General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.
A procedure is developed to extract cross-band textural features from ERTS multi-spectral scanner imagery. Evolving from a single-image texture extraction procedure which uses spatial dependence matrices to measure relative co-occurrence of nearest neighbor grey tones, the cross-band texture procedure uses the distribution of neighboring grey tone N-tuple differences to measure the spatial inter-relationships, or co-occurrences, of the grey tone N-tuples present in a texture pattern. In both procedures, texture is characterized in such a way as to be invariant under linear grey tone transformations. However, the cross-band procedure compliments the single-image procedure by extracting texture information and spectral information contained in ERTS multi-images. Classification experiments show that when used alone, without spectral processing, the cross-band texture procedure extracts more information than the single-image texture analysis. Results show an improvement in average correct classification from 86.2% to 88.3% for ERTS image no. 1021-16333 with the cross-band texture procedure. However, when used together with spectral features, the single-image texture plus spectral features perform better than the cross-band texture plus spectral features, with an average correct classification of 93.8% and 91.6% respectively.
KANSAS ENVIRONMENTAL AND RESOURCE STUDY: A GREAT PLAINS MODEL

SPECTRAL AND TEXTURAL PROCESSING OF ERTS IMAGERY

PREFACE

A procedure is developed to extract cross-band textural features from ERTS multi-spectral scanner imagery. Evolving from a single-image texture extraction procedure which uses spatial dependence matrices to measure relative co-occurrence of nearest neighbor grey tones, the cross-band texture procedure uses the distribution of neighboring grey tone N-tuple differences to measure the spatial inter-relationships, or co-occurrences, of the grey tone N-tuples present in a texture pattern. In both procedures, texture is characterized in such a way as to be invariant under linear grey tone transformations. However, the cross-band procedure compliments the single-image procedure by extracting texture information and spectral information contained in ERTS multi-images. Classification experiments show that when used alone, without spectral processing, the cross-band texture procedure extracts more information than the single-image texture analysis. Results show an improvement in average correct classification from 86.2% to 88.8% for ERTS image no. 1021-16333 with the cross-band texture procedure. However, when used together with spectral features, the single-image texture plus spectral features perform better than the cross-band texture plus spectral features, with an average correct classification of 93.8% and 91.6%, respectively.
KANSAS ENVIRONMENTAL AND RESOURCE STUDY: A GREAT PLAINS MODEL
MARCH 1974
Type III Final Report for the Period
August 1, 1972 - March 17, 1974

Prepared for:
National Aeronautics and Space Administration
Goddard Spaceflight Center
Greenbelt, Maryland 20771

Contract No. NAS 5-21822

RECEIVED
NOV 08 1974
SIS/902.6

THE UNIVERSITY OF KANSAS CENTER FOR RESEARCH, INC.
2385 Irving Hill Rd.—West Campus Lawrence, Kansas 66044
Use of Feature Extraction Techniques for the Texture and Context Information in ERTS Imagery:

Spectral and Textural Processing of ERTS Imagery

R. M. Haralick, Principal Investigator
University of Kansas Center for Research, Inc.
Remote Sensing Laboratory
c/o Space Technology Center
Nichols Hall
2291 Irving Hill Drive-Campus West
Lawrence, Kansas 66045

March 17, 1974
Type III Final Report for the Period August 1, 1972 - March 17, 1974

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Goddard Spaceflight Center
Greenbelt, Maryland 20771
Type III Final Report
for the period: August 1, 1972 - March 17, 1974

Title of Investigation: Use of Feature Extraction Techniques for Texture Context Information in ERTS Imagery

ERTS-A Proposal Number: 60-1
Task Number: 1
Co-Investigators: R. M. Haralick and G. L. Kelly

NASA-GSFC PI ID Number: UN 094

Report Prepared by:

Robert J. Bosley
Research Assistant

Report Approved by:

R. M. Haralick
Co-Principal Investigator
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. TEXTURE</td>
<td>2</td>
</tr>
<tr>
<td>III. REVIEW OF PAST WORK ON TEXTURE</td>
<td>3</td>
</tr>
<tr>
<td>IV. TEXTURAL FEATURES</td>
<td>4</td>
</tr>
<tr>
<td>IV.1 Spatial Grey Tone Dependence Matrices</td>
<td>4</td>
</tr>
<tr>
<td>IV.2 Textural Features for Multi-Images</td>
<td>11</td>
</tr>
<tr>
<td>IV.3 Textural Feature Extraction Procedure</td>
<td>14</td>
</tr>
<tr>
<td>V. GROUND TRUTH ASSIGNMENT PROCEDURE</td>
<td>17</td>
</tr>
<tr>
<td>VI. IDENTIFICATION PROCEDURE</td>
<td>19</td>
</tr>
<tr>
<td>VII. RESULTS OF CLASSIFICATION EXPERIMENTS</td>
<td>23</td>
</tr>
<tr>
<td>VII.1 Summary of Classification Results</td>
<td>30</td>
</tr>
<tr>
<td>VIII. CONCLUSION</td>
<td>31</td>
</tr>
<tr>
<td>APPENDIX I - Textural Features Obtained from the Grey Tone Dependence Matrix</td>
<td>47</td>
</tr>
<tr>
<td>APPENDIX II - N-Dimensional Spherical Coordinate Systems and Ellipsoidally Symmetric Distribution</td>
<td>51</td>
</tr>
<tr>
<td>APPENDIX III - Normalization Procedure to Make Covariance Matrix Invariant Under Translating and Scaling Transformations</td>
<td>65</td>
</tr>
<tr>
<td>APPENDIX IV - Computer Program Documentation &amp; Listings</td>
<td>67</td>
</tr>
<tr>
<td>APPENDIX V - Glossary and Index to Remotely Sensed Image Pattern Recognition Concepts</td>
<td>247</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>254</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Figure 1</td>
<td>Eight Nearest Neighbor Resolution Cells</td>
</tr>
<tr>
<td>Figure 2</td>
<td>The Set of All Distance 1 Horizontal Neighboring Resolution Cells on a 4 × 4 Image</td>
</tr>
<tr>
<td>Figure 3</td>
<td>The General Form of Any Grey Tone Spatial Dependence Matrix</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Textural Feature for Two Different Land-Use Category Images</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Example of a Nearest Neighbor Grey Tone Spatial Dependence Matrix</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Scattergram of the Nearest Neighbor Differences for Distance 1 on MSS Bands 4 and 5</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Histogram the Distribution of Differences</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Retrieval Program Printout of a Portion of Image No. 1021-16333 over Kansas City</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Identification Scheme for Classification</td>
</tr>
<tr>
<td>Figure 10</td>
<td>ERTS Image No. 1002-18134</td>
</tr>
<tr>
<td>Figure 11</td>
<td>ERTS Image No. 1330-16515</td>
</tr>
<tr>
<td>Figure 12</td>
<td>ERTS Image No. 1021-16333</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Cross-Band Texture Feature Component Designation</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Examples of Cross-Band Texture Features</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Comparison of the Performance of the Cross-Band Texture Procedure and the Single-Image Texture Procedure</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Three Dimensional Spherical Coordinate System</td>
</tr>
</tbody>
</table>
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Contingency Table Using the 6 Multi-Image Textural Features Over Monterey Bay, California</td>
<td>34</td>
</tr>
<tr>
<td>Table 2</td>
<td>Results of Land-Use Classification Experiments Over Monterey Bay, California</td>
<td>34</td>
</tr>
<tr>
<td>Table 3</td>
<td>Contingency Table Using the 6 Distance 1 Multi-Image Textural Features Over Garden City, Kansas</td>
<td>35</td>
</tr>
<tr>
<td>Table 4</td>
<td>Results of Land-Use Classification Experiments for Distance 1 Over Garden City, Kansas</td>
<td>35</td>
</tr>
<tr>
<td>Table 5</td>
<td>Contingency Table Using the 6 Distance 8 Multi-Image Textural Features Over Garden City, Kansas</td>
<td>36</td>
</tr>
<tr>
<td>Table 6</td>
<td>Results of Land-Use Classification Experiments for Distance 8 Over Garden City, Kansas</td>
<td>36</td>
</tr>
<tr>
<td>Table 7</td>
<td>Contingency Table Using Both Distance 1 and 8 Multi-Image Textural Features Over Garden City, Kansas</td>
<td>37</td>
</tr>
<tr>
<td>Table 8</td>
<td>Results of Land-Use Classification Experiments for Distances 1 and 8 over Garden City, Kansas</td>
<td>37</td>
</tr>
<tr>
<td>Table 9</td>
<td>Contingency Table Using Distance 1 Multi-Image Textural Features Over Garden City, Kansas, with the Bayes Classifier</td>
<td>38</td>
</tr>
<tr>
<td>Table 10</td>
<td>Contingency Table Using Distance 8 Multi-Image Textural Features Over Garden City, Kansas, with the Bayes Classifier</td>
<td>38</td>
</tr>
<tr>
<td>Table 11</td>
<td>Contingency Table Using the Bayes Classifier for 9 of the 29 Cross-Band Textural Features Over Garden City, Kansas</td>
<td>39</td>
</tr>
<tr>
<td>Table 12</td>
<td>Contingency Table Using All 17 Distance 1 Texture Features on MSS Band 5 Over Kansas City</td>
<td>40</td>
</tr>
<tr>
<td>Table 13</td>
<td>Contingency Table Using All 29 Distance 1 Cross-Band Texture Features Over Kansas City</td>
<td>40</td>
</tr>
<tr>
<td>Table 14</td>
<td>Contingency Table Using the 4 Spectral Features Over Kansas City</td>
<td>41</td>
</tr>
<tr>
<td>Table 15</td>
<td>Contingency Table Using All 17 Texture Features Plus the 4 Spectral Features Over Kansas City</td>
<td>41</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 16. Contingency Table Using the First 26 of the 29 Cross-Band Texture Features Plus the 4 Spectral Features Over Kansas City 42
Table 17. Contingency Table Using the First 8 of the 17 Texture Features from MSS Band 5 Over Kansas City 42
Table 18. Contingency Table Using the First 9 of the 29 Cross-Band Texture Features Over Kansas City 43
Table 19. Contingency Table Using the First 9 of the 29 Cross-Band Texture Features Over Kansas City with the Bayes Classifier 43
Table 20. Contingency Table Using the First 6 of the 29 Cross-Band Texture Features Plus the 4 Spectral Features Over Kansas City 44
Table 21. Contingency Table Using the First 9 of the 17 Texture Features from Band 5 plus the 4 Spectral Features Over Kansas City 44
Table 22. Contingency Table Using the First 11 of the 29 Cross-Band Texture Features Plus the 4 Spectral Features Over Kansas City 45
Table 23. Contingency Table Using the First 16 of the 29 Cross-Band Texture Features Plus the 4 Spectral Features Over Kansas City 45
Table 24. Contingency Table Using the First 21 of the 29 Cross-Band Texture Features Plus the 4 Spectral Features Over Kansas City 46
I. INTRODUCTION

A procedure is developed to extract textural features for automatic analysis of ERTS multi-spectral scanner imagery. Previous work (Haralick, 1973; Haralick, 1972) indicates that useful textural features can be computed from the co-occurrence matrices for grey tones in specific spatial relationships on an image. The performance of the land-use classification algorithm using these textural features from only one band is encouraging; up to 75 per cent of the images were correctly classified (Haralick, 1973). Since textural features and spectral features of ERTS multi-images provide different kinds of information, a significant increase in identification accuracy will occur when both features are used together.

Adoption of the texture procedure for multi-images leads to excessive amounts of storage for the grey tone N-tuple co-occurrence matrices. Therefore, to solve the storage problem we measure grey tone N-tuple differences instead of grey tone N-tuples and assume an ellipsoidally symmetric functional form for the co-occurrence distribution of multi-image grey tone N-tuple differences.

It is expected that the estimated parameters of the ellipsoidally symmetric distribution will lead to textural features that can distinguish between texturally distinct categories on ERTS MSS images over Kansas. In order to obtain more texture information, the dimensionality of the grey tone N-tuples was increased from the original four MSS bands by the addition of cross-band product terms for higher order components. This procedure for cross-band texture analysis of multi-images provides a natural extension of the single-image texture analysis while retaining its advantages: invariance under translating and scaling transformations, low storage requirements, and direct proportionality between the number of operations required to process an image and the number of resolution cells present in an image.
Texture and tone are two fundamental pattern elements used in the interpretation of image data. The concept of tone is concerned with the whiteness, greyness, or blackness of resolution cells of the image. The concept of texture is concerned with the spatial distribution of the grey tones. Tone is based upon the varying shades of grey of the resolution cells in the image, while texture is based upon the spatial distribution of grey tones. However, texture and tone are not independent concepts but are intrinsically related to one another. Although either property can dominate the other depending upon the image context, texture and tone are always present.

When one attempts to objectively use tone and texture pattern elements, the texture-tone concept must be explicitly defined. This can be visualized as follows. When a small area patch of an image has little variation of features of discrete grey tone, then that area is dominated by tonal properties. As the number of distinguishable features of discrete grey tone increases within the patch, then the texture properties will dominate. The size of the small area patch, the relative sizes of the discrete features and the number of distinguishable discrete features are all crucial in this distinction. When the size of the small area patch is reduced to one resolution cell, the only property present is tone. When there is no spatial pattern in the tonal features and the grey tone variation between features is wide, a fine texture results. And as the spatial pattern becomes more defined using more and more resolution cells, then a coarser texture results.

Texture can be termed as being fine, coarse, smooth, rippled, mottled irregular, or lineated. Texture is a property of nearly all surfaces, the grain of wood, the weave of fabric, the pattern of crops in a field, etc. Although texture is quite easy for humans to recognize and describe, it is quite subjective by its nature and is extremely difficult to precisely define and analyze by digital computers. Since the texture of images contains important information for discrimination purposes, textural features could be very useful.
III. REVIEW OF PAST WORK ON TEXTURE

To date there has been at least six different approaches to the problems of measuring and characterizing texture of images: autocorrelation functions, optical transforms, digital transforms, edgeness, structural elements, and spatial grey tone co-occurrence probabilities. The first three approaches all measure spatial frequency either directly or indirectly. Spatial frequency is related to texture because fine textures are rich in high frequencies while coarse textures are rich in low frequencies.

One alternative approach to viewing texture as spatial frequency distribution is to view texture as the amount of edgeness per unit area. Fine textures have a high number of edges per unit area whereas coarse textures have a small number of edges per unit area.

The structural element approach uses a matching procedure to detect the spatial regularity of shapes called structural elements in a binary image. When the structural elements themselves are single resolution cells, the information provided by this approach is the autocorrelation function of the binary image. By using larger and more complex shapes, a more generalized autocorrelation can be computed.

The grey tone spatial dependence approach characterizes texture by the spatial distribution of its grey tones. In coarse textures the distribution changes only slightly with distance, but for fine textures it changes rapidly with distance.

Because of our familiarity with the concepts of spatial frequency and edgeness, these approaches to texture characterizations are readily employed. However, an inherent problem exists with these approaches in regard to grey tone calibration of the image and they are not invariant under even a linear grey tone translation. And the price paid for invariance by compensating with quantization is a loss of grey tone precision in the quantized image.

The power of the structural element approach is that it emphasizes the shape aspects of the discrete tonal features. Weakness of this approach lies in that it can only do so for binary images.

The power of the spatial grey tone co-occurrence approach lies in characterizing the spatial inter-relationships of the grey tones in a texture pattern in such a way that is invariant under monotonic grey tone transformations. Weakness of the approach lies in failure to capture the shape aspects of the discrete tonal features.
IV. TEXTURAL FEATURES

The above description of texture is an idealization of what actually occurs, a gross simplification. Discrete tonal features are actually quite subjective in that they do not necessarily stand out as entities by themselves. Therefore, the texture analysis presented here is concerned with more general or macroscopic concepts rather than discrete tonal features. The procedure developed by Haralick (Haralick, 1972) for obtaining the textural features of an image is based on the assumption that the texture information on an image is contained in the overall spatial co-occurrence relationship which the greytones in the image have to one another. More specifically, we assume that this texture information is adequately specified by a set of spatial grey tone dependence matrices, which are computed for various angular relationships and distances between neighboring resolution cell pairs on the image. All of the textural features are then derived from these angular nearest neighbor spatial grey tone dependence matrices.

IV.1 Spatial Grey Tone Dependence Matrices

Let $G = \{0, 1, \ldots, Ng\}$ be the set of possible grey tones that each resolution cell can take on after image normalization by equal probability quantizing to $Ng$ levels. It can be shown that this quantization guarantees that images which are a monotonic transformation of one another, such as lighter or darker images due to variations in film, lighting, or development, will produce the same results. Let $Nx$ be the number of resolution cells in the horizontal direction and $Ny$ the number of resolution cells in the vertical direction in the image to be analyzed so that $Lx = \{1, 2, \ldots, Nx\}$ and $Ly = \{1, 2, \ldots, Ny\}$ are the horizontal and vertical spatial domains. Then $Ly \times Lx$ will be the set of resolution cells of the image. And the image $I$ can be represented as a function which assigns some grey tone in $G$ to each resolution cell or pair of coordinates in $Ly \times Lx$; $I: Ly \times Lx \rightarrow G$.

Essential to our conceptual framework of texture are four closely related measures called angular nearest neighbor grey tone spatial dependence matrices. The concept of angular nearest neighbor for a resolution cell is the adjacent resolution cell for a given angle, as shown in Figure 1.
FIGURE 1. Eight nearest neighbor resolution cells of cell '•'. Resolution cells 1 and 5 are the 0-degree nearest neighbors to resolution cell '•', resolution cells 2 and 6 are the 135-degree nearest neighbors, etc. Note that this information is purely spatial, having nothing to do with grey tone values.

We assume that the texture information in our image 1 is contained in the overall or "average" spatial relationship which the grey tones in image 1 have to one another. Specifically, we shall assume that this information is adequately specified by the matrix of relative frequencies $P_{ij}$ with which two neighboring resolution cells separated by distance $d$ occur on the image, one with grey tone $i$ and the other with grey tone $j$. These matrices of spatial grey tone dependence frequencies are a function of the angular relationship between the neighboring resolution cells as well as a function of the distance between them. Figure 2 illustrates the set of all horizontal neighboring resolution cells separated by distance 1. This set along with the image grey tones would be used to calculate a distance $i$ horizontal spatial grey tone co-occurrence matrix. Formally, for angles quantized to 45° intervals the unnormalized frequencies are defined by:

\[
P(i,j,d,0°) = \# \{(k,l), (m,n)) \in (L_y \times L_x \times L_y \times L_x) \mid k-m=0, \ |l-n|=d, \ l(k,l)=i, \ l(m,n)=j\}
\]

\[
P(i,j,d,45°) = \# \{(k,l), (m,n)) \in (L_y \times L_x \times L_y \times L_x) \mid (k-m=d, \ l-n=-d) \text{ or } (k-m=-d, \ l-n=d), \ l(k,l)=i, \ l(m,n)=j\}
\]

\[
P(i,j,d,90°) = \# \{(k,l), (m,n)) \in (L_y \times L_x \times L_y \times L_x) \mid |k-n|=d, \ l(n,l)=0, \ l(k,l)=i, \ l(m,n)=j\}
\]

\[
P(i,j,d,135°) = \# \{(k,l), (m,n)) \in (L_y \times L_x \times L_y \times L_x) \mid (k-m=d, \ l-n=d) \text{ or } (k-m=-d, \ l-n=-d), \ l(k,l)=i, \ l(m,n)=j\}
\]

Note that these matrices are symmetric; $P(i,j,d,a) = P(j,i,d,a)$. The distance metric $d$ implicit in the above equations can be explicitly defined by $d((k,l), (m,n)) = \max \{|k-m|, |l-n|\}$. 

\[d((k,l), (m,n)) = \max \{|k-m|, |l-n|\}.\]
\[ L_y = \{1, 2, 3, 4\} \]

\[ L_x = \{1, 2, 3, 4\} \]

\[
\begin{array}{cccc}
(1,1) & (1,2) & (1,3) & (1,4) \\
(2,1) & (2,2) & (2,3) & (2,4) \\
(3,1) & (3,2) & (3,3) & (3,4) \\
(4,1) & (4,2) & (4,3) & (4,4) \\
\end{array}
\]

\[
R_H = \{(k,1), (m,n) \in (L_y \times L_x) \times (L_y \times L_x) | k-m = 0, |l-n| = 1\}
\]

\[
= \{(1,1), (1,2), (1,1), (1,2), (1,3), (1,3), (1,2), (1,3), (1,4), (1,4), (1,3), (2,2), (2,2), (2,1), (2,1), (2,2), (2,3), (2,3), (2,2), (2,4), (2,4), (2,3), (3,1), (3,2), (3,1), (3,2), (3,3), (3,3), (3,2), (3,3), (3,4), (3,4), (3,3), (4,1), (4,4), (4,2), (4,2), (4,1), (4,2), (4,3), (4,3), (4,2), (4,3), (4,4), (4,4), (4,3)\}
\]

**FIGURE 2.** Illustrates the set of all Distance 1 Horizontal Neighboring Resolution Cells on a 4 x 4 Image.
For an example of the four distance 1 grey tone spatial dependence matrices, consider Figure 3. Figure 3-a represents a $4 \times 4$ image with four grey tones, ranging from 0 to 3. Figure 3-b shows the general form for any grey tone spatial dependence matrix. For example, the element in the $(2, 1)$-st position of the distance 1 horizontal $P_H$ matrix is the total number of times two grey tones of value 2 and 1 occurred horizontally adjacent to each other. To determine this number, we count the number of pairs of resolution cells in $R_H$ such that the first resolution cell of the pair has grey tone 2 and the second resolution cell of the pair has grey tone 1. Figure 3-c through 3-f shows all four distance 1 grey tone spatial dependence matrices.

From the grey tone dependence matrices a set of 17 textural features is derived. The equations defining these 17 features are given in Appendix 1. To illustrate the significance of these features, three are defined as follows:

$$f_1 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \left( \frac{P(i, j)}{\#_R} \right)^2,$$

$$f_2 = \sum_{n=0}^{N_g - 1} n^2 \left\{ \sum_{|i-j|=n} \left( \frac{P(i, j)}{\#_R} \right) \right\},$$

$$f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} ij P(i, j)}{\#_R} - \mu_x \mu_y \frac{\sigma_x \sigma_y}{\#_R},$$

where $\#_R$ = number of resolution cells pairs, and $\mu_x$ and $\sigma_x$ are the mean and standard deviation of the marginal distribution $P_x$ defined by

$$P_x(i) = \sum_{j=1}^{N_g} \frac{P(i, j)}{\#_R},$$

and $\mu_y$ and $\sigma_y$ are the mean and standard deviation of the marginal distribution $P_y$ defined by:

$$P_y(j) = \sum_{i=1}^{N_g} \frac{P(i, j)}{\#_R}.$$
Figure 3-a.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 3-b. This shows the general form of any grey tone spatial dependence matrix for an image with integer grey tone values 0 to 3. #(i, j) stands for number of times grey tones i and j have been neighbors.

<table>
<thead>
<tr>
<th>Grey Tone</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>#(0, 0)</td>
<td>#(0, 1)</td>
<td>#(0, 2)</td>
<td>#(0, 3)</td>
</tr>
<tr>
<td>1</td>
<td>#(1, 0)</td>
<td>#(1, 1)</td>
<td>#(1, 2)</td>
<td>#(1, 3)</td>
</tr>
<tr>
<td>2</td>
<td>#(2, 0)</td>
<td>#(2, 1)</td>
<td>#(2, 2)</td>
<td>#(2, 3)</td>
</tr>
<tr>
<td>3</td>
<td>#(3, 0)</td>
<td>#(3, 1)</td>
<td>#(3, 2)</td>
<td>#(3, 3)</td>
</tr>
</tbody>
</table>

Figure 3-c.

\[
P_H = \begin{pmatrix}
4 & 2 & 1 & 0 \\
2 & 4 & 0 & 0 \\
1 & 0 & 6 & 1 \\
0 & 0 & 1 & 2
\end{pmatrix}
\]

\[
P_{90} = \begin{pmatrix}
6 & 0 & 2 & 0 \\
0 & 4 & 2 & 0 \\
2 & 2 & 2 & 2 \\
0 & 0 & 2 & 0
\end{pmatrix}
\]

Figure 3-d.

\[
P_{135} = \begin{pmatrix}
2 & 1 & 3 & 0 \\
1 & 2 & 1 & 0 \\
3 & 1 & 0 & 2 \\
0 & 0 & 2 & 0
\end{pmatrix}
\]

\[
P_{45} = \begin{pmatrix}
4 & 1 & 0 & 0 \\
1 & 2 & 2 & 0 \\
0 & 2 & 4 & 1 \\
0 & 0 & 1 & 0
\end{pmatrix}
\]

Figure 3-e.

Figure 3-f.

Reproducibility of the original page is poor.
To explain the significance of these features, let us consider the kind of values they take on two different land use category images. Figure 4 shows the digital printout of two sub-images of size 64 x 64 resolution cells (approximately 8.5 square mile area) from MSS band 5 of 1002-18134, see Figure 10. The image shown in 4(a) belongs to the grass land category and the image in Figure 4(b) is mostly water. Values of the features $f_1$, $f_{12}$, and $f_3$ are also shown for these images in Figure 4.

The angular second moment feature (ASM), $f_1$, is a measure of homogeneity of the image. In a homogeneous image, such as shown in 4(b), there are very few dominant grey tone transitions. Hence, the $P$ matrix for this image will have fewer entries of large magnitude. For an image like the one shown in Figure 4(a), the $P$ matrix will have a large number of small entries and hence the ASM feature which is the sum of squares of the entries in the $P$ matrix will be smaller. A comparison of the ASM values given below the images in Figure 4 shows the usefulness of the ASM feature as a measure of the homogeneity of the image.

The contrast feature, $f_{12}$, is obtained as a difference moment of the $P$ matrix and is a measure of the contrast or the amount of boundaries present in an image. Since there is a large amount of boundaries present in the image 4(a) compared to the image shown in 4(b), the contrast feature for the grassland image has consistently higher values compared to the water body image.

The correlation feature, $f_3$, is a measure of linear grey tone dependencies in the image. For both the images shown in Figure 4, the correlation feature is somewhat higher in the horizontal ($0^\circ$) direction, along the line of scan. The water body image consists mostly of a constant grey tone value for the water plus some additive noise. Since the noise samples are mostly uncorrelated, the correlation features for the water body image have lower correlation values compared to the grassland image. Also the grassland image has a considerable amount of linear structure along $45^\circ$ lines across the image and hence the value of the correlation feature is higher along this direction compared to the values for $90^\circ$ and $135^\circ$ directions.

The various features presented here are all functions of distance and angle. The angular dependencies present a special problem. Suppose image $A$ has features $a$, $b$, $c$, $d$ for angles $0^\circ$, $45^\circ$, $90^\circ$, and $135^\circ$ and image $B$ is identical to $A$ except that $B$ is rotated $90^\circ$ with respect to $A$. Then $B$ will have features $c$, $d$, $a$, $b$, for angles $0^\circ$, $45^\circ$, $90^\circ$, and $135^\circ$ respectively. Since the texture
Figure 4. Textural Features for Two Different Land Use Category Images.

a. Grassland

<table>
<thead>
<tr>
<th>Angle</th>
<th>ASM</th>
<th>Contrast</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>.0128</td>
<td>3.048</td>
<td>.8075</td>
</tr>
<tr>
<td>45°</td>
<td>.0080</td>
<td>4.011</td>
<td>.6366</td>
</tr>
<tr>
<td>90°</td>
<td>.0077</td>
<td>4.014</td>
<td>.5987</td>
</tr>
<tr>
<td>135°</td>
<td>.0064</td>
<td>4.709</td>
<td>.4610</td>
</tr>
<tr>
<td>Avg.</td>
<td>.0087</td>
<td>3.945</td>
<td>.6259</td>
</tr>
</tbody>
</table>

b. Water Body

<table>
<thead>
<tr>
<th>ASM</th>
<th>Contrast</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1016</td>
<td>2.153</td>
<td>.7254</td>
</tr>
<tr>
<td>.0771</td>
<td>3.057</td>
<td>.4768</td>
</tr>
<tr>
<td>.0762</td>
<td>3.113</td>
<td>.4646</td>
</tr>
<tr>
<td>.0741</td>
<td>3.129</td>
<td>.4650</td>
</tr>
<tr>
<td>.0822</td>
<td>2.863</td>
<td>.5327</td>
</tr>
</tbody>
</table>
context of A is the same as the texture context of B, any decision rule using the angular features a, b, c, d must produce the same results for c, d, a, b, or for that matter b, c, d, a (45° rotation) and d, a, b, c, (135° rotation). To guarantee this, we do not use the angularly dependent features directly. Instead, we merge the four arrays by summing corresponding elements. The merged array is then used for computing the 17 texture features defined in Appendix I.

IV.2 Textural Features for Multi-Images

Results of previous work in texture using the spatial grey tone dependence matrices as the basis from which all textural features are extracted has been very encouraging (Haralick, 1973). The good performance of these texture features can be seen by the results of the classification experiments. The computational aspects of the procedure are also notable. The number of operations required to process an image using the spatial grey tone dependence matrices is directly proportional to the number of resolution cells, N, present in an image. In comparison, the number of operations needed to use Fourier or Hadamard transforms to extract texture information are of the order of \( N \log N \). And, to compute the entries in the spatial grey tone dependence matrices, one needs to keep only two lines of image data in core at a time, keeping storage requirements to a minimum.

Even with these advantages, however, the extraction of texture information from multi-images, as in the case of ERTS MSS data, forces a new approach to the measurement of grey tone N-tuple co-occurrences. The use of the spatial dependence matrices requires that they be stored in the computer. For multi-images containing grey tone N-tuples, we have too many possible grey tone N-tuples which can neighbor each other and as a result, the dependence matrices will be very large. For example, for four MSS bands in which each grey tone can range through 64 levels, each matrix would have \( 64^4 \times 64^4 \) elements. Even using the symmetry of the matrices to reduce the number of entries does not help since there would be on the order of \( 10^{15} \) entries.

The spatial dependence matrices, however, provide a way of escape. In using these matrices, it was observed that they are heavily weighted along the diagonal with decreasing entries farther from the diagonal. Figure 5 gives an
example of one of these matrices. Note that the number of entries decreases as we move away from the diagonal. This suggests that neighboring resolution cells are similar. Choosing any resolution cell in an image at random, we are very likely to find nearly identical neighbors to the cell in all directions and less likely to find dissimilar neighbors. Clearly, a measure which indicates how similar the neighboring $N$-tuples are and how fast the similarity drops off with distance must contain textural information about the object imaged.

It is therefore reasonable to measure the difference between neighboring grey tone $N$-tuples and observe this distribution instead of computing the number of times each $N$-tuple neighbors every other $N$-tuple. In both cases we measure the co-occurrence of nearest neighbor grey tone $N$-tuples.

Since the textural features are based on the spatial dependence of grey tone $N$-tuples, our first step must be to define a binary relation between neighboring resolution cells on which the co-occurrence of grey tone $N$-tuples can be counted. As above, let $L_x = \{1, 2, \ldots, N_x\}$ and $L_y = \{1, 2, \ldots, N_y\}$ be the set of column and row indexes, respectively, so that $L_y \times L_x$ is the set of resolution cells in the image. Let $G = \{0, 1, \ldots, N_g\}$ be the set of possible grey tones that each component of every grey tone $N$-tuple can be assigned. Then, the image $I$ can be defined by $I : L_y \times L_x \rightarrow G \times G \times \ldots \times G$.

Let $R$ be the set of all pairs of resolution cells in a specified spatial relation. Then $R$ is a binary relation on the set $L_y \times L_x$; $R \subseteq (L_y \times L_x) \times (L_y \times L_x)$. For example, the set of all distance 1 horizontally neighboring pairs of neighboring resolution cells would be defined by:

$$R = \{(k, 1), (m, n) \in (L_y \times L_x) \times (L_y \times L_x) \mid k-m=0, \|i-n\|=1\}.$$  

The co-occurrence frequency of grey tone $N$-tuples $(i_1, i_2, \ldots, i_N)$ and $(j_1, j_2, \ldots, j_N)$ in spatial relation defined by $R$ is

$$P((i_1, \ldots, i_N), (j_1, \ldots, j_N)) = \frac{\# \{(k, 1), (m, n) \in R \mid I(k, 1) = (i_1, \ldots, i_N), I(m, n) = (j_1, \ldots, j_N)\}}{\#_R},$$

where $\#$ denotes the number of elements in the set.

Note that this $R$ is symmetric. Assume that $((k, 1), (m, n))$ is in $R$. Then $k-m=0$, and $\|i-n\|=1$ from the definitions of $R$. But $\|i-n\|=1$ when $\|n-1\|=1$. 


<table>
<thead>
<tr>
<th>FIGURE 5. Example of Nearest Neighbor Grey Tone Dependence Matrices, Taken from Processing ERTS Data.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
</tr>
</tbody>
</table>

**Note:** The table above represents a matrix where each element is a digit from 0 to 9, indicating the grey tone dependence. The values represent the nearest neighbor values in a grid, used for processing ERTS data.
And if \(|n-1| =1\) and \(k-m =0\), then \((m, n), (k, l)\) is in \(R\). Thus, \(R\) is symmetric.
In fact, by the symmetry of any distance function, \(R\), in general, must be symmetric.
And since \(R\) is symmetric, \(P\) is also symmetric.

IV.3 Textural Feature Extraction Procedure

Let \(R\) be a symmetric binary relation pairing nearby neighboring resolution cells. We define the frequency of grey tone \(N\)-tuple differences co-occurring in the spatial configuration defined by \(R\) as

\[
P(x_1, x_2, \ldots, x_N) = \frac{\# \left\{ (i, j), (m, n) \in R \mid 1(i, j) - 1(m, n) = x_N \right\}}{\#R}
\]

Note that \(P\) is an even function since

\[
P(x_1, x_2, \ldots, x_N) = \# \left\{ (i, j), (m, n) \in R^{-1} \mid 1(i, j) - 1(m, n) = x_N \right\} / \#R
\]

Referring to the monotonic behavior of nearly every column in the matrices of Figure 5, and assuming that this behavior occurs on every band of the ERTS multimages, it is reasonable to assume that the even frequency distribution \(P(x_1, \ldots, x_N)\) of the nearby grey tone \(N\)-tuple differences can be adequately approximated using an ellipsoidally symmetric distribution; thus we may write

\[
P(x_1, x_2, \ldots, x_N) = f(x^T Ax)
\]

for some monotonically decreasing function \(f\).
This implies that only the function \( f \) and the matrix \( A \) need to be determined. We take \( f \) to be one of the two forms \( e^{-\frac{1}{2} \| \|_2^2} \), \( (1+\| \|_2^m) \). Figure 6 is a scattergram of the differences for the first two bands of distance 1 horizontally neighboring resolution cells of a 6x64 sample image. Figures 6 and 7 clearly show the ellipsoidally symmetric functional form of the distribution of neighboring differences. In Appendix II N-dimensional spherical coordinate systems and ellipsoidally symmetric distributions are discussed and it is shown that the matrix \( A \) is proportional to the inverse covariance matrix of the N-tuple differences. Thus, we estimate \( A \) by a matrix proportional to the inverse of the estimate for the covariance matrix.

Therefore, if the image is blocked into subimages of small area so that each subimage is essentially of one category, we can expect the distribution of grey tone N-tuple differences over each subimage to be a function only of the assumed form of the function \( f \) and the covariance matrix of the difference vectors for grey tone N-tuples in a specified spatial relationship within the subimage. This leads us to consider textural features for multi-images based upon the elements of this spatial-spectral covariance matrix.

