBROUTINE CV(A,B,C,D,E,LIN)
        COMMON ISIZ,C1,C2,C3,DC
        INTEGER A(512),B(512),C(512),D(512),E(512),O(512)
*
        O(1)=C(1)
        O(2)=C(2)
        O(ISIZ)=C(ISIZ)
        IR=ISIZ-1
        O(IR)=C(IR)
*
        IR=ISIZ-4
        DO 10 I=1,IR
        I1=I+1
        I2=I+2
        I3=I+3
        I4=I+4
*
        S1=0.
        S2=0.
        DO 20 K=I1,I3
        S1=S1+A(K)+E(K)
20      S2=S2+B(K)+D(K)
        S1=S1+B(I)+C(I)+D(I)+B(I4)+C(I4)+D(I4)
        S2=S2+C(I1)+C(I3)
        S1=DC+C3*C(I2)+C1*S1+C2*S2
        IF(S1.GT.255.) S1=255.
        IF(S1.LT.0.) S1=0.
10      O(I2)=S1
*
        CALL IPRESS(O,ISIZ,1)
        L=LIN-2
        CALL DSKIO(O,ISIZ,L,0,2,8)
        RETURN
        END
```

```
C   <ROBINSON>RATIO.FOR;4   WED 11-JUN-75 4:17PM
C
C   SCALED RATIO OF PIXEL AMP_ITUDES
C
C   INPUT ON UNITS 1 & 2, OUTPUT UN UNIT 3
C
      DIMENSION V(512),W(512),RATIO(512)
      TYPE 495
  495 FORMAT(1X,'ENTER SIZE AND SCALE FACTOR')
      ACCEPT *,N,SCALE
      DO 100 I = 1,N
      CALL DSKIO(V,N,I,1,1)
      CALL EXPAND(V,N,1)
      CALL DSKIO(W,N,I,1,2)
      CALL EXPAND(W,N,1)
      DO 50 J = 1,N
      IF(W(J).EQ.0.) W(J)=1.
      RATIO(J)=ABS(V(J)*SCALE/W(J))
      IF(RATIO(J) .LE. 255.) GO TO 50
      RATIO(J) = 255.
   50 CONTINUE
      CALL PRESS(RATIO,N,1)
      CALL DSKIO(RATIO,N,I,0,3)
  100 CONTINUE
      CALL FINI
      STOP
      END
```

```
        DIMENSION A(512),B(512),C(512)
        REAL LGT(256)
        INTEGER SIZ,BITI,BITO
*
        TYPE 100
100     FORMAT(' ENTER PICSIZE, BITSIZ IN, OUT')
        ACCEPT *,SIZ,BITI,BITO
*
*       SET UP LOG LOOKUP TABLE
        CST=128./ALOG(256.)
        DO 10 I=1,256
        ARG=I
10      LGT(I)=ALOG(ARG)*CST
*
        DO 20 LIN=1,SIZ
        CALL DSKIO(A,SIZ,LIN,1,1,BITI)
        CALL DSKIO(B,SIZ,LIN,1,2,BITI)
        IF(BITI.EQ.36) GOTO 25
        CALL EXPAND(A,SIZ,1)
        CALL EXPAND(B,SIZ,1)
25      DO 30 I=1,SIZ
        I1=A(I)
        I2=B(I)
        C(I)=128.+LGT(I1)-LGT(I2)
        IF(C(I).LT.0.) C(I)=0.
30      IF(C(I).GT.255) C(I)=255
*
        IF(BITO.EQ.36) GOTO 20
        CALL PRESS(C,SIZ,1)
20      CALL DSKIO(C,SIZ,LIN,0,3,BITO)
        CALL FINI
        END
```