Consider each covariance matrix as a vector. Consider the distribution of the set of covariance matrices from the blocked image. Since the entries of the covariance matrix are the parameters of the distribution, we would like to have these entries invariant with respect to scale changes on the grey tone N-tuple differences. In order to do this, we scale the grey tone N-tuple differences so that all components have variance 1. The covariance matrix of these normalized differences is equivalent to the correlation matrix. Appendix III shows that this normalization procedure makes the covariance matrix invariant with respect to translating and scaling transformations on the grey tone N-tuples. The normalized covariance matrix can be considered as an extracted texture feature vector in an \( N(N-1)/2 \) dimensional hyperspace.

Initial classification experiments indicated a need for more textural information and in order to provide this the dimensionality, \( N \), of each resolution cell was increased from the four provided by the four MSS bands to eight by appending higher order terms and cross-band product terms for each cell. The resultant increase in correct classification accuracy can be seen in section VII.
FIGURE 6. Scattergram showing ellipsoidally symmetric distribution of differences for Bands 4 and 5 over a sample 64 x 64 ERTS image.

FIGURE 7. Histogram of the distribution of differences on Band 5 for distance 1 horizontally neighboring resolution cells.
V. GROUND TRUTH ASSIGNMENT PROCEDURE

For all classification experiments, the only method available to determine ground truth was based upon the 9 inch by 9 inch ERTS image transparency. Initially, the land-use category names were determined with the aid of two photo interpreters and, when available, contour maps published by the U.S. Geological Survey.

After the categories to be used were decided upon, each ERTS image was processed. The image was blocked into 64 x 64 subimages for texture analysis on MSS band 5 and 32 by 32 subimages for cross-band texture analysis of MSS bands 4 thru 7. The ERTS Retrieval Program was then used to printout a picture of the same area that was processed. This was subdivided into 64 x 64 subimages. In this way, the location of each subimage was known and could be fairly accurately determined on a 7 inch by 7 inch print. Then, on the basis of its location, the subimage is assigned a ground truth category. Figure 8 shows a picture printout by the ERTS Retrieval Program for a portion of image 1021-16333 (see Figure 12) over Kansas City. The picture has been blocked into 64 x 64 subimages and by referring to the original image in Figure 12, each subimage was assigned a ground truth category.

Figures 8 and 12 also illustrate the major problem with ground truth assignment. This is the determination of ground truth when a subimage covers more than one ground truth category. The irregular boundary around the urban area of Kansas City makes ground truth assignment difficult. The deciding factor was the amount of area within the subimage from each category. The subimage was assigned to the category which had the largest area within its boundary. For this reason the accuracy with which the location of every subimage is determined becomes very important, and the picture printouts from the retrieval programs aided in this.

Since the starting and ending image row and column coordinates were known, the Retrieval Program could use the same coordinates for the picture printout. This printout could then be accurately divided into 64 by 64 subimages. The actual location of these subimages could then be determined fairly well, although it is sometimes difficult to locate objects in the printout that are on the original image. To aid in this, a few of the well defined objects were used to construct a 64 by 64 grid on the 7 inch by 7 inch
Figure 8. Printout of a Portion of Image 1021-16333 Over Kansas City by the ERTS Retrieval Program for Use in Ground Truth Assignment.
print. When there was a question in the ground truth, the printout could be referred to for the more accurate subimage location. The size of the subimage also relates directly to the accuracy with which its location on the print can be determined. It was found that 32 lines by 32 points per line was about as small as a subimage can be for good accuracy in ground truth assignment.

VI. IDENTIFICATION PROCEDURE

In the classification experiments two different classification algorithms were used, a Bayes classifier and a piecewise linear classifier. The Bayes classifier assumes a multivariate normal distribution and randomly chooses pattern vectors for the training set where the piecewise linear classifier does not.

The problem of developing procedures for categorizing environmental units consists of the following.

With reference to Figure 9, the Universe $\mathcal{U}$ consists of environmental units (for example rocks) $U_1, U_2, \ldots, U_T$ which belongs to one of $R$ possible categories $C_1, C_2, \ldots, C_R$ (different land use categories). Of the large number of environmental units present in the universe, we observe a smaller subset of units $U_1, U_2, \ldots, U_N$. Our observations consist of a set of measured values of $n$ features $f_1, f_2, \ldots, f_n$ for each unit $U$ sampled. Based on the information contained in the feature vectors $F_1, F_2, \ldots, F_N$, the categories of the environmental units which produce these measurements being known, we want to develop an algorithm to identify the categories of new units based on the measurements they produce.
Nonlinear Feature Extraction Transformation Identification

Universe $U = \{U_1, U_2, \ldots, U_T\}$

Feature Space $\mathcal{F} = \{F_1, F_2, \ldots, F_T\}; \quad F = [f_1, f_2, \ldots, f_n]^T$

Transformed Feature Space $\mathcal{X} = \{X_1, X_2, \ldots, X_T\}; \quad X = [x_1, x_2, \ldots, x_m]^T$

The vectors $F_i$ and $X_i$ are usually referred to as feature vector and pattern vector respectively.

Figure 9. Identification Scheme.
The decision rule which assigns categories based on the values of features may be implemented in the feature space \( \mathcal{F} \) by partitioning \( \mathcal{F} \) into various regions and assigning categories to new units based on the regions to which their feature vectors belong. Efficient partitioning of the feature space may require complicated nonlinear decision boundaries (discriminant functions). Instead of deriving a decision rule in the feature space \( \mathcal{F} \), we may transform the feature vectors into a new space \( \mathcal{X} \) and implement a decision rule in the new space \( \mathcal{X} \). By using appropriate nonlinear transformations, we may be able to implement nonlinear decision boundaries in \( \mathcal{F} \) as linear decision boundaries in \( \mathcal{X} \). Several procedures are available for deriving linear decision boundaries for partitioning \( \mathcal{X} \) into various regions, based on the information contained in a set of sample patterns \( X_1, X_2, \ldots, X_N \) whose categories are known.

Identification Algorithms: In a widely used algorithm (Fukunaga 1972, Fu and Mendel 1970, Miesel 1972), the pattern space \( \mathcal{X} \) is separated into a number of regions using a set of hyperplanes (decision boundaries) whose locations are determined by the sample patterns. Each region is dominated by sample patterns of a particular category. When a new pattern is presented for identification, it is assigned a category depending on the region in which it belongs. If the new pattern \( X \) is located in a region dominated by sample patterns of category \( c_j \), then \( X \) is classified as coming from category \( c_j \).

To illustrate the procedure for obtaining the hyperplanes, consider the problem of separating the sample patterns \( X_1, X_2, \ldots, X_{n_i} \) belonging to category \( c_i \) and \( X_{n_i+1}, X_{n_i+2}, \ldots, X_{n_i+n_i} \) belonging to category \( c_j \). We can write the linear discriminant function (hyperplane) which separates the patterns belonging to categories \( c_i \) and \( c_j \) as

\[
h_{ij}(X) = v_{ij}^T X + v_{ij}^o \geq 0 \text{ for } X \in c_i,
\]

\[
h_{ij}(X) = v_{ij}^T X + v_{ij}^o < 0 \text{ for } X \in c_j.
\]

The vector \( v_{ij} \) and the scalar \( v_{ij}^o \) are to be determined from the information contained on the sample patterns.

If we introduce a new form to express the pattern vectors as

\[
Z = [1 \ x_1 \ x_2 \ \ldots \ x_n]^T \text{ for } X \in c_i,
\]

\[
Z = [-1 \ -x_1 \ -x_2 \ \ldots \ -x_n]^T \text{ for } X \in c_j.
\]
then the discriminant function can be written as

\[ h_{ij}(Z) = W_{ij}^T Z = 0 \]  

(1)

where \( W_{ij} \) is referred to as a weight vector and

\[ h_{ij}(Z) = W_{ij}^T Z = 0 \]

is the equation of a hyperplane in the transformed feature space.

The weight vector \( W_{ij} \) is chosen so as to satisfy equation 1 for as many training patterns as possible. Usually we do not know the precise form of \( h_{ij} \). But, given our knowledge of the categories of the training patterns, we can postulate reasonable values \( g_{ij}(Z_k) \) for \( h_{ij}(Z_k) \) and choose \( W_{ij} \) to minimize the mean square error given by

\[ 
\epsilon^2 = \frac{1}{n_i + n_j} \sum_{k=1}^{n_i + n_j} (W_{ij}^T Z_k - g_{ij}(Z_k))^2 .
\]

Usually \( g_{ij}(Z_k) \) is taken to be +1 for \( k = 1, 2, \ldots, n_i + n_j \). We can rewrite \( \epsilon^2 \) as,

\[ 
\epsilon^2 = \frac{1}{(n_i + n_j)} \left[ W_{ij}^T Y - G_{ij}^T \right] \left[ Y^T W_{ij} - I \right]
\]

(2)

where

\[ Y = [Z_1 Z_2 \cdots Z_{n_i + n_j}], \text{ and} \]

\[ G_{ij} = [g_{ij}(Z_1) g_{ij}(Z_2) \cdots g_{ij}(Z_{n_i + n_j})] . \]

The weight vector which minimizes \( \epsilon^2 \) given in equation 2 is given by

\[ W_{ij} = (Y Y^T)^{-1} Y G_{ij} \]

which is the well-known normal equation set from linear least square theory.

For the multiclass problem involving \( N_R \) categories, a total of \( N_R (N_R - 1)/2 \) hyperplanes must be determined using the procedure described above. After the hyperplanes are determined, the classification of new patterns is done as follows. For each category \( e_i \), the number of hyperplanes, \( V_i \), which give a positive response when the new pattern \( X \) is presented are determined using

\[ V_i = \sum_{j=1}^{N_R} \frac{|W_{ij}^T Z| + W_{ij}^T Z}{2|W_{ij}^T Z|} ; i = 1, 2, \ldots, N_R \]

22
where \( Z = \begin{bmatrix} 1 \\ X \end{bmatrix} \).

\( X \) is assigned to category \( c_j \) if

\[
V_j = \max_i \{ V_i \}
\]

If there is a tie between categories \( c_m \) and \( c_n \), then \( X \) is assigned to \( c_m \) if \( W_{mn}^T Z \geq 0 \) or to \( c_n \) if \( W_{mn}^T Z < 0 \). Several modifications of the linear discriminant function method and a multitude of other classification procedures may be found in the references cited.

**VII. RESULTS OF CLASSIFICATION EXPERIMENTS**

Tables 1 thru 24 show the results of the classification experiments. Unless otherwise stated, the contingency tables are determined using the piecewise linear classification programs (RCLASS) given in Appendix IV. In each contingency table the number of errors (\( \#\text{ERR} \)) and percent error (\( \%\text{ERR} \)) is shown for both errors of commission and errors of omission. The final entry in the percent error column is the average of the percent error for each category. The average correct classification is simply the number of correct classifications divided by the number of incorrect classifications. The final column in each contingency table (\( \%\text{SD} \)) is an estimate for the standard deviation of the probability for correct classification. This was taken from Afarani (Afarani, 1972) where he gives an estimate for the variance of the probability of correct classification when the sample size is fixed as:

\[
\sigma^2 = \frac{(\sum n_{ii})(\sum_{i=j} n_{ij})}{n^3} = \frac{(\text{no. correct})(\text{no. errors})}{(\text{total no.})^3}
\]

where \( n \) is the fixed sample size and \( n_{ij} \) is the number of classifications of units assigned to category \( j \) whose true category is \( i \). Used directly, this gives an estimate over all categories for the entire contingency table. This is the last entry in the percent standard deviation (\( \%\text{SD} \)) column. The remaining entries were determined by fixing \( i \) and estimating the standard deviation for each category \( i \). This estimate says that as the number of samples tested increases and the number of correct classifications increase, the variance of the probability of correct classification decreases, as one would expect.
In order to obtain an initial estimate of performance of the multi-image texture features an experiment was performed on ERTS satellite imagery over Monterey Bay, California, image number 1002-18134 (see Figure 10) taken on July 25, 1972. Using a small set of 64 sampled 32 x 32 subimages and training on 34 of these, 80 per cent of the remaining 30 test samples were correctly classified according to four land-use categories: coastal forest, annual grassland, urban area, and water, as shown in Table 1. This is encouraging since previous accuracy using spatial dependence matrices on band 5 with 64 by 64 subimages over the same general area was only 70.5 per cent as shown in Table 2 (Haralick, 1973).

The ability to obtain good ground truth and several distinct categories in the California data was not the case for an ERTS image over Finney County, Kansas, which was used in later experiments. Approximately a 40 mile by 60 mile section near Garden City, Kansas, on image number 1330-16515 (see Figure 11), taken on June 18, 1973, was processed with initially four categories: grassland, large fields, small fields, and water. Both texture procedures, using the multi-image texture features with 32 by 32 subimages and the single-image texture analysis on MSS band 5 with 64 by 64 subimages, were used on the image. Tables 3 and 4 show the results of classification for distance 1 resolution cells while Tables 5 and 6 show distance 8 results. In both cases the single image classification is higher. However, when both distances 1 and 8 are used together, classification accuracy for both procedures is nearly identical, as shown in Tables 7 and 8, about 70 per cent. Tables 9 and 10 show results using the Bayes Classifier.

This implies that more information is contained in the single-band texture features than the multi-image texture features. In order to add more texture information, a measure of entropy (Kullbach, 1959), given by

\[ E = -\frac{1}{2} \log |P| \]

where P is the correlation matrix, was added to the cross-band texture feature set. Also, higher order components were appended to each grey tone N-tuple by squaring the grey tones and getting cross-band product terms. Only a few of these were added, increasing the grey tone N-tuple dimensionality from 4 to 8, which results in an increase in the number of feature vector components (elements in the correlation matrix) from 6 to 28 plus the entropy measure. The eight components in each grey tone N-tuple are: MSS Band 5, Band 6, Band 7, (Band 5)^2, (Band 6)^2, (Band 7)^2, (Band 5) × (Band 6),
Figure 13 gives an illustration of a correlation matrix with the feature vector component designation to be used in indicating the components selected for input to the classification programs.

Table 11 shows the contingency table for the cross-band texture features using 9 of the 29 components (1, 2, 5, 6, 9, 12, 20, 24, and 27 of figure 13) for a portion of the Garden City, Kansas, data. The increase in identification accuracy between the large and small fields results in an increase in overall correct classification, up to 87.1%.

The final classification experiment to test the cross-band texture analysis was made on ERTS image 1021-16333 (see Figure 12) taken on August 13, 1973, over Kansas City. Four land-use categories were chosen: cropland (directly north and south of Kansas City), urban area (Kansas City, Topeka), grassland (southwestern corner of the image), and water (Perry reservoir plus several small lakes). These areas were processed three separate ways:

1) spectrally, 32 by 32 subimages
2) texturally, 64 by 64 subimages on band 5
and 3) cross-band texturally, 32 by 32 subimages.

Because of problems with ground truth and a small data set for water, that category was later dropped. The spectral processing involved obtaining the average grey tone over the subimage for each spectral band, giving 4 components for each feature vector. The textural processing was over a larger subimage than either the spectral or the cross-band textural processing. This was chosen because the 64 by 64 subimages have performed better in the past than the 32 by 32 subimages for the single-image texture analysis. The smaller subimage size was chosen for the cross-band texture processing so that the subimage would more likely be from only one category. The cross-band texture method uses spectral information which the single-image texture procedure does not have available. The smaller subimage size was also chosen for the spectral processing in order to provide an estimate of the amount of spectral information contained in the four MSS bands that is available to the cross-band texture analysis.

Single-image texture analysis was done for distance 1 nearest neighbors at all angles and all of the 17 texture features defined in Appendix I were computed for each subimage. The cross-band texture processing was also done for distance 1 using horizontally adjacent nearest neighbor grey tone N-tuples.
Figure 10. Part of ERTS Image No. 1002-18134 (MSS Band 5) Over Monterey Bay, California, Taken on July 25, 1972.
Figure 11. ERTS Image No. 1330-16515 (MSS Band 5) Over Garden City, Kansas, Taken on June 18, 1973.
Figure 12. ERTS Image No. 1021-16333 Over (MSS Band 5) Kansas City, Taken on August 13, 1972.
<table>
<thead>
<tr>
<th></th>
<th>Band 5</th>
<th>Band 6</th>
<th>Band 7</th>
<th>((\text{Band 5})^2)</th>
<th>((\text{Band 6})^2)</th>
<th>((\text{Band 7})^2)</th>
<th>(\times(\text{Band 6}))</th>
<th>(\times(\text{Band 7}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Band 5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Band 6</td>
<td>F(2) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Band 7</td>
<td>F(3) F(9) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((\text{Band 5})^2)</td>
<td>F(4) F(10) F(15) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((\text{Band 6})^2)</td>
<td>F(5) F(11) F(16) F(20) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((\text{Band 7})^2)</td>
<td>F(6) F(12) F(17) F(21) F(24) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((\text{Band 5})(\text{Band 6}))</td>
<td>F(7) F(13) F(18) F(22) F(25) F(27) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>((\text{Band 5})(\text{Band 7}))</td>
<td>F(8) F(14) F(19) F(23) F(26) F(28) F(29) 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Entropy = F(1)

**Figure 13.** Designation of the entropy measure and elements of the correlation matrix as feature vector components for cross-band texture analysis of multi-images.
Table 12 shows the contingency table for all 17 single-image texture features over image 1021-16333 with an average correct classification of 86.2%. Table 13 shows the resulting contingency table over the same image for the 29 cross-band texture features with an increased correct classification of 88.8%. The spectral processing on the four MSS bands for the same image gave four spectral features—the mean grey tone over the subimage for each spectral band. The resulting contingency table using these four spectral features is shown in Table 14. The relatively good performance, 73.9% correct, shows a significant amount of land-use information is contained in the four spectral bands, accounting for the better performance of the cross-band texture analysis over the single-image texture analysis.

It is interesting to see the classification accuracy of the single-image texture greatly improve to 93.8% with the addition of the four spectral features to the original 17 texture features, as shown in Table 15. As expected, the addition of the four spectral features to the first 26 of the 29 cross-band texture features does not improve the classification accuracy as well as with the single-image texture. Table 16 shows the resulting contingency table with an average correct classification of 91.6%. Note the higher estimate for the standard deviation of the probability for correct classification with the textural plus spectral features.

These tables show that the single-image texture procedure does well in extracting texture information, but for this data set, the cross-band texture procedure by itself performs slightly better by extracting more information texturally and spectrally.

The remaining tables show the effect upon classification accuracy of using fewer features. In each case it can be seen that the fewer the number of features used, the lower the average correct classification. Also, it can be seen that the Bayes classifier performs slightly better than the piecewise linear classifier.

Tables 20 thru 24 show that as the number of components is increased, the spectral plus cross-band texture reaches a limiting accuracy of approximately 92%. Ground truth assignment errors could easily account for the remaining 8% error.

VII.1 Summary of Classification Results

It is apparent from these classification experiments that both texture extraction procedures complement each other in that they extract different kinds of texture information. When used without spectral features, the cross-band texture procedure performed better. However, when the spectral features were added, the single-image texture plus spectral features performed better than the cross-band texture plus spectral features. This indicates that the cross-band texture procedure does well in
extracting more information texturally and spectrally than the single-image texture procedure. However, when the spectral information is made available to the single-image texture procedure, it performs better than the cross-band texture plus spectral features.

Programs used in these experiments can be found in Appendix IV. The single-image texture programs are under the Texture Analysis Programs with mainline MAINLN. And the cross-band texture programs are in the Cross-Band Texture Analysis Programs with mainline SPECTR. RCLASS is the mainline program for the piecewise linear classifier. Some of the cross-band texture feature components can be seen in Figure 14. Figure 14a shows the ground truth assignment for the image 1021-16333 (Figure 12) for the area near Kansas City. At the bottom of the image, grassland was inserted to give an idea of how well the features separate the three categories: cropland, urban area, and grassland.

VIII. CONCLUSION

The procedure developed here for the extraction of texture information from ERTS multi-images gives encouraging results. The classification experiments show that the cross-band texture procedure can be used successfully in automatic land-use classification of multi-images over Kansas.

The cross-band texture procedure is a natural extension of the previous single-image texture extraction procedure based upon angular nearest neighbor grey tone spatial dependence matrices. It retains the power of the previous approach to texture by characterizing the spatial inter-relationships, or co-occurrences, of the grey tone N-tuples present in a texture pattern in such a way as to be invariant under linear grey tone transformations. And both procedures are simple to employ, economical, and require a minimum of core-storage (see Figure 15).

Both procedures complement each other by extracting different kinds of textural information with the cross-band texture procedure using the cross-band spectral information contained in ERTS multi-images. Results indicate that the cross-band texture procedure does well by extracting more information texturally and spectrally than the single-image texture procedure when used alone. However, when the spectral information is made available to the single-image texture procedure, it performs better than the cross-band texture plus spectral features in classifying texturally distinct land-use categories from ERTS multi-images over Kansas.
Figure 14-a. Ground Truth Map

Figure 14-b. F(16)

Figure 14-d. F(24)

Figure 14-c. F(20)

Figure 14-e. F(29)

Figure 14. Examples of Cross-Band Texture Feature Vector Components; Resolution Cell Size is 32 x 32
<table>
<thead>
<tr>
<th>TEXTURE</th>
<th>CORE REQUIRED</th>
<th>TOTAL PROCESSING TIME</th>
<th>TOTAL JOB COST</th>
<th>COST PER RESOLUTION CELL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SINGLE-IMAGE TEXTURE ((64\times64))</td>
<td>43K</td>
<td>0.788 HR</td>
<td>$223.</td>
<td>$0.000252</td>
</tr>
<tr>
<td>CROSS-BAND TEXTURE ((32\times32))</td>
<td>25K</td>
<td>0.884 HR</td>
<td>$216.</td>
<td>$0.000244</td>
</tr>
</tbody>
</table>

**Figure 15.** Comparison of the Performance of the Single-Image Texture Analysis Programs on MSS Band 5 with the Cross-Band Texture Analysis Programs on MSS Bands 4 thru 7 in Processing the Same Area, Near Kansas City, on ERTS Image No. 1021-16333 using a Honeywell 635 Computer.
### Table 1. Contingency Table for Image No. 1002-18134 Using Multi-Image Textural Features, 6 Components, Average Correct Classification on Test Set = 80.0%*

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coastal Forest</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Annual Grassland</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Urban Area</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Water</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>5</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors of Commision</th>
<th>#Errors</th>
<th>%Err</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Table 2. Results of Land-use Classification Experiments from ERTS Image No. 1002-18134 Over Monterey Bay, California.**

<table>
<thead>
<tr>
<th>Features</th>
<th>No. of Samples in Training Set</th>
<th>No. of Samples in Test Set</th>
<th>Overall Accuracy of Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Image Textural</td>
<td>34</td>
<td>30</td>
<td>80.0%</td>
</tr>
<tr>
<td>Single-Image Textural</td>
<td>260</td>
<td>172</td>
<td>70.5%</td>
</tr>
</tbody>
</table>

*Unless stated otherwise, all contingency tables are determined using the piecewise linear classification programs (RCLASS).*

Reproducibility of the original page is poor.
### Table 3. Contingency Table for Image No. 1330-16515 Using Multi-Image Textural Features, Distance 1, 6 Components. Average Correct Classification = 66.5% |

| True Category | Assigned Category | |
|---------------|------------------|------------------|------------------|------------------|------------------|------------------|
|               | Grassland        | Large Fields     | Small Fields     | Water            | Total            | #Err            | %Err            | %SD            |
| Grassland     | 47               | 11               | 5                | 0                | 63               | 16              | 25.4            | 5.5            |
| Large Fields  | 3                | 142              | 32               | 1                | 178              | 36              | 20.2            | 3.0            |
| Small Fields  | 17               | 46               | 52               | 0                | 115              | 63              | 31.3            | 4.6            |
| Water         | 0                | 1                | 6                | 1                | 8                | 2               | 25.0            | 11.7           |
| Total         | 67               | 200              | 95               | 2                | 364              |                 |                 |                |

<table>
<thead>
<tr>
<th>Errors of Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Errors</td>
</tr>
<tr>
<td>%Error</td>
</tr>
</tbody>
</table>

### Table 4. Results of Land-Use Classification Experiments for ERTS Image No. 1330-16515 at Distance 1.

<table>
<thead>
<tr>
<th>Features</th>
<th>No. of Samples in Training Set</th>
<th>No. of Samples in Test Set</th>
<th>Overall Accuracy of Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-Image Textural</td>
<td>548</td>
<td>364</td>
<td>66.5%</td>
</tr>
<tr>
<td>Single-Image Textural</td>
<td>140</td>
<td>88</td>
<td>76%</td>
</tr>
</tbody>
</table>

Table 4. Results of land-use classification experiments for ERTS image no. 1330-16515 at distance 1.
Table 5. Contingency table for image no. 1330-16515 using multi-image textural features, distance 8, 6 components. Average correct classification = 63.7%.

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th>Errors of Commission</th>
<th>#Err</th>
<th>%Err</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grassland</td>
<td>Grassland</td>
<td>32</td>
<td>31</td>
<td>49.2</td>
<td>6.3</td>
</tr>
<tr>
<td></td>
<td>Large Fields</td>
<td>7</td>
<td>31</td>
<td>17.4</td>
<td>2.8</td>
</tr>
<tr>
<td></td>
<td>Small Fields</td>
<td>18</td>
<td>62</td>
<td>53.9</td>
<td>4.6</td>
</tr>
<tr>
<td></td>
<td>Water</td>
<td>0</td>
<td>4</td>
<td>50.0</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>57</td>
<td>42.6</td>
<td>2.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 6. Results of land-use classification experiments for ERTS image no. 1330-16515 at distance 8.
<table>
<thead>
<tr>
<th>TRUE CATEGORY</th>
<th>ASSIGNED CATEGORY</th>
<th>ERRORS OF COMMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>LARGE FIELDS</td>
</tr>
<tr>
<td>GRASSLAND</td>
<td>52</td>
<td>7</td>
</tr>
<tr>
<td>LARGE FIELDS</td>
<td>2</td>
<td>145</td>
</tr>
<tr>
<td>SMALL FIELDS</td>
<td>12</td>
<td>42</td>
</tr>
<tr>
<td>WATER</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TOTAL</td>
<td>66</td>
<td>195</td>
</tr>
</tbody>
</table>

Table 7. Contingency table for image no. 1330-16515 using multi-image textural features, distances 1 and 8, 12 components. Average correct classification = 71%.

<table>
<thead>
<tr>
<th>FEATURES</th>
<th>NO. OF SAMPLES IN TRAINING SET</th>
<th>NO. OF SAMPLES IN TEST SET</th>
<th>OVERALL ACCURACY OF TEST SET</th>
</tr>
</thead>
<tbody>
<tr>
<td>MULTI-IMAGE TEXTURAL</td>
<td>548</td>
<td>364</td>
<td>71%</td>
</tr>
<tr>
<td>SINGLE-IMAGE TEXTURAL</td>
<td>140</td>
<td>88</td>
<td>73%</td>
</tr>
</tbody>
</table>

Table 8. Results of land-use classification experiments for ERTS image no. 1330-16515 using both distances 1 and 8.
TABLE 9. Contingency table for image no. 1330-16515 using multi-image textural features, distance 1.6 components, average correct classification = 74.0% using Bayes.

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th></th>
<th>#Err</th>
<th>%Err</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grassland</td>
<td>Grassland</td>
<td>49</td>
<td>4</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Large Fields</td>
<td>6</td>
<td>150</td>
<td>27</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Small Fields</td>
<td>10</td>
<td>34</td>
<td>67</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Water</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>Grassland</td>
<td>65</td>
<td>192</td>
<td>100</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors of Commission</th>
<th>#Errors</th>
<th>%Errors</th>
<th>%Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Errors</td>
<td>16</td>
<td>42</td>
<td>33.0</td>
</tr>
</tbody>
</table>

TABLE 10. Contingency table for image no. 1330-16515 using multi-image textural features, distance 8.6 components, average correct classification = 61.1% using Bayes.

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th>#Err</th>
<th>%Err</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grassland</td>
<td>Grassland</td>
<td>31</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>Large Fields</td>
<td>5</td>
<td>140</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>Small Fields</td>
<td>12</td>
<td>46</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>Water</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>Grassland</td>
<td>49</td>
<td>200</td>
<td>110</td>
</tr>
</tbody>
</table>
### Table II. Errors of Assigned Category

<table>
<thead>
<tr>
<th>TRUE CATEGORY</th>
<th>GRASSLAND</th>
<th>LARGE FIELDS</th>
<th>SMALL FIELDS</th>
<th>TOTAL</th>
<th>ERRORS</th>
<th>%ERR</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRASSLAND</td>
<td>35</td>
<td>3</td>
<td>0</td>
<td>38</td>
<td>3</td>
<td>7.9</td>
<td>4.4</td>
</tr>
<tr>
<td>LARGE FIELDS</td>
<td>2</td>
<td>12</td>
<td>2</td>
<td>16</td>
<td>4</td>
<td>25.0</td>
<td>10.8</td>
</tr>
<tr>
<td>SMALL FIELDS</td>
<td>0</td>
<td>2</td>
<td>14</td>
<td>16</td>
<td>2</td>
<td>12.5</td>
<td>8.3</td>
</tr>
<tr>
<td>TOTAL</td>
<td>37</td>
<td>17</td>
<td>16</td>
<td>70</td>
<td>15</td>
<td>15.1</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Table 11. Contingency table for image no. 1330-16515 using 9 of the 29 cross-band textural features, distance 1. #TRAIN=103, #TEST=70, AVERAGE CORRECT CLASSIFICATION = 87.1% USING THE BAYES CLASSIFIER. THE 9 FEATURE COMPONENTS USED WERE: 1, 2, 5, 6, 9, 12, 20, 24, 27.
### Table 12. Contingency Table for ERTS Image 1021-16333 Using All 17 Single-Band Texture Features at Distance 1.

<table>
<thead>
<tr>
<th>TRUE CATEGORY</th>
<th>ASSIGNED CATEGORY</th>
<th>ERRORS OF COMMISSION</th>
<th>#ERR</th>
<th>%ERR</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CROPLAND</td>
<td>61</td>
<td>3</td>
<td>4.7</td>
<td>2.6</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>6</td>
<td>6</td>
<td>25.0</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>6</td>
<td>7</td>
<td>25.0</td>
<td>8.2</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>73</td>
<td>12</td>
<td>16.4</td>
<td>11.7</td>
</tr>
</tbody>
</table>

Table 13. Contingency Table for ERTS Image 1021-16333 Using All 29 Cross-Band Texture Features at Distance 1.

<table>
<thead>
<tr>
<th>TRUE CATEGORY</th>
<th>ASSIGNED CATEGORY</th>
<th>ERRORS OF COMMISSION</th>
<th>#ERR</th>
<th>%ERR</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CROPLAND</td>
<td>150</td>
<td>18</td>
<td>10.7</td>
<td>2.4</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>11</td>
<td>11</td>
<td>12.0</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>11</td>
<td>13</td>
<td>11.2</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>172</td>
<td>22</td>
<td>12.8</td>
<td>11.0</td>
</tr>
</tbody>
</table>

#TRAIN = 569, #TEST = 376, AVERAGE CORRECT CLASSIFICATION = 88.8%
Table 14. Contingency table for ERTS image 1021-16333 using spectral features only. #TRAIN=564, #TEST=372
AVERAGE CORRECT CLASSIFICATION = 73.9%

Table 15. Contingency table for ERTS image 1021-16333 using all 17 single-image texture plus the 4 spectral features, 21 components. #TRAIN=175, #TEST=112
AVERAGE CORRECT CLASSIFICATION = 93.8%
### Table 16. Contingency Table for ERTS Image 1021-16333 Using First 26 Cross-Band Texture Plus the 4 Spectral Features, 30 Components. #Train=567, #Test=369

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th>#Err</th>
<th>%Err</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropland</td>
<td>Cropland</td>
<td>155</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Urban</td>
<td>12</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Grassland</td>
<td>4</td>
<td>2</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>171</td>
<td>94</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>168</td>
<td>92</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>7.7</td>
<td>2.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>13.0</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>5.5</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.7</td>
<td>1.4</td>
<td></td>
</tr>
</tbody>
</table>

AVERAGE CORRECT CLASSIFICATION = 91.6%

### Table 17. Contingency Table for ERTS Image 1021-16333 Using First 8 of the 17 Single-Image Texture Features. #Train=178, #Test=116. AVERAGE CORRECT CLASSIFICATION = 84.5%

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th>#Err</th>
<th>%Err</th>
<th>%SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropland</td>
<td>Cropland</td>
<td>60</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Urban</td>
<td>8</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Grassland</td>
<td>5</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>73</td>
<td>18</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>64</td>
<td>24</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>6.2</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>33.3</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>21.4</td>
<td>7.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20.3</td>
<td>3.4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors of Commission</th>
<th>#Errors</th>
<th>%Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13</td>
<td>17.8</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>11.1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>12.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.6</td>
</tr>
</tbody>
</table>
Table 18. Contingency table for ERTS image 1021-16333 using first 9 of the 29 cross-band texture features, 
#train=569, #test=376, average correct classification = 79.0%  

<table>
<thead>
<tr>
<th>TRUE CATEGORY</th>
<th>ASSIGNED CATEGORY</th>
<th>ERRORS OF COMMISSION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CROPLAND</td>
<td>URBAN</td>
</tr>
<tr>
<td>CROPLAND</td>
<td>138</td>
<td>22</td>
</tr>
<tr>
<td>URBAN</td>
<td>21</td>
<td>68</td>
</tr>
<tr>
<td>GRASSLAND</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>TOTAL</td>
<td>179</td>
<td>95</td>
</tr>
<tr>
<td>#ERRORS</td>
<td>41</td>
<td>27</td>
</tr>
<tr>
<td>%ERROR</td>
<td>22.9</td>
<td>28.4</td>
</tr>
</tbody>
</table>

Table 19. Contingency table for ERTS image 1021-16333 using first 9 of the 29 cross-band texture features, 
#train=562, #test=374, average correct classification = 81.0% using the Bayes classifier.
### Table 20.
Contingency table for ERTS image 1021-16333 using first 6 of the 29 cross-band texture plus the 4 spectral features, #train=569, #test=367, average correct classification = 84.2%

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th>Errors of Commission</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROPLAND</td>
<td>CROPLAND</td>
<td>158</td>
</tr>
<tr>
<td>URBAN</td>
<td>CROPLAND</td>
<td>13</td>
</tr>
<tr>
<td>GRASSLAND</td>
<td>CROPLAND</td>
<td>9</td>
</tr>
<tr>
<td>TOTAL</td>
<td>CROPLAND</td>
<td>180</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>113</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>367</td>
</tr>
<tr>
<td></td>
<td>#ERROR</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>%ERROR</td>
<td>12.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 21.
Contingency table for ERTS image 1021-16333 using first 9 of the 17 single-image texture plus the 4 spectral features, #train=175, #test=112, average correct classification = 92.9%

<table>
<thead>
<tr>
<th>True Category</th>
<th>Assigned Category</th>
<th>Errors of Commission</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROPLAND</td>
<td>CROPLAND</td>
<td>61</td>
</tr>
<tr>
<td>URBAN</td>
<td>CROPLAND</td>
<td>3</td>
</tr>
<tr>
<td>GRASSLAND</td>
<td>CROPLAND</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>CROPLAND</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>URBAN</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>GRASSLAND</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>#ERROR</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>%ERROR</td>
<td>4.7</td>
</tr>
</tbody>
</table>

---

44
### Table 22. Contingency Table for ERTS Image 1021-16333 Using First 11 of the 29 Cross-Band Texture Plus the 4 Spectral Features, \#Train=564, \#Test=372, Average Correct Classification = 90.1%

<table>
<thead>
<tr>
<th>Assigned Category</th>
<th>Cropland</th>
<th>Urban</th>
<th>Grassland</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropland</td>
<td>161</td>
<td>7</td>
<td>0</td>
<td>168</td>
</tr>
<tr>
<td>Urban</td>
<td>15</td>
<td>77</td>
<td>0</td>
<td>92</td>
</tr>
<tr>
<td>Grassland</td>
<td>9</td>
<td>6</td>
<td>97</td>
<td>112</td>
</tr>
<tr>
<td>Total</td>
<td>185</td>
<td>90</td>
<td>97</td>
<td>372</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors of Commission</th>
<th>#Errors</th>
<th>%Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>13.0</td>
<td>14.4</td>
</tr>
</tbody>
</table>

### Table 23. Contingency Table for ERTS Image 1021-16333 Using First 16 of the 29 Cross-Band Texture Plus the 4 Spectral Features, \#Train=565, \#Test=371, Average Correct Classification = 90.3%

<table>
<thead>
<tr>
<th>Assigned Category</th>
<th>Cropland</th>
<th>Urban</th>
<th>Grassland</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropland</td>
<td>159</td>
<td>8</td>
<td>0</td>
<td>167</td>
</tr>
<tr>
<td>Urban</td>
<td>15</td>
<td>76</td>
<td>1</td>
<td>92</td>
</tr>
<tr>
<td>Grassland</td>
<td>7</td>
<td>5</td>
<td>100</td>
<td>112</td>
</tr>
<tr>
<td>Total</td>
<td>181</td>
<td>89</td>
<td>101</td>
<td>371</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors of Commission</th>
<th>#Errors</th>
<th>%Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>22</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>12.2</td>
<td>14.6</td>
</tr>
<tr>
<td>TRUE CATEGORY</td>
<td>ASSIGNED CATEGORY</td>
<td>ERRORS OF COMMISSION</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td>CROPLAND</td>
<td>URBAN</td>
</tr>
<tr>
<td>CROPLAND</td>
<td>156</td>
<td>9</td>
</tr>
<tr>
<td>URBAN</td>
<td>13</td>
<td>79</td>
</tr>
<tr>
<td>GRASSLAND</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>TOTAL</td>
<td>175</td>
<td>89</td>
</tr>
<tr>
<td>#ERRORS</td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td>%ERROR</td>
<td>10.9</td>
<td>11.2</td>
</tr>
</tbody>
</table>

Table 24. Contingency table for ERTS image 1021-16333 using first 21 of the 29 cross-band texture plus the 4 spectral features, #TRAIN=568, #TEST=368, average correct classification = 91.9%
APPENDIX I
TEXTURAL FEATURES OBTAINED FROM THE GREY TONE DEPENDENCE MATRIX

In this appendix, we define 17 textural features which are computed for each of the four angular grey tone dependence matrices.

The following notation will be used in defining the 17 textural features.

\[ P(i, j) \] - \((i, j)^{th}\) entry in a particular grey tone dependence matrix.

\[ P_x(i) \]
\[ \#_R \]

\[ P_y(i) \]
\[ \#_R \]

\[ \#_R \] - number of resolution cell pairs which were considered in computing the entries in \(P(i, j)\).

\[ N_g \] - number of distinct grey tone values in the image.

\[ \mu \] - mean of \(P(i, j)/\#_R\).

\[ P_{x+y}(i) \]
\[ \#_R \]

\[ P_{x-y}(i) \]
\[ \#_R \]

\[ P_{x-y}(i) \] - \((i, j)^{th}\) entry in the distribution of the sum of grey tones of neighboring resolution cells.

\[ P_{x-y}(i) \] - \((i, j)^{th}\) entry in the distribution of the absolute differences in the grey tones of neighboring resolution cells.
TEXTURAL FEATURES

1. Angular Second Moment:

\[ f_1 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \left( \frac{P(i,j)}{\#_R} \right)^2 \]

2. Entropy:

\[ f_2 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} - \left( \frac{P(i,j)}{\#_R} \right) \log \left( \frac{P(i,j)}{\#_R} \right) \]

3. Correlation:

\[ f_3 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i \cdot j \left( \frac{P(i,j)}{\#_R} \right) - \mu_x \mu_y}{\sigma_x \sigma_y} \]

where \( \mu_x \) and \( \sigma_x \) are the mean and standard deviation of \( P_x \), and \( \mu_y \) and \( \sigma_y \) are the mean and standard deviation of \( P_y \).

4. Sum of Squares on \( x \):

\[ f_4 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-\mu)^2 \left( \frac{P(i,j)}{\#_R} \right) \]

5. Product Moment:

\[ f_5 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-\mu)(j-\mu) \left( \frac{P(i,j)}{\#_R} \right) \]
6. Inverse Moment:
\[
f_6 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1+(i-j)^2} \left\{ \frac{P(i,j)}{\#R} \right\}
\]

7. Difference Moment:
\[
f_7 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-j)^2 \left\{ \frac{P(i,j)}{\#R} \right\}
\]

8. Sum Average:
\[
f_8 = \sum_{i=1}^{N_g} \frac{\frac{P_{x+y}^i(i)}{\#R} \{P(i,j)\}}{\#R}
\]

9. Mean:
\[
f_9 = \frac{1}{N_g} \sum_{i=1}^{N_g} \left\{ \frac{P(i,j)}{\#R} \right\}
\]

10. Sum Variance:
\[
f_{10} = \text{variance of } P_{x+y}/\#R
\]

11. Sum Entropy:
\[
f_{11} = \sum_{i=1}^{2N_g} - \left\{ \frac{P_{x+y}(i)}{\#R} \right\} \log \left\{ \frac{P_{x+y}(i)}{\#R} \right\}
\]

12. Contrast:
\[
f_{12} = \sum_{n=0}^{N_g-1} n^2 \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \left( \frac{P(i,j)}{\#R} \right)_{|i-j|=n}
\]
13. Difference Variance:

\[ f_{13} = \text{variance of } \frac{p_{x-y}}{\#R} \]

14. Difference Entropy:

\[ f_{14} = \sum_{i=0}^{N-1} - \left\{ \frac{p_{x-y}(i)}{\#R} \right\} \log \left\{ \frac{p_{x-y}(i)}{\#R} \right\} \]

15, 16, 17. Additional Measures of Correlation:

\[ f_{15} = \frac{HXY - HXY_1}{\max HX, HY} \]

\[ f_{16} = \sqrt{1 - \exp[-2.0(HXY_2 - HXY)]} \]

\[ f_{17} = \sqrt{\text{second largest eigenvalue of } QQ^T} \]

where \( HX \) and \( HY \) are the entropies of the marginals of the transition matrix before quantization, \( HXY \) is the entropy of the transition matrix, and \( HXY_2 \) is the entropy of the product distribution of the marginals before quantization;

\[ Q(i, j) = \frac{p(i, j)}{\sqrt{p_x(i)p_y(j)}} \]

\(*f_{17} \text{ is the maximal correlation coefficient.}\)
APPENDIX II

N-DIMENSIONAL SPHERICAL COORDINATE SYSTEMS
AND ELLIPSOIDALLY SYMMETRIC DISTRIBUTIONS

We illustrate the N-dimensional spherical coordinate system in the calculation
of the volume of the N-dimensional hypersphere. Next we show how suitable functions
can be used to define ellipsoidally symmetric density functions and we determine the
normalizing constant for any function. Finally, we show that for any ellipsoidally
symmetric density \( f(\sqrt{x^T A x}) \), the matrix \( A \) is proportional to the inverse covariance
matrix of \( x \) and we determine the constant of proportionality.

II.1 Volume of an N-dimensional Hypersphere

Let \( V \) be the volume of a N-dimensional hypersphere of radius \( r_0 \). By
definition

\[
v = \int \int \ldots \int_{\sum_{i=1}^{N} x_i^2 \leq r_0} dx_1 \, dx_2 \ldots dx_N
\]

To evaluate this N-fold integral, we transform to spherical coordinates.

\[
x_1 = r \cos \theta_1 \cos \theta_2 \ldots \cos \theta_{N-3} \cos \theta_{N-2} \cos \theta_{N-1}
\]
\[
x_2 = r \cos \theta_1 \cos \theta_2 \ldots \cos \theta_{N-3} \cos \theta_{N-2} \sin \theta_{N-1}
\]
\[
x_3 = r \cos \theta_1 \cos \theta_2 \ldots \cos \theta_{N-3} \sin \theta_{N-2}
\]
\[
\vdots
\]
\[
x_j = r \cos \theta_1 \cos \theta_2 \ldots \cos \theta_{N-3} \sin \theta_{N-2}
\]
\[
\vdots
\]
\[
x_N = r \sin \theta_1
\]

Figure 16 illustrates the geometry of the spherical coordinate system we use for a
3-dimensional system.
Three-Dimensional Spherical Coordinate System

Transformation between rectangular coordinate system and spherical coordinate system.

Figure 16 Three-Dimensional Spherical Coordinate System
The Jacobian $J$ of this transformation is defined by the determinant $J$.

$$
J = \begin{vmatrix}
\frac{\partial x_1}{\partial r} & \frac{\partial x_2}{\partial r} & \cdots & \frac{\partial x_N}{\partial r} \\
\frac{\partial x_1}{\partial \theta_1} & \frac{\partial x_2}{\partial \theta_1} & \cdots & \frac{\partial x_N}{\partial \theta_1} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial x_1}{\partial \theta_{N-1}} & \frac{\partial x_2}{\partial \theta_{N-1}} & \cdots & \frac{\partial x_N}{\partial \theta_{N-1}} \\
\end{vmatrix}
$$

$$
= \begin{vmatrix}
\cos \theta_1 \cos \theta_2 \cdots \cos \theta_{N-1} & \cos \theta_1 \cos \theta_2 \cdots \cos \theta_{N-2} \sin \theta_{N-1} & \cdots & \sin \theta_1 \\
-r \sin \theta_1 \cos \theta_2 \cdots \cos \theta_{N-1} & -r \sin \theta_1 \cos \theta_2 \cdots \cos \theta_{N-2} \sin \theta_{N-1} & \cdots & r \cos \theta_1 \\
-r \cos \theta_1 \sin \theta_1 \cdots \cos \theta_{N-1} & -r \cos \theta_1 \sin \theta_2 \cdots \cos \theta_{N-2} \sin \theta_{N-1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
-r \cos \theta_1 \cos \theta_2 \cdots \sin \theta_{N-1} & r \cos \theta_1 \cos \theta_2 \cdots \cos \theta_{N-2} \cos \theta_{N-1} & \cdots & 0 \\
\end{vmatrix}
$$

To find the value of the Jacobian, factor $r$ out of the last $(N-1)$ rows and from each column factor out its first entry.
\[ J = r^{N-1} \cos^{N-1} \theta_1 \cos^{N-2} \theta_2 \cdots \cos \theta_{N-1} \sin \theta_1 \sin \theta_2 \cdots \sin \theta_{N-1} \]

<table>
<thead>
<tr>
<th></th>
<th>(1)</th>
<th>(1)</th>
<th>(1)</th>
<th>(1)</th>
<th>(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-\tan \theta_1)</td>
<td>(-\tan \theta_1)</td>
<td>(-\tan \theta_1)</td>
<td>(\cot \theta_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-\tan \theta_2)</td>
<td>(-\tan \theta_2)</td>
<td>(-\tan \theta_2)</td>
<td>(0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-\tan \theta_{N-1})</td>
<td>(\cot \theta_{N-1})</td>
<td>(0)</td>
<td>(\vdots)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-\tan \theta_{N-2})</td>
<td>(\cot \theta_{N-2})</td>
<td>(0)</td>
<td>(\vdots)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-\tan \theta_{N-3})</td>
<td>(\cot \theta_{N-3})</td>
<td>(0)</td>
<td>(\vdots)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subtracting column 2 from column 1, column 3 from column 2, \ldots\, column \(N\) from column \(N-1\) there results

<table>
<thead>
<tr>
<th></th>
<th>(0)</th>
<th>(0)</th>
<th>(0)</th>
<th>(0)</th>
<th>(0)</th>
<th>(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
<td>(0)</td>
<td>(\cdots)</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\cdots)</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\cdots)</td>
</tr>
<tr>
<td>(-\tan \theta_{N-1})</td>
<td>(-\tan \theta_{N-1})</td>
<td>(\cot \theta_{N-1})</td>
<td>(\cot \theta_{N-1})</td>
<td>(0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-\tan \theta_{N-2})</td>
<td>(-\tan \theta_{N-2})</td>
<td>(\cot \theta_{N-2})</td>
<td>(\cot \theta_{N-2})</td>
<td>(0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-\tan \theta_{N-3})</td>
<td>(-\tan \theta_{N-3})</td>
<td>(\cot \theta_{N-3})</td>
<td>(\cot \theta_{N-3})</td>
<td>(0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-\tan \theta_{N-4})</td>
<td>(-\tan \theta_{N-4})</td>
<td>(\cot \theta_{N-4})</td>
<td>(\cot \theta_{N-4})</td>
<td>(0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Since all entries in the upper left triangle are zero, the value of the determinant is easily found as minus one times the product of entries on the lower left to upper right diagonal.
\[ J = r^{N-1} \cos^{N-1} \theta_1 \cos^{N-2} \theta_2 \ldots \cos \theta_{N-2} \sin \theta_1 \sin \theta_2 \ldots \sin \theta_{N-1} (-1)^{\frac{N-1}{2}} \frac{\tan^3 \theta - \cot \theta}{N-1} \]

Notice that \( \tan \theta + \cot \theta = \frac{1}{\sin \theta \cos \theta} \). Now upon simplifying we obtain

\[ J = (-1)^{N-1} r^{N-1} \cos^{N-2} \theta_1 \cos^{N-3} \theta_2 \ldots \cos \theta_{N-2} \]

and \( |J| = r^{N-1} \cos^{N-2} \theta_1 \cos^{N-3} \theta_2 \ldots \cos \theta_{N-2} \) since

\[ \cos \theta_i > 0 \text{ for } -\pi/2 < \theta_i < \pi/2, \ i=1, 2, \ldots, N-2. \]

In spherical coordinates the volume \( V \) of the \( N \)-dimensional hypersphere of radius \( r_o \) is readily evaluated.

\[
V = \int_{r=0}^{r=r_o} \int_{\theta_1=0}^{\theta_1=\pi/2} \int_{\theta_2=0}^{\theta_2=\pi/2} \ldots \int_{\theta_{N-1}=0}^{\theta_{N-1}=\pi/2} r^{N-1} \cos^{N-2} \theta_1 \cos^{N-3} \theta_2 \ldots \cos \theta_{N-2} \ dr d\theta_1 \ldots d\theta_{N-1}
\]

Separating the integrations,

\[
V = \int_{r=0}^{r=r_o} \int_{\theta_1=-\pi/2}^{\theta_1=\pi/2} \int_{\theta_2=-\pi/2}^{\theta_2=\pi/2} \ldots \int_{\theta_{N-1}=-\pi/2}^{\theta_{N-1}=\pi/2} r^{N-1} \cos^{N-2} \theta_1 \cos^{N-3} \theta_2 \ldots \cos \theta_{N-2} \ dr d\theta_1 \ldots d\theta_{N-1}
\]

Since \( \int_{\theta=0}^{\pi/2} \cos \theta d\theta = \frac{\Gamma\left(\frac{N+1}{2}\right) \Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{N+2}{2}\right)} \),

\[ \theta = \frac{\pi}{2} \]
\[ V = \frac{r^N_o}{N} \left[ \frac{\Gamma\left(\frac{N-1}{2}\right) \Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \right] \left[ \frac{\Gamma\left(\frac{N-2}{2}\right) \Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{N-1}{2}\right)} \right] \ldots \cdot \frac{\Gamma\left(\frac{2}{2}\right) \Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{3}{2}\right)} \right]^{2\pi} \]

\[ = \frac{r^N_o}{N} \left[ \frac{\Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \right]^{N-2} \frac{\Gamma(1)}{\Gamma\left(\frac{N}{2}\right)} \]

But \( \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi} \) and \( \Gamma(1) = 1 \)

\[ V = \frac{N-2}{N} \frac{2\pi \pi^2}{\Gamma\left(\frac{N-1}{2}\right)} = \frac{2r^N_o}{N} \frac{\pi^2}{\Gamma\left(\frac{N}{2}\right)} \]
11.2 Suitable functions for Ellipsoidally Symmetric Distribution.

Suppose $f$ is a real function, defined on domain $R$, a subset of $[0, \infty]$, and satisfying $f(\mu) \geq 0$ for all $\mu$ in $R$ and $\mu^k f(\mu) d\mu$ is finite for $k \leq N+1$. We show that $f$ is suitable for defining an ellipsoidally symmetric density function and we determine the constant $c$ so that $(\sqrt{x^T A x})$ is an ellipsoidally symmetric density.

Let $A$ be a $N \times N$ symmetric positive definite matrix and $X$ an $N \times 1$ vector. Consider the ellipsoidally symmetric function $f(\sqrt{x^T A x})$. We wish to determine a constant $C$ such that $(\sqrt{x^T A x})$ is a density function.

It is clear that $C = \frac{1}{\int \cdots \int f(\sqrt{x^T A x}) \, dx_1 \cdots dx_N}$.

To determine the value of the integral, we will make a transformation which rotates and scales. Let $T$ be an orthonormal matrix such that $T' A T = D$, where $D$ is a diagonal matrix. Make the change of variables

$$X = T D^{-\frac{1}{2}} z.$$

The Jacobian $J$ of this transformation is

$$J = \begin{vmatrix}
\frac{\partial x_1}{\partial z_1} & \frac{\partial x_1}{\partial z_2} & \cdots & \frac{\partial x_1}{\partial z_N} \\
\frac{\partial x_2}{\partial z_1} & \frac{\partial x_2}{\partial z_2} & \cdots & \frac{\partial x_2}{\partial z_N} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial x_N}{\partial z_1} & \frac{\partial x_N}{\partial z_2} & \cdots & \frac{\partial x_N}{\partial z_N}
\end{vmatrix} = |TD^{-1/2}| = |T| |D|^{-1/2}.$$
Since $T$ is an orthonormal matrix, $|T| = 1$ and

$$|D| = |T'AT| = |T'| |A| |T| = |A|.$$  

So the Jacobian is the determinant $|A|^{-1/2}$ which is positive since $A$ is positive definite.

$$1 = \int \cdots \int f\left(\sqrt{x'Ax}\right) \, dx_1 \cdots dx_N$$

$\sqrt{x'Ax} \in \mathbb{R}$

$$= |A|^{-1/2} \int \cdots \int f\left(\sqrt{z'D^{-1/2}T'ATD^{-1/2}}z\right) \, dz_1 \cdots dz_N$$

$\sqrt{z'D^{-1/2}T'ATD^{-1/2}}z \in \mathbb{R}$

$$= |A|^{-1/2} \int \cdots \int f(z') \, dz_1 \cdots dz_N$$

$\sqrt{z'z} \in \mathbb{R}$

Now change to spherical coordinates.

$$z_1 = r \cos \theta_1 \cos \theta_2 \cdots \cos \theta_{N-1}$$

$$z_2 = r \cos \theta_1 \cos \theta_2 \cdots \cos \theta_{N-2} \sin \theta_{N-1}$$

$$\vdots$$

$$z_j = r \cos \theta_1 \cdots \cos \theta_{N-j} \sin \theta_{N-j+1}$$

$$\vdots$$

$$z_N = r \sin \theta_1$$
The Jacobian of this transformation is \((-1)^N\cos^{N-2} \theta_1 \cos^{N-2} \theta_2 \cdots \cos^{N-2} \theta_{N-2}\).

\[ I = |A|^{-1/2} \int_{r \in \mathbb{R}} \int_{\theta_1 = -\pi/2}^{\pi/2} \int_{\theta_2 = -\pi/2}^{\pi/2} f(r) r^{N-1} \cos^{N-2} \theta_1 \cos^{N-2} \theta_2 \cdots \]

\[ \int_{\theta_{N-2} = -\pi/2}^{\pi/2} \int_{\theta_{N-1} = 0}^{\pi/2} \cos^{N-2} \theta_1 d\theta_1 \cdots \cos^{N-2} \theta_2 d\theta_2 \]

\[ = |A|^{-1/2} \int_{r \in \mathbb{R}} f(r) dr \int_{\theta_1 = -\pi/2}^{\pi/2} \cos^{N-2} \theta_1 d\theta_1 \int_{\theta_2 = -\pi/2}^{\pi/2} \cos^{N-2} \theta_2 d\theta_2 \]

\[ \cdots \int_{\theta_{N-2} = -\pi/2}^{\pi/2} \int_{\theta_{N-1} = 0}^{\pi/2} \cos^{N-2} \theta_{N-2} d\theta_{N-2} \cdots \]

Since \( \int_{\theta = -\pi/2}^{\pi/2} \cos^N \theta d\theta = \frac{\Gamma\left(\frac{N+1}{2}\right)}{\Gamma\left(\frac{N+2}{2}\right)} \), the integrals are readily evaluated.
\[
I = |A|^{-1/2} \int_{\mathbb{R}^n} r^{N-1} f(r) dr \left[ \frac{\Gamma\left(\frac{N-1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \frac{\Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \ldots \frac{\Gamma\left(\frac{2}{2}\right)}{\Gamma\left(\frac{3}{2}\right)} \right]^{2\pi} 
\]

\[
= |A|^{-1/2} \int_{\mathbb{R}^n} r^{N-1} f(r) dr \frac{\Gamma\left(\frac{N-2}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} 2\pi 
\]

\[
= \frac{2^{(n)N/2}}{|A|^{1/2} \Gamma\left(\frac{N}{2}\right)} \int_{\mathbb{R}^n} r^{N-1} f(r) dr 
\]

Therefore, the constant \(c\) is

\[
c = \frac{|A|^{-1/2} \Gamma\left(\frac{N}{2}\right)}{2^{(n)N/2} \int_{\mathbb{R}^n} r^{N-1} f(r) dr} 
\]
Next we determine the normalizing constant $c$ for the forms $e^{-\frac{1}{2}u^2}$ and $(1 + u^2)^{-m}$.

Case 1. Multivariate Normal

The density function for the multivariate normal distribution is of the form

$$f(\sqrt{x'Ax}) = e^{-\frac{1}{2}x'Ax}, \quad 0 \leq x'Ax \leq \infty$$

or,

$$f(r) = e^{-\frac{1}{2}r^2}, \quad 0 \leq r \leq \infty.$$  

Since

$$\int_{r \in R} r^{N-1} f(r) dr = \int_{1}^{\infty} (2u)^{\frac{N-2}{2}} e^{-u} du = 2^{\frac{N-2}{2}} \Gamma\left(\frac{N}{2}\right)$$

then the normalizing constant is

$$c = \frac{\Gamma\left(\frac{N}{2}\right)}{2\pi^{\frac{N}{2}} |A|^{-\frac{1}{2}} \int_{r \in R} r^{N-1} f(r) dr} = \frac{\Gamma\left(\frac{N}{2}\right)}{2\pi^{\frac{N}{2}} |A|^{-\frac{1}{2}} \Gamma\left(\frac{N}{2}\right)}$$

and,

$$f(\sqrt{x'Ax}) = \frac{1}{(2\pi)^{\frac{N}{2}} |A|^{-\frac{1}{2}}} e^{-\frac{1}{2}x'Ax}, \quad 0 \leq x'Ax \leq \infty.$$ 

Case 2. Multivariate Pearson Type VII.

Let

$$f(\sqrt{x'Ax}) = (1 + x'Ax)^{-m}, \quad 0 \leq x'Ax \leq \infty.$$  

then,

$$\int_{r \in R} r^{N-1} f(r) dr = \int_{1}^{\infty} r^{N-1} (1+r^2)^{-m} dr = \int_{0}^{1} \left(1+u^{2}\right)^{-\frac{N-2}{2}} u^m du = \frac{\Gamma\left(\frac{N}{2}\right) \Gamma\left(\frac{N}{2} - m\right)}{\Gamma\left(\frac{N}{2} - m + 1\right)}$$

And the normalizing constant is

$$c = \frac{\Gamma\left(\frac{N}{2}\right) \Gamma\left(m\right)}{2\pi^{\frac{N}{2}} |A|^{-\frac{1}{2}} \Gamma\left(\frac{N}{2}\right) \Gamma\left(m\right)} = \frac{\Gamma\left(\frac{N}{2}\right) \Gamma\left(m\right)}{\Gamma\left(\frac{N}{2} - m + 1\right)}$$

and,

$$f(\sqrt{x'Ax}) = \frac{\Gamma\left(\frac{N}{2}\right) |A|^{\frac{1}{2}} \Gamma\left(\frac{N}{2} - m\right)}{\pi \Gamma\left(\frac{N}{2}\right) \Gamma\left(m\right)} (1 + x'Ax)^{-m}, \quad m > \frac{N}{2}, \quad 0 \leq x'Ax \leq \infty.$$ 

II.3 Covariance Matrix For Multivariate Distributions.

Given the density function $f(\sqrt{x'Ax})$ we want to find the covariance matrix $\gamma$,

$$\gamma = E(xx') = c \int_{\sqrt{x'Ax} \in R} \cdots \int_{\sqrt{x'Ax} \in R} f(\sqrt{x'Ax}) dx_1 \cdots dx_N$$
where \( c \) is a normalizing constant and \( N \) is the dimension of \( x \). Using the orthonormal transformation \( T'AT = D \), where \( D \) is a diagonal matrix, and scaling with \( x = TD^{-\frac{1}{2}}z \), we have

\[
\mathbb{\mathcal{I}} = c \int \cdot \cdot \cdot \int \left( TD^{-\frac{1}{2}}z \right) \left( zD^{-\frac{1}{2}}T' \right) f(\sqrt{z'z}) |A|^{-\frac{1}{2}}dz_1 \cdot \cdot \cdot dz_N
\]

since

\[
x'AX = z'D^{-\frac{1}{2}}T'ATD^{-\frac{1}{2}}z = z'D^{-\frac{1}{2}}D\frac{1}{2}z = z'z
\]

and

\[
J = \begin{bmatrix}
\frac{\partial x_1}{\partial z_1} & \cdots & \frac{\partial x_1}{\partial z_N} \\
\frac{\partial x_2}{\partial z_1} & \cdots & \frac{\partial x_2}{\partial z_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial x_N}{\partial z_1} & \cdots & \frac{\partial x_N}{\partial z_N}
\end{bmatrix} = |T D^{-\frac{1}{2}}| = |T| |D|^{-\frac{1}{2}} = |T'AT|^{-\frac{1}{2}} = |A|^{-\frac{1}{2}}.
\]

Rearranging, \( \mathbb{\mathcal{I}} = c |A|^{-\frac{1}{2}} T D^{-\frac{1}{2}} \int \cdot \cdot \cdot \int z' z' f(\sqrt{z'z}) dz_1 \cdot \cdot \cdot dz_N D^{-\frac{1}{2}} T' \),

since \( z'z \) is an \( N \times N \) matrix. Looking at the off diagonal terms, for \( i \neq j \),

\[
\int \cdot \cdot \cdot \int z_i z_j f(\sqrt{z_i z_i}) dz_1 \cdot \cdot \cdot dz_N = 0 \quad \text{since we are integrating an odd function over even limits.}
\]

function over even limits.

For terms of \( \mathbb{\mathcal{I}} \) along the diagonal, for \( i=j \),

\[
\int \cdot \cdot \cdot \int z_i^2 f(\sqrt{z_i z_i}) dz_1 \cdot \cdot \cdot dz_N = \int \cdot \cdot \cdot \int z_i^2 f(\sqrt{z_i z_i}) dz_1 \cdot \cdot \cdot dz_N
\]

and changing to spherical coordinates,

\[
= \int \int \frac{\pi}{2} \cdot \cdot \cdot \int \frac{\pi}{2} \int r^2 \cos^2 \theta_1 \cdot \cdot \cdot \cos^2 \theta_{N-1} f(r) r^{N-1} \cos N-2 \theta_1 \cdot \cdot \cdot \cos \theta_{N-2} dr \theta_1 \cdot \cdot \cdot d \theta_{N-1}
\]

\[
r \in R \quad \theta_1 = -\frac{\pi}{2} \quad \theta_{N-2} = \frac{\pi}{2} \quad \theta_{N-1} = 0
\]

\[
= \int r^{N+1} f(r) dr \int \int \frac{\pi}{2} \cos^2 N \theta_1 d \theta_1 \cdot \cdot \cdot r \frac{\pi}{2} \cos^3 N-2 \theta_1 d \theta_{N-2} r^{N-2} d \theta_{N-2} \cdot \cdot \cdot N-1
\]

\[
\theta_{N-1} = -\frac{\pi}{2}
\]

Since \( \int \int r^{N+1} f(r) dr = \frac{(N+1)(N+2)}{2} \),

\[
\int \frac{\pi}{2} \cos^k \theta d \theta = \frac{(k+1)(N+2)}{2N}.
\]
then
\[
\int \cdots \int \frac{1}{(\sqrt{2\pi})^n} f(\mathbf{z}) \, dz_1 \cdots dz_N = \int \int f(\mathbf{r}) \, dr_{\mathbf{r}} \frac{\Gamma\left(\frac{N+1}{2}\right) \Gamma\left(\frac{1}{2}\right) \Gamma\left(\frac{N}{2}\right) \Gamma\left(\frac{1}{2}\right) \cdots \Gamma\left(\frac{4}{2}\right) \Gamma\left(\frac{1}{2}\right)}{\Gamma\left(\frac{k+2}{2}\right) \Gamma\left(\frac{N+1}{2}\right) \Gamma\left(\frac{3}{2}\right) \Gamma\left(\frac{1}{2}\right) \Gamma\left(\frac{4}{2}\right)}
\]
\[
= \frac{2\pi^{\frac{N}{2}}}{\Gamma\left(\frac{N}{2}\right)} \int f(\mathbf{r}) \, dr_{\mathbf{r}}
\]

From II.2,
\[
c = \frac{\Gamma\left(\frac{N}{2}\right) |A|^\frac{1}{2}}{2 \pi^{\frac{N}{2}} \int f(\mathbf{r}) \, dr_{\mathbf{r}}}
\]

so that
\[
\Sigma = \frac{\Gamma\left(\frac{N}{2}\right) |A|^\frac{1}{2}}{2 \pi^{\frac{N}{2}} \int f(\mathbf{r}) \, dr_{\mathbf{r}}} |A|^{-\frac{1}{2}} TD^{-\frac{1}{2}} \frac{\Gamma\left(\frac{N}{2}\right)}{\Gamma\left(\frac{N}{2}\right)} \int f(\mathbf{r}) \, dr_{\mathbf{r}}
\]

and,
\[
\Sigma = TD^{-1}T \frac{\int f(\mathbf{r}) \, dr_{\mathbf{r}}}{N\int f(\mathbf{r}) \, dr_{\mathbf{r}}}
\]

where \( T^T A T = D \), or \( D^{-1} = T^T A^{-1} T \).

So that
\[
\Sigma = \frac{\int f(\mathbf{r}) \, dr_{\mathbf{r}}}{N\int f(\mathbf{r}) \, dr_{\mathbf{r}}}
\]

Since the integrals are constants for any \( f \), the covariance matrix is directly proportional to \( A^{-1} \). We determine the constant of proportionality for the multivariate normal and Pearson Type VII distributions.
Case 1. Multivariate Normal

For the multivariate normal, the density is of the form \( f(r) = ce^{-\frac{1}{2}r^2} \) where \( r^2 = x'Ax, 0 \leq r \leq \infty \).

Since, \( \int_{0}^{\infty} e^{-\frac{1}{2}r^2} dr = 2 \cdot \Gamma \left( \frac{k+1}{2} \right) \)

then \( \int_0^{\infty} f(r) dr = c \cdot 2^{\frac{N}{2}} \cdot \Gamma \left( \frac{N+k+2}{2} \right) \)

and \( \int_0^{\infty} N \cdot f(r) dr = c \cdot 2^{\frac{N-2}{2}} \cdot \Gamma \left( \frac{N}{2} \right) \)

so that \( \mathbf{f} = A^{-1} \cdot 2^{\frac{N}{2}} \cdot \frac{\Gamma \left( \frac{N}{2} \right)}{\Gamma \left( \frac{N+k+2}{2} \right)} \cdot \frac{\Gamma \left( \frac{N}{2} \right)}{\Gamma \left( \frac{N}{2} \right)} \)

and thus, \( \mathbf{f} = A^{-1} \)

or, \( A = f^{-1} \).

Case 2. Multivariate Pearson Type VII.

\( f(r) = c \cdot (1 + r^2)^{-m}, 0 \leq r \leq \infty \).

Since \( \int_0^{\infty} r^{N+1} f(r) dr = \frac{c}{2} \cdot \frac{\Gamma \left( \frac{N}{2} + 1 \right) \cdot \Gamma \left( \frac{N-m}{2} - 1 \right)}{\Gamma \left( m \cdot \frac{N}{2} \right)} \)

and \( \int_0^{\infty} r^{N-1} f(r) dr = \frac{c}{2} \cdot \frac{\Gamma \left( \frac{N}{2} \right) \cdot \Gamma \left( \frac{N-m}{2} \right)}{\Gamma \left( m \cdot \frac{N}{2} \right)} \)

then, \( \mathbf{f} = A^{-1} \cdot \frac{\mathbf{c} \cdot N \cdot \Gamma \left( \frac{N}{2} \right) \cdot \Gamma \left( \frac{N-m}{2} - 1 \right)}{2N \cdot \mathbf{c} \cdot \Gamma \left( m \cdot \frac{N}{2} \right)} \cdot \frac{2 \cdot \Gamma \left( m \cdot \frac{N}{2} \right)}{\Gamma \left( m \cdot \frac{N}{2} - 1 \right) \cdot \Gamma \left( m \cdot \frac{N}{2} - 1 \right)} \)

\( = A^{-1} \cdot \frac{1}{2} \cdot \frac{1}{\left( m \cdot \frac{N}{2} \right) - 1} \)
APPENDIX III

NORMALIZATION PROCEDURE TO MAKE COVARIANCE MATRIX INvariant UNDER TRANSLATING AND SCALING TRANSFORMATIONS

Let \( \mathbf{T}_x \) be a covariance matrix for the difference vectors of grey tone N-tuples in a specified spatial relationship within a subimage. We transform the covariance matrix to obtain the normalized covariance matrix \( \mathbf{T}_y \) using \( y = \mathbf{D} x \), where \( x \) is the difference vector and \( \mathbf{D} \) is diagonal. Thus, assuming zero mean,

\[
\mathbf{T}_y = E(y y') = E(D x x' D') = D E(x x') D' = D \mathbf{T}_x D \quad \text{since} \quad D' = D.
\]

For normalization, we have

\[
d_{ii} = \frac{1}{\sqrt{\sigma_{ii}^2}}
\]

where \( \sigma_{ii}^2 \) is the \( ii^{th} \) element of \( \mathbf{T}_x \) and is the variance \( \sigma_i^2 \) of the \( i^{th} \) component of \( x \).