```
C   <ROBINSON>KL.FOR;1    TUE 26-AUG-75 2:59PM                    PAGE 1
C
C          PRINCIPAL COMPONENT ANALYSIS OF ERTS PICTURES.
C          PROGRAM DEVELOPED BY GUNER ROBINSON , JUNE 1975 .
C          COMBINES THE PROGRAMS KL1, KL2 AND PC TO FORM ONE PROGRAM.
C          DSKIO UNITS 1-4 ARE FOR THE INPUT FOUR 512*512 BANDS.
C          PRINCIPAL COMPONENTS APPEAR ON UNITS 5-8.
C          UNITS 11-14 ARE USED FOR TEMPORARY STORAGE OF UNNORMALIZED
C          PRINCIPAL COMPONENTS.
C          FORTRAN UNIT 35 IS OUTPUT DATA SET FOR LISTING INFORMATION.
C
C        PROGRAM KL1
C
      DIMENSION DATA(64,64,4),C(4,4),V(512),EVAL(4),VAR(4)
      DIMENSION A(4,4),R(10),CT(4,4),PC(64,64,4)
      DATA IBSIZE/64/,NUMF/4/,  JUMP1/1/, JUMP2/0/, IUNIT /35/
      NBLKS=512/IBSIZE
      BLKS=FLOAT(NBLKS)
      FRAMES=FLOAT(NUMF)
      BSIZE=FLOAT(IBSIZE)
C
C   64 X 64 BLOCK SIZE
C
C   JUMP1 = 0   GIVES INTERMEDIATE PRINTOUTS
C   JUMP1 = 1   ELIMINATES INTERMEDIATE PRINTOUTS
C   JUMP2 = 0   DOES NOT GENERATE OUTPUT IMAGE FILES
C   JUMP2 = 1   WRITES OUTPUT IMAGE FILES
C
C
C
C
      DO 203 I1=1,NUMF
      DO 203 I2=1,NUMF
  203 CT(I1,I2)=0.
C
C
      DO 1000 K = 1,NBLKS
      IROWLO = (K-1)*IBSIZE + 1
      IROWHI = IROWLO + IBSIZE - 1
C
C   REDUCTION OF THE NUMBER OF LINES
C
C
      WRITE (IUNIT,55), IROWLO, IROWHI
   55 FORMAT (///////,10X,'LINES',I5,3X,'THROUGH',I5,3X,'PROCESSED',//)
      DO 999 J = 1,NBLKS
      ICOLLO = (J-1)*IBSIZE + 1
      ICOLHI = ICOLLO + IBSIZE - 1
C
C   REDUCTION OF THE NUMBER OF COLUMNS
C
C
      WRITE (IUNIT,56), ICOLLO, ICOLHI
   56 FORMAT (/////,20X,'COLUMNS',I5,3X,'THROUGH',I5,3X,'PROCESSED',//)
C
C   READ ORIGINAL IMAGE FILES
```