Assume that all grey tone N-tuples have a scale factor \( a \) and an additive factor \( c \) so that for N-tuples \( a x_1 + c \) and \( a x_2 + c \), the difference becomes

\[
y = (a x_1 + c) - (a x_2 + c) = a(x_1 - x_2).
\]

Hence, translational effects due to bias terms are cancelled but scaling effects are marked by the diagonal transformation \( y = \mathbf{A} x \) so that the elements of the covariance matrix become

\[
\mathbf{t} = E(\mathbf{y} \mathbf{y}') = E(\mathbf{A} x x' \mathbf{A}') = \mathbf{A} \mathbf{T}_x \mathbf{A}
\]
where $A$ is a diagonal matrix. We must show that $\mathbf{\bar{y}}_N$, the normalized covariance matrix of $\mathbf{y}$, is identical to $\mathbf{\bar{y}}_y$. Normalizing $\mathbf{\bar{y}}_x$ we have

$$\mathbf{\bar{y}}_N = D \mathbf{\bar{y}}_x D$$

$$= D (A \mathbf{\bar{y}}_x A) D$$

where $D$ is again diagonal but in this case,

$$d_{ii} = \frac{1}{\sqrt{\sigma_{ii} \sigma_{ii}^2}}$$

with $\sigma_{ii}$ the $ii$th element of diagonal matrix $A$. For the $ij$th element of $\mathbf{\bar{y}}_N$ we have

$$\sigma_{Nij} = d_{ii} \sigma_{ij} \sigma_{jj} d_{jj}$$

$$= \frac{\sigma_{ij}}{\sqrt{\sigma_{ii} \sigma_{ii}^2}} \frac{\sigma_{jj}}{\sqrt{\sigma_{jj} \sigma_{jj}^2}}$$

$$= \sigma_{ij}$$

Thus, this procedure of normalization makes the entries of the normalized covariance matrix invariant with respect to translating and scaling transformations on the grey tone $N$-tuples.
APPENDIX IV
COMPUTER PROGRAM DOCUMENTATION & LISTINGS

IV.1 PROGRAM DOCUMENTATION

IV.1-a ERTS Retrieval Programs
IV.1-b Texture Analysis Programs
IV.1-c Cross-Band Texture Analysis Programs
IV.1-d Piecewise Linear Classification Programs

IV.2 PROGRAM LISTINGS

IV.2-a ERTS Retrieval Programs
IV.2-b Texture Analysis Programs
IV.2-c Cross-Band Texture Analysis Programs
IV.2-d Piecewise Linear Classification Programs
IV.1-a ERTS Retrieval Programs - Documentation

RETV
  ERTS
  PIXEY
  ZEQUAN
  PITCHR
  WRTDSK
  RDDSK1
  KEQUAN
  RDDSK2
ERTS RETRIEVAL PROGRAMS

PROGRAM TITLE: RETV
VERSION: 1
DATE: January, 1974
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW635
PURPOSE:

This is the mainline for the ERTS Retrieval Programs which retrieve ERTS MSS data from standard NASA digital tapes and outputs the image data in a picture, or greytone listing, or copies it onto an output tape.

INPUT PARAMETERS under NAMELIST 'PARAM':

NBAND Band number, 1 through 4 for MSS band 4 through 7, to be selected. Set to 5 for all 4 bands, assumed to be 2.
IRSTART, ICSTART Row, column starting coordinates.
IRSTOP, ICSTOP Row, column stopping coordinates.
TITLE 80 Column title for output list.
MILLI TRUE if coordinates are in millimeters; assumed to be FALSE, coordinates are row, column points in ERTS image.
SMALL TRUE if millimeter coordinates are from a small 70 mm by 70 mm negative; assumed to be FALSE, coordinates from a 7 inch by 7 inch print.
PRNT TRUE for grey tone listing; assumed FALSE.
TAPE TRUE for tape output; assumed FALSE.
REQUIREMENTS:

1. ERTS input tape must be on file code 'ES'.
2. Four disc files must be on files 11, 12, 13 and 14.
3. Any output tape must be positioned on file code 'IFIL'.
4. All coordinates must be determined relative to the input tape rather than the print -- that is, ICSTOP must not exceed 824 points or 46 mm.
5. Core = 22 k
6. Subprograms required:
   - RETV
   - ERTS
   - PIXEY
   - ZEQUAN
   - ERTS
   - PITCHR
   - WRTDSK
   - RDDSK1
   - KEQUAN
   - PITCHR
   - RDDSK2

COMMENTS:

For efficiency, data is read by RETV in blocks of 41 lines by 41 points. One ERTS tape (one-fourth of an image) will then be covered by 20 horizontal blocks, leaving 4 points left at the end of each line. Note --- sometimes on the first tape of an image, the first four points are greytones of 255 and can adversely affect a picture printout. If so, set ICSTRT = 5.
Also note that if the point ICSTRT is not a multiple of 41 from the end of the line, then the last points may not necessarily be listed since blocks are determined starting from ICSTRT.

A special picture run can be made to print out and reduce any image over the entire tape by setting SPIC to TRUE and specifying under namelist PARAM the following parameters: IRSTRT, IRSTOP, ICSTRT, ICSTOP, NBPAND, QUAN. Then under namelist PICTUR, parameters for PITCHR are specified. See PIXEY for details.
ERTS RETRIEVAL PROGRAM

SUBPROGRAM TITLE: ERTS
VERSION: II
DATE: September, 1972
UPDATE: November, 1973
AUTHOR: G. Gunnels
DOCUMENTED BY: R. Bosley
PROGRAM LANGUAGE: GMAP
IMPLEMENTED ON: HW635
PURPOSE: To read 7-track ERTS MSS data tapes.

ENTRY POINTS:
CALL EINIT (NOLS)
CALL ESKIP (NOSK)
CALL EREAD (I, LN)
CALL EREWIND

ARGUMENTS:
NOLS Number of words per scan line; returned by EINIT.
NOSK The number of records to skip.
I The array into which the NOLS words of data
LN Returned by EREAD giving the line number of the
from a line of ERTS data is placed.
line of data returned. If LN = 0, the end of file
was reached on the ERTS tape.

ERROR FLAGS:
MB EREAD buffer is not large enough for a block of
AI EINIT was called twice.
NI EINIT was not called before calling EREAD,
ESKIP, or EREWIND.
UE  EOF encountered while reading ID or annotation blocks on ERTS tape.

EF  EOF encountered while trying to skip records in ESKIP.

COMMENTS:
EINIT initializes the ERTS tape so that data may be read, and must be called first. ESKIP skips over NOSK records (scan lines). EREWIND rewinds the ERTS tape. EINIT must not be called twice. The data placed into array I by EREAD is in standard corresponding point forms. Since the ERTS MSS data has four channels, there are actually NDLS/4 points or cells per scan line.

REQUIREMENTS:
ERTS tape must be on file code 'ES'.
ERTS RETRIEVAL PROGRAMS

SUBPROGRAM TITLE: PIXEY
VERSION: 1
DATE: January, 1974
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW635

PURPOSE: To provide a flexible option for printing a picture of ERTS MSS data using user specified parameters to PITCHR.

ENTRY POINT:
CALL PIXEY (ILINE, IMAGE, IRSTRT, IRSTOP, ICSTRT,
ICSTOP, QUAN, NBAND)

INPUT ARGUMENTS:
ILINE Array the ERTS line is read into.
IMAGE Array containing one line of the image.
IRSTRT, IRSTOP Starting, stopping row in the image.
ICSTRT, ICSTOP Starting, stopping column in the image.
QUAN TRUE for equal probability quantizing of the image.
NBAND Bond number to be processed; set to 5 for all 4 MSS bands.

INPUT PARAMETERS: under NAMELIST 'PICTUR':
LNSKIP, Line and column increment for ERTS data;
KOLSKIP assumed to be 4,3.
ICELL, JCELL  Number of rows, columns in image to be printed; assumed to be 1,256.

INIT  Number of times entry to PITCHR is made at SNAP: must be greater than 1; assumed to be 304.

IMIN, IMAX  Minimum, maximum greytones in image; assumed to be 0,75 or 12 if QUAN is true.

NROW  Number of rows to be printed; equal to ICELL.

NFFILES  Number of output files available to PITCHR; set to 0 for all output on file code 06, set to 2 for files 06 and 42; assumed to be 2.

IFIL(10)  Array containing output file codes; assumed to be 06 and 42.

NULW, NULD  Number of columns, rows per output page; assumed to be 129, 60.

AMAG, DMAG  Width, length magnification for output picture; assumed to be 1.

SAMPLE RUN:

$PARAM SPIC=T, QUAN=T, IRSTRT=1, IRSTOP=1216, ICSTRT=5, ICSTOP=772, NBAND=2$END/$PICTUR INIT=304, NFFILES=2, LNSKI=4, KOLSKP=3, JCELL=256$END.

This run will print out on files 06 and 42 a picture 256 points wide by 304 lines long. Note that 1216/4 = 304 and 786/3 = 256 gives the values for LNSKI=4, KOLSKP=3, JCELL=256, INIT=304. Also, using these values for LNSKI and KOLSKP will result in a picture in proportion to the same area on an ERTS image print, approximately twelve 64 by 64 subimages across by 19 subimages down the tape.

COMMENTS:

If SPIC=T in the $PARAM card, then a $PICTUR card must follow. Note that the parameters IRSTRT, IRSTOP, ICSTRT, ICSTOP, QUAN, NBAND are supplied on the $PARAM card while LNSKI, KOLSKP and all PITCHR parameters are supplied on the $PICTUR card.
The program will output a negative picture but a positive picture can be made by setting IMAX=0 and IMIN=75 (or 12 if quantization is used). The parameters are initialized to output an area approximately twelve 64 by 64 subimages wide by 19 down. For a complete description of PITCHR parameters, see PITCHR.

**REQUIREMENTS:**

1. Processor time for one bond with parameters as shown in the sample run is 0.075 hr with 3k lines of output.
2. Subprograms required are ZEQUAN, PITCHR.

**CALLED BY:**

RETV
ERTS RETRIEVAL PROGRAM

SUBPROGRAM TITLE: ZEQUAN
VERSION: 1
DATE: September, 1973
UPDATE: September, 1973
AUTHOR: Z. Dinstein
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW635

PURPOSE: To equal probability quantize a large image on disc to NQ levels.

ENTRY POINT:
CALL ZEQUAN (LINE, NUMLIN, NUMPPL, NCOMP, ICOMP,
LEFT, NQ, INFILE, IOUTFL)

INPUT ARGUMENTS:
LINE Array to store one line of the image.
NUMLIN Number of lines in the image.
NUMPPL Number of columns in the image.
NCOMP Number of components in the image.
ICOMP The component to be quantized.
LEFT Left-most cell desired in the line.
NQ Number of quantized levels.
INFILE File code of disc containing the image to be quantized.

OUTPUT ARGUMENTS:
IOUTFL Output file code for the quantized image.
COMMENTS: Processing is done line by line after an initial pass through the image to determine the number of grey tone levels in the image. The minimum and maximum grey tones are printed. The number of grey tone levels should not exceed 512. Input data on disc INFILE must be in binary.

CALLED BY: PIXEY
ERTS RETRIEVAL PROGRAM

SUBPROGRAM TITLE: PITCHR
VERSION: II
DATE: July, 1969
UPDATE: November, 1970
AUTHOR: R. Cowles
DOCUMENTED BY: G. Elliott
PROGRAM LANGUAGE: GMAP
IMPLEMENTED ON: HW635
PURPOSE:
To print out images in 13 grey levels.

ENTRY POINTS:
CALL PITCHR (IRRAY, ICELL, JCELL, INIT, IT, IMIN, IMAX,
NROW, NFIL, IFIL, NULW, NULD, AMAG, DMAG, *)
CALL SNAP
CALL SNAPA (IARRAY)
CALL ENDBNR

ARGUMENTS:
IRRAY Array to be printed, either integer or floating point.
ICELL Number of rows in array. (row dimension)
JCELL Number of columns in array. (column dimension)
INIT = 0 if all of image to be printed out is in core at time of call. Output will be done before
return to calling program.
=1 for reinitialization entry. Any of the arguments previously specified with INIT = 0 with the
exception of INIT may now be changed. Return will be made to the calling program without
any output. This is especially useful if the image is read into core in pieces and the last piece
does not completely fill the array. >1 for initialization
entry. INIT will reflect the number of times entry is made at SNAP before final border is to be printed. Return is to the calling routine without any output. INIT = 0 is assumed.

II

= 0 if array is floating point
= 1 if array is integer
II = 1 is assumed.

IMIN
Minimum brightness level in array. Type of IMIN should correspond to that indicated by II.
IMIN = 0 is assumed.

IMAX
Maximum brightness level in array. Type of IMAX should correspond to that indicated by II.
IMAX = 12 is assumed.

NROW
Number of rows of array to be printed if full array is not to be printed. This allows for partial printing.
NROW = ICELL is assumed.

NFIL
Number of output files available if image is to be output in strips that are NULW lines wide.
= 0 for all output on file code 06
= 1 for all output on file code IFIL
1 for outputs in strips, on file codes specified in array IFIL. NFIL = 0 is assumed.

IFIL
Array containing output file codes. Ignored unless NFIL = 0. IFIL must be a variable rather than a literal, since SNAP alters the value(s) of IFIL to contain the location of the relevant file control block in the upper half of the word.

NULW
Number columns per output page. MAX = 129
NULW = 120 unless otherwise specified.

NULD
Number of rows to be printed before a slew on the top of the next page is given. NULD = ICELL*DMAG + 1.
AMAG
Floating point magnification in width.
AMAG = 1 is assumed.

DMAG
Floating point magnification in length.
DMAG = 1 is assumed.

Error return if not enough output files are available. Return is made with NARG = -10.
If not specified, a message will be printed out on the accounting report and an NF report will terminate execution.
ERTS RETRIEVAL PROGRAMS

SUBPROGRAM TITLE: WRTDSK
VERSION: II
DATE: September, 1972
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW635

PURPOSE:
To write one line of ERTS data onto disc files, one file for each band.

ENTRY POINT:
CALL WRTDSK (ILINE, NHOR, IPSTR, IPEND, NBAND)

INPUT ARGUMENTS:
ILINE Array containing one line of ERTS data
NHOR Number of horizontal blocks of 41 columns in the line.
IPSTR, IPEND Starting and ending points in the ERTS line.
NBAND The desired band; set to 5 for all 4 MSS bands.

COMMENTS:
After reading a line of data into ILINE, RETV calls WRTDSK to pick out the segment of NHOR blocks in the line and write it onto disc. If only one band is desired, only that band is put onto disc. Disc files 11, 12, 13, and 14 must be present.

CALLED BY:
RETV

82
ERTS RETRIEVAL PROGRAMS

SUBPROGRAM TITLE: RDDSKL
VERSION: II
DATE: September, 1972
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW635

PURPOSE: To read the ERTS data from disc and print out a picture of the image, proceeding vertically and then from left to right.

ENTRY POINT:

CALL RDDSKL (IMAGE, QUAN, NHOR, NVERT, NDSK)

INPUT ARGUMENTS:

IMAGE Array used to store two 41 x 41 blocks.
QUAN TRUE for equal probability quantization of the image.
NHOR Number of horizontal 41 x 41 blocks.
NVERT Number of vertical 41 x 41 blocks.
NDSK File code of disc to be processed.

ERROR FLAGS:

Same as for PITCHR.

COMMENTS:

Since blocks are 41 columns wide, then two horizontal blocks are processed together, except for the final strip of blocks when NHOR is an odd number. It is recommended that QUAN be set to true since transmission errors result in very high and very low greytones along the line which will make the true image features indistinguishable unless equal probability quantizing is used.
SUBPROGRAMS CALLED:
  KEQUAN
  PITCHR

CALLED BY:
  RETV
ERTS RETRIEVAL PROGRAMS

SUBPROGRAM TITLE: KEQUAN
VERSION: II
DATE: September, 1971
UPDATE: June, 1973
AUTHOR: G. Elliot
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW635

PURPOSE:
To equal probability quantize the input array to NQ levels.

ENTRY POINT:
CALL KEQUAN (IA, NGL, NQ, IASIZE)

ARGUMENTS:
IA Input array which is returned quantized.
NGL Number of greytone levels in IA.
NQ Number of quantizing levels.
IASIZE Size of array IA.

ERROR FLAGS:
If the number of greytone levels exceeds 512, an error message is printed.

CALLED BY:
RDDS1
ERTS RETRIEVAL PROGRAMS

SUBPROGRAM TITLE: RDDSK2

VERSION: II

DATE: September, 1972

UPDATE: January, 1974

AUTHOR: Robert J. Bosley

DOCUMENTED BY: Robert J. Bosley

PROGRAM LANGUAGE: FORTRAN IV

IMPLEMENTED ON: HW635

PURPOSE:

To read ERTS data from disc and print out the grey tones and/or copy the data onto an output tape.

ENTRY POINT:

CALL RDDSK2 (IMAGE, IRSTRT, IRSTOP, NHOR, NDSK, PRNT, TAPE, IFIL)

INPUT ARGUMENTS:

IMAGE Array data is read into.
IRSTRT, IRSTOP Starting, stopping lines of data.
NHOR Number of horizontal blocks of 41 columns.
NDSK File code of the disc to be read.
PRNT TRUE for greytone listing.
TAPE TRUE for tape output
IFIL Output tape file code.

COMMENTS:

If neither the grey tone listing nor the output tape is desired, execution is returned to the calling program, RETV.

CALLED BY:

RETV
IV.1-b Texture Analysis Programs - Documentation

MAINLN
ERTS
MAING
KEQUAN
PITCHR
FPLXIT
INDEX
IMOMTR
COR
IEQPQ1
RITOWT
ERTS TEXTURE ANALYSIS

PROGRAM TITLE: MAINLN
VERSION: II
DATE: September, 1972
UPDATE: June, 1973
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Craig Paul
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE:
To read the Texture Analysis parameters, read in subimages from ERTS input tape, and send these subimages to MAING for processing.

INPUT PARAMETERS:
1. Title Card: Up to 80 columns for use as identification information.

2. Parameters under NAMELIST /PARAM/:
   IBAND  Band used for processing:
      = 1 for MSS Band 4
      = 2 for MSS Band 5
      = 3 for MSS Band 6
      = 4 for MSS Band 7
      assumed to be 2.
   NUMPPL  Number of points per line of each subimage;
      assumed to be 64.
   NUMLIN  Number of lines of each subimage; assumed to be 64.
   NBVERT  Number of the last vertical subimage to be processed; assumed to be 36.
   NUMIM  Number of subimages taken horizontally in one run; assumed to be 3. Note:
      NUMPPL x NUMIM must not exceed 192 points.
N11 The upper-left column coordinate for the vertical strip (of NUMIM horizontal subimages) being processed in this run.

NUMSTR The number of the vertical strip being processed; assumed to be 1.

NBSKIP The number of vertical rows of subimages to skip before beginning processing; assumed to be 0.

PNCH Specifies the output option:
= 1HY for card output
= 1HT for tape output on file code IF
= 1HN for neither card nor tape but a listing of the LEX arrays; assumed to be tape (1HT).

Note: PNCH must be denoted as a Hollerith constant in the $PARAM card.

IF Output tape file code which is assumed to be in position (it is not rewound); assumed to be '03'.

NRED The base used for image reduction; assumed to be 1.

NSTART, NTIMES Each subimage is processed NSTART through NTIMES times, each time with a new reduction factor NFT = NRED ** NLAYER where NLAYER goes from NSTART - 1 through NTIMES - 1; both assumed to be 1.

PICTUR TRUE for a picture of each subimage; assumed FALSE.

NQUANT Number of quantization levels of the probability function in IMOMTR; assumed to be 16.

MERGE TRUE for all four LEX arrays to be merged into one array; assumed to be TRUE.
REQUIREMENTS:

1. ERTS input tape must be on file code 'ES'.
2. Core - 43k.
3. Disc file must be on file code '02'.
4. Subprograms required:
   MAINLN
   ERTS
   MAING
   KEQUAN
   PITCHR
   FPLXIT
   INDEX
   IMOMTR
   INDEX
   COR
   IEQPQ1
   RITOWT

CARD SETUP FOR SAMPLE RUN:

$ IDENT 9999, ANYNAME
$ LIBRARY LB
$ OBJECT ERTS TEXTURE ANALYSIS PROGRAMS
$ DKEND
$ EXECUTE
$ PRMFL LB, R, S, PATTERN/GEE/LIB
$ TAPE ES, A5DD, 60500, ERTS00, INPUT
$ FILE 02, A2R, 2L
$ LIMITS 20, 43k, 10k
$ INCODE IBMF

TEST-SETUP FOR TEXTURE ANALYSIS PROGRAMS
$ PARAM N11=1, PNCH = 1HN, PICTUR = TSEND
$ ENDJOB

This run of the texture analysis programs will process the ERTS image in 64 x 64 subimages, giving only printed output plus a picture of each subimage.
Comments:

This is the mainline of the texture analysis programs. Each ERTS image is divided into 4 vertical strips, each put onto a 7-track digital tape. This image is divided up into subimages for processing. For example, if the subimages are 64 columns by 64 lines, then the first tape contains subimages 1 through 12, the second contains 13 through 24, and so on up to 48 horizontal subimages. And, each tape contains 36 vertical subimages. Often, the first 8 points at the beginning of each line have grey tones of 255 and will adversely affect processing. Therefore, the first eight points of each line are skipped.

Each tape contains usually 3296 points of 4 bands, or 824 points per line for one band. Skipping the first eight points leaves 816 points per line. If subimages are each 64 columns, then 12 will occupy 768 columns, leaving 48 points at edge of each tape unused.

Due to core limitations, each input tape is processed in vertical strips of up to 192 horizontal points. This is 3,64 by 64 subimages, or 6,32 by 32 subimages in one run. For example, if NUMPPL = 64 and NUMIM = 3, then the tape is processed as follows: Run 1 - (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), ..., (36, 1), (36, 2), (36, 3). Run 2 - (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), ..., (36, 4), (36, 5), (36, 6). Run 3 - (1, 7), (1, 8), (1, 9), (2, 7), (2, 8), (2, 9), (3, 7), ..., (36, 7), (36, 8), (36, 9). Run 4 - (1, 10), (1, 11), (1, 12), ..., (36, 10), (36, 11), (36, 12). This completes the processing of the first tape. Note that N11 is determined relative to the entire image while NUMSTR is relative to the input tape. That is, for tape 1, N11 and NUMSTR are both 1 for Run 1. But for Run 2, N11 is 4 while NUMSTR is 2 and for Run 3, N11 is 7 while NUMSTR is 3, etc.
Continuing with the example, type 2 will be processed as follows: Run 1 (N11 = 13, NUMSTR = 1) - (1, 13), (1, 14), (1, 15), ..., (36, 13), (36, 14), (36, 15). Run 2 (N11 = 16, NUMSTR = 2) - (1, 16), (1, 17), (1, 18), ..., (36, 16), (36, 17), (36, 18), etc.

See Figure 1 for an illustration of an ERTS image divided into 64 by 64 subimages. Note – the size of each subimage must not exceed 4096 points.
EROS IMAGE

4 Tapes for Each EROS Image

<table>
<thead>
<tr>
<th>64 x 64 SUBIMAGE</th>
<th>UNUSED 48 x 64 SUBIMAGE</th>
<th>824 Points</th>
<th>3 x 192 Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>14</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>24</td>
<td>27</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>34</td>
<td>37</td>
<td>40</td>
</tr>
<tr>
<td>41</td>
<td>44</td>
<td>47</td>
<td>50</td>
</tr>
<tr>
<td>51</td>
<td>54</td>
<td>57</td>
<td>60</td>
</tr>
<tr>
<td>61</td>
<td>64</td>
<td>67</td>
<td>70</td>
</tr>
</tbody>
</table>

1 Strip

| TAPE 1 | TAPE 2 | TAPE 3 | TAPE 4 |

Figure 1.
SUBPROGRAM TITLE: ERTS

VERSION: II

DATE: September, 1972

UPDATE: November, 1973

AUTHOR: G. Gunnels

DOCUMENTED BY: R. Bosley

PROGRAM LANGUAGE: GMAP

IMPLEMENTED ON: HW635

PURPOSE:

To read 7-track ERTS MSS data tapes.

ENTRY POINTS:

CALL EINIT (NOLS)
CALL ES Skip (NOSK)
CALL EREAD (I, LN)
CALL EREWND

ARGUMENTS:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOLS</td>
<td>Number of words per scan line; returned by EINIT.</td>
</tr>
<tr>
<td>NOSK</td>
<td>The number of records to skip.</td>
</tr>
<tr>
<td>I</td>
<td>The array into which the NOLS words of data from a line of ERTS data is placed.</td>
</tr>
<tr>
<td>LN</td>
<td>Returned by EREAD giving the line number of the line of data returned. If LN = 0, the end of file was reached on the ERTS tape.</td>
</tr>
</tbody>
</table>

ERROR FLAGS:

<table>
<thead>
<tr>
<th>Error Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB</td>
<td>EREAD buffer is not large enough for a block of ERTS data.</td>
</tr>
<tr>
<td>AI</td>
<td>EINIT was called twice.</td>
</tr>
<tr>
<td>NI</td>
<td>EINIT was not called before calling EREAD, ESkip, or EREWND.</td>
</tr>
</tbody>
</table>
UE
EOF encountered while reading ID or annotation blocks on ERTS tape.

EF
EOF encountered while trying to skip records in ESKIP.

COMMENTS:
EINIT initializes the ERTS tape so that data may be read, and must be called first. ESKIP skips over NOSK records (scan lines). EREWIND rewinds the ERTS tape. EINIT must not be called twice. The data placed into array 1 by EREAD is in standard corresponding point forms. Since the ERTS MSS data has four channels, there are actually NDLS/4 points or cells per scan line.

REQUIREMENTS:
ERTS tape must be on file code 'ES'.
ERTS TEXTURE ANALYSIS

SUBPROGRAM TITLE: MAING
VERSION: III
DATE: October, 1971
UPDATE: June, 1973
AUTHOR: R. M. Haralick
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE:
To print out a picture of the subimage, copy it to file 2, and then process it through the texture subroutines FPLXIT to get the LEX arrays, IMOMTR to calculate the texture features, and RITOWT to output the results.

ENTRY POINT:
CALL MAING (IWORK, MERR, MERGE, PICTUR, IF)

ARGUMENTS:
IWORK
NUMLIN by NUMPPL subimage array.
MERR
Set to 1 if an error occurs in fitting the LEX arrays into IWORK.
MERGE
TRUE to merge the four LEX arrays into one.
PICTUR
TRUE for a picture of the subimage.
IF
File code for the output tape.

COMMENTS:
The subimage sent in IWORK is scaled to fill a page for the picture printout, and at the same time it is copied to a scratch disc on file code '02'. If the size of the LEX arrays is greater than NUMPPL x NUMLIN, then an error message is printed and processing is terminated by MAINLN, after putting EOF marks on the output tape. Each subimage is quantized by KEQUAN to 32 levels.
CALLED BY:

MAINLN

SUBPROGRAMS REQUIRED:

KEQUAN
PITCHR
FPLXIT
IMOMTR
RITOWT
SUBPROGRAM TITLE: KEQUAN
VERSION: II
DATE: September, 1971
UPDATE: June, 1973
AUTHOR: G. Elliot
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW635
PURPOSE: To equal probability quantize the input array to NQ levels.

ENTRY POINT: CALL KEQUAN (IA, NGL, NQ, IASIZE)

ARGUMENTS:

IA: Input array which is returned quantized.
NGL: Number of greytone levels in IA.
NQ: Number of quantizing levels.
IASIZE: Size of array IA.

ERROR FLAGS:
If the number of greytone levels exceeds 512, an error message is printed.
SUBPROGRAM TITLE: PITCHR
VERSION: II
DATE: July, 1969
UPDATE: November, 1970
AUTHOR: R. Cowles
DOCUMENTED BY: G. Elliott
PROGRAM LANGUAGE: GMAP
IMPLEMENTED ON: HW635
PURPOSE: To print out images in 13 grey levels.

ENTRY POINTS:
CALL PITCHR (IRRAY, ICELL, JCELL, INIT, IT, IMIN, IMAX,
NROW, NFIL, IFIL, NULW, NULD, AMAG, DMAG, *)
CALL SNSAP
CALL SNSAPA (IARRAY)
CALL ENDBNR

ARGUMENTS:
IRRAY Array to be printed, either integer or floating point.
ICELL Number of rows in array. (row dimension)
JCELL Number of columns in array. (column dimension)
INIT =0 if all of image to be printed out is in core at time of call. Output will be done before return to calling program.
=1 for reinitialization entry. Any of the arguments previously specified with INIT =0 with the exception of INIT may now be changed. Return will be made to the calling program without any output. This is especially useful if the image is read into core in pieces and the last piece does not completely fill the array. >1 for initialization
entry. INIT will reflect the number of times entry is made at SNAP before final border is to be printed. Return is to the calling routine without any output.

INIT =0 is assumed.

II
=0 if array is floating point
=1 if array is integer
II =1 is assumed.

IMIN
Minimum brightness level in array. Type of IMIN should correspond to that indicated by II.
IMIN =0 is assumed.

IMAX
Maximum brightness level in array. Type of IMAX should correspond to that indicated by II.
IMAX =12 is assumed.

NROW
Number of rows of array to be printed if full array is not to be printed. This allows for partial printing.
NROW = ICELL is assumed.

NFIL
Number of output files available if image is to be output in strips that are NULW lines wide.
=0 for all output on file code 06
=1 for all output on file code IFIL
>1 for outputs in strips, on file codes specified in array IFIL. NFIL=0 is assumed.

IFIL
Array containing output file codes. Ignored unless NFIL= 0. IFIL must be a variable rather than a literal, since SNAP alters the value(s) of IFIL to contain the location of the relevant file control block in the upper half of the word.

NULW
Number columns per output page. MAX =129
NULW =120 unless otherwise specified.

NULD
Number of rows to be printed before a slew on the top of the next page is given. NULD = ICELL*DMAG +1.
AMAG: Floating point magnification in width. AMAG = 1 is assumed.

DMAG: Floating point magnification in length. DMAG = 1 is assumed.

Error return if not enough output files are available. Return is made with NARG = -10. If not specified, a message will be printed out on the accounting report and an NF report will terminate execution.
ERTS TEXTURE ANALYSIS

SUBPROGRAM TITLE: FPLXIT
VERSION: II
DATE: September, 1971
UPDATE: June, 1973
AUTHOR: R. M. Haralick
DOCUMENTED BY: R. J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE:
To compute the four nearest neighbor greytone matrices LEX1, LEX2, LEX3, LEX4, for angles of 90, 0, 135, and 45 degrees.

ENTRY POINT:
CALL FPLXIT (IDATA, LEX1, LEX2, LEX3, LEX4, NUMPPL, MERGE)

ARGUMENTS:
IDATA
LEX1 - LEX4
NUMPPL
MERGE

Scratch array holding two lines of the subimage.
Address indices for the four LEX arrays.
Number of points in each line of the subimage.
TRUE to merge the four LEX arrays into one array.

COMMENTS:
This subroutine reads two lines at a time from the subimage being processed, which is now on scratch disc file 02. After all the LEX arrays are created, they can be merged into one by setting MERGE to TRUE in MAINLN. The merge is performed by adding each array term by term and putting the total into LEX1.
SUBPROGRAMS REQUIRED:
INDEX

CALLED BY:
MAING
ERTS TEXTURE ANALYSIS

SUBPROGRAM TITLE: INDEX
VERSION: 1
DATE: September, 1971
UPDATE: September, 1971
AUTHOR: R. M. Haralick
DOCUMENTED BY: R. J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635
PURPOSE: To return to the calling program the single subscript for the LEX array that indicates where element (I, L) can be found, given its row and column subscripts I and L.

ENTRY POINT: INDEX (I, J)

ARGUMENTS:

I Row subscript for an element in the LEX array.
J Column subscript for an element in the LEX array.

COMMENTS: This subprogram is a FUNCTION.

CALLED BY: FPLXIT
IMOMTR

104
ERTS TEXTURE ANALYSIS

SUBPROGRAM TITLE: IMOMTR
VERSION: II
DATE: September, 1971
UPDATE: June, 1973
AUTHOR: R. M. Haralick
DOCUMENTED BY: R. J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE:

To calculate the moment texture statistics.

ENTRY POINT:

CALL IMOMTR (LEX1, LEX2, LEX3, LEX4, F, IQ, MERGE)

ARGUMENTS:

LEX1 - LEX4 Address indices for the four LEX arrays.
F Cumulative distribution function.
IQ Quantized output array of IEPFG1,
MERGE TRUE indicates the four LEX arrays have been merged into one.

TEXTURE FEATURES:

1. Angular Second Moment

\[ \text{ANGMOM} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p^2(i, j) \]

where \( N_g \) is the number of grey tone levels, and
\( p(i, j) \) is the array of joint probabilities.
2. Entropy

\[ \text{ENTROPY} = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i, j) \log P(i, j) \]

3. Mean

\[ \text{AMEAN} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} i \cdot P(i, j) \]

4. Variance

\[ \text{SGMASQ} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \text{AMEAN})^2 P(i, j) \]

5. Covariance

\[ \text{SGMAXY} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \text{AMEAN})(j - \text{AMEAN}) P(i, j) \]

6. Correlation

\[ \text{RATIO} = \frac{\text{SGMAXY}}{\text{SGMASQ}} \]

7. Inverse Moment

\[ \text{IVDMOM} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{P(i, j)}{1+(i-j)^2} \]

8. Average Controst

\[ \text{DIFAVE} = \sum_{k=1}^{N_g-1} k \cdot \text{DIF}(k) \]

where \( \text{DIF}(k) = \sum_{|i-j|=k} P(i, j) \)
9. Variance of DIF

\[ \text{DIFVAR} = \sum_{k=1}^{N_{g-1}} k^2 \cdot \text{DIF}(k) - \left( \sum_{k=1}^{N_{g-1}} k \cdot \text{DIF}(k) \right)^2 \]

10. Entropy of DIF

\[ \text{DIFENT} = - \sum_{k=1}^{N_{g-1}} \text{DIF}(k) \cdot \log(\text{DIF}(k)) \]

11. Average of Intensity

\[ \text{SUMAVE} = \sum_{k=2}^{2N} k \cdot \text{SUM}(k) \]

where \( \text{SUM} = \sum_{i+j=k} P(i,j) \)

12. Variance of SUM

\[ \text{SUMVAR} = \sum_{k=2}^{2N} k^2 \cdot \text{SUM}(k) - \left( \sum_{k=2}^{2N} k \cdot \text{SUM}(k) \right)^2 \]

13. Entropy of SUM

\[ \text{SUMENT} = - \sum_{k=2}^{2N} \text{SUM}(k) \cdot \log(\text{SUM}(k)) \]

14. True mean of probability function

\[ \text{TMEAN} = \frac{1}{N_g} \sum_{i=1}^{N_g} F(i) \]
COMMENTS:
The three remaining texture features are computed in subroutine COR: CORINF, CORMUT, and CORMAX. If MERGE is TRUE, then these features are computed for only the merged array, LEX1. Otherwise they are computed for each LEX array, corresponding to each of four angles.

SUBPROGRAMS REQUIRED:
INDEX
IEQPQI
COR

CALLED BY:
MAING
ERTS TEXTURE ANALYSIS

SUBPROGRAM TITLE: COR
VERSION: II
DATE: November, 1972
UPDATE: June, 1973
AUTHOR: Sam Shanmugam
DOCUMENTED BY: R. J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE:
To calculate three measures of correlation between two discrete random variables X and Y whose joint probabilities of occurrence are known.

ENTRY POINT:
CALL COR(PXY, N, IOPT, Q, COR1, COR2, COR3)

INPUT ARGUMENTS:
PXY Array of joint probabilities.
N Size of the array PXY
IOPT Option flag - if IOPT = 0, then only COR1 and COR2 will be calculated; if IOPT = 1, then COR3 will also be calculated.
Q Scratch array of size N x N. This array is used only if IOPT is non-zero.

OUTPUT ARGUMENTS:
COR1 Maximal correlation measure.
COR2 Information measure of correlation.
COR3 Second type of maximal measure.
COMMENTS:

These three correlation measures are the last three texture features. For details of the measures see "Mutual Information and Maximal Correlation As Measure of Dependence," by C. B. Bell, in the Annals of Mathematical Statistics, vol. 43, 1962.

CALCULATIONS:

1. \( \text{COR1} = \frac{H(x, y) - H_1(x, y)}{\max(H(x), H(y))} \)

where \( H(x, y) = \sum_i \sum_j \log (p_{xy}^2(i, j)) \)

\( H_1(x, y) = \sum_i \sum_j \left( \log \left( p_x(i) p_y(j) \right) \right) p_{xy}(i, j) \)

\( H(x) = \sum_i \left( \log p_x(i) \right) p_x(i) \)

and \( H(y) = \sum_j \left( \log p_y(j) \right) p_y(j) \)

2. \( \text{COR2} = \sqrt{1-e^{-2R}} \)

where \( R = H_2(x, y) - H(x, y) \)

\( H_2(x, y) = \sum_i \sum_j \left( \log \left( p_x(i) p_y(j) \right) \right) p_{xy}(i, j) \)

and \( p_x(i) = \sum_i p_{xy}(i, j), p_y(j) = \sum_j p_{xy}(i, j). \)
3. COR3 is computed using the eigenvector corresponding to the second largest eigenvalue of $Q Q^T$, where

$$Q(I, J) = \frac{P_{xy}(i, j)}{\sqrt{P_x(i) P_y(j)}}$$

CALLED BY:

IMOMTR
To determine $k$ levels of equal probability quantization for an array for which the cumulative distribution function is known for all elements.

**ENTRY POINT:**

```plaintext```
CALL IEQPQ1 (N, K, F, IQ, IMIN)
```

**ARGUMENTS:**

- **N**
  - Number of items in array $F$ to be equal probability quantized.

- **K**
  - Number of quantizing levels.

- **F**
  - Input array to be quantized.

- **IQ**
  - Output array of quantized levels.

- **IMIN**
  - The lowest possible level in the input data.

**CALLED BY:**

```plaintext```
IMOMTR
```

**ERTS TEXTURE ANALYSIS**

<table>
<thead>
<tr>
<th>SUBPROGRAM TITLE:</th>
<th>RITOWT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VERSION:</td>
<td>II</td>
</tr>
<tr>
<td>DATE:</td>
<td>September, 1971</td>
</tr>
<tr>
<td>UPDATE:</td>
<td>June, 1973</td>
</tr>
<tr>
<td>AUTHOR:</td>
<td>R. M. Haralick</td>
</tr>
<tr>
<td>DOCUMENTED BY:</td>
<td>R. J. Bosley</td>
</tr>
<tr>
<td>PROGRAM LANGUAGE:</td>
<td>FORTRAN IV</td>
</tr>
<tr>
<td>IMPLEMENTED ON:</td>
<td>HW 635</td>
</tr>
<tr>
<td>PURPOSE:</td>
<td>To output onto printer, cards, or tape the texture features.</td>
</tr>
</tbody>
</table>

**ENTRY POINT:**

```
CALL RITOWT (LEX1, LEX2, LEX3, LEX4, G, IQ, MERGE, IF, PICTUR)
```

**ARGUMENTS:**

- **LEX1 - LEX4**: Address indices for the LEX arrays.
- **G**: CDF for the image data
- **IQ**: Quantized output of IEQPQ1 of NQUANT levels.
- **MERGE**: TRUE indicates that the four LEX arrays have been merged into one array.
- **IF**: File code for output tape.
- **PICTUR**: TRUE indicates that a picture of the subimage has been printed.

**COMMENTS:**

The output format for the listing is slightly different depending upon the PICTUR and merge options. The PNCH option determines whether cards or tape or neither are used to output the 17
texture features for each subimage. In any case, the texture features are listed on the printer. If PNCH = 1HY in the $PARAM card in MAINLN, then cards are punch according to the following formats:

1. for MERGE = TRUE: M1, N1, NFT, ANGMOM, ENTROP, RATIO, SGMASQ, SGMAXY, AMEAN, VIDMOM, KOUNT/TMEAN, DIFENT, DIFAVE, DIFVAR, SUMENT, SUMAVE, SUMVAR, KOUNT+1/CORINF, CORMUT, CORMAX, KOUNT+2.

   FORMAT (I2, 1X, 2I2, 1X, 7F9.5, 19/8X, 7F9.5, 19/8X, 3F9.5, 36X, 19), where (M1, N1) is the subimage row, column coordinate.


   FORMAT ('THE SCENE', 12, 1, 1, 12, 1) HAS BEEN REDUCED BY', 15/8(1X, 8F9.5, 17/)/1X, 4F9.4, 38X, 15).

If tape output on file code 'IF' is selected by PNCH = 1HT, then the texture features are written in binary as follows:

   WRITE(IF) M1, N1, NFT, ANGMOM(K), ENTROP(K), RATIO(K), SGMASQ(K), SGMAXY(K), AMEAN(K), VIDMOM(K), TMEAN(K), DIFENT(K), DIFAVE(K), DIFVAR(K), SUMENT(K), SUMAVE(K), SUMVAR(K), CORINF(K), CORMUT(K), CORMAX(K), where K is one for MERGE = TRUE and is 4 for MERGE = FALSE denoting the number of values for each measure.

   If neither cards nor tape output is selected, then the LEX arrays are listed after the texture features.

CALLED BY:

MAING
IV.1-c Cross-Band Texture Analysis Programs - Documentation

SPECTR
GETIM / GETIT
ERTS
DIFFER
COVAR
MNCVIN / MNCV
CORREL
CROSS-BAND TEXTURE ANALYSIS

PROGRAM TITLE: SPECTR
VERSION: 1
DATE: January, 1974
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTY
IMPLEMENTED ON: HW635

PURPOSE:
This program is the mainline of the spectral-textural analysis which obtain spectral-textural features for land-use classification of ERTS MSS data.

ENTRY POINT:
CALL SPECTR (IMAGE, X, ILINE, IXDIM, IYDIM, NDIN)

INPUT ARGUMENTS:
IMAGE
   Array containing a subimage.
   (IYDIM, IXDIM, IDIN)
X
   Array containing the difference image.
   (IYDIM, IXDIM, IDIN)
ILINE
   Array one ERTS line is read into.
IXDIM
   Column dimension of IMAGE, X.
IYDIM
   Row dimension of IMAGE, X.
NDIN
   Number of components, bands, in IMAGE, X.

INPUT PARAMETERS: under NAMELIST 'PARAM':
NDIM
   Number of components desired in IMAGE:
   Assumed to be NDIN.
NUMLIN
   Number of lines in subimage; Assumed to be IYDIM.
NUMPPL: Number of columns in subimage; Assumed to be IYDIM.

FMT: Format used to output elements of covariance matrix; assumed to be 'E11.4'.

TITLE: 80 column title for run.

OPT: TRUE to print covariance matrix; assumed FALSE.

IDIST: Distance between neighboring cells for difference array; assumed to be 1.

IRSTR: Starting row in ERTS image; assumed to be 1.

IRSTOP: Stopping row in ERTS image; assumed as last row.

LAPHOR: Number of horizontal points that subimages overlap; assumed to be 0.

LAPVER: Number of vertical points that subimages overlap; assumed to be 0.

PNCH: TRUE for output on cards, FALSE for output to file code 01 for tape or disc.

EXAMPLE OF DRIVER:

DIMENSION IMAGE (16, 17, 8), X (16, 17, 8), ILINE (3300)
EQUIVALENCE (IMAGE, X, ILINE(130))
IXDIM =17
IYDIM =16
NDIN =8
CALL SPECTR (IMAGE, X, ILINE, IXDIM, IYDIM, NDIN)
STOP
END

This driver will set up the spectral-textural analysis mainline SPECTR to process 16 x 16 subimages over 8 components with IDIST =1. Note IXDIM must include NUMPPL plus IDIST points, and array ILINE must have at least NUMPPL x NDIM points outside of any other array. These points form array XLINE in COVAR.
REQUIREMENTS:

1. Core - 25k for IMAGE (32, 33, 8)
2. ERTS input tape must be on file code 'ES'
3. Random access disc file on file 11, eg. $FILE II AIIR, OR.
4. Subprograms required:
   DRIVER
   SPECTR
   GETIM
   SETDIM (Fortran callable program to initialize HEMP package)
   GETIT
   ERTS
   DIFFER
   COVAR
   MNCVIN
   MNCV
   CORREL
   SFA07F
   HEMDET (Fortran callable program from HEMP library to solve for determinant of matrix).

COMMENTS:

These analysis programs obtain a series of NUMLIM by NUMPPL by NDIM subimages from the ERTS input tape and outputs a feature vector with \((1 + \text{NDIM} \times (\text{NDIM}-1)/2)\) components for each subimage. The input data is processed in horizontal rows of subimages that may overlap both horizontally and vertically. The distance between neighboring resolution cells used to get the difference array is variable. Note that IXDIM must include NUMPPL plus IDIST points, and that the array ILINE must have at least NUMPPL \times \text{NDIM} points outside of any other array because these are used for array XLINE in COVAR. Other than this, arrays IMAGE, X, and ILINE may be equivalence
to conserve core, as in the example for a DRIVER. The first feature component on the output file is the entropy measure, and the remaining \( \text{NDIM} \) \( (\text{NDIM}-1)/2 \) components are elements of the correlation matrix. See the GETIM subprogram for a listing of all 8 possible components for a subimage.

CALLED BY: DRIVER
SUBPROGRAM TITLE: GETIM
VERSION: 1
DATE: January, 1974
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTY
IMPLEMENTED ON: HW635

PURPOSE:
To get a row of subimages from the ERTS input tape and copy them onto a random access disc file.

ENTRY POINTS:
CALL GETIM (ILINE, IDIST, NDIM, IRSTRT, IRSTOP, NUMLIN, NUMPPL, LAPVER, LAPTOR, NHOR, INCR, IPEND)
CALL GETIT

INPUT ARGUMENTS:
ILINE Array into which one ERTS line is read.
IDIST Distance between neighboring resolution cells
whose differences form the difference array, X.
NDIM Number of components in each resolution cell.
IRSTRT Starting line of ERTS data.
IRSTOP Stopping line of ERTS data.
NUMLIN Number of lines in each subimage.
NUMPPL Number of points per line (columns) in each subimage.
LAPVER Number of lines that subimages overlap.
LAPTOR Number of columns that subimages overlap.
**OUTPUT ARGUMENTS:**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NHOR</td>
<td>Number of horizontal overlapping subimages in image.</td>
</tr>
<tr>
<td>INCR</td>
<td>Horizontal increment to the first column of the next subimage in the row.</td>
</tr>
<tr>
<td>IPEND</td>
<td>Last cell in the row.</td>
</tr>
</tbody>
</table>

**REQUIREMENTS:**

Random access disc file on file code II: and FILE II, AIIR, IOR.

**COMMENTS:**

This program is initialized by calling GETIM which initializes the ERTS input tape and sets up the disc on file II for random access with fixed length records of NDIM words, up to a maximum of NDIM = 8. The eight possible greytone components are:

1. Band 2
2. Band 3
3. Band 4
4. Band 2 x Band 2
5. Band 3 x Band 3
6. Band 4 x Band 4
7. Band 2 x Band 3
8. Band 2 x Band 4, where band 1 through 4 is MSS band 4 through 7.

It is suggested that all eight components be used and the best of these be selected for feature vector components. This gives a total of 29 components, including the entropy measure.

After calling GETIM, all further calls are made to GETIT which goes down the input data file line by line copying to the random access disc an entire row of NHOR subimages of NDIM components. Note that MSS band 4 (band 1 here) has been deleted because of its high correlation with MSS band 5 (band 2 here).

**CALLED BY:**

SPECTR
SUBPROGRAM TITLE: ERTS
VERSION: II
DATE: September, 1972
UPDATE: November, 1973
AUTHOR: G. Funnels
DOCUMENTED BY: R. Bosley
PROGRAM LANGUAGE: GMAP
IMPLEMENTED ON: HW635
PURPOSE: To read 7-track ERTS MSS data tapes.

ENTRY POINTS:
CALL EINIT (NOLS)
CALL ESKIP (NOSK)
CALL EREAD (I, LN)
CALL EREWND

ARGUMENTS:
NOLS Number of words per scan line; returned by EINIT.
NOSK The number of records to skip.
I The array into which the NOLS words of data
     from a line of ERTS data is placed.
LN Returned by EREAD giving the line number of the
     line of data returned. If LN = 0, the end of file
     was reached on the ERTS tape.

ERROR FLAGS:
MB EREAD buffer is not large enough for a block of
     ERTS data.
AI EINIT was called twice.
NI EINIT was not called before calling EREAD,
     ESKIP, or EREWND.
UE

EOF encountered while reading ID or annotation blocks on ERTS tape.

EF

EOF encountered while trying to skip records in ESKIP.

COMMENTS:

EINIT initializes the ERTS tape so that data may be read, and must be called first. ESKIP skips over NOSK records (scan lines). EREWND rewinds the ERTS tape. EINIT must not be called twice.

The data placed into array 1 by EREAD is in standard corresponding point forms. Since the ERTS MSS data has four channels, there are actually NDLS/4 points or cells per scan line.

REQUIREMENTS:

ERTS tape must be on file code 'ES'.

123
CROSS-BAND TEXTURE ANALYSIS

SUBPROGRAM TITLE: DIFFER
VERSION: 1
DATE: January, 1974
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTY
IMPLEMENTED ON: HW635

PURPOSE:
To calculate the nearest neighbor difference array.

ENTRY POINT:
CALL DIFFER (IA, X, IXDIM, IYDIM, NDIM, IDIST, NUMPPL, NUMLIN)

INPUT ARGUMENTS:
IA(IYDIM, IXDIM, NDIM) Array containing the subimage being processed.
IXDIM Column dimension of IA and X.
IYDIM Row dimension of IA and X.
NDIM Number of components of each resolution cell.
NUMPPL Number of columns in the subimage.
NUMLIN Number of lines in the subimage.

OUTPUT ARGUMENTS:
X(IYDIM, IXDIM, NDIM) Array of nearest neighbor differences.
COMMENTS:

This subroutine will replace the original subimage in array IA with the nearest neighbor horizontal difference: (I1 - J1, I2 - J2, ..., IN - JN) where I and J are N-dimensional horizontally neighboring resolution cells separated by distance IDIST. Arrays IA and X may be equivalenced to occupy the same area of core. Note that the absolute value is used to get the differences. This gives only the positive half of the distribution of differences I-J and J-I. This shifts the mean of the distribution from the origin and must be accounted for in COVAR when the covariance matrix of the difference array is calculated.

CALLED BY:

SPECTR
CROSS-BAND TEXTURE ANALYSIS

SUBPROGRAM TITLE: COVAR
VERSION: 1
DATE: January, 1974
UPDATE: January, 1974
AUTHOR: Robert J. Bosley
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTY
IMPLEMENTED ON: HW635

PURPOSE: To calculate the covariance matrix of the difference array.

ENTRY POINT:
CALL COVAR (XLINE, NDIM, NUMPPL, X, IXdIM, IYDIM, NUMLIN, NDIN, COV)

INPUT ARGUMENTS:
XLINE(NDIM, NUMPPL) Array used to send one line of the difference array to MNCVIN.
NDIM Number of components of each vector in X.
NUMPPL Number of columns of vectors in X.
X Nearest neighbor difference array.
IXDIM, IYDIM Column, row dimensions of array X.
NUMLIN Number of rows of vectors in X.
NDIN Dimension of COV array.

OUTPUT ARGUMENTS:
COV (NDIN, NDIN) Covariance matrix of the difference array X.

REQUIREMENTS:
Subroutine MNCVIN.
COMMENTS:

Array XLINE is formed from the first NDIM x NUMPPL points of array ILINE in SPECTR. Hence at least the first NDIM x NUMPPL words of ILINE must not be equivalenced into array X.
Since only the positive differences were used to make array X by DIFFER, the mean is reset to zero for each component.

CALLED BY:

SPECTR
CROSS-BAND TEXTURE ANALYSIS

SUBPROGRAM TITLE: MNCVIN
VERSION: 1
DATE: August, 1973
UPDATE: August, 1973
AUTHOR: James D. Young
DOCUMENTED BY: James D. Young
PROGRAM LANGUAGE: FORTRAN IV or FORTY
IMPLEMENTED ON: HW635

PURPOSE:
To calculate the mean vector and covariance matrix for each category of a set of vectors, based on a specified percentage of the vectors randomly chosen within the set.

ENTRY POINTS:
CALL MNCVIN (NVPCAL, NDIM, NCALL, PERCNT, NCAT, X, NTRUTH, COV, XMEAN, SCTMEN, SAMSZ, IERROR, JERROR)

CALL MNCV

INPUT ARGUMENTS:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVPCAL</td>
<td>Number of vectors per call.</td>
</tr>
<tr>
<td>NDIM</td>
<td>Dimension of data vectors.</td>
</tr>
<tr>
<td>NCALL</td>
<td>Number of calls.</td>
</tr>
<tr>
<td>PERCNT</td>
<td>Percentage of total number of vectors from which the mean and covariance matrices will be calculated.</td>
</tr>
<tr>
<td>NCAT</td>
<td>Number of categories considered; set to 1 if only one set of statistics will be calculated for all data, set to the number of categories in data set if one set of statistics will be calculated for each category.</td>
</tr>
</tbody>
</table>
Matrix containing input data vectors in its columns.

Vector containing the ground truth integers, 1 through NCAT, associated with the data vectors of X; if NCAT is 1 this vector is ignored.

**OUTPUT ARGUMENTS:**

- **COV (NDIM, NDIM, NCAT)**: Matrix containing covariance matrices of the data.
- **XMEAN (NDIM, NCAT)**: Matrix containing mean vectors of the data.
- **SCTMEN (NDIM, NCAT)**: Scratch matrix.
- **SAMSZ (NCAT)**: Vector with the number of vectors used to calculate the statistics for each category.
- **ERROR**: Error flag when returned non-zero:
  1. if NVPCAL .LE. 0
  2. if NDIM .LE. 0
  3. if NCAT .LE. 0
  4. if PERCNT .GE. 100. or so small that less than 2 vectors will be used to calculate all the statistics.
  5. if NCAT .LE. 0
- **JERROR**: Error flag when returned non-zero:
  1. if an illegal ground truth label is formed.

**COMMENTS:**

One call to MNCVIN initializes this routine. Calls to MNCV should be performed NCALL times, each with the next line of vectors in X. After MNCV has been called NCALL times, the mean vector and covariance matrix for each category is completed. For use in the Spectral-Textural Analysis programs, PERCNT is set to 100 and NCAT is set to 1.
CROSS-BAND TEXTURE ANALYSIS

SUBPROGRAM TITLE: CORREL

VERSION: 1

DATE: January, 1974

UPDATE: January, 1974

AUTHOR: Robert J. Bosley

DOCUMENTED BY: Robert J. Bosley

PROGRAM LANGUAGE: FORTY

IMPLEMENTED ON: HW 635

PURPOSE: To calculate the correlation matrix given the covariance matrix of the difference array.

ENTRY POINT:

CALL CORREL (COV, NDIM, COR)

INPUT ARGUMENTS:

COV Covariance matrix of the difference array
NDIM Order of matrix COV.

OUTPUT ARGUMENTS:

COR Correlation matrix of COV.

CALLED BY:

SPECTR
IV.1-d Piecewise Linear Classification Programs - Documentation

RCLASS
XIN
LINEAR
WEIGHT
LINEAR DISCRIMINANT CLASSIFIER

SUBPROGRAM TITLE: RCLASS
VERSION: 1
DATE: September, 1972
UPDATE: November, 1972
AUTHOR: Sam Shanmugam
DOCUMENTED BY: Sam Shanmugam
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE:
This subroutine is the mainline for routines which implement a decision rule using piecewise linear discriminant functions. It calls subroutine XIN to read in the pattern vectors, and calls LINEAR to get the decision rule and classify the pattern vectors.

ENTRY POINT:
CALL RCLASS (WORK, ISIZE)

ARGUMENTS:
WORK Scratch array of size ISIZE which holds the training vectors and the weight vectors.
ISIZE Dimension of array WORK.

INPUT PARAMETERS:
Card 1. Format (411)
NOPT1 Set to 1 to print out training patterns; otherwise set to zero.
NOPT2 Set to 1 to print out test patterns; otherwise set to zero.
NOPT3 Set to 0 to list only the contingency table for the training patterns; otherwise set to 1 and the classification of each training pattern is listed as well as the contingency table.
NOPT4  Set to 0 for only the contingency table of the test set; otherwise set to 1 for the classification of each test pattern as well.

Card 2. Format (515)

NTOT  The total number of pattern vectors in the data set.

NPART  NPART out of every ten pattern vectors in the data set will be used for training. The remaining will be used as test patterns.

NDIM  The number of measurements per vector plus two.

NC  The number of ground truth categories.

NPAIR  Twice the maximum number of training patterns in any one category.

REQUIREMENTS:

1. Maximum number of categories is 15.
2. Maximum number of components, NDIM, is 100.
3. Pattern vectors must be sorted by category.
4. A scratch disc file must be on 02.
5. ISIZE must be at least
   \[ \text{ISIZE} \geq \text{NDIM} (\text{NTRAIN} + 10) + 1000 \\
   + (\text{NC} (\text{NC} + 1)/2) \text{ND} + \text{NPAIR} \times \text{ND} \]
   where \( \text{ND} = \text{NDIM} - 1 \)
   and \( \text{NTRAIN} \) = number of training patterns.
6. Pattern vectors must be written in binary to disc file 01 as follows:
   \[ \text{WRITE (01) IG}T, M1, N1, NFT, (\text{FEAT}(I), I = 1, \text{NMEAS}) \]
   where
   \( \text{IGT} \) is the ground truth category.
   \( M1, N1, NFT \) are not used - may be used as ID tags
   \( \text{FEAT} \) is the feature vector
   \( \text{NMEAS} \) is the number of measurements per feature vector.
7. Subprograms required
   DRIVER
   RCLASS
   XIN
   LINEAR
   WEIGHT

ERROR FLAGS:

If ISIZE is too small, processing is terminated and an error message is listed.

THEORY:

Using a regression type algorithm the program obtains a set of hyperplanes for separating the training patterns of different category pairs. A total of NC(NC-1)/2 hyperplanes are determined. Test patterns are identified by taking a majority vote on this set of hyperplanes. For complete details, see "Introduction to Statistical Pattern Recognition" by Y. Fukunaga, Academic Press, 1972.

COMMENTS:

The input data file on 01 should have a total of NTOT logical records in binary. Each logical record must be of length NDIM + 2 words, where word 1 is the ground truth category and words set is sorted into training and test sets according to NPART. Training vectors are stored in WORK and test vectors are copied to disc 02.
CALLED BY: ~ DRIVER

EXAMPLE OF DRIVER:

DIMENSION WORK (10000)
ISIZE = 10000
CALL RCLASS (WORK, ISIZE)
STOP
END
LINEAR DISCRIMINANT CLASSIFIER

SUBPROGRAM TITLE: XIN
VERSION: II
DATE: September, 1972
UPDATE: December, 1973
AUTHOR: Sam Shanmugam
DOCUMENTED BY: Robert J. Bosley
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE: To read the parameter cards and the Input data set, copying the test vectors to disc.

ENTRY POINT:

CALL XIN (WORK, U)

ARGUMENTS:

WORK Array training vectors are read into
U Scratch array

CALLED BY: RCLASS

COMMENTS: This version does not use subroutine POST to position the input file 01. Hence the input data file must be the first file on 01.
LINEAR DISCRIMINANT CLASSIFIER

SUBPROGRAM TITLE: LINEAR
VERSION: I
DATE: September, 1972
UPDATE: November, 1972
AUTHOR: Sam Shanmugam
DOCUMENTED BY: Sam Shanmugam
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635

PURPOSE: Using the set of training patterns, this program obtains a set of hyperplanes for pairwise separation of training patterns of different categories. The program also identifies the test patterns on a majority vote on the hyperplanes and outputs contingency table.

ENTRY POINT:

CALL LINEAR (XTRAIN, XTEST, W, U, DUMMY)

ARGUMENTS:

XTRAIN Matrix containing training patterns.
XTEST Matrix containing test patterns.
W Array of weight vectors.
U Matrix used for calculating the boundary between category pairs.
DUMMY Scratch array.

CALLED BY:

RCLASS

SUBPROGRAMS REQUIRED:

WEIGHT
LINEAR DISCRIMINANT CLASSIFIER

SUBPROGRAM TITLE: WEIGHT
VERSION: I
DATE: September, 1972
UPDATE: November, 1972
AUTHOR: Sam Shanmugam
DOCUMENTED BY: Sam Shanmugam
PROGRAM LANGUAGE: FORTRAN IV
IMPLEMENTED ON: HW 635
PURPOSE: To find the minimum mean square fit hyperplane for separating the training patterns of two different categories.

ENTRY POINT:
CALL WEIGHT (U, DUMMY, WT, ND, NIJ)

INPUT ARGUMENTS:
U Array containing the patterns of category 1 and category J.
DUMMY Scratch array.
ND ND+1 Is the dimension of the pattern vector.
NIJ Number of vectors in U.

OUTPUT ARGUMENTS:
WT Weight vector which defines the hyperplane separating categories 1 and J.

CALLED BY:
LINEAR

SUBPROGRAMS REQUIRED:
MINV Matrix inversion program from the IBM Scientific Subroutine Package.
IV.2-a  ERTS Retrieval Program Listings

RETV
ERTS
PIXEY
ZEQUAN
PITCHR
WRTDSK
RDDSK1
KEQUAN
RDDSK2
DESCRIPTION OF PROGRAM.

THIS IS THE MAINLINE FOR ERTS RETRIEVAL PROGRAMS WHICH WILL
RETRIEVE DATA FROM STANDARD NASA ERTS DATA TAPES AND OUTPUT THE
IMAGE DATA IN A PICTURE, OR GREYTONE LIST NG, OR COPY IT ONTO AN
OUTPUT TAPE ON FILE CODE "IF.~1". COORDINATES MAY BE EITHER IN
MILLIMETERS OR IN ROW, COLUMN COUNT. FOR REASONABLE EFFICIENCY
DATA IS READ IN BLOCKS OF 41 LINES BY 41 POINTS. ONE ERTS TAPE
WILL THEN BE COVERED BY 27 HORIZONTAL BLOCKS, LEAVING 4 POINTS
LEFT AT THE END OF EACH LINE. NOTE—SOMETIMES ON THE FIRST TAPE
OF AN IMAGE, THE FIRST FOUR POINTS ARE 255 AND WILL ADVERSELY
AFFECT A PICTURE PRINTOUT. IF SO, SKIP THE FIRST FOUR POINTS BY
SETTING ICSTRT=5.

NOTE—IF THE POINT ICSTRT IS NOT A MULTIPLE OF 41 FROM THE
END OF THE LINE, THEN THE LAST POINTS MAY NOT NECESSARILY BE
OUTPUTTED SINCE THE BLOCKS ARE DETERMINED BY THE ICSTRT POSITION.
THEORETICALLY IT IS SUGGESTED THAT IF THE DATA OUT TO THE END OF EACH
LINE IS DESIRED, THAT THE COORDINATES BE SPECIFIED IN ROW,COLUMN
FORMAT AND THAT THEY BE PREDETERMINED SO AS TO OUTPUT THE DESIRED
POINTS IN THE ERTS LINE.

A SPECIAL PICTURE RUN TO PRINT OUT AND REDUCE ANY IMAGE OVER
THE ENTIRE TAPE CAN BE MADE BY SETTING SPIC TO TRUE, IN SPARAM
AND THEN SPECIFYING PICTURE PARAMETERS UNDER NAMELIST SPICT.

SEE THE PIXEY PROGRAM FOR AN EXAMPLE OF A SAMPLE RUN.

INPUT PARAMETERS UNDER NAMELIST SPARAM.

NBAND
BAND NUMBER TO BE SELECTED
 RETV0029
set=5 FOR ALL BANDS  RETV0030

IRSTRT,ICSTRT
ROW,COLUMN STARTING COORDINATES
 RETV0031
IRSTOP,ICSTOP
ROW,COLUMN STOPPING COORDINATES
 RETV0032

NHOR
NUMBER OF HORIZONTAL 41 X 41 BLOCKS
 RETV0033

NVERT
NUMBER OF VERTICAL BLOCKS COVERING
 RETV0034

A AREA SELECTED
 RETV0035

IPSTR,IPEND
STARTING,ENDING POINTS IN ERTS LINE
 RETV0036

TITLE
TITLE FOR THIS RUN
 RETV0037

IFIL
OUTPUT FILE FOR TAPE, SET TO 02
 RETV0038

LINE
ARRAY WHERE ERTS LINE IS READ INTO
 RETV0039

IMAGE
ARRAY TO STORE 41 X 41 SUM IMAGE BLOCK
 RETV0040

PIC
TRUE FOR PICTURE OUTPUT
 RETV0041

SPIC
TRUE FOR SPECIAL PICTURE RUN
 RETV0042

QUAN
TRUE FOR QUANTIZED PICTURE OUTPUT
 RETV0043

PPNT
TRUE FOR GREYTONE LISTING
 RETV0044

TAPE
TRUE FOR TAPE OUTPUT
 RETV0045

WILL
TRUE IF COORD ARE SPECIFIED IN MM.
 RETV0046

SMALL
TRUE IF MM. COORD COME FROM A SMALL
 RETV0047

70 X 70 NEGATIVE—OTHERWISE, SE ASSUME
 RETV0048
TO BE FALSE -- FROM A 7 X 7 INCH PRINT

SUBPROGRAMS REQUIRED.

RETV
ERTS (WITH EREWNO)
PIXKEY
ZEQUAN
ERTS
PITCHR
WRD5K
R5D5K1
KEDUAN
PITCHR
R5D5K2

DIMENSION IMAGE(41,82),ILINE(3301),TITLE(14)
EQUIVALENCE (IMAGE(1,1),ILINE(1))
LOGICAL EOF,PIC,SPIC,QUAN,PRNT,TAPE,SMALL, MILLI
DATA BLANK/*
NAMELIST /PARAM/NBAND,PI,C,QUAN,PRNT,TAPE,IFIRST,IRSTOP,
ICSTOP,IFIL,TITLE,MILLI,SPIC

****** SECTION I --- SET UP PARAMETERS FOR PROCESSING ******

CALL EINIT(LENGTH)
INITIALIZE ERTS AND EOFFLAG
CALL FGEOF(05,EOF)

1001 PIC=.TRUE.
SPIC=.FALSE.
QUAN=.TRUE.
PRNT=.FALSE.
TAPE=.FALSE.
SMALL=.FALSE.
MILLI=.FALSE.
DO 7 I=1,14
10 TITLE(I)=BLANK
7 IFIL=2
NBAND=2
IRSTOP=1

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
READ(05,PARAH)
C IF(EOF) STOP
C WRITE(6,1) (TITL(I),I=1,14),NBAND
IF(NBAND.EQ.5) WRITE(6,2)
IF(TAPE) WRITE(6,3) IF.
IF(PIC) WRITE(6,4) QUAN
WRITE(6,5) IRSRT,ISTRT,ICSTRT,ICSTOP
1 FORMAT(20X,'ERTS RETRIEVAL PROGRAM VERSION II'///IX,14A6///) RETV0114
1 ' BAND NUMBER IS ',NBAND)
2 FORMAT(' ALL FOUR ERTS BANDS WILL BE PROCESSED')
3 FORMAT(' NEW OUTPUT TAPE WILL BE CREATED ON FILE ') I3)
4 FORMAT(' PICTURE QUANTIZATION IS ',.LI)
5 FORMAT(' STARTING ROW IS ',86.10X,'ENDING ROW IS ',86/.
5 ' STARTING COLUMN IS ',86.10X,'ENDING COLUMN IS ',6)
C FORMAT(' COORD IN MILLI-METERS')
C IF(.NOT.SMALL) GO TO 10
C WRITE(6,6)
C IF(SMALL) GO TO 10
C IRSTRT=IPSTRT*2336/180-12
10 IRSTOP=IPSTOP*2336/180
1 IRSRT=ICSTRT*824/46-17
11 ICSTOP=ICSTOP*824/46
12 GO TO 15
C IRSTRT=IPSTRT*2336/70-31
13 IRSTOP=IPSTOP*2336/70
14 IRSRT=ICSTRT*824*2/35-46
15 ICSTOP=ICSTOP*824*2/35
16 GO TO 15
C WRITE(6,16) IRSRT,ISTRT,ICSTRT,ICSTOP
16 FORMAT(' COORDINATES IN ROW AND COLUMN FORMAT ARE---'///
17 ' STARTING ROW IS ',16.10X,'ENDING ROW IS ',16/.
17 ' STARTING COLUMN IS ',16.10X,'ENDING COLUMN IS ',16)
18 FORMAT(' DO SMALL NEGATIVE COORDINATES')
C A NEGATIVE IS 70MM BY 70MM
C WRITE(6,17) IRSRT,ISTRT,ICSTRT,ICSTOP
17 FORMAT(' WRITE THE NEW COORDINATES')
18 FORMAT(' CHECK COORDINATES SO THEY FALL IN ONE TAPE')
19  IF (ICSTOP .GT. 24) GO TO 20  
       CHECK PARAMETERS  
       IF (NBAND .GT. 5) GO TO 20  
       IF (NBAND .LT. 1) GO TO 20  
       IF (IRSTRT .LT. 1) GO TO 20  
       IF (ICSTOP .LT. ICSTRFT) GO TO 20  
       IF (ICSTOP .GT. ICSTRFT) GO TO 22  
20  WRITE (6, 21)  
21  FORMAT (/// /// **EXECUTION TERMINATED FOR THIS RUN--ERROR IN PARAMETERS**  
       //1EPS*:1-1)  
       GO TO 1001  
       CHECK FOR SPECIAL PICTURE RUN  
       IF (NOT. SPIC) GO TO 18  
       CALL PIKEY (INLINE,ILINE,IRSTRT,IRSTOP,ICSTRFT,ICSTOP,QUAN,NBAND)  
       GO TO 1001  
       POSITION THE INPUT TAPE  
       NOSK = IRSTPT - 1  
       CALL EREWNO  
       IF (NOSK .NE. 0) CALL ESKIP (NOSK)  
       WRITE (6, 23) LENGTH  
23  FORMAT (// LENGTH OF ONE ERTS LINE IS \$I\$, \$I\$)  
       IF (LENGTH .LE. 3300) GO TO 29  
       WRITE (6, 24)  
24  FORMAT (/// LENGTH OF LINE ON ERTS EXCEEDS DIMENSION OF ARRAY ILN)  
       IF (-- EXECUTION TERMINATED)  
       STOP  
       DETERMINE THE NUMBER OF BLOCKS  
       ALLOW 15 PTS TO BE CUT OFF BEFORE  
       STARTING A NEW STRIP OF BLOCKS  
       NHOR = ((ICSTOP - ICSTRFT - 15) / 41) + 1  
       RESET ENDING COLUMN  
       ICSTOP = NHOR * 41 + ICSTRFT - 1  
       IF (ICSTOP .LE. 824) GO TO 26  
       N=OR = NHOR - 1  
       GO TO 27  
25  WRITE (6, 25) NHOR, NVERT, ICSTOP  
       SET STARTING AND ENDING PTS IN INLINE  
       ISPRT = (ICSTRFT - 1) * 4  
26  WRITE (6, 26) NVERT, ISPRT  
       WRITE OUT NO OF BLOCKS AND STOPS  
       DATEW (6, 28) NHOR, NVERT, ICSTOP  
27  FORMAT (/// NUMBER OF HORIZONTAL BLOCKS \$5, \$8/  
       1 * NUMBER OF VERTICAL BLOCKS \$5, \$8/  
       1 * REVISED STOPPING ROW IS NOW \$16/  
       1 * REVISED STOPPING COLUMN IS NOW \$16)  
       ***** SECTION II --- PROCESS THE ERTS DATA TAPE *****

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR.
PART 1--READ THE DATA
GO THRU EACH VERTICAL BLOCK

IF(LN,NE,9) GO TO 90
WRITE(6,80)LN
FORMAT('///' ***EXECUTION TERMINATED---EOF DETECTED ON ERTS TAPE*)
GO TO 301
WRITE NHOK RECORDS ON DISC, ONE DISC PER BAND
CALL WRDISK(ILINE,NHOR,IPSTR,PEND,NBAND)

PART 2--OUTPUT THE DATA
SET UP NO OF TIMES AND DISC FILE CODE
NTIMES=1
NOSK=10*NBAND-1
IF(NBAND,NE,5) GO TO 250
NTIMES=4
NOSK=10
GO THRU ONE TIME FOR EACH BAND
DO 300 ITIME=1,NTIMES
NOSK=NOSK+1
IF(PIC) CALL RDISK1(IMAGE,QUAN,NHOR,NVERT,NOSK)
WRITE OUT THE GREY-TONES
CALL RDISK2(IMAGE,IRSTR,IRSTOP,NHOR,NOSK,PRNT,TAPE,IFIL)
CONTINUE
GO BACK FOR ANOTHER RUN

IF(TAPE) END FILE IFIL
WRITE(6,302)
FORMAT('///' END OF THIS RUN'/1M1)
GO TO 1001
END

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
CALL EINIT(NOLS)
EINIT initializes the ERTS tape so that data may be read. It returns the number of words per scan line in the variable NOLS.