```
C
      DO 100 I = 1,NUMF
      INDEXR = 0
      DO 50 L = IROWLO,IROWHI
      INDEXR = INDEXR + 1
      CALL DSKIO(V,512,L,1,I)
      CALL EXPAND(V,512,1)
      INDEXC = 0
      DO 50 M = ICOLLO,ICOLHI
      INDEXC = INDEXC + 1
   50 DATA(INDEXR,INDEXC,I) = V(M)
      IF (JUMP1 .EQ. 1) GO TO 100
      WRITE (IUNIT,863),I
  863 FORMAT (///,30X,'INPUT ARRAY',5X,'I =',I3,/)
      WRITE (IUNIT,865),((DATA(LL,MM,I), LL = 1,16), MM = 1,16)
  100 CONTINUE
  865 FORMAT(1X,16F8.1)
C
C  COMPUTE SPECTRAL MEAN AND VARIANCES
C
      DO 120 I=1,NUMF
      EVAL(I)=0.
      VAR(I)=0.
      DO 130 L=1,IBSIZE
      DO 130 M=1,IBSIZE
  130 EVAL(I)= EVAL(I)+DATA(L,M,I)
  120 EVAL(I)=EVAL(I)/(BSIZE*BSIZE)
      DO 125 I=1,NUMF
      DO 140 L=1,IBSIZE
      DO 140 M=1,IBSIZE
  140 VAR(I) = VAR(I) +(DATA(L,M,I)-EVAL(I))*
     *(DATA(L,M,I)-EVAL(I))
  125 VAR(I) = VAR(I)/(BSIZE*BSIZE)
      IF(JUMP1.EQ.1) GO TO 149
      WRITE(IUNIT,806)
  806 FORMAT(///,30X,'SPECTRAL MEAN ARRAY',//)
      WRITE(IUNIT,856),(EVAL(LL),LL=1,4)
      WRITE(IUNIT,807)
  807 FORMAT(///,30X,'SPECTRAL VARIANCE ARRAY'//)
      WRITE(IUNIT,856),(VAR(LL),LL=1,4)
  149 CONTINUE
C
C  COMPUTE SPECTRAL COVARIANCE ARRAY
C
      NUMF1=NUMF-1
      DO 150 I1=1,NUMF
  150 C(I1,I1)=VAR(I1)
      DO 200 I1=1,NUMF1
      DO 205 I2=I1+1,NUMF
      SUM=0.
      DO 210 L = 1,IBSIZE
      DO 210 M = 1,IBSIZE
  210 SUM = SUM + (DATA(L,M,I1)-EVAL(I1))*(DATA(L,M,I2)-EVAL(I2))
      SUM=SUM/(BSIZE*BSIZE)
  205 C(I1,I2)=SUM
  200 CONTINUE
      DO 250 I1=1,NUMF
      DO 250 I2=I1+1,NUMF
  250 C(I2,I1)=C(I1,I2)
C
C  STORE FOR COMPUTING TYPICAL COVARIANCE MATRIX
C
      DO 260 I1=1,NUMF
      DO 260 I2=1,NUMF
  260 CT(I1,I2)=CT(I1,I2)+C(I1,I2)
      IF(JUMP1.EQ.1) GO TO 999
      WRITE (IUNIT,808)
  808 FORMAT(///,30X,'SPECTRAL COVARIANCE ARRAY',//)
  856 FORMAT (1X,4F20.5)
      WRITE (IUNIT,856),((C(LL,MM), LL = 1,4), MM = 1,4)
      CALL MSTR(C,R,NUMF,0,1)
      CALL EIGEN(R,A,NUMF,0)
      WRITE(IUNIT,811)
  811 FORMAT(///,30X,' THE  KL  MATRIX',//)
      WRITE(IUNIT,856),((A(LL,MM), LL=1,4), MM=1,4)
      WRITE(IUNIT,812)
  812 FORMAT(//,30X,' THE  EIGENVALUES',/)
      DO 255 I1=1,NUMF
      DO 255 I2=1,NUMF
  255 C(I1,I2)=0.
      CALL MSTR(R,C,NUMF,1,0)
      WRITE(IUNIT,856),(C(LL,LL),LL=1,4)
  999 CONTINUE
 1000 CONTINUE
C
C  COMPUTE TYPICAL COVARIANCE MATRIX
C
      DO 1260 I1=1,NUMF
      DO 1260 I2=1,NUMF
      CT(I1,I2)=CT(I1,I2)/(BLKS*BLKS)
 1260 C(I1,I2)=CT(I1,I2)/SQRT(CT(I1,I1))/SQRT(CT(I2,I2))
      WRITE (IUNIT,818)
  818 FORMAT(///,30X,'AVERAGED SPECTRAL COVARIANCE MATRIX',//)
  819 FORMAT(///,30X,'NORMALIZED SPECTRAL COVARIANCE MATRIX',//)
      WRITE (IUNIT,856),((CT(LL,MM), LL = 1,NUMF), MM = 1,NUMF)
      WRITE(IUNIT,819)
      WRITE (IUNIT,856),((C(LL,MM), LL = 1,NUMF), MM = 1,NUMF)
      CALL MSTR(CT,R,NUMF,0,1)
      CALL EIGEN(R,A,NUMF,0)
      WRITE(IUNIT,811)
      WRITE(IUNIT,856),((A(LL,MM), LL=1,4), MM=1,4)
      WRITE(IUNIT,812)
      DO 1255 I1=1,NUMF
      DO 1255 I2=1,NUMF
 1255 CT(I1,I2)=0.
      CALL MSTR(R,CT,NUMF,1,0)
      WRITE(IUNIT,856),(CT(LL,LL),LL=1,4)
C
C  WRITE OUT THE KL MATRIX
C
C       DO 1257 I1=1,NUMF
C       DO 1258 I2=1,NUMF
C 1258 R(I2)=A(I1,I2)
C 1257 CALL DSKIO(R,NUMF,I1,0,20,36)
        CALL KL2( IBSIZE , NUMF , A ,  IUNIT )
        CALL PCPROG( NUMF  , IUNIT )
        CALL FINI
      STOP
      END
C
        SUBROUTINE      KL2( IBSIZE , NUMF , A , IUNIT )
      DIMENSION DATA(64,64,4),C(4,4),V(512),EVAL(4),VAR(4)
      DIMENSION A(4,4),R(4),CT(4,4),PC(64,64,4)
        DATA JUMP1 / 1 / , JUMP2 / 1 /
      NBLKS=512/IBSIZE
      BLKS=FLOAT(NBLKS)
      FRAMES=FLOAT(NUMF)
      BSIZE=FLOAT(IBSIZE)
C
C  64 X 64 BLOCK SIZE
C
C  JUMP1 = 0  GIVES INTERMEDIATE PRINTOUTS
C  JUMP1 = 1  ELIMINATES INTERMEDIATE PRINTOUTS
C  JUMP2 = 0  DOES NOT GENERATE OUTPUT IMAGE FILES
C  JUMP2 = 1  WRITES OUTPUT IMAGE FILES
C
  55 FORMAT (///////,10X,'LINES',I5,3X,'THROUGH',I5,3X,'PROCESSED',//)
  56 FORMAT (/////,20X,'COLUMNS',I5,3X,'THROUGH',I5,3X,'PROCESSED',//)
 865 FORMAT(1X,16F8.1)
 856 FORMAT (1X,4F20.5)
C
C  READ IN THE KL MATRIX
C
C       DO 1257 I1=1,NUMF
C       CALL DSKIO(R,NUMF,I1,1,20,35)
C       DO 1258 I2=1,NUMF
C 1258 A(I2,I1)=R(I2)
C 1257 CONTINUE
      WRITE(IUNIT,811)
 811 FORMAT(///,30X,' THE  KL  MATRIX',//)
      WRITE(IUNIT,856),((A(LL,MM), LL=1,NUMF), MM=1,NUMF)
        CALL DOPEN( 11 , "1000000 , 'PC1.DAT;-1;T' )
        CALL DOPEN( 12 , "1000000 , 'PC2.DAT;-1;T' )
        CALL DOPEN( 13 , "1000000 , 'PC3.DAT;-1;T' )
        CALL DOPEN( 14 , "1000000 , 'PC4.DAT;-1;T' )
```