TO SKIP A NUMBER OF RECORDS
CALL ESKIP(NOSK)
NOSK is the number of records to skip.

IF THE END OF FILE IS ENCOUNTERED BEFORE NOSK RECORDS ARE SKIPPED, EREAD IS ABORTED.

TO READ A LINE
CALL EREAD(IN, LN)

THIS IS THE ARRAY INTO WHICH THE NOLS WORDS OF DATA FROM A LINE OF ERTS DATA IS PLACED. THE DATA IS PLACED INTO THIS ARRAY IN STANDARD CORRESPONDING POINT FORM.

THE ERTS DATA HAS FOUR CHANNELS, SO THERE ARE ACTUALLY NOLS/4 DATA POINTS PER SCAN LINE.

LN THIS IS RETURNED BY EREAD. IF LN IS RETURNED AS ZERO, THE END OF FILE HAS BEEN REACHED ON THE ERTS TAPE. IF IT IS RETURNED NON-ZERO, THEN IT IS THE LINE NUMBER OF THE LINE OF DATA RETURNED.

NOTE-- THE ERTS TAPE MUST HAVE FILE CODE "ES" FOR THIS PROGRAM.

ABORT CODES POSSIBLE FROM THIS SUBROUTINE ARE--
MB EREAD BUFFER "DATA" IS NOT LARGE ENOUGH FOR A BLOCK OF ERTS DATA. IT MUST BE INCREASED IN SIZE.
AI EINIT WAS CALLED TWICE.
NI EINIT WAS NOT CALLED BEFORE CALLING EREAD OR ESKIP.
UE END OF FILE ENCOUNTERED WHILE READING.
ID OR ANNOTATION BLOCKS ON ERTS TAPE.
EF END OF FILE ENCOUNTERED WHILE TRYING TO SKIP RECORDS ON ERTS TAPE IN ESKIP.
ERTS READING PROGRAM-READ,ESKIP,EINIT

STORAGE

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>EINIT</td>
<td>0</td>
</tr>
<tr>
<td>54</td>
<td>RECNO.</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>FDW</td>
<td>18/FCB,1/0,1/1,2/0,1/0,1/0,1/1</td>
</tr>
<tr>
<td>56</td>
<td>FILC8</td>
<td>FCB,ES,...,1024,...,1,...,1</td>
</tr>
<tr>
<td>57</td>
<td>CNT1</td>
<td>0</td>
</tr>
<tr>
<td>58</td>
<td>CNT2</td>
<td>0</td>
</tr>
<tr>
<td>59</td>
<td>CNT3</td>
<td>0</td>
</tr>
<tr>
<td>60</td>
<td>EOFA</td>
<td>0</td>
</tr>
<tr>
<td>61</td>
<td>MME</td>
<td>0</td>
</tr>
<tr>
<td>62</td>
<td>DATA</td>
<td>0</td>
</tr>
<tr>
<td>63</td>
<td>OT</td>
<td>0</td>
</tr>
<tr>
<td>64</td>
<td>IT</td>
<td>0</td>
</tr>
<tr>
<td>65</td>
<td>STS</td>
<td>0</td>
</tr>
<tr>
<td>66</td>
<td>ST</td>
<td>0</td>
</tr>
<tr>
<td>67</td>
<td>STK</td>
<td>0</td>
</tr>
</tbody>
</table>

DATA BUFFER AREA DATA,9 DATA,139 DATA,**

DOw FOR READING ID DOw FOR READING ANNOTATION BLOCK DOw FOR READING DATA BLOCK

OUTPUT TALLY INPUT TALLY

STACK TALLY STACK TALLY STACK AREA

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
PAGE 5

ERTS READING PROGRAM-READ,ESKIP,EINIT

EINIT

69 EINIT SAVE

70 Szn .EINIT HAVE WE BEEN CALLED BEFORE
71 TNZ $ABA YES
72 STC1 .EINIT MARK THAT WE'VE BEEN HERE ONCE
73 EAXO 2,1*
74 STXO EIP SAVE ADDRESS
75 CALL OPEN(FDW,1) OPEN FILE

76 CALL READ(FCB,CNT1) READ ID RECORD

77 CALL WAIT(FCB,EOF) AND WAIT FOR IT TO GET DONE

78 LDQ DATA+3 GET RECORD LENGTH
79 ANQ =0177777,DL ISOLATE IT
80 EIP STO ** GIVE IT TO CALLER
81 QLS 6+18
82 STCQ ADDCNT,70 SAVE TALLY COUNT
83 ORL 6+18-1 MULTIPLY BY 2
84 DIV 9,DL AND DIVIDE BY 9
85 ARL 0 IS REMAINDER ZERO
86 TZE *+2 YES
87 ADD 1,DL NO, SO INCREMENT QUOTIENT
88 CMPQ 1025,DL IS IT TOO BIG
89 TNC *+3 NO
90 LDQ =3H0MB,DL YES, SO ABORT
91 MNE GE80RT
92 STCQ CONT,07 SAVE IT IN DW
93 MNE GESNAP
94 ZERO DATA,9 SNAP OUT ID BLOCK
95 CALL READ(FCB,CNT2) READ ANNOTATION BLOCK

96 CALL WAIT(FCB,EOF) WAIT ON IT

147
ERTS READING PROGRAM-ERead,ESkIp,EInit

EInit

97  MHE  GESnap  SNAP OUT ANNOTATION BLOCK
98  ZERO  DATA,139  AND RETURN
99  RETURN  EInit
100  ESkiP  SAVE

101  SZN  EInit  ARE WE INITIALIZED
102  TZE  EABT3  NO
103  L0Q  2,1*  GET NUMBER RECORDS TO SKIP
104  EComp  TZE  ESKR  NONE TO 00, SO RETURN
105  CMPQ  64,DL  IS IT > 63
106  TRC  ESK64  YES
107  ASQ  RECNO.  INCREMENT RECORD COUNT
108  EAX1  0,DL
109  STX1  +5
110  CALL  FSREC(FGB,.ESkIp,.Eofs)

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

111  ESkiP  RETURN  ESkiP  NO. OF RECORDS TO SKIP
112  .ESkiP  3SS  1
113  Eofs  LDQ  =3HoFF,DL  UNEXPECTED EOF
114  NMe  GEBORT
115  * TRYING TO SKIP 64 OR MORE RECORDS
116  ESK64  LDA  63,DL  INCREMENT RECORD COUNT
117  ASa  RECNO.
118  SPO  63,DL
119  STQ  .ESkiP  SAVE FOR LATER
120  CALL  FSREC(FGB,63,.Eofs)

121  LDQ  .ESkiP  TRIED TO CALL EInit TWICE
122  TPa  ECmp
123  *  EABTA  LDQ  =3HOAI,DL
124  NMe  GEBORT  DIDN'T CALL EInit
125  EABTB  LDQ  =3HOHi,DL

148
EREAD
129 EREAD SAVE

130 SZN .EINIT ARE WE INITIALIZED
131 TZE INIT3 NO
132 EAQ 2,1* GET ARRAY ADDR.
133 ADDCNT ORQ **,DL ADD TALLY COUNT
134 STQ OT AND SAVE IT
135 EAQ DATA SET UP INPUT TALLY
136 STQ IT INSERT ADDRESS OF COUNT NO
137 EAQX 3,1* ERTS0129
138 STX0 ERP ERTS0129
139 CALL READ(FGB,CONT) READ NEXT LINE OF DATA

140 CALL WAIT(FGB,EOF) WAIT ON IT

141 AQS RECN0. INCREMENT LINE NO.
142 LDQ RECN0. GET LINE NO.
143 ERP STQ ** RETURN TO CALLER
144 STZ LINIT INITIALIZE EXPANDER
145 NEXT LDQ STS
146 STQ ST
147 NXTONE TSX1 L
148 STA ST,ID PUT INTO STACK TALLY
149 TTF *+2 GET POINT
150 DRL ERROR AND PUT IT IN ARRAY
151 TSX1 L GET NEXT POINT
152 STA ST,ID PUT INTO STACK TALLY
153 TTF NXTONE GO PROCESS NEXT
154 LDQ STS TALLY RUNOUT
155 STQ ST SO REINITIALIZE TALLY
156 AGAIN LDQ ST,ID PICK UP POINTS
157 TTF MORE GOT ONE
158 STQ ST,ID TALLY RUNOUT
159 TTF NEXT GO PROCESS NEXT EIGHT POINTS
160 RETURN EREAD TALLY RUNOUT, SO WE'RE DONE.
161 MORE STQ ST,ID SAVE IT
162 TTF AGAIN GO GET NEXT ONE STACKED
163 DRL ERROR ERTS0156
<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>165</td>
<td>LSAVE 9SS 1</td>
<td>OR saved from last TSX1 L</td>
</tr>
<tr>
<td>166</td>
<td>LIMIT 9SS 1</td>
<td>Shift indicator for L</td>
</tr>
<tr>
<td>167</td>
<td>L</td>
<td>Are we ready for next pair</td>
</tr>
<tr>
<td>168</td>
<td>L</td>
<td>YES</td>
</tr>
<tr>
<td>169</td>
<td>SZN LIMIT</td>
<td>NO, so get next word</td>
</tr>
<tr>
<td>170</td>
<td>LDQ IT,IO</td>
<td>Increment limit</td>
</tr>
<tr>
<td>171</td>
<td>AOS LIMIT</td>
<td>DONE for this one</td>
</tr>
<tr>
<td>172</td>
<td>LINIT</td>
<td>Increment again.</td>
</tr>
<tr>
<td>173</td>
<td>LCONT AOS</td>
<td>IS this 5th time</td>
</tr>
<tr>
<td>174</td>
<td>LDQ LIMIT</td>
<td>YES</td>
</tr>
<tr>
<td>175</td>
<td>CMPQ 5,DL</td>
<td>OR 9th time</td>
</tr>
<tr>
<td>176</td>
<td>TZE L5</td>
<td>NO</td>
</tr>
<tr>
<td>177</td>
<td>CMPQ 9,DL</td>
<td>REinitialize</td>
</tr>
<tr>
<td>178</td>
<td>TNZ +2</td>
<td>Get saved QR</td>
</tr>
<tr>
<td>179</td>
<td>STZ LIMIT</td>
<td>Shift over</td>
</tr>
<tr>
<td>180</td>
<td>LDQ LSAVE</td>
<td>Save QR for next time</td>
</tr>
<tr>
<td>181</td>
<td>LRET LDA 0,DL</td>
<td>AND return</td>
</tr>
<tr>
<td>182</td>
<td>LLS 8</td>
<td>Restore QR</td>
</tr>
<tr>
<td>183</td>
<td>STQ LSAVE</td>
<td>THIS one is divided over</td>
</tr>
<tr>
<td>184</td>
<td>TRA 0,1</td>
<td>WORD BOUNDARIES</td>
</tr>
<tr>
<td>185</td>
<td>L5 LDA 0,DL</td>
<td>RETURN</td>
</tr>
<tr>
<td>186</td>
<td>LDQ LSAVE</td>
<td>MARK EOF TO CALLER</td>
</tr>
<tr>
<td>187</td>
<td>LLS 4</td>
<td>AND RETURN</td>
</tr>
<tr>
<td>188</td>
<td>LDQ IT,IO</td>
<td>EOF processor for EREAD</td>
</tr>
<tr>
<td>189</td>
<td>LLS 4</td>
<td>EOF</td>
</tr>
<tr>
<td>190</td>
<td>STQ LSAVE</td>
<td>EOF</td>
</tr>
<tr>
<td>191</td>
<td>TRA 0,1</td>
<td>RETURN</td>
</tr>
<tr>
<td>192</td>
<td>E</td>
<td>END</td>
</tr>
</tbody>
</table>

The above assembly is designed to handle EOF processing. In the assembly, the process begins by saving the limit (L) and checking if it's ready for the next pair (SZN). If yes, it increments the limit (IT,IO) and checks for the 5th time (CMPQ 5,DL). If this is the 5th time, it reinitializes (STZ LIMIT). If not, it shifts over (TZE L5). The process continues by checking if it's the 9th time (CMPQ 9,DL) and if not, it returns the saved QR (Tnz +2). If yes, it gets the saved QR (LDQ LSAVE). The process then checks if it's the 5th time again (STQ LSAVE) and increments if necessary (TRA 0,1). If it's the 5th time, it restores the QR (LDA 0,DL) and checks if it's the 9th time (LDS 4). If yes, it returns the saved QR (STQ LSAVE) and increments again (TRA 0,1). The process repeats until EOF is reached (LDQ IT,IO).
SPECIAL PICTURE ROUTINE

CPIXEO    SPECIAL PICTURE ROUTINE

WRITTEN BY RJ DOSLEY      JAN 1974

DESCRIPTION OF PROGRAM:

THIS SUBPROGRAM OUTPUTS A PICTURE OF THE ERTS DATA FROM USER
SPECIFIED PARAMETERS TO PITCHR. INPUT PARAMETERS FOR PITCHR PLUS
THE PARAMETERS NNSKIP AND KOLSKP ALL MUST BE SUPPLIED TO THIS
SUBPROGRAM UNDER THE NAMESPACE PICTUR FORMAT. FOR A COMPLETE
DESCRIPTION OF INPUT ARGUMENTS TO PITCHR, SEE THE PITCHR WRITEUP.

BY THE USER SETTING IMAX=0 AND IMIN=M MAX GREY TONE LEVEL, IT IS
POSSIBLE TO OBTAIN A POSITIVE IMAGE INSTEAD OF A NEGATIVE.

NOTE--DISC FILES MUST BE ON FILE CODES 13 AND 14.

INTERNAL PARAMETERS.

IBAND    BAND NUMBER BEING PROCESSED
IS,IE    STARTING, ENDING BAND NUMBERS
LINES    NUMBER OF LINES IN IMAGE PRINTED
NNSK    NUMBER OF LINES TO SKIP IN ERTS TAPE
INSTR,IPEND    STARTING, ENDING POINTS IN ERTS LINE
LNSKIP,KOLSKP    LINE AND COLUMN INCREMENT
ISKIP    COLUMN INCREMENT FOR ERTS LINE
KP    NUMBER OF COLUMNS IN IMAGE

INPUT ARGUMENTS TO PITCHR.

IMAGE  ARRAY TO BE PRINTED
ICELL, JCELL  NUMBER OF TIMES ENTRY IS MADE AT SNAP
INIT  MUST BE GREATER THAN 1
IMIN,IMAX  MINIMUM, MAXIMUM GREY TONES IN ARRAY
NROW  NUMBER OF ROWS TO BE PRINTED=ICELL
NFIL  NUMBER OF OUTPUT FILES AVAILABLE
NUM  SET = 0 FOR ALL OUTPUT ON FILE CODE 6
      SET = 2 FOR FILES 06 AND 42
      IF .GT. 2, USER MUST SUPPLY FILES
IFIL    ARRAY CONTAINING OUTPUT FILE CODES
NULW,NULD  NUMBER OF COLUMNS, ROWS PER OUTPUT PAGE
AMAG,DMAG  WIDTH, LENGTH JOHN MAGNIFICATION

ENTRY POINT.

CALL PXEY(IUINE, IMAGE, ISTRRT, ISTOP, ICSTRT, ICSTOP, QUAN, N0BAND)

INPUT ARGUMENTS.

ILINE  ARRAY ERTS LINE IS READ INTO
IMAGE  IMAGE ARRAY FOR PITCHR, HOLDS ONE LINE
ISTRRT, ISTOP  STARTING, STOPING ROW IN IMAGE
ICSTRT, ICSTOP  STARTING, STOPING COL IN IMAGE
QUAN  TRUE FOR EQUAL PROBABILITY QUANTIZING
N0BAND  BAND NUMBER TO BE PROCESSED
SPECIAL PICTURE ROUTINE

SET=5 FOR ALL BANDS

EXAMPLE OF SAMPLE RUN.

CONTROL CARDS---

PARAM SPIC=T,QUAN=T,IRSTRT=1,IRSTOP=1216,ICSTRT=5,
ICSTOP=777,NBAND=23END

SPICRUN INIT=304,NFILES=2,LNSKIP=4,KOLSKP=3,JCELL=256END

THIS RUN WILL PRINT OUT 44 FILES 06 AND 42 A PICTURE 256 PTS
WIDE BY 304 LINES. NOTE THAT 1216/4=304 AND 765/3=256, GIVING THE
VALUES FOR LNSKIP=4,KOLSKP=3,JCELL=256,INIT=304. ALSO, USING
THESE VALUES FOR LNSKIP AND KOLSKP, THE PICTURE WILL BE IN
PROPORTION TO THE EDTS PRINT.

SUBROUTINE PIXEY(IIMAGE,IRSTRT,IPSTOP,ICSTRT,ICSTOP,QUAN,
1
N3BAND)

DIMENSION ILINE(1),IFIL(10),IMAGE(1)

LOGICAL QUAN

NAMELIST /PICTURE/NFILES,IFIL,JCELL,JCELL,NINIT,NMIN,NMAX,NROW,
1
NULW,NULD,AMAG,DMAG,LNSKIP,KOLSKP

INITIALIZE PARAMETERS

ICELL=1
JCELL=256
NINIT=304
INMAX=0
INMAX=75
IF(QUAN)INMAX=12
NFILES=2
IFIL(1)=6
IFIL(2)=42
NROW=JCELL
NULW=129
NULD=60
AMAG=1
DMAG=1
LNSKIP=4
KOLSKP=3

READ PARAMETERS

WRITE(I6,1)
WRITE(I6,PICTURE)

FORMAT(///20X,'SPECIAL PICTURE ROUTINE'///)

FORMAT(1H1,'BAND NUMBER'1I)

SET UP FOR THE DESIRED BAND

IS=1
IF=4
LNSKIP=4*KOLSKP
IF(ISUBAND.EQ.5) GO TO 5

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
SPECIAL PICTURE ROUTINE

IC=NPANO
IE=IS

GO THRU EACH BAND

DO 100 IBAND=IS,IE
RE WIND 13
REWIND 14

WRITE SUBTITLE

WRITE (6,2) IBAND
IPSTR=(ICSTOP-1)*4+IBAND
IPEND=ICSTOP*4

INITIALIZE SNAP IN PITCH

CALL PITCHR(IMAGE,JCELL,JCELL,INIT,1,IMAX,IMIN,NROW,NFILES,IFIL)

1
NU LW,NUL D,AMAG,OMAG,

POSITION THE INPUT TAPE

NOSK=IRSTRT-1
CALL EPFWND
IF(NOSK.NE.0) CALL ESKIP(NOSK)

GO THRU EACH LINE

DO 50 LINE=IPSTR,IRSTRT+LNSKIP
READ LNSKIP LINES

GET THIS LINE FOR THE IMAGE

DO 20 IP=IPSTR,IPEND,LNSKIP
KP=KP+1
IMAGE(KP)=ILINE(IP)
Continu E

WRITE THIS LINE TO SCRATCH FILE 14

WRITE(14) (IMAGE(IP),IP=1,JCELL)
CONTINUE

ENDFILE 14
REWIND 14

QUANTIZE FILE 14 AND PUT OUT ON 13

IF(QUAN) CALL ZEQUAN(IMAGE,INIT,JCELL,1,1,1,13,14,13)
REWIND 13

READ THE QUANTIZED IMAGE

DO 90 I=1,INIT
READ(13) (IMAGE(K),K=1,JCELL)

SNAP OUT THIS LINE

CALL SNAP

CONTINUE
RETURN
END
MODIFIED FROM KEQUAN BY Z DIENSTEIN  

DESCRIPTION OF PROGRAM. 

THIS SUBROUTINE QUANTIZE AN IMAGE ON FILE "INFILE" BY EQUAN. 

PROBABILITY TO NG LEVLES AND OUTPUTS IT TO FILE "IOUTFL". THIS 

PROCESSING IS DONE LINE BY LINE AFTER A FIRST PASS IS MADE THRU 

THE IMAGE TO DETERMINE THE MINIMUM AND MAXIMUM GREY TONES. 

ENTRY POINT. 

CALL ZEQUAN(LINE,NUMLIN,NUMPPL,ICOMP,LEFT,NQ,INFILE, 

IOUTFL) 

INPUT ARGUMENTS. 

LINE = ARRAY ONE LINE OF IMAGE IS READ INTO 

NUMLIN= NUMBER OF LINES IN IMAGE 

NUMPPL= NUMBER OF COLUMNS IN IMAGE 

NCOMP= NUMBER OF COMPONENTS IN ORIGINAL IMAGE 

ICOMP = THE COMPONENT TO BE QUANTIZED 

LEFT = LEFT-MOST CELL IN LINE DESIRED 

NQ = NUMBER OF QUANTIZED LEVELS 

INFILE= INPUT FILE CONTAINING IMAGE TO BE QUANTIZED 

OUTPUT ARGUMENTS. 

IOUTFL= OUTPUT FILE CONTAINING QUANTIZED IMAGE 

SUBROUTINE ZEQUAN(LINE,NUMLIN,NUMPPL,ICOMP,LEFT,NQ,INFILE, 

IOUTFL) 

C DIMENSION LINE(NCOMP,1),KN(512) 

DO 1 I=1,512 

KN(I)=0 

1 CONTINUE 

MIN=10000 

MAX=-10000 

REWO INFILE 

C DO 2 II=1,NUMLIN 

READ(INFILE) (LINE(J,L),J=1,NCOMP),L=1,NUMPPL 

GO THRU EACH LINE 

C IRIGHT=LEFT+NUMPPL-1 

SET STARTING,ENDING PTS IN LINE 

C DO 2 II=LEFT,IRIGHT 

GET MIN,MAX AND HISTOGRAM 

C IF(MAX.LT.J) MAX=J 

J=LINE(ICOMP,I) 

IF(MIN.GT.J) MIN=J 

KN(J+1)=KN(J+1)+1
IMPROBABILITY OF THE
ORIGINAI PAGE IS POOR/

02-12-74 19.149

`ZEQUAN`

2 CONTINUE
NGL=MAX
WRITE((16,100)) MIN,MAX
100 FORMAT(1X,*MIN, MAX APE '',Z4)
C NP=NUMLIN*NUMPL
J=I
MQ=NQ
C DO 3 I=1,NQ
N.=NP
C 4 N=NL-MQ*KN(J)
NP=NP-KN(J)
KN(J)=1-1
J=J+1
C IF(J.J.GT.NGL) GO TO 6
IF(MQ*KN(J).LE.NL*2)GO TO 4
C DETERMINE LEVELS
DECREASE THE NO. OF LEVELS LEFT
C IF LAST LEVEL, SKIP
INCREMENT THE LEVEL AGAIN
C GET NEW LEVEL.
RESET THE LAST LEVEL
C GO TO 8
C N=(NO-I)/2
IF(N,.LT.1)GO TO 8
DO 7 I=1,NGL
7 KN(I)=KN(I)+N
C CONTINUE
RECEIVE DISC FILES
C REWIND INFIL
ASSIGN QUANTIZED LEVELS LINE BY LINE
C REWIND IOUTFL
DO 11 II=1,NUMLIN
READ(INFILE) ((LINE(J,L),J=1,NCOMP),L=1,NUMPL)
DO 9 I=LEFT,RIGHT
J=LINF(lCOMP+I)
9 LINE(lCOMP,I)=KN(J+1)
C WRITE OUT QUANTIZED FILE
C WRITE(IOUTFL) ((LINE(lCOMP,K),K=LEFT,RIGHT)
C CONTINUE
C END FILE IOUTFL
C REWIND IOUTFL
C RETURN
C END
REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

51 FOR INTEGER, 0 FOR FLOATING
POINT

SHIFT UP NEXT CHARACTER

ADRESSES OF A SHORT ARRAY

PUT IN SHORT ARRAY

PUT BLANKS IN JUSTIFIED, BLANK FILLED

MAKE NO LEFT JUSTIFIED BY FIVE

RETURN IF ENTRIES AT BORDER

IS IT DIVISIBLE BY TEN

CURRENT CHARACTER FOR VERTICAL
BORDER

KEystone ENTER SIDE OF IMAGE DATA

DATA FOR VERTICAL BORDER

COPY BIG FOR 5 MEDIAN

1.6.12
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>LOAD</td>
<td>3514.24</td>
<td>0</td>
</tr>
<tr>
<td>157</td>
<td>FAD</td>
<td>0.00</td>
<td>0</td>
</tr>
<tr>
<td>158</td>
<td>FSTR</td>
<td>E</td>
<td>2</td>
</tr>
<tr>
<td>159</td>
<td>LDD</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>160</td>
<td>LLG</td>
<td>5,1</td>
<td>4</td>
</tr>
<tr>
<td>161</td>
<td>TNZ</td>
<td>Z</td>
<td>5</td>
</tr>
<tr>
<td>162</td>
<td>EA0</td>
<td>F</td>
<td>6</td>
</tr>
<tr>
<td>163</td>
<td>STX0</td>
<td>G1</td>
<td>7</td>
</tr>
<tr>
<td>164</td>
<td>STX1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>165</td>
<td>TRA</td>
<td>0,1</td>
<td>9</td>
</tr>
<tr>
<td>166</td>
<td>ZZ</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>EUX</td>
<td>SX</td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>E1X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>E2X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>AMIN</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>SLVK</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>EVD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
WRITE ERTS DATA ONTO DISC

VERSION II WRITTEN BY RJ OSLEY JAN 1974

DESCRIPTION OF PROGRAM:
THIS SUBROUTINE WRITES ONE LINE OF ERTS DATA FROM ARRAY ILINE
ONTO DISC FILES, ONE FILE FOR EACH BAND. DATA IS WRITTEN IN NHOR
BLOCKS.

INTERNAL PARAMETERS:
DISC FILE CODE, 11 THRU 14
IP
POINT INDEX FOR ERTS LINE
IPSTR,IPSTOP
STARTING,STOPPING POINTS IN ILINE, DEPENDENT UPON THE BAND

ENTRY POINT:
CALL WRTDSK(ILINE,NHOR,IPSTR,IPEND,NBAND)

INPUT ARGUMENTS:
ARRAY CONTAINING ERTS LINE OF DATA
ILINE
NUMBER OF HORIZONTAL BLOCKS OF 41 COL
NHOR
START AND END POINTS IN ERTS LINE
IPSTR,IPEND
THE BAND DESIRED, SET=5 FOR ALL FOUR
NBAND

SUBROUTINE WRTDSK(ILINE,NHOR,IPSTR,IPEND,NBAND)

DIMENSION ILINE(1)

IF(NBAND.EQ.5) GO TO 100 WRITE ONE BAND ONLY

NDSK=10*NHOR

DO 10 J=1,NHOR
IPSTRT=(J-1)*41*4+NBAND+IPSTR
IPSTOP=IPSTR+40*4
WRITE(NDSK) (ILINE(IP),P=IPSTR,IPSTOP,4)
10 CONTINUE
RETURN

DO 100 I=1,4
NDSK=10*I

DO 101 J=1,NHOR
IPSTRT=(J-1)*41*4+I+IPSTR
IPSTOP=IPSTR+40*4
WRITE(NDSK) (ILINE(IP),P=IPSTR,IPSTOP,4)
101 CONTINUE
RETURN
END

REPRODUCIBILITY OF THIS
ORIGINAL PAGE IS POOR
DESCRIPTION OF PROGRAM.

This subroutine reads the ERTS data from the disc files and then prints out a picture of the data, proceeding vertically and then from left to right. Since the blocks are 41 cols then two will fit on one page, eliminating one half of the total lines.

INTERNAL PARAMETERS.

- NGL: Number of grey levels in image
- NO: Number of quantizing levels
- IASIZE: Size of array to be quantized
- MHOR: Half of Hor
- NCOL: Number of columns used in image
- NSkip: Number of records to skip to stay in the same strip
- LAST: True indicates the last strip

ENTRY POINT.

CALL RDOSK1(Image, Quan, NHOR, NVERT, NDSK)

INPUT ARGUMENTS.

- Image: Array to store two 41 x 41 blocks
- Quan: True for equal probability quantization of the image
- Max: Maximum grey tone in image
- M:N: Minimum grey tone in image
- NHOR: Number of horizontal blocks
- NVERT: Number of vertical blocks
- NDSK: File code of disc to be processed

SUBROUTINE RDOSK1(Image, Quan, NHOR, NVERT, NDSK)

1. DIMENSION Image(41,82)
2. LOGICAL EOF, Quan, LAST
3. EOF=.FALSE.
4. LAST=.FALSE.
5. CALL FLEGEND(NDSK, EOF)

Determine the number of levels

- MAX=63
- M:N=0
- IF(Quan) MAX=11

Set up for quantization

- NGL=75
- NO=12
- MHOR=NHOR/2
- NCOL=82

Reproducibility of the original page is poor.
IF(NHOP.NE.1) GO TO 2
3 NCOL=41
MHO=1
IASIZE=41*NCOL
C NSKIP=NHOP-(NCOL/41)
C SET THE NUMBER OF RECORDS TO SKIP
C TO REMAIN IN THE SAME STRIP
C GO THRU HORIZONTALLY
C
C DO 100 J=1,NHOR
C REWIND NOSK
C K=J
C IF(LAST) K=(NHOR/2)+1
C IF(K,EQ.1) GO TO 5
C DO 1=2,K
C IF(NHOR,GT.1) READ(NOSK)
1 C REASNOSK)
C IF ONLY ONE BLOCK VERTICALLY, DO NOT
C INITIALIZE SNAP IN PITCHR
C REASNOSK)
C IF(NVERT,NE.1)
C 1 CALL PITCHR(IMAGE,41,NCOL,NVERT,,MAX,MIN,1,21,,)
C READ IN TWO 41 BY 41 BLOCKS
C DO 50 II=1,NVERT
C GO 20 JJ=1,41
C READ NCOL COLUMNS
C READ(NOSK) IMAGE(JJ,KOL),KOL=1,41)
C IF(NCOL,NE.82) READ(NOSK) IMAGE(JJ,KOL),KOL=42,82)
C SKIP OVER RECORDS NOT WANTED
C IF(NSKIP,EQ.0) GO TO 20
C DO 16 N=1,NSKIP
C READ(NOSK)
C IF(EOF) GO TO 20
16 CONTINUE
C CONTINUE
C EOF=.FALSE.
C QUANTIZE THE IMAGE
C IF(QUAN) CALL KEQUAN(IMAGE,NSL,NQ,IASIZE)
C SNAP OUT A 41 BY 82 BLOCK
C IF(NVERT,NE.1) CALL SNAP
C IF(NVERT,NE.1) CALL PITCHR(IMAGE,41,NCOL,1,MAX,41:M,,1,21,,)
C CONTINUE
C CONTINUE
C CHECK TO SEE IF NHOR IS EVEN
C IF(NHOR.EQ.1) GO TO 101
C IF(NHOR.EQ.(NHOR*2)) GO TO 101
C IF NOT, DO THE LAST STRIP OF 41 COLS
C IF(LAST) GO TO 101
C LAST=.TRUE.
C GO TO 3
READ DISC AND OUTPUT PICTURE

WRITE THE BAND NUMBER

101 IGAND=NOSK-10
WRITE(6,102) IGAND
102 FORMAT(6X,'PICTURE FROM BAND NUMBER',2)
RETURN
END
CKEQUAN

KEQUAN01
KEQUAN02
KEQUAN03
KEQUAN04
KEQUAN05
KEQUAN06
KEQUAN07
KEQUAN08
KEQUAN09
KEQUAN10
KEQUAN11
KEQUAN12
KEQUAN13
KEQUAN14
KEQUAN15
KEQUAN16
KEQUAN17
KEQUAN18
KEQUAN19
KEQUAN20
KEQUAN21
KEQUAN22
KEQUAN23
KEQUAN24
KEQUAN25
KEQUAN26
KEQUAN27
KEQUAN28
KEQUAN29
KEQUAN30
KEQUAN31
KEQUAN32
KEQUAN33
KEQUAN34
KEQUAN35
KEQUAN36
KEQUAN37
KEQUAN38
KEQUAN39
KEQUAN40
KEQUAN41
KEQUAN42
KEQUAN43
KEQUAN44
KEQUAN45
KEQUAN46
KEQUAN47
KEQUAN48
KEQUAN49
KEQUAN50

C

WRITTEN BY G. ELLIOT

VERSION II BY RJ BOSLEY FOR IASIZE

DESCRIPTION OF PROGRAM.

THIS PROGRAM WILL QUANTIZE BY EQUAL PROBABILITY THE INPUT

ARRAY IA TO NO LEVELS.

ENTRY POINT.

CALL KEQUAN(IA,NGL,NQ,IASIZE)

ARGUMENTS.

IA

INPUT ARRAY TO BE CONVERTED TO

QUANTIZED ARRAY

NGL

NUMBER OF GREY TONE LEVELS IN IA

NQ

NUMBER OF QUANTIZING LEVELS

IASIZE

SIZE OF ARRAY IA

SUBROUTINE KEQUAN(IA,NGL,NQ,IASIZE)

DIMENSION IA(1),KN(512)

IF(NGL.GT.512) WRITE(6,10)

COUNT EACH GREY LEVEL

DO 1 I=1,NGL

1

KN(I)=0

GO 1 I=1,NGL

J=IA(I)

KN(J+1)=KN(J+1)+1

NP=IASIZE

J=1

MQ=NQ

GO THRU NQ LEVELS

DO 3 I=1,NQ

2

NLP=NP

GET NEW LEVEL

NL=NL-MQ*KN(J)

IF (NL.GT.NGL) GO TO 6

INCREMENTS AGAIN FOR LEVEL

IF(MQ*KN(J).LE.NL) GO TO 4

DECREASE NO. OF LEVELS LEFT

3

MQ=MQ-1

DO 5 I=J,NGL

4

KN(I)=MQ-1

GO TO 8

5

GO TO 8
C
6 N=(NQ-I)/2
IF(N.LT.1)GO TO 8
DO 7 I=1,NGL
   KN(I)=KN(I)+N
7 C
8 DO 9 I=1,..ASIZE
    J=IA(I)
9   A(I)=KN(J+1)
RETURN
END
SUBROUTINE RDOSK2(IMAGE,IRSTRT,IRSTOP,NHOR,NJSK,PRNT,TAPE,IFIL)

INPUT ARGUMENTS.
IMAGE
IRSTRT,IRSTOP
NHOR
NJSK
PRNT
TAPE
IFIL

SUBROUTINE WRITES DATA FROM DISC FILE NJSK AND PRINTS OUT THE GREY TONES AND WRITES THEM OUT ON AN OUTPUT TAPE.

ENTRY POINT.
CALL RDOSK2(IMAGE,IRSTRT,IRSTOP,NHOR,NJSK,PRNT,TAPE,IFIL)

VERSION II WRITTEN BY RJ DOSLEY JAN 1974

DESCRIPTION OF PROGRAM.
THIS SUBROUTINE READS DATA FROM DISC FILE NJSK AND PRINTS OUT THE GREY TONES AND WRITES THEM OUT ON AN OUTPUT TAPE.

INTERNAL PARAMETERS.
K
LINE

ENTRY POINT.
CALL RDOSK2(IMAGE,IRSTRT,IRSTOP,NHOR,NJSK,PRNT,TAPE,IFIL)

INPUT ARGUMENTS.
IMAGE
IRSTRT,IRSTOP
NHOR
NJSK
PRNT
TAPE
IFIL

IF(PRNT) GO TO 2
IF(TAPE) GO TO 3
RETURN

WRITE(6,1) K
FORMAT(h1,'LINE STRIP',20X,'BAND NUMBER IS',I2)
DO 50 LINE=IRSTRT,IRSTOP

IF(PRNT) WRITE(6,100) LINE,J,IMAGE(1,KOL),KOL=1,41
IF(TAPE) WRITE(IFIL) LINE,J,IMAGE(1,KOL),KOL=1,41
CONTINUE
FORMAT(1X,I5,I3,4I3)
RETURN
END
IV.2-b  Texture Analysis Program Listings

MAINLN
ERTS (see IV.2-a)
MAING
KEQUAN (see IV.2-a)
PITCHR (see IV.2-a)
FPLXIT
INDEX
IMOMTR
COR
IEQPQi
RITOWT
This TEXTURE ANALYSIS PACKAGE was written in order to process ERTS imagery data using pattern recognition techniques.

METHOD.


INPUT.

PARAMETER CARDS.

1. TITLE CARD. This card is used for title information and is listed on the output listing.

2. PARAMETERS according to the FORTRAN NAME.1ST FORMAT under the NAME PARAM, see below.

OTHER INPUT.

ERTS IMAGE DATA TAPE ON INPUT FILE CODE 'ES'.

ABOOUTS.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.
SPECIFIED IN OUTPUT LISTING.

OUTPUT.

PRINTER OUTPUT.
1. PARAMETER VALUES.
2. PICTURE OF IMAGE BEING PROCESSED.
3. COORDINATES OF THE IMAGE.
4. VALUES OF THE EXTRACTED FEATURES.
5. LISTING OF LFX ARRAYS.

CARD OUTPUT.
1. COORDINATES OF THE IMAGE BEING PROCESSED.
2. VALUES OF THE FEATURES EXTRACTED.
3. CARD COUNT.

TAPE OUTPUT.
1. COORDINATES OF THE IMAGE BEING PROCESSED.
2. VALUES OF THE FEATURES EXTRACTED.

RESTRICTIONS.
1. DATA SET IS STANDARD ERTS DATA TAPE FROM NASA.
2. THE NUMBER OF POINTS IN ONE LINE OF THE STRIP BEING PROCESSED MUST NOT EXCEED 192 POINTS.
3. THE SIZE OF EACH IMAGE MUST NOT EXCEED 4096 POINTS.
4. QUANTIZATION MUST BE 32 LEVELS OR LESS.
5. THE ERTS INPUT TAPE MUST HAVE A FILE CODE 'ES'.
6. ICRU MUST NOT EXCEED FOUR.

SUBPROGRAMS REQUIRED.
- MAINLN
- ERTS READ PROGRAM
- MAING
- KEQUAN
- PICTUR
- FPLXIT
- INDEX
- IMOTR
- INDEX
- COR
- IEQPO1
- RITCOT

CARD SETUP FOR SAMPLE RUN.

$ IDENT 9999, ANYNAME
$ LIBRARY LB
$ OBJECT M-A-I-N-L-N
$ ERTS TEXTURE ANALYSIS PROGRAMS...
$ EXECUTE

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
This run of the Texture Analysis Programs will process the ERTS data in 64 x 64 images, giving only printed output, plus a picture of the images.

BIBLIOGRAPHY

DESCRIPTION OF PROGRAM
This is the mainline of the ERTS Texture Analysis Programs. Each ERTS image is divided into 4 strips, each put onto one tape. The first tape is divided into subimages 1 to 12, the second from 13 to 24, etc., up to 48 horizontal images.

Due to core limitations, each input tape is processed in strips with each strip totaling up to 192 points, horizontally. If the images are 64 x 64, each strip will contain 3 subimages. Images will be processed as follows, run 1 (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), ..., (36,1), (36,2), (36,3), run 2 (1,4), (1,5), ..., (36,4), (36,5), (36,6), etc.

Note that for run 1, N11 is 1 and NUMSTR is 1. For run 2, N11 is 4 and NUMSTR is 2. And for the second tape, for run 1, N11 is 13 and NUMSTR is 1. N11 is the upper-left column coordinate and is relative to the entire image, where NUMSTR is relative to the data tape. Also note that the first 8 points at the beginning of each line are left out. It is possible to have less than one full image at the end of each tape that cannot be processed.

DESCRIPTION OF INPUT PARAMETERS UNDER NAMELIST /PARAM/.

NOUANT NO. OF QUANTIZING LEVELS IN IEOQP1, SET TO 16.

NUMIM THE NUMBER OF IMAGES TAKEN HORIZONTALLY IN ONE PASS.

NUMSTR THE STRIP NUMBER OF THE RUN IN RELATION TO THE ERTS INPUT TAPE.

NEVERT THE NUMBER OF IMAGES IN A VERT COL OF THE STRIP.
THE NUMBER OF LINES IN EACH NUMIN X NUMPPL SUB-IMAGE
NUMPPL
THE NUMBER OF PTS PER LINE IN EACH SUB-IMAGE
***NOTE***NUMPPL*NUMIN JUST NOT EXCEED 192
***EXAMPLE***IF NUMPPL=64,THEN NUMIN=3
***EXAMPLE***IF NUMPPL=32,THEN NUMIN=4
INAND
THE SPECTRUM BAND TO BE PROCESSED, FROM 1 TO 4
PNCH
SPECIFIES THE OUTPUT OPTION----Y FOR CARDS, T FOR TAPE, AND N FOR PRINTER ONLY
***NOTE***PNCH MUST BE DENOTED AS A TOLERANCE CONSTANT IN THE DATA CARD
NI1
THE UPPER LEFT COLUMN COORDINATE FOR THE STRIP BEING PROCESSED
IF
THE FILE CODE OF THE OUTPUT TAPE--ASSUMED TO BE IN POSITION
NESKIP
THE NUMBER OF VERTICAL ROWS OF SUB-IMAGE TO BE SKIPPED PRIOR TO EXECUTION
MERGE
IF .TRUE., THE LEX ARRAYS WILL BE MERGED TO ONE PICTUR
IF .TRUE., A PICTURE OF EACH SUB-IMAGE WILL BE PRINTED
***NOTE***PROCESSING IS, APPROXIMATELY--
100 PERCENT= MERGE-OFF, PICTUR-ON
125 PERCENT= MERGE-ON, PICTUR-ON
158 PERCENT= MERGE-ON, PICTUR-OFF

D_MENSION ILINE(4096),TITLE(14)
COMM COMMON /Q/ NQUANT
COMMON M1,M1,F(15),LMAX,LMIN,NUMPPL,NUMIN,NB0BL,IR1,IR2,IR3,IR4,
COMMON M1,MI,F(15),LMAX,LMIN,NUMPPL,NUMIN,NB0BL,IR1,IR2,IR3,IR4,
1 COMMON /E/ ENTRNP(4),DIFENT(4),DIFAVE(4),DIFFAR(4),SUMNT(4),
COMMON /E/ ENTRNP(4),DIFENT(4),DIFAVE(4),DIFFAR(4),SUMNT(4),
1 SUMAVE(4),SUMVAR(4)
COMMON /CORRE/CORINF(4),ORMIT(4),CORMAX(4)
COMMON IMAGE(64,192)
LOGICAL MERGE, PICTUR
NAMEL,TH/ / /NAMTH/NUMTH/NUMSTR/NVERT/NBAND/N11,MERGE,PICTUR,NESKIP,
1 PNCH,NUMPPL,NUMIN,NPLOT,VLAYE,NTIME,NTIME,NQUANT,F
DATA IDAN/NUMIN,NVERT,NESKIP,MERGE,PICTUR,N11/2,3,36,0.,TRUE.,
1 .FALSE./,1/TAPE/1HT/,NUMSTR/1/IF/03/,Y/Y/14Y/,N/1HN/
PICH=TAPE
NUMPPL=64
NUMIN=64
NPLT=1
NPLT=1
NSTART=1
NTIME=1
NQUANT=16

**************SECTION I--PREPARE FOR ERTS READING AND PROCESSING********

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
READ TITLE AND WRITE IT OUT

READ(S,F) (TITLF(I),I=1,14)
WRITE(6,21)
WRITE(6,7) (TITLF(I),I=1,14)
FORMAT(/////040X,'ERTS TEXTURE ANALYSIS'////)

READ ERTS TEXTURE PARAMETERS

READ(5,PAPAM)
M=NUMH*NUMPPR
IF(M.LE.192) GO TO 2
WRITE(6,1)
FORMAT(' NUMBER OF IMAGES TIMES THE NUMBER OF POINTS _Y EACH LINE
MUST NOT EXCEED 192 ')
STOP

INITIALIZE THE ERTS READ PROGRAM AND WRITE OUT PARAMETERS

CALL INIT(LENGTH)
WRITE(6,11) LENGTH,NUMH,NUMSTR,NUMVERT,IBAND,NBSKP
WRITE(6,101) PUNCH,NUMPPR,NUMIN,NRED,NSTAPT,NTIMES,NQUANT
FORMAT(10X,'LENGTH OF ERTS LINE IS',I5,' POINTS'/10X,'NUMBER OF HORIZONTAL IMAGES,NUMIM,IS',I2,/210X,'THIS STRIP IS NUMBER',I2,'10X,'NUMBER OF VERTICAL IMAGES FORMAINJ120
STAINED IN STRIP IS',I2,'10X,'PROCESSING WILL BE ON BAND',I2,/410X,'SKIPPED DOWN ',I3,' VERTICAL IMAGES BEFORE START.'14X///)
FORMAT(10X,'PUNCH=',A3,' NUMPP=',I4,' NUMIN=',I4,' NRED=',I4,
ISTICX,'NQUANT=',I4)
IF(MERGE) WRITE(6,3)
FORMAT///010X,'FOUR EX ARRAYS HAVE BEEN MERGED INTO ONE ARRAY'MAINJ175
1 
IF(.NOT.PICTUR) WRITE(6,4)
FORMAT( ' THE PICTURE OPTION IS OFF' ///)
4
SKIP THE FIRST NBSKP ROWS OF IMAGES

NBSK=NUMIN*N3SKIP
CALL ESKP(NBSK)

GO DOWN THE STRIP
IMAGE GORDINATES (M1,N1) ARE TRANSFERRED IN COMMON
M1 GIVES THE ROW COUNT GOING DOWN THE STRIP

IBEGIN=((NUMSTR-1)*(192*4))+IBAND
MOVE THE IMAGE TO THE RIGHT BY EIGHT POINTS

BEGIN=IBegin+32
JSTOP=JStop+((NUMIM*NUMPPL)*4)-1BAND
N9=NSkip+1
DO 99 M1=M8, NOVERT

*********SECTION II--READ ERTS AND MOVE DATA INTO IMAGE*********

LINE IS THE ARRAY INTO WHICH THE ERTS DATA IS READ--ILINE AND
WORK USE THE SAME STORAGE SPACE.
MDOWN GIVES THE ROW COUNT AN IMAGE FROM 1 DOWN TO NUMLIN
DO 99 MDOWN=1, NUMLIN
READ ERTS LINE BY LINE
CALL EREAD(ILINE, LN)
LN, RETURNED BY EREAD, GIVES THE LINE NUMBER, OR ERROR INDICATION
IF (LN.FEQ.0) GO TO 996

MOVE EVERY POINT IN THE LINE THAT BELONGS TO 1BAND, INTO IMAGE
JSTOP GIVES THE STOPPING POINT IN ILINE FOR THE TRANSFER
THE TRANSFER IS INCREMENTED BY 4, THE NUMBER OF BANDS
COUNT GOES FROM 1 TO NUMPPL*NUMIM, GIVING THE LENGTH COUNT

Lcount=0
DO 80 Ipoint=Begin, JStop, 4
Lcount=Lcount+1
IMAGE(MDOWN, LCOUNT)=YES(LINE(I.POINT))
CONTINUE
CONTINUE

IMAGE IS FULL. START TEXTURE ANALYSIS

*********SECTION III--PROCESS IMAGE, BLOCK BY BLOCK*********

BLOCk IMAGE INTO NUMLIN X NUMPPL BLOCKS FOR PROCESSING

KF=0
DO 60 Jblock=1, NUMIM
KS=KS+1
KE=KS*NUMPPL-1
KL=0
DO 59 Kline=1, NUMLIN

K. GOES FROM 1 TO NBR OF Pts WHILE KCOL GIVES THE COLUMN COUNT

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
DO 58 KCO=K-1,K1
K2=K2+1
I.LINE(KL)=IMAGE(K.LINE,KCO)
58 CONTINUE
59 CONTINUE
M1=(M1-1)+JBLCK
C
USE ILINE AS A DUMMY ARRAY TO SEND WORK TO MAING
C
CAL. MAING(ILINE,MERR,MERGE,PICTUR,IF)
C
CHECK FOR ERROR CONDITION IN MAING
C
IF(MERR.EQ.1) GO TO 992
60 CONTINUE
99 CONTINUE

**********SECTION IV--END OF STRIP, FINISH UP**********
C
PUT AN EOF MARK ON OUTPUT FILE IF AND WRITE ANOTHER RECORD
C
ENDFILE IF
WRITE(IF) (ILINE(K),K=1,10)
STOP
C
ERROR DETECTED--WRITE FILE MARK AND ANOTHER RECORD
C
992 WRITE(6,993) M1,N1
993 FORMAT(10X,'ERROR IN SUBROUTINE MAING, LAST IMAGE WAS "',
12,X,'",')
ENDFILE IF
WRITE(IF) (ILINE(K),K=1,10)
STOP
C
ERROR DETECTED--WRITE FILE MARK AND ANOTHER RECORD
C
996 WRITE(6,997) M1
997 FORMAT(10X,'UNEXPECTED EOF ON ERTS, LAST ROW COMPLETED WAS ",,13)
ENDFILE IF
WRITE(IF) (ILINE(K),K=1,10)
STOP
END
DESCRIPTION OF PROGRAM.

THIS SUBROUTINE PREPARES THE IMAGE IN ARRAY IWORK FOR PITCHR AND THEN PROCESSES IT, ACCORDING TO THE MERGE OPTION. CALLING SUBROUTINES FOR THE LEX ARRAYS, CALCULATING THE TEXTURE FEATURES AND THEN WRITING OUT THE RESULTS.

ENTRY POINT.

CALL MAING(IWORK,MERR,MERGE,PICTUR,IF)

ARGUMENTS.

IWORK THE NUM*N*NUMPP_ IMAGE ARRAY
MERR ERROR FLAG FOR LEX ARRAY SIZE
MERGE OPTION TO MERGE THE FOUR LEX ARRAYS
PICTUR OPTION TO PRINT PICTURE OF THE IMAGE
IF FILE COEF FOR OUTPUT TAPE IN RITOW

INTERNAL PARAMETERS.

NUMLIN THE NUMBER OF LINES IN THE IMAGE
NUMPP_ THE NUMBER OF POINTS PER IMAGE LINE
NUM_ MAXIMUM ALLOWABLE SIZE OF THE LEX ARRAYS, NUMPP_*NUMLIN
NQ NUMBER OF QUANTIZING LEVELS FOR KFQUAN
IMAX MAXIMUM GREY TONE LEVEL IN IWORK
IMIN MINIMUM GREY TONE LEVEL IN IWORK
IS STARTING PT OF A ROW IN IWORK
IE ENDING POINT OF THE ROW IN IWORK
NGL NUMBER OF GREY TONE LEVELS IN IWORK
IASIZE SIZE OF IWORK ARRAY
HORIZ HORIZONTAL SCALE FACTOR FOR PITCHR
VERT VERTICAL SCALE FACTOR FOR PITCHR
LEAST1 ONE GREY TONE LEVEL BELOW IMIN
NOLV NUMBER OF GREY TONE LEVELS IN IWORK
NBUL THE NUMBER OF LEVELS IN THE TRIANGULAR LEX ARRAY
L1...L6 ADDRESS INDEXS FOR THE LEX ARRAYS

SUBROUTINE MAING(IWORK,MERR,MERGE,PICTUR,IF)

LOGICAL MERGE,PICTUR
DIMENSION IWORK(4096)

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
DIMENSION G(64),IQ(64)
COMMON /A/ NQANT
COMMON NI,N1,F(I15),IMAX,MIN,NUNPL,NUMLIN,NQ0PL,IR1,IR2:R3:R4,
COMMON SUM(4),SUMVAR(4)
COMMON /C/ENTROPI(4),DIFENT(4),FAV(4),SUHEMT(4),SUMA(4)
COMMON /COR/ CORR(4),CORMAX(4)
COMMON IMAGE(64,192)
COMMON =NUMPL*NUNLIN
NQ=32

REWRITE SCATCH FILE, COPY/IMAGE IN A LINE BY LINE FASHION ONTO
THE SCATCH FILE, AND DETERMINE THE MINIMUM AND MAXIMUM GREY TONE
REWRITE 2

FIRST, QUANTIZE THE ARRAY TO NQ LEVELS FOR PICTURE AND EFFICIENCY
IMAX=-10000
MIN=10000
DO 13 J=1,NUMPL
IE=NUNPL*J
DO 12 K=IS,IE IF(IIMIN.GT.IWORK(K)) IMIN=IWORK(K)
IF(IIMAX.GT.IWORK(K)) IMAX=IWORK(K)
CONTINUE
12 CONTINUE
13 CONTINUE
NGL=IMAX+1
IASIZE=NUNPL*NUNLIN
CALL KEQUA(IWORK,NGL,NQ,IASIZE)

COPY IMAGE ON SCATCH FILE AFTER QUANTIZATION
DO 20 I=1,NUMLIN
IS=NUNPL*(I-1)+1
IE=NUNPL*I+1
WRITE(2) (IWORK(K),K=IS,IE)
CONTINUE
20 CONTINUE

THE MAXIMUM AFTER KEQUAN QUANTIZES TO NQ LEVELS IS NQ-1, MINIMUM=0
IMAX=NQ-1
MIN=0

TEST FOR PICTURE OPTION
IF(.NOT.PICT) GO TO 16

TRANSPOSE IWORK FOR PITCH

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
L=0
DO 14 I=1,NUM-IN
L=L+1
IS=NUMPL*{I-1}+1
IE=NUMPL*I
K=0
DO 15 J=IS,IE
K=K+1
IMAGE(I,K)=IWORK(J)
15 CONTINUE
CONTINUE
PRINT OUT PICTURE OF THE IMAGE
HORZ=(64.0*0.90)/FLOAT(NUMPL)
VERT=(64.0*0.75)/FLOAT(NUMLIN)
CALL PITCHR(IMAGE,NUMLIN,NUMPL,0,1,0,IMAX,,HORZ,VERT,)
LEAST1=IMIN-1
NOBL IS THE NUMBER OF BRIGHTNESS LEVELS
NOBL=IMAX-LEAST1
NBUBL IS THE NUMBER OF LEVELS IN THE TRIANGULAR LEX ARRAY
NBUBL=NOBL*(NOBL+1)/2
SET UP THE INDEXS FOR THE LEX ARRAYS
L1=1
L2=L1+NUMPL*2
L3=L2+NBUBL
L4=L3+NBUBL
L5=L4+NBUBL
L6=L5+NBUBL-1
CHECK THE SIZE
IF(L6.GT.NDIM) GO TO 78
DO 4 NN=NSTART,NTIMES
REWO 2
NLAYER=NN-1
GET THE LEX ARRAYS
CALL PFLEXIT(IWORK(L1),IWORK(L2),IWORK(L3),IWORK(L4),IWORK(L5),
1 NUMPL,HERGE)
CALCULATE THE TEXTURE FEATURES
CALL IMONTR(IWORK(L2),IWORK(L3),WORK(L4),IWORK(L5),G,17,MERGE)

OUTPUT THE TEXTURE DATA

CALL RITOWN(IWORK(L2),IWORK(L3),WORK(L4),IWORK(L5),G,17,MERGE,IF)

CONTINUE

SET ERROR INDICATOR TO NO ERRORS

MFRR=0
RETURN

WRITE(6,104) NOIM,L5
104 FORMAT(6H NOIM=,15,16H NOIM MUST BE = ,17)

MERR=1
RETURN
END
DESCRIPTION OF PROGRAM.

This subroutine computes four nearest neighbor grey tone matrices, LEX1, LEX2, LEX3, and LEX4, for angles of 90-degrees, 90-degrees, 135-degrees, and 45-degrees respectively.

Included in this subroutine is an option to merge the four LEX arrays into one, LEX1.

ENTRY POINT.

CALL FPLXIT(IDATA, LEX1, LEX2, LEX3, LEX4, NUMPPL, MERGE)

ARGUMENTS.

IDATA WORKING ARRAY FOR TWO LINES OF IMAGE
LEX1-LEX4 ADDRESS INDEXES FOR LEX ARRAYS
NUMPPL NUMBER OF PTS PER IMAGE LINE
MERGE OPTION TO MERGE THE FOUR LEX ARRAYS INTO ONE ARRAY, LEX1

INTERNAL PARAMETERS.

NUMUBL NUMBER OF LEVELS IN THE TRIANGULAR LEX ARRAYS
IST POINTER TO FIRST LINE
NND POINTER TO SECOND LINE
NRED BASE FOR IMAGE REDUCTION
NLAYER THE POWER TO WHICH NRED IS RAISED
MM AMOUNT OF REDUCTION OF THE IMAGE
FILE 2 SCRATCH FILE CONTAINING THE IMAGE
INDEX(I,J) GREY TONE VALUES OF NEIGHBORING RESOLUTION CELLS, ONE TO EACH ANGLE
SUBSCRIPT FOR THE LEX ARRAY INDICATING WHERE ELEMENT (I,J) CAN BE FOUND

SUBROUTINE FPLXIT(IDATA, LEX1, LEX2, LEX3, LEX4, NUMPPL, MERGE)

DIMENSION IDATA(NUMPPL,2), LEX1(N1), LEX2(N1), LEX3(N1), LEX4(N1)

COMMON ML,N1, TYPE,F(14)
COMMON I1,I2,NUMPPL,NUMMIN,NUBUL,IR1,IR2,IR3,IR4,DUMMY(29)
COMMON LAYER, NRED, NSTART, NITEMS

LOGICAL MERGE

INITIALIZE LEX1, LEX2, LEX3, AND LEX4 ARRAYS TO ZERO

DO 10 I = 1, NUMUBL
C

IST POINTS TO FIRST LINE, NNO POINTS TO SECOND LINE

IST = 1
NNO = 2

MLAYER INDICATES BY HOW MUCH THE IMAGE WILL BE REDUCED.
NRED IS THE FACTOR BY WHICH THE IMAGE WILL BE REDUCED. (IT IS THE FPLXIT67
BASE WHICH IS RAISED TO THE POWER MAYER.) THEN, BY DEFINING THE FPLXIT68
QUANTITY MM, WHERE MM = NRED**MLAYER, WE HAVE A SINGLE FACTOR FPLXIT69
THAT DETERMINES THE REDUCTION BASE AND THE AMOUNT OF THE REDUC-
ION. IF, FOR EXAMPLE, NRED = 2, AND MLAYER RANGES FROM 0 TO 3 -- FPLXIT70
THIS RANGE IS DETERMINED BY THE PARAMETER TIMES (SEE *MAIN*), FPLXIT71
THE RESULTANT PROCESSING WILL YIELD FOUR IMAGES THAT WILL BE SUB-FPLXIT72
CESSIVELY REDUCED BY 1, 1/2, 1/4, AND 1/8 RESPECTIVELY. FPLXIT73

MM = NRED**MLAYER
NUMPL2 = NUMPL/MM
DO 111 KK=1,MM
DO 111 KK=1,MM
C

GET THE FIRST LINE OF DATA FROM DISC FILE 02
C

DO 1 LL=1,KK1
8 READ(2) (IDATA(L,IST),L=1,NUMPL1)
N = 0
DO 29 J=KK2,NUMPL1,MM
N = N + 1
29 IDATA(N,IST) = IDATA(J,IST)
C

MM=M+KK1
DO 1 LCNT = MMM,NUMLIN,MM
C

GET THE SECOND LINE OF DATA. AFTER EACH ITERATION, THE OLD SEC-
C
OND LINE BECOMES THE NEW FIRST LINE.
C

DO 1 LL=1,MM
18 READ(2) (IDATA(L,NND),L=1,NUMPL1)
N = 0
DO 19 J=KK2,NUMPL1,MM
N = N + 1

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
SET I, L, J, AND K EQUAL TO THE (NORMALIZED) VALUES OF GREY TONES OF RESOLUTION CELLS IN POSITIONS (I, ISTI), (1, NNDI), (2, ISTI), AND (Z, NNDI) RESPECTIVELY.

I = IDATA(1, ISTI) - LEAST1
J = IDATA(2, ISTI) - LEAST1
L' = IDATA(1, NNDI) - LEAST1
K = IDATA(2, NNDI) - LEAST1

OUT TWO DIMENSIONAL INFORMATION INTO ONE DIMENSIONAL FORM. THE FUNCTION NEEDED TO CONVERT A DOUBLE SUBSCRIPTED ARRAY, IMM(X,Y), INTO A SINGLE SUBSCRIPTED ARRAY, IMM(Z), IS OF THE FORM G(X) + F(Y), WHERE G(X) = (X-1)X/2, AND F(Y) = Y. THEREFORE,

Z = (X-1)X/2 + Y

THIS IS DONE IN THE PROGRAM BY THE EXTERNAL FUNCTION, INDEX(X,Y).


IMM(1,1) = IMM(1),
IMM(2,1) = IMM(2),
IMM(2,2) = IMM(3),
IMM(3,1) = IMM(4),

IMM(NNOB, NOBL) = IMM(NNOB), WHERE NNOB = NOB(NOBL + 1)/2, AND NOBL IS THE TOTAL NUMBER OF GREY TONES IN THE ARRAY.

THE SCANNING PROCEEDURE, THAT IS, THE METHOD BY WHICH THE PAIR-WISE COMPARISONS ARE MADE, IS DESCRIBED BELOW FOR THE GENERAL CASE.

CONSIDER A RESOLUTION CELL WITH SPATIAL COORDINATES (4,N), AND CALL THIS CELL 'I'. THE SCANNING OPERATION BEGINS IN THE UPPER LEFT HAND CORNER OF THE IMAGE (THE FIRST POSITION OF 'I') IS IN THAT OF RESOLUTION CELL (1,1) AND IT THEN PROCEEDS BY COMPARING THE GREY TONE OF 'I' WITH, AT MOST, FOUR GREY TONES OF ITS NEIGHBORING RESOLUTION CELLS. THAT 'I' NEVER NEEDS TO CONSIDER MORE THAN FOUR NEAREST NEIGHBORS.

CAN BE SEEN FROM THE DIAGRAM OF THE SEARCH PATTERN SHOWN BELOW.

```
I
J
H
L
K
```
ON A GIVEN ITERATION, 'I' WILL LOOK FIRST AT ITS NEAREST VERTICAL
NEIGHBOR ('L'), NEXT AT ITS NEAREST HORIZONTAL NEIGHBOR ('J'),
THIRD AT ITS LOWER RIGHT NEIGHBOR ('K'), AND FOURTH AT ITS LOWER
LEFT DIAGONAL NEIGHBOR ('M'). 'I' THE MOVES INTO THE POSITION OF 'I'
THE LEFT-MOST RESOLUTION CELL OF THE PREVIOUSLY SCANNED SECOND
ROW (THE POSITION OCCUPIED BY 'M'). THE OPERATION IS REPEATED UN-
TIL ALL NEIGHBORING PAIRS OF RESOLUTION CELLS HAVE BEEN EXAMINED.

MAKE COUNT FOR THE FIRST TWO COLUMNS.

IL = INDEX(I,L)
COUNT VERTICALLY ADJACENT (90-DEGREE) NEAREST NEIGHBORS FOR FIRST
TWO COLUMNS.
LEX1(IL) = LEX1(IL) + 1
IR1 = IR1 + 1
IJ = INDEX(I,J)
COUNT HORIZONTALLY ADJACENT (0-DEGREE) NEAREST NEIGHBORS FOR THE
FIRST TWO COLUMNS.
LEX2(IJ) = LEX2(IJ) + 1
IR2 = IR2 + 1
IK = INDEX(I,K)
COUNT 'LEFT DIAGONALLY' ADJACENT (135-DEGREE) NEAREST NEIGHBORS
FOR FIRST TWO COLUMNS.
LEX3(IK) = LEX3(IK) + 1
IR3 = IR3 + 1

NOW SHIFT ONE COLUMN TO THE RIGHT AND CONTINUE THE
PROCEDURE FOR GENERAL CASE IN WHICH A RESOLUTION CELL (I) HAS
ONE VERTICAL NEAREST NEIGHBOR (L), ONE HORIZONTAL (J), ONE LOW-
ER RIGHT DIAGONAL (K), AND ONE LOWER LEFT DIAGONAL (M). ITERATE
UP TO NEXT TO LAST COLUMN.

DO 2 N = 3,NUMPL
I = J
M = L
L = K
J = IDATA(N,IST) - LEAST1
K = IDATA(N,NN) - LEAST1
IL = INDEX(I,L)

COUNT VERTICALLY ADJACENT (90-DEGREE) NEAREST NEIGHBORS.
LEX1(IL) = LEX1(IL) + 1
IR1 = IR1 + 1

IJ = INDEX(I,J)

COUNT HORIZONTALLY ADJACENT (0-DEGREE) NEAREST NEIGHBORS.
LEX2(IJ) = LEX2(IJ) + 1
IR2 = IR2 + 1
IK = INDEX(I,K)

COUNT 'LEFT DIAGONALLY' ADJACENT (135-DEGREE) NEAREST NEIGHBORS.
LEX3(IK) = LEX3(IK) + 1
IR3 = IR3 + 1
IM = INDEX(I,M)

COUNT 'RIGHT DIAGONALLY' (45-DEGREE) ADJACENT NEAREST NEIGHBORS.
LEX4(IM) = LEX4(IM) + 1
IR4 = IR4 + 1

CONTINUE

MAKE COUNT FOR LAST COLUMN.

I = J
M = L
L = K

IL = INDEX(I,L)

COUNT VERTICALLY ADJACENT NEAREST NEIGHBORS FOR LAST COLUMN.
LEX1(IL) = LEX1(IL) + 1
IR1 = IR1 + 1
IM = INDEX(I,M)

COUNT 'RIGHT DIAGONALLY' ADJACENT NEAREST NEIGHBORS FOR THE LAST COLUMN.
LEX4(IM) = LEX4(IM) + 1
IR4 = IR4 + 1
INTERCHANGE THE LINE POINTERS.

MN = IST
IST = MNO
MNO = MN

1 CONTINUE

MAKE COUNT FOR LAST ROW.
I = IDATA(1,IST) - LEAST1
J = IDATA(2,IST) - LEAST1
IJ = INDEX(I,J)

COUNT HORIZONTALLY ADJACENT NEAREST NEIGHBORS FOR FIRST TWO COLUMNS OF LAST ROW.
LEX2(IJ) = LEX2(IJ) + 1
IR2 = IR2 + 1

COMPLETE COUNT FOR LAST ROW.
DO 12 N = 3,NUMPL2
I = J
J = IDATA(N,IST) - LEAST1
IJ = INDEX(I,J)

COUNT HORIZONTALLY ADJACENT NEAREST NEIGHBORS FOR REMAINDER OF LAST ROW.
LEX2(IJ) = LEX2(IJ) + 1
IR2 = IR2 + 1
12 CONTINUE
RESTART 02
111 CONTINUE

NOW DOUBLE THE DIAGONAL TO MAKE EVERYTHING COME OUT RIGHT

NOBL = I1 - I2 + 1
DO 100 I1 = 1,NOBL
II = INDEX(I1,II)
LEX1(II) = ? * LEX1(II)
LEX2(II) = ? * LEX2(II)
LEX3(II) = LEX3(II)*?
LEX4(II) = LFX4(II)*2
100 CONTINUE
IF (.NOT. MERGE) RETURN
C
C IF MERGE IS TRUE, SUM ALL APrAYS INTO LEX1
C
DO 112 I=1,NBUWL
LEX1(I)=LEX1(I)+LEX2(I)+LEX3(I)+LEX4(I)
112 CONTINUE
RETURN
END

769 WORDS OF MEMORY USED BY THIS COMPILATION

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
INDEX I-N-D-E-X
C
C    WRITTEN BY RMH
C    SEPT 1971
C
C    GIVEN THE ROW AND COLUMN SUBSCRIPTS I AND L, INDEX RETURNS
C    THE SINGLE SUBSCRIPT FOR THE LEX ARRAY INDICATING WHERE
C    ELEMENT (I,L) CAN BE FOUND.
C
1 2 3
4 5 6
7 8 9 10
C
FUNCTION INDEX(I,L)
INDEX1(I,L) = (I-1)*I/2 + L
IF(I.GT.L) GO TO 1
INDEX = INDEX1(L,I)
RETURN
1 INDEX = INDEX1(I,L)
RETURN
END

INDEX001
INDEX002
INDEX003
INDEX004
INDEX005
INDEX006
INDEX007
INDEX008
INDEX009
INDEX010
INDEX011
INDEX012
INDEX013
INDEX014
INDEX015
INDEX016
INDEX017
INDEX018
INDEX019
INDEX020
INDEX021
INDEX022
INDEX023
DESCRIPTION OF PROGRAM.

THIS PROGRAM CALCULATES THE MOMENT TEXTURE STATISTICS (AS DEFINED BELOW UNDER TEXTURAL FEATURES) FROM THE LEX ARRAYS, ACCORDING TO THE MERGE OPTION.

ENTRY POINT.

CALL IMOMTR(LEX1, LEX2, LEX3, LEX4, F, IQ, MERGE)

ARGUMENTS.

LEX1-LEX4
F
IQ
MERGE

ADDRESS INDEXS FOR LEX ARRAYS
CUMULATIVE DISTRIBUTION FUNCTION
QUANTIZED OUTPUT ARRAY OF IECQP01
OPTION TO MERGE THE FOUR LEX ARRAYS INTO ONE ARRAY

INTERNAL PARAMETERS.