```
C
C  CONSTRUCT PRINCIPAL COMPONENTS
C
      DO 2000 K = 1,NBLKS
      IROWLO = (K-1)*IBSIZE + 1
      IROWHI = IROWLO + IBSIZE - 1
C
C  RECONSTRUCT  ROWS
C
      WRITE (IUNIT,55), IROWLO,IROWHI
      IFIRST = 0
      DO 1999 J = 1,NBLKS
      ICOLLO = (J-1)*IBSIZE + 1
      ICOLHI = ICOLLO + IBSIZE - 1
C
C  RECONSTRUCT  COLUMNS
C
      WRITE(IUNIT,56),ICOLLO,ICOLHI
C
C  READ ORIGINAL IMAGE FILES
C
      DO 1100 I = 1,NUMF
      INDEXR = 0
      DO 1050 L = IROWLO,IROWHI
      INDEXR = INDEXR + 1
      CALL DSKIO(V,512,L,1,I)
      CALL EXPAND(V,512,1)
      INDEXC = 0
      DO 1050 M = ICOLLO,ICOLHI
      INDEXC = INDEXC + 1
 1050 DATA(INDEXR,INDEXC,I) = V(M)
      IF(JUMP1 .EQ. 1) GO TO 1100
      WRITE(IUNIT,863),I
  863 FORMAT(///,30X,'INPUT ARRAY',5X,'I =',I3,/)
      WRITE(IUNIT,865),((DATA(LL,MM,I), LL = 1,16), MM = 1,16)
 1100 CONTINUE
C
C  PRINCIPAL COMPONENTS
C
      DO 508 L=1,IBSIZE
      DO 508 M=1,IBSIZE
      DO 509 I=1,NUMF
      PC(L,M,I)=0.
      DO 509 JI=1,NUMF
  509 PC(L,M,I)=PC(L,M,I)+A(JI,I)*DATA(L,M,JI)
  508 CONTINUE
      DO 1515 I=1,NUMF
C     DO 1510 L=1,IBSIZE
C     DO 1510 M=1,IBSIZE
C     PC(L,M,I)=ABS(PC(L,M,I))
```

```
C1510 IF(PC(L,M,I).GT.255.) PC(L,M,I)=255.
      IF(JUMP1.EQ.1) GO TO 1530
      WRITE(IUNIT,864),I
  864 FORMAT(///,30X,'PRINCIPAL COMPONENT',5X,'I =',I3,/)
      WRITE(IUNIT,865),((PC(LL,MM,I),LL=1,16),MM=1,16)
 1530 IF(JUMP2 .EQ. 0) GO TO 1515
      INDEXR = 0
      DO 1536 L = IROWLO,IROWHI
      INDEXR = INDEXR + 1
C      N = I + NUMF
        N = 10 + I
      IF(IFIRST .EQ. 1) GO TO 1546
      CALL DSKIO(V,512,L,1,1)
      CALL EXPAND(V,512,1)
      GO TO 1547
 1546 CALL DSKIO(V,512,L,1,N,36)
 1547 CONTINUE
      INDEXC = 0
      DO 1535 M = ICOLLO,ICOLHI
      INDEXC = INDEXC + 1
      V(M) = PC(INDEXR,INDEXC,I)
 1535 CONTINUE
      CALL DSKIO(V,512,L,0,N,36)
 1536 CONTINUE
 1515 CONTINUE
      IFIRST = 1
 1999 CONTINUE
 2000 CONTINUE
      RETURN
      END
C
C        PROGRAM PC
C
        SUBROUTINE PCPROG( NUMF , IUNIT )
      DIMENSION PC1(512),PC2(512),PC3(512),PC4(512)
      N=512
      FHI=0.
      FNHI=0.
      FLO=100.
      FNLO=255.
      WRITE(IUNIT,98)
   98 FORMAT(//,5X,' MIN AND MAX BEFORE NORMALIZATION',//)
   99 FORMAT(//,5X,' MIN AND MAX AFTER NORMALIZATION',//)
      DO 100 I = 1,N
      CALL DSKIO(PC1,N,I,1,11,36)
      CALL DSKIO(PC2,N,I,1,12,36)
      CALL DSKIO(PC3,N,I,1,13,36)
      CALL DSKIO(PC4,N,I,1,14,36)
      DO 50 J = 1,N
      FHI=AMAX1(PC1(J),PC2(J),PC3(J),PC4(J),FHI)
   50 FLO=AMIN1(PC1(J),PC2(J),PC3(J),PC4(J),FLO)
      WRITE(IUNIT,60),I,FLO,FHI
   60 FORMAT(5X,'LINE=',I4,5X,'MIN=',F8.1,5X,'MAX=',F8.1)
 100 CONTINUE
     WRITE(IUNIT,99)
     DO 200 I = 1,N
     CALL DSKIO(PC1,N,I,1,11,36)
     CALL DSKIO(PC2,N,I,1,12,36)
     CALL DSKIO(PC3,N,I,1,13,36)
     CALL DSKIO(PC4,N,I,1,14,36)
     DO 250 J = 1,N
     PC1(J)=(PC1(J)-FLO)*255./(FHI-FLO)
     PC2(J)=(PC2(J)-FLO)*255./(FHI-FLO)
     PC3(J)=(PC3(J)-FLO)*255./(FHI-FLO)
     PC4(J)=(PC4(J)-FLO)*255./(FHI-FLO)
     FNHI=AMAX1(PC1(J),PC2(J),PC3(J),PC4(J),FNHI)
 250 FNLO=AMIN1(PC1(J),PC2(J),PC3(J),PC4(J),FNLO)
     WRITE(IUNIT,60),I,FNLO,FNHI
     CALL PRESS(PC1,N,1)
     CALL PRESS(PC2,N,1)
     CALL PRESS(PC3,N,1)
     CALL PRESS(PC4,N,1)
     CALL DSKIO(PC1,N,I,0,05)
     CALL DSKIO(PC2,N,I,0,06)
     CALL DSKIO(PC3,N,I,0,07)
     CALL DSKIO(PC4,N,I,0,08)
 200 CONTINUE
     RETURN
     END
```

```
C   <ROBINSON>WIENER.FOR;1   THU 4-SEP-75 11:16AM                     PAGE 1

C         WIENER FILTERING
C
C         THIS PROGRAM FIRST GETS VARIOUS INFORMATION. THEN IT
C         CREATES A MATRIX WHICH WILL BE USED LATER IN THE FILTERING
C         PROCESS.  THEN IT GETS A LINE AND TAKES THE LARGEST POSSIBLE
C         SEGMENT FROM IT (MUST BE A POWER OF 2 PLUS SOME SMALL INTEGER
C         (SAY 14)) BUT IT CANT BE THE WHOLE LINE (BECAUSE THAT MUST BE
C         ALSO A POWER OF 2, ELSE 'POWER' BLOWS UP) SO ONLY ABOUT A
C         QUARTER OF THE PICTURE IS CHANGED.
C         THEN HAVING DONE THE LINES, IF THE PICTURE IS SQUARE AND THE
C         USER HAS APPROVED, THE COLUMN GET THE SAME TREATMENT.

          EXTERNAL RTN
          REAL ARR(1024,13)
          INTEGER OBYTSZ,OUTTYP,OUTDS,HPSTN,VPSTN,VMX2
      DIMENSION INV(256),W(30),S(256),GIRL(1024),M(3)
          EQUIVALENCE (H,ARR(1,1)),(GIRL,ARR(1,5))
          COMPLEX HT,H(1024),KR(1024),G(1024),RSN
          EQUIVALENCE (G(1),KR(1))
          INTEGER WID,LEN,TURN


          INDS = 3
          OUTDS = 4

          DATA M/0,0,0/
          TYPE 1000
1000      FORMAT(' ENTER INPUT FILE NAME-')
          CALL DOPEN(3,"160C03000000,0)
          TYPE 1500
1500      FORMAT (' IS IT REAL(1), OR PACKED(2)?')
          ACCEPT *,INTYP
          IBYTSZ = 64 - 28 * INTYP
          TYPE 2000
2000      FORMAT (' ENTER OUTPUT FILE NAME-')
          CALL DOPEN(4,"460C03000000,0)
          TYPE 1500
          ACCEPT *,OUTTYP
          OBYTSZ = 64 - 28 * OUTTYP
          NBW = OUTTYP * -3 + 7
          TYPE 1
 1        FORMAT(' ENTER THE BLUR, AND THE CORRELATION COEFFICIENT.')
          ACCEPT *,F,P
23        IF (P .LT. 1.0 .AND. P .GT. 0) GO TO 15
          TYPE 20
20        FORMAT (' THE CORR. COEFF. MUST BE LESS THAN 1.0: REENTER')
          ACCEPT *,P
          GO TO 23
15        TYPE 26
26        FORMAT (' NOW ENTER THE SIGNAL/NOISE RATIO')
          ACCEPT *,SN
```

```
C   BE SURE AND CHECK TO SEE THE RANGE HERE.
          TYPE 3,F,P,SN
 3        FORMAT(' THE BLUR, THE CORR. COEFF., THE SIGNAL TO
     * RATIO'/2F10.2,E12.3)
          TYPE 3000
3000      FORMAT (' ENTER LENGTH AND WIDTH OF PICTURE-')
          ACCEPT *,LEN,WID
          N = WID
C   HERE IS WHERE ONE DECIDES IF WANT TOO USE WHOLE THING
          CALL POWER(N,M(1))
          TURN = 1
          IF (WID .NE. LEN) GO TO 3500
          TYPE 3450
3450      FORMAT (' DO YOU WANT ONE OR TWO DIMENSIONAL DEBLUR?')
          ACCEPT *,TURN
C GENERATE THE IMPULSE RESPONSE OF THE BLUR
3500      TYPE 350
350       FORMAT (' DO YOU WANT THE CHANGED PORTION TO BE '/'
     A     (1) AT THE RIGHT, (2) IN THE MIDDLE OR (3) ON THE LEFT?')
          ACCEPT *,HPSTN
          IF (TURN .EQ. 1) GO TO 3550
          TYPE 255
255       FORMAT ( ' DO YOU WANT IT (1) AT THE TOP, (2) IN THE CENTER'/'
     A     OR (3) AT THE BOTTOM OF THE PICTURE?')
          ACCEPT *,VPSTN
3550      TYPE 4000
4000      FORMAT (' ENTER ''IMPULSE RESPONSE'' MATRIX SIZE')
          ACCEPT *,L

          IF (L .GT. 31) GO TO 3500

          NL = N/2 + 1
          NK = NL + L-1
          FL=(L-1.)/2.
          LF = FL + .5
          IF (HPSTN .EQ. 2) MX2 = N/4 - 1
          IF (HPSTN .EQ. 1) MX2 = LF+1
          IF (HPSTN .EQ. 3) MX2 = WID -LF-NL
          MX = MX2 - LF
          IF (VPSTN .EQ. 2) VMX2 = LEN/2 - N/4 -1
          IF (VPSTN .EQ. 1) VMX2 = LF+1
          IF (VPSTN .EQ. 3) VMX2 = LEN -LF-NL
```

```
C         NOW HAVE GOTTEN ENOUGH INFO TO START. SO BEGIN CALCULATIONS
          DO 42 I=1,N
 42       H(I) = 0.0

          H(LF + 1) = 1.0
          T1=0
          IF(F .EQ. 0.) GO TO 40
          DO 5 I=1,LF
          A1=(I-(FL+1.))**2/(2*F*F)
          IF (A1.GT.(89.416))GO TO 5
          H(I)=EXP(-A1)
  5       T1=T1+H(I)
          T1 = T1 * 2.0 + 1 - 2*(LF-FL)
          DO 4 I=1,LF+1
  4       H(I) = H(I)/T1
          DO 55 I = 1,LF
 55       H(L+1-I) = H(I)
 40       LL=L-1
          W(1)=H(1)
          DO 30 I=2,LL
  30      W(I)=W(I-1)+H(I)
          CALL HARM(H,M,INV,S,1,IFERR)
C GENERATE THE CORRELATION MATRIX
          IF (SN .GE. 10000000000.0) GO TO 135
          KR(1) = 1.0
          DO 8 I=2,NL
  8       KR(I)=P*KR(I-1)
          N3=NL-2
          DO 9 I=1,N3
  9       KR(NL+I)=KR(NL-I)
          CALL HARM(KR,M,INV,S,2,IFERR)
          DO 13 I=1,N
          HT = CONJG(H(I))
          RSN=1.0/(KR(I)*SN)
 13       G(I)=HT/(H(I)*HT+RSN)
          GO TO 139
135       DO 137 I=1,N
137       G(I) = 1.0/H(I)
139       CONTINUE
          NKX = NK + 1


C         THE FILTER HAS BEEN SET UP FOR USE. NOW READ IN
C         THE PICTURE AND USE IT.

          CALL DOPEN(5,"610001000000,'50KSD.TAD')
          CALL DOPEN(6,"610001000000,'60KSD.TAD')
```

```
222       CONTINUE


       DO 16 I=1,LEN
       CALL DSKIO(GIRL,WID,I,1,INDS,IBYTSZ)
          IF (INTYP.EQ.2) CALL EXPAND(GIRL,WID,1)
          IF (I .LT. VMX2 .OR. I .GT. VMX2+NL) GO TO (67,93) OUTTYP
          DO 333 J=L,NK-LL
 333      H(J)=GIRL(J+MX)
          DO 96 J=NKX,N
96        H(J)=0
          DO 25 J=2,LL
          H(J)=GIRL(J+MX)-W(L-J)*GIRL(1+MX)
 25       H(NK+1-J)=GIRL(NK+1-J+MX)-W(L-J)*GIRL(NK+MX)
          H(1)=W(1)*GIRL(1+MX)
          H(NK)=W(1)*GIRL(NK+MX)
           CALL HARM(H,M,INV,S,2,IFERR)
          DO 18 J=1,N
 18       H(J)=G(J)*H(J)
          CALL HARM(H,M,INV,S,-2,IFERR)
          IF (OUTTYP .EQ. 1) GO TO 63
          DO 60 J=2,NL-1
          A = H(J)
          IF (A.LT.0) A=0
          IF (A .GT. 255.0) A=255.0
60        GIRL(J+MX2) = A
93        CALL PRESS(GIRL,WID,1)
          GO TO 67
63        DO 65 J=2,NL-1
65        GIRL(J+MX2) = H(J)
67        CALL DSKIO(GIRL,WID,I,0,OUTDS,OBYTSZ)
 16       CONTINUE

          IF (TURN .EQ. 1) GO TO 99999
          INDS = OUTDS
          OUTDS = 5
          IF (TURN .EQ. 3) OUTDS = 4
          CALL TRNSPS(ARR,1024*33,LEN,WID,NBW,RTN,RTN,INDS,OUTDS,6)
          IBYTSZ = OBYTSZ
          IF (TURN .EQ. 3) GO TO 99999
          INDS = 5
          TURN = 3
          I   = MX2
          MX2 = VMX2
          MX = MX2 - LF
          VMX2 = I
          GO TO 222

99999     CALL FINI
          STOP
          END


          SUBROUTINE RTN
          RETURN
          END
```

```
        COMMON TAB(0:255),IGAM(0:255),ISIZ
        TYPE 200
200     FORMAT(' SIZE? (256 OR 512)')
        ACCEPT *,ISIZ
*       SET UP DENSITY LOCK-UP TABLE
        CALL DENS
*       SET UP GAMMA LOCK-UP TABLE
        CALL GAMM
*       WRITE LINEAR DENSITY SCALE
        CALL SCAL
        TYPE 100
100     FORMAT(' SCALE DONE')
*       PRE-DISTORT FOR DENSITY,VIGNETTING AND GAMMA
        CALL VIGN
        END
*
        SUBROUTINE DENS
*       SETS UP DENSITY LOOK-UP TABLE
        COMMON TAB(0:255),IGAM(0:255),ISIZ
        DIMENSION A(9),V(9)
        DATA A/4.,13.,37.,57.,72.,86.,95.,104.,110./
        DATA V/0.,17.,51.,85.,119.,153.,187.,221.,255./
*
        SC=106./255.
        I1=1
        I2=2
        SL=(V(I2)-V(I1))/(A(I2)-A(I1))
*
        DO 10 I=0,255
        X=4.+I*SC
        IF(X.LE.A(I2)) GOTO 20
        I1=I2
        I2=I2+1
        SL=(V(I2)-V(I1))/(A(I2)-A(I1))
*
20      DX=X-A(I1)
10      TAB(I)=V(I1)+SL*DX
        RETURN
        END
*
        SUBROUTINE GAMM
        COMMON TAB(0:255),IGAM(0:255),ISIZ
        CST=SQRT(255.)
        DO 10 I=0,255
        ARG=I
10      IGAM(I)=CST*SQRT(ARG)
        RETURN
        END
*
        SUBROUTINE SCAL
        COMMON TAB(0:255),IGAM(0:255),ISIZ
```

```
C   <FREI>PHOTO.FOR;12    SAT 24-MAY-75 2:15PM                      PAGE 1:1
*       SUPERIMPOSES A LINEAR GREY SCALE ON PACKED DATASETS
        INTEGER OUT(512)
        DIMENSION ISC(16)
        DATA ISC/255,238,221,204,187,170,153,136,
     F119,102,85,68,51,34,17,0/
        IF(ISIZ.EQ.256) ISC(15)=0
*
        IS=ISIZ/32
        DO 10 I=1,IS
        ISS=I
        IF(ISIZ.EQ.256) ISS=I*2-1
        I32=(I-1)*32
        DO 10 J=1,32
        IND=I32+J
10      OUT(IND)=ISC(ISS)
        CALL IPRESS(OUT,ISIZ,1)
        LUP=ISIZ-20*ISIZ/256
        DO 20 L=LUP,ISIZ
20      CALL DSKIO(OUT,ISIZ,L,0,1,8)
        END
*
        SUBROUTINE VIGN
        COMMON TAB(0:255),IGAM(0:255),ISIZ
        DIMENSION II(512),A(512)
*
        C1=1.25
        C2=(C1-1.)/((ISIZ**2)/2.)
*
        M=ISIZ/2
        DO 10 I=1,ISIZ
10      A(I)=C2*(I-M)**2
*
        DO 20 LIN=1,ISIZ
        CALL DSKIO(II,ISIZ,LIN,1,1,8)
        CALL IXPAND(II,ISIZ,1)
25      DO 30 IP=1,ISIZ
*       DENSITY LOCK-UP
        DEN=TAB(II(IP))
*       VIGNETTING
        IVIG=DEN/(C1-A(LIN)-A(IP))
        IF(IVIG.GT.255.) IVIG=255.
30      II(IP)=IGAM(IVIG)
        CALL IPRESS(II,ISIZ,1)
20      CALL DSKIO(II,ISIZ,LIN,0,2,8)
        CALL FINI
        END
```