ANGHOM...CORMAX
NQUANT
NQUANT2
IMAX
IMIN
IR1-IR4
R1-R4
QO
P(XY)
NOBL
NAD01-NAD04

NUMBER OF QUANTIZING LEVELS FOR IECQP01
MAXIMUM NUMBER OF GREY TONE LEVELS
MINIMUM NUMBER OF GREY TONE LEVELS
THE NUMBER OF RESOLUTION CELL PAIRS
INVERSE OF IR1-IR4
COUNTED IN EACH LEX ARRAY
SCRATCH ARRAY USED BY SUBROUTINE COR
ARRAY OF JOINT PROBABILITIES
NUMBER OF GREY TONE LEVELS
SUM OF ELEMENTS OF THE LEX ARRAY

TEXTURAL FEATURES.

ANGHOM= SUM SUM P(I,J)*P(I,J)
                           I J

AMEAN = SUM SUM I*P(I,J)
                        I J

AMEAN = SUM SUM J*P(I,J)
                        I J

SGMASQ = SUM SUM ((I-AMEAN)**2)*P(I,J)

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
\[ \text{SGMASQ} = \sum_{i,j} ((J - \text{AMEAN})^2 \cdot P(i,j)) \]
\[ \text{SGMAXY} = \sum_{i,j} (i - \text{AMEAN}) \cdot (j - \text{AMEAN}) \cdot P(i,j) \]
\[ \text{IVOMOM} = \sum_{i,j} (P(i,j)/(i+1 \cdot j)^2) \]
\[ \text{RATIO} = \text{SGMAXY} / \text{SGMASQ} \]

\[ \text{ENTROP} = -\sum_{i,j} P(i,j) \cdot \log(P(i,j)) \]

\[ \text{DIF(K)} = \sum_{i,j} P(i,j) \]
\[ \text{SUM(K)} = \sum_{i,j} P(i,j) \]
\[ \text{DIFENT} = \sum_{k} \text{DIF(K)} \cdot \log(D.F(K)) \cdot (-1) \]
\[ \text{DIFVAR} = \sum_{k} k \cdot \text{DIF(K)} \]

DIFVAR IS THE VARIANCE OF THE DISTRIBUTION DIF

NOTE THAT
\[ \text{DIFMOM} = 2 \cdot (\text{SGMASQ} - \text{SGMAXY}) \]

THE INTEGERS 1, 2, 3, 4 FOLLOWING THE VARIABLE NAMES CORRESPOND TO
THE FOUR ANGLES:

<table>
<thead>
<tr>
<th>SUBSCRIPT</th>
<th>LEX ARRAY</th>
<th>MOMENT ARRAY</th>
<th>ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>90 DEGREES</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0 DEGREES</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>135 DEGREES</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>45 DEGREES</td>
<td></td>
</tr>
</tbody>
</table>

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
IF MERGE IS .TRUE., THE FOUR LEX ARRAYS HAVE BEEN MERGED INTO LEX1.

SUBROUTINE IMOMT(LEX1, LEX2, LEX3, LEX4, F, IN, MERGE)

REAL IVDOMH
LOGICAL MERGE
DIMENSION LEX1(1), LEX2(1), LEX3(1), LEX4(1)
DIMENSION (64), I7(64)
DIMENSION DIF1(64), DIF2(64), DIF3(64), DIF4(64)
DIMENSION SUM1(128), SUM2(128), SUM3(128), SUM4(128)
DIMENSION P(3600), Q(300)
EQUIVALENCE (P(1), Q(1))
COMMON /Q/ NQUANT
COMMON M1, N1, TYPE, G(14)
COMMON MAX, MIN, NUMP+, NUML, NUBL, IR1, IR2, IR3, IR4, ANGMH(4)
COMMON AMEAN(4), SGMASQ(4), SGMAXY(4), DIFMOM(4), Ratio(4), IVDOMH(4), THEAN
COMMON /C/ ENTROPI(4), D. FENT(4), DIFAVE(4), DIFVAR(4), SUMENT(4)
COMMON /S/ SUMAVE(4), SUMVAR(4)
COMMON /CORREL/ CORINF(4), CORMUT(4), CORMAX(4)

INITIALIZE ARRAYS TO ZERO

DO 1 I = 1, 4
IVDOMH(I) = 0
ANGM(1) = 0
AMEAN(I) = 0
SGMASQ(I) = 0
SGMAXY(I) = 0
ENTROPI(I) = 0.0
DIFENT(I) = 0.0
DIFAVE(I) = 0.0
DIFVAR(I) = 0.0
SUMAVP(I) = 0.0
SUMVAR(I) = 0.0
1 RATIO(I) = 0
DO 86 K = 1, NQUANT
D.F1(K) = 0.0
DIF2(K) = 0.0
DIF3(K) = 0.0
D.F4(K) = 0.0
86 NQUANT2 = 2*NQUANT
DO 87 KS = 1, NQUANT2
SUM1(KS) = 0.0
SUM2(KS) = 0.0
SUM3(KS) = 0.0
SUM4(KS) = 0.0
87
GET THE NUMBER OF BRIGHTEST LEVELS, NOBL
IF THE LEX ARRAY WERE SQUARE AND NOT COMPACTED, IT WOULD BE
DIMENSIONED NOBL BY NOBL

NOBL = IMAX - IMIN + 1

NOW DETERMINE THE TOTAL NUMBER OF RESOLUTION CELL PAIRS
COUNTED IN EACH OF THE LEX ARRAYS

I31 = 0
I32 = 0
I33 = 0
I34 = 0
IF (.NOT. MERGE) GO TO 40
DO 42 I = 1, NOBL
DO 42 J = 1, NOBL
IJ = INDEX (I, J)
IR1 = IR1 + LFX1 (IJ)
42 CONTINUE
R1 = 1./FLOAT (IR1)
GO TO 41
40 DO 5 I = 1, NOBL
DO 5 J = 1, NOBL
IJ = INDEX (I, J)
IR1 = IR1 + LFX1 (IJ)
IR2 = IR2 + LFX2 (IJ)
IR3 = IR3 + LFX3 (IJ)
5 IR4 = IR4 + LFX4 (IJ)

GET R1, R2, R3, R4 TO SAVE DIVISIONS

R1 = 1./FLOAT (IR1)
R2 = 1./FLOAT (IR2)
R3 = 1./FLOAT (IR3)
R4 = 1./FLOAT (IR4)

FIND THE CORRELATION MEASURES
PUT THE LEX ARRAYS IN P MATRIX AND CALL CORRELATION ROUTINE

DO LEX2 ARRAY

JJ = 0
DO 201 I = 1, NOBL
DO 201 J = 1, NOBL
IJ = INDEX (I, J)
JJ = JJ + 1
201 P (JJ) = FLOAT (LEX2 (IJ)) * R1
CALL CO3 (P, NO9L + 1, NO, COR1, COR2, COR3)
COR1NF (1) = COR1
CORHUT(1)=COR2
CORMAX(1)=COR3
JJ=0

DO LEX4 ARRAY

DO 211 I=1,N0BL
DO 211 J=1,N0BL
IJ=INDEX(I,J)
JJ=JJ+1
211 P(JJ) = FLOAT(LEX4(I,J)) * R
CALL COR1(I,P,NOB1+1,Q0, COR1, COR2, COR3)
CORINF(2) = COR1
CORMUT(2) = COR2
CORMAX(2) = COR3
JJ=0

DO LEX1 ARRAY

DO 221 I=1,N0BL
DO 221 J=1,N0BL
IJ=INDEX(I,J)
JJ=JJ+1
221 P(JJ)=FLOAT(LEX1(I,J))*R1
CALL COR1(I,P,NOB1+1,Q0, COR1, COR2, COR3)
CORINF(3) = COR1
CORMUT(3) = COR2
CORMAX(3) = COR3
IF (MERGE) GO TO 43
JJ=0

DO LEX3, ARRAY

DO 231 I=1,N0BL
DO 231 J=1,N0BL
IJ=INDEX(I,J)
JJ=JJ+1
231 P(JJ)=FLOAT(LEX3(I,J))*R4
CALL COR1(I,P,NOB1+1,Q0, COR1, COR2, COR3)
CORINF(4) = COR1
CORMUT(4) = COR2
CORMAX(4) = COR3

GET THE PROBABILITY FUNCTION IN F

DO 379 I=1,N0BL
379 F(I)=0

IF MERGE, GO TO SECTION 41 TO MAKE THE COMPUTATIONS

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
C
I
F(MERGE) GO TO 911
DD 6 I=1,N0SL
IA=0
DO 7 J=1,N0SL
IJ=INDEX(' ','J)
7 IA=IA+LEX1(IJ)+LEX2(IJ)+LEX3(IJ)+LEX4(IJ)
6 F(I)=FLOAT(IA)/FLOAT(IR1+IR2+IR3+IR4)
C
C C
FIRST COMPUTE THE TRUE MEAN
C
TMEAN=O
D0 10 I=1,N0SL
10 TMEAN=TMEAN+F(I)*FLOAT(:)
TMEAN=TMEAN+FLO(T(0MIN-1)
C
C
GET CUMULATIVE DISTRIBUTION FUNCTION IN F
D0 8 I=2,N0SL
8 F(I)=F(I)+F(I-1)
C
C
DETERMINE THE QUANTIZING FUNCTION
CALL IEQPO1(N0SL,NQUANT,F,IQ,MIN)
C
C
NEXT COMPUTE THE QUANTIZED TRANSLATED MEAN FOR EACH ARRAY
D0 2 I=1,NQUANT
DO 2 J=1,NQUANT
NSI=1
IF(I.NE.1) NSI=IQ(I-1)+2-IMIN+1
NSJ=1
IF(J.NE.1) NSJ=IQ(J-1)+2-IMIN+1
IF(NSI.GT.NEI) GO TO 2
IF(NSJ.GT.NEJ) GO TO 2
NADO1=0
NADO2=0
NADO3=0
NADO4=0
DO 9 NJ=NSJ,NEJ
DO 9 NI=NSI,NEI
IJ=INDEX('I',NJ)
NADO1=NADO1+LEX1(IJ)
NADO2=NADO2+LEX2(IJ)
NADO3=NADO3+LEX3(IJ)
NADO4=NADO4+LEX4(IJ)
9 AMEAN(I)=AMEAN(I)+FLOAT(NADO2* I)
AMEAN(4)=AMEAN(4)+FLOAT(NADO3* I)
AHEAN(2)=AHEAN(2)+FLOAT(NADO4* I)
AHEAN(3)=AHEAN(3)+FLOAT(NADO1* I)
2 CONTINUE

NOW NORMALIZE TO GET THE MEANS
AHEAN(1)=AHEAN(1)*R2
AHEAN(2)=AHEAN(2)*P4
AHEAN(3)=AHEAN(3)*P1
AHEAN(4)=AHEAN(4)*R3

NOW DO MOMENT CALCULATIONS
DO 3 I=1,NQUANT
DO 3 J=1,NQUANT
NSI=1
IF(I.NE.1) NSI=IQ(I-1)+2-IMIN
NEI=IQ(I) -IMIN +1
NSJ=1
IF(J.NE.1) NSJ=IQ(J-1)+2 -IMIN
NEJ=IQ(J) -IMIN +1
IF(NSI.GT.NEI) GO TO 3
IF(NSJ.GT.NEJ) GO TO 3
NADO1=0
NADO2=0
NADO3=0
NADO4=0
DO 13 NI=NSI,NEI
DO 13 NJ=NSJ,NEJ
 NINJ=INDEX(NI,NJ)

SUM UP THE ELEMENTS IN EACH LEX ARRAY
NADO1=NADO1+LEX1(NINJ)
NADO2=NADO2+LEX2(NINJ)
NADO3=NADO3+LEX3(NINJ)
13 NADO4=NADO4+LEX4(NINJ)

NORMALIZE
RL1=FLOAT(NADO1)*R1
RL2=FLOAT(NADO2)*R2
RL3=FLOAT(NADO3)*R3
RL4=FLOAT(NADO4)*R4

CALCULATE THE MOMENTS
ANGHMOM(1)=ANGHMOM(1)+PL2**2
ANGHMOM(2)=ANGHMOM(2)+PL4**2
ANGHMOM(3)=ANGHMOM(3)+RL1**2
ANGMOM(4)=ANGMOM(4)+PL3**2
SGMASQ(1)=SGMASQ(1)+(FLOAT(1)-AMEAN(1))**2*PL2
SGMASQ(2)=SGMASQ(2)+(FLOAT(1)-AMEAN(2))**2*PL4
SGMASQ(3)=SGMASQ(3)+(FLOAT(1)-AMEAN(3))**2*PL1
SGMASQ(4)=SGMASQ(4)+(FLOAT(1)-AMEAN(4))**2*RL3
SGMAXY(1)=SGMAXY(1)+(FLOAT(1)-AMEAN(1))*PL2
SGMAXY(2)=SGMAXY(2)+(FLOAT(1)-AMEAN(2))*PL4
SGMAXY(3)=SGMAXY(3)+(FLOAT(1)-AMEAN(3))*RL1
SGMAXY(4)=SGMAXY(4)+(FLOAT(1)-AMEAN(4))*RL3
TVDOMOM(1)=TVDOMOM(1)+PL2/(1.+FLOAT();J)**2
TVDOMOM(2)=TVDOMOM(2)+PL4/(1.+FLOAT();J)**2
TVDOMOM(3)=TVDOMOM(3)+PL1/(1.+FLOAT();J)**2
TVDOMOM(4)=TVDOMOM(4)+RL3/(1.+FLOAT();J)**2
IF(RL2.LT.0.000001)GO TO 50
ENTROP(1)=ENTROP(1)-RL2*LOG10(RL2)
IF(RL4.LT.0.000001)GO TO 51
ENTROP(2)=ENTROP(2)-RL4*LOG10(RL4)
IF(RL1.LT.0.000001)GO TO 52
ENTROP(3)=ENTROP(3)-RL1*LOG10(RL1)
IF(RL3.LT.0.000001)GO TO 53
ENTROP(4)=ENTROP(4)-RL3*LOG10(RL3)
CONTINUE
C
C SET UP THE SUM ARRAY
C K=IABS(I-J)+1
C
C SET UP THE DIFFERENCE ARRAY
C KS=IABS(I+J)+1
DIF1(K)=DIF1(K)+RL2
DIF2(K)=DIF2(K)+RL4
DIF3(K)=DIF3(K)+RL1
DIF4(K)=DIF4(K)+RL3
SUM1(KS)=SUM1(KS)+RL2
SUM2(KS)=SUM2(KS)+RL4
SUM3(KS)=SUM3(KS)+RL1
SUM4(KS)=SUM4(KS)+RL3
CONTINUE
DO 4 I=1,4
RATIO(I)=SGMAXY(I)/SGMASQ(I)
4 CALCULATE THE ENTROPY,AVERAGE,AND THE VARIANCE OF THE DIFFERENCE ARRAY
DO 31 K=1,NOANT
IF(DIF1(K).LT.0.000001)GO TO 54
DIFENT(1)=DIFENT(1)-DIF1(K)*LOG10(DIF1(K))
IF(DIF2(K).LT.0.000001)GO TO 55
DIFENT(2)=DIFENT(2)-DIF2(K)*LOG10(DIF2(K))

C

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
DO 55 K=1,NOUAN2
IF(SUM1(K).LT.0.000001) GO TO 58
SUHENT(1)=SUHENT(1)-SUH(K)*ALOG(SUM1(K))
58 IF(SUM2(K).LT.0.000001) GO TO 59
SUHENT(2)=SUHENT(2)-SUH(K)*ALOG(SUM2(K))
59 IF(SUM3(K).LT.0.000001) GO TO 60
SUHENT(3)=SUHENT(3)-SUH(K)*ALOG(SUM3(K))
60 IF(SUM4(K).LT.0.000001) GO TO 61
SUHENT(4)=SUHENT(4)-SUH(K)*ALOG(SUM4(K))
61 CONTINUE
G=FLOAT(K)
SUMAVE(1)=SUMAVE(1)+(G*SUM1(K))
SUMAVE(2)=SUMAVE(2)+(G*SUM2(K))
SUMAVE(3)=SUMAVE(3)+(G*SUM3(K))
SUMAVE(4)=SUMAVE(4)+(G*SUM4(K))
SUMVAR(1)=SUMVAR(1)+(G*SUM1(K))
SUMVAR(2)=SUMVAR(2)+(G*SUM2(K))
SUMVAR(3)=SUMVAR(3)+(G*SUM3(K))
SUMVAR(4)=SUMVAR(4)+(G*SUM4(K))
DO 32 KK=1,4
32 SUMVAR(KK)=SUMVAR(KK)-(SUMAVE(KK)*SUMAVE(KK))
RETURN

SECTION 11 IMOMTR FOR THE MERGED LEX ARRAY

GET THE PROBABILITY FUNCTION IN F FOR MERGE OPTION

DO 16 I=1,NOBL
IA=0
FIRST COMPUTE THE TRUE MEAN

THEAN=0
DO 90 J=1,NOL
90 THEAN=THEAN+F(I)*FLOAT(I)
THEAN=THEAN+FLOAT(IMIN-1)

GET CUMULATIVE DISTRIBUTION FUNCTION IN F
DO 91 I=2,NOL
91 F(I)=F(I)+F(I-1)

DETERMINE THE QUANTIZING FUNCTION
CALL IEQPQI(NOL,NQUANT,F,IO,MIN)

NEXT COMPUTE THE QUANTIZED TRANSLATED MEAN
DO 92 I=1,NQUANT
92 J=1,NQUANT
NSI=1
IF(I.NE.1) NSI=IQ(I-1)+2-MIN
NEI=IQ(I)-MIN+1
NSJ=1
IF(J.NE.1) NSJ=IQ(J-1)+2-MIN
NEJ=IQ(J)-MIN+1
IF(NSI.GT.NEI) GO TO 92
IF(NSJ.GT.NEJ) GO TO 92
NADDI=0
DO 93 NI=NSI,NEI
93 NJ=NSJ,NEJ
IJ=INDEX(NI,NJ)
NADDI=NADDI+FLOAT(IJ)
AHEAN(3)=AHEAN(3)+FLOAT(NADDI*I)
CONTINUE

NOW normalize to get the MEANS
AHEAN(3)=AHEAN(3)*R1

NOW do MOMENT CALCULATIONS
DO 95 I=1,NQUANT
IF(I.NE.1) NSI=IQ(I-1)+2-IMIN
NEI=IQ(I) -IMIN +1
NSJ=1
IF(J.NE.1) NSJ=IQ(J-1)+2 -IMIN
NEJ=IQ(J) -IMIN +1
IF(NSI.GT.NEI) GO TO 95
IF(NSJ.GT.NEJ) GO TO 95
NADD=0
DO 96 NI=NSI,NEI
DO 96 NJ=NSJ,NEJ
NINJ=INDEX(NI,NJ)
C SUM UP THE ARRAY
96 NADD=NADD+EX1(NINJ)
C NORMALIZE
RL1=FLOAT(NADD1)*R1
C COMPUTE MOMENTS
ANGMOM(3)=ANGMOM(3)+R.1**2
SGMASQ(3)=SGMASQ(3)+((FLOAT(I)-AMEAN(3))**2)*RL1
SGMAXY(3)=SGMAXY(3)+((FLOAT(I)-AMEAN(3))*(FLOAT(J)-AMEAN(3)))*RL1
TVOMOM(3)=TVOMOM(3)+RL1/(1+FLOAT(I-J)**2)
IF(RL1.LT.0.000001)GO TO 533
ENTROP(3)=ENTROP(3)-RL1*ALOG(RL1)
CONTINUE
533 SET UP THE SUM ARRAY
K=IABS(I-J)+1
C SFT UP THE DIFFERENCE ARRAY
KS=IABS(I+J)+1
DIFF3(K)=0.F3(K)+RL1
SUM3(KS)=SUM3(KS)+RL1
95 CONTINUE
RATIO(3)=SGMAXY(3)/SGMASQ(3)
C CALCULATE THE ENTROPY, AVERAGE, AND THE VARIANCE OF THE DIFFERENCE ARRAY
C DO 97 K=1,NQUANT
IF(DIF3(K).LT.0.000001) GO TO 577
DIFFNT(3)=DIFFNT(3)-DIF3(K)*ALOG(DIF3(K))
577 CONTINUE
G=FLOAT(K)
DIFAVE(3) = DIFAVE(3) + (G*DIF3(K))
DIFVAR(3) = DIFVAR(3) + (G*G)*DIF3(K)
D.FVAR(3) = D.FVAR(3) - (DIFAVE(3)*DIFAVE(3))

CALCULATE THE ENTROPY, AVERAGE, AND THE VARIANCE OF THE SUM

ARRAY

DO 98 K = 1, NQUMP
   IF (SUM3(K), LT, 0, 000001) GO TO 99
   SUMENT(3) = SUMENT(3) - SUM3(K)*AOG(SUM3(K))
CONTINUE
   G = FLOAT(K)
   SUMAVE(3) = SUMAVE(3) + (G*SUM3(K))
   SUMVAR(3) = SUMVAR(3) + (G*G)*SUM3(K)
   SUMVAR(3) = SUMVAR(3) - (SUMAVE(3)*SUMAVE(3))
RETURN
END

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
SUBROUTINE COR

WRITTEN BY SAM SHANMUGAM

NOV 1972

OBJECTIVE.

This program calculates three measures of CORRELATION COR1, COR2, COR3, between two DISCRETE RANDOM VARIABLES X and Y whose JOINT PROBABILITIES of occurrence are STORED IN THE ARRAY PXY.

ENTRY POINT.

CALL COR(PXY, N, IOPT, Q, COR1, COR2, COR3).

INPUT ARGUMENTS.

PXY: ARRAY OF JOINT PROBABILITIES.
N: SIZE OF THE ARRAY PXY.
IOPT: OPTION FLAG--IF IOPT=0 THEN COR1 AND COR2 ONLY WILL BE COMPUTED. IF IOPT=1, THEN COR3 WILL BE COMPUTED.
Q: SCRATCH ARRAY OF SIZE N X 4. THIS ARRAY IS NEEDED ONLY IF IOPT IS NON-ZERO. IF IOPT IS ZERO THEN A DUMMY VARIABLE MAY BE SUBSTITUTED FOR THE ARGUMENT Q.

OUTPUT ARGUMENTS.

COR1: MAXIMAL CORRELATION MEASURE.
COR2: INFORMATION MEASURE OF CORRELATION.
COR3: SECOND TYPE OF MAXIMAL MEASURE.

BIBLIOGRAPHY.

ANNALS OF MATHEMATICAL STATISTICS, VOL. 43, 1962, COR0004.
P. 587. 'MUTUAL INFORMATION AND MAXIMAL CORRELATION AS MEASURE OF DEPENDENCE' by C.J. BELL.

CAUTION.
1. THE ARRAY PXY MUST HAVE A DIMENSION OF COR00052
N X N IN THE CALLING PROGRAM. OR PXY SHOULD BE A ONE DIMENSIONAL VECTOR, CONTAINING THE COR00053
JOINT PROBABILITIES IN A COLUMN BY COLUMN ARRANGEMENT.

2. IF N IS LARGE, THE COMPUTATIONS FOR COR00057
COR3 WILL TAKE CONSIDERABLE TIME. HENCE THE USE OF THIS ROUTINE IS RESTRICTED TO N LESS THAN OR EQUAL TO 32

COMPUTATIONS:

PX(I) = SUM PXY(I,J)
PY(J) = SUM PXY(I,J)
HXY = SUM SUM LOG(PXY(I,J)) PXY(I,J)
I J
HXY1 = SUM SUM LOG((PX(I)*PY(J)) PXY(I,J)
I J
HXY2 = SUM SUM LOG(PX(I)*PY(J)) PX(I)
I J
HX = SUM (LOG(PX(I)) PX(I)
I
HY = SUM (LOG(PY(J)) PY(J)
J
R = HXY2 - HXY, EMAX = MAX(HX, HY)
COR1 = (HXY - HXY1)/EMAX
COR2 = SQRT(1.0 - EXP(-2.0*EMAX))
COR3 = IS COMPUTED USING THE EIGENVECTOR CORRESPONDING TO THE SECOND LARGEST EIGEN VALUE OF Q*Q^T.
WHERE Q(I,J) = PXY(I,J)/SQRT(PX(I)*PY(J))

** ***

DIMENSION PXY(1),Q(1)
DIMENSION PX(64),PY(64),E(64),V(128),B(64),C(64),D(64),F(54)
DIMENSION IZERO(32)
***

DO 80 I=1,N
PY(I)=0.0
80  DO 80 I=1, N
PY(I)=0.0
SUBROUTINE OR

HX=0.0
HY=0.0
HXY=0.0
HXY1=0.0
HXY2=0.0

COMPUTE THE MARGINALS AND THEIR ENTROPY

DO 82 I=1,N
DO 81 J=1,N
IJ=(J-1)*N+I
81 PX(I)=PX(I)*PXY(IJ)
IF (PX(I) .LT. 0.0000001) GO TO 82
HX=HX-(ALOG(PX(I)))*PX(I))
82 CONTINUE
DO 84 J=1,N
DO 83 I=1,N
IJ=(J-1)*N+I
83 PY(J)=PY(J)*PXY(IJ)
IF (PY(J) .LT. 0.0000001) GO TO 84
HY=HY-(ALOG(PY(J)))*PY(J)
84 CONTINUE

COMPUTE THE ENTROPY OF THE JOINT DISTRIBUTION

DO 69 I=1,N
DO 68 J=1,N
IJ=(J-1)*N+I
IF (PXY(IJ) .LT. 0.0000001) GO TO 68
HXY=HXY-(ALOG(PXY(IJ)))*PXY(IJ)
PXPy=PXY(IJ)*PY(J)
IF (PXPy .LT. 0.0000001) GO TO 69
HXY1=HXY1-(ALOG(PXPy))*PXY(IJ)
HXY2=HXY2-(ALOG(PXPy))*PXY(IJ)
68 CONTINUE

COMPUTE COR1 AND COR2

EMAX=HX
IF (HX .LT. HY) EMAX=HY
COR1=(HXY-HXY1)/EMAX
R=HXY2-HXY
COR2 = SQRT(1.0-EXP(-2.0*R))

IF COR3 NOT ASKED FOR RETURN

SCAN PXPy AND DELETE ROWS OF ZEROS
AND COLUMNS OF ZEROS

DO 599 INDX=1,32
02-12-74 20:060 SUBROUTINE FOR

599 IZERO(NDFX)=0
NZERO=0
DO 600 I=1,N
I=1
IF (PX(I).GT.0.000001) GO TO 601
NZERO=NZERO+1
IERO(NZERO)=II
601 CONTINUE
600 CONTINUE
IF (NZERO.EQ.0) GO TO 651
JJ=0
DO 650 J=1,N
DO 650 I=1,N
DO 640 KK=1,NZERO
NDEX=ZERO(KK)
IF ((I.EQ.NDEX).OR.(J.EQ.NDEX)) GO TO 649
649 CONTINUE
JJ=JJ+1
IJ=(J-1)*N+I
PX(IJ)=PX(IJ)
649 CONTINUE
650 CONTINUE
651 CONTINUE
C

REMOVE ZERO ENTRIES IN THE MARGINALS

C

JJ=0
DO 661 I=1,N
IF (PX(I).LT.0.000001) GO TO 662
JJ=JJ+1
PX(IJJ)=PX(IJJ)
662 CONTINUE
661 CONTINUE

NORMALIZE PXI AND STORE IN Q. SAVE PXI

C

C

C

DO 58 I=1,N
DO 58 J=1,N
IJ=(J-1)*N+I
CON=SQRT(PY(IJJ))
Q(IJ)=PX(IJJ)/CONS
COMPUTE THE UPPER/DIAG ELEMENTS OF Q*QT
STORE IN Q

DO 49 I=1,N
DO 51 J=I,N
B(IJ)=0.0
DO 52 K=1,N
IK=(K-1)*N+I
JK=(K-1)*N+J
Q(JK)=B(JK)+Q(IK)*Q(IJK)
CONTINUE
DO 50 J=1,N
IJ=(J-1)*N+I
50 Q(IJ)=B(IJ)
CONTINUE

FILL IN THE BELOW DIAG ELEMENTS OF Q*QT

DO 48 J=1,N
DO 49 I=J,N
IJ=(I-1)*N+I
J=(I-1)*N+J
48 Q(IJ)=Q(IJ)
CONTINUE

FORM SQRT(PX(I)) * Q*QT * SQRT(PY)
STORE IN Q

DO 91 I=1,N
DO 92 J=1,N
IJ=(J-1)*N+I
J=(J-1)*N+J
91 Q(IJ)=Q(IJ)/SQRT(PX(I)*PX(J))
CONTINUE

GET THE EIGEN VECTORS AND EIGEN VALUES
OF Q*QT

CALL THE SUBROUTINE TO GET THE EIGENVALUES.
GET A MAX OF 5 EIGENVALUES. IF ALL FIVE
ARE NEAR UNITY, SET COR3=9.9999, RETURN.
IF ALL OF THEM (OTHER THAN THE FIRST ONE)
ARE LESS THAN 0.001, SET COR3=0.0001 AND
RETURN. THE EIGEN VALUES ARE CALCULATED
WITH AN ACCURACY OF 0.0001.

MAX=5
IF(N.LT.MAX) MAX=N
CR=0.0001
EPS=0.00001
CALL SFA02D(Q,N,N,CR,EPS,MAX,NE,EV,B,C,O,F,E)
SUBROUTINE COP

C IF(N.E.GT.1) GO TO 60
C SET COR3=0.0001 AND RETURN
C COR3=.01
C WRITE(6,200)
200 FORMAT(1H1,1.10X,* ALL EIGEN VALUES OTHER THAN THE FIRST ARE LESS THAN COR0025)
1AN 0.001, COR3 IS SET=0.0001/10X,*ABORT AA*)
N=NNNN RETURN
C CONTINUE
C FIND THE EIGEN VALUE CLOSEST TO 1.0
C DO 63 I=1,NE
C DIF=ABS(E(I)-1.0)
C JJE=I
C IF(DIF.GT.0.00003) GO TO 64
C CONTINUE
C IF PROGM IS AT THIS POINT, THEN THE FIRST 5 EIGEN VALUES ARE CLOSE TO 1.0. SET COR3=0.9999 AND return
C COR3=0.9999
C WRITE(6,202)
202 FORMAT(1H1,1.10X,* THE FIRST 5 EIGEN VALUES ARE NEAR UNITY. */10X,* COR0025)
1 COR0025 IS SET =0.999.....ABORT BB*)
N=NNNN RETURN
CFOUND A PROPER EIGEN VALUE
C IFOUND=JJE
C COR3=SORT(E(IFOUND))
C N=NNNN RETURN END

02-12-74 20.860
I-E-Q-P-O-1

I-F-O-P-O-1

written by denish goel

September 1971

description of program:

this subroutine determines k levels of quantizing for an array

for which the cumulative distribution function of all the el-

ments have already been obtained.

entry point:

call ieopq1(n,k,f,iq,imin)

arguments:

n number of items to be quantized, the

dimension of the f array.

k the number of quantizing levels.

f input array of cumulative distribution.

iq output array of quantizing levels.

imin the lowest possible level in

the input data.

subroutine ieopq21(n,k,f,iq,imin)

dimension f(1),iq(1)

dif=10.**6

obtain the first quantizing level.

go thru the whole array of c.d.f.'s

do 1 j=1,n

find percentage of distribution for first quantizing level and

check for the nearest c.j.f.

x=a0s(1./float(k)-f(j))

if(dif.le.x)go to 1

dif=x

isave=j

1 continue

first quantizing level

in(1)=isave+imin-1

to go for next level
LFTOFF=ISAVE
DO 2 I=2,K
   DECIDE FOR OTHER QUANTIZING LEVELS IN THE SIMILAR WAY.
   DIF=10.**6
   DO 3 J=LFTOFF,N
      THE PERCENTAGE OF DISTRIBUTION FOR NEXT QUANTIZING LEVEL WILL
      BE DECIDED AMONG REST OF ELEMENTS.
      X=ABS(((1.-F(LFTOFF))/FLOAT(K-I+1))+F(LFTOFF)-F(J))
      IF(DIF.LE.X)GO TO 3
      DIF=X
      ISAVE=J
   3 CONTINUE
   IQ(I)=ISAVE+IMIN-1
   LFTOFF=ISAVE
2 CONTINUE
RETURN
END
SUBROUTINE RITOWT(LEX1, LEX2, LEX3, LEX4, G, IQ, MERGE, IF, PICTUR)

DIMENSION LEX1(1), LEX2(1), LEX3(1), LEX4(1), G(64), IQ(64), 3(4)
COMMON M1, N1, TYPEF(14), JO(9), ANGMOM(4), AMFAN(4), SGMAQ(4),
1 SGMAXY(4), DIFMON(4), RATIO(4), VIDMON(4), THEAN, LEAST1, NRED
COMMON NAYFR, NSTART, NMES, ND, PNCH
COMMON CE, ENTROP(4), CIFENT(4), Q, FAVE(4), CIFVAR(4), SUMENT(4),
1 SUMAVE(4), SUMVAR(4)
"

--- --....---- - ._--- ---~"" ---- ---- ----...---'.....- ..... ---.
.' -..--.--..--

~ ...

-.~,~..... : ..

--

.

, ;

02-12 -74

JUtlE 117 J

R-!- T-O-I" - r

20.71. 3

~!TOWrrH

COt1tiON l.;nR~EL/CO~INF (I.~ ,':'OP.MIJT C4' ,CORMAXC4'
LOGICAl. MEPGE ,PiCTlJ R
OAT A 8 I 1. , ,£3 ( 2' ,[3 ( :; I , [! (4 ) I 0 • , I~ 5 • ,9 0 • , t 35 .1
DATA TAPF. /lHll
DATA Y/lHY I
DATA KnU~T/O/.iHl/OI

RrlOW T?2
RiTaW T53
Rl lOIH sr.
RITOWT5t;
R! TOIHr; 6
RI TOIHS 7
R~ TOIHSI I

NFl=NRE)··N~AYE~

N=4
K!(J=S O
H=3

c
c
c
c
c
c
c
c
c

I

!
I
.I
I
i

CHECK

RlToWT'i~

FO~

:FlHER GE)

THE MERGE OPTIO N
GO TO 22

RnOW T60
RiTOI H6t
RI TOIH6 2
RiTOW T63
R:: TOWT61.
RITOI H65
R~ TOWT or,
R!TOWTf,7
Ri.:TO~Tfj 8
RITOr/ Too
RITOW T70
RI TOWT71
RITOI H72
Rl TO,H7 3
RITOH l 74
R! TO',H7~
RITOW Tn
IUTOW T77
R~ TOWT7P
RITO\H 7C.
RITOIH Be
RITOIH 31
RITOWT8Z

REPRODUOIBILITY OF THE
ORIGINAL ~AGE IS POOR

PUNCH OR NO PUNCH
IFCPNC~.HE.Y)

GO TO 400

PUN~H TEXTURE

106

~EAT~RES FOR

AL~ FOUR ANGLES

Hl.Nl, NFT
WR!TE(4~,60)
FO~~AT(lX,2A?,l2HIS COMP~ETED)
KOUNT :KOUN T+l

t'.KOUNT

ENT~OP'K),K=1,l
W~I.TE'43,600 )CANGMOH(KI,K=1,4,,(
+1
T
KOUNT ::KOU~J

(

,4) ,KOUNT
WRITE 143,60 0) (~AT~O IK) ,K=1, 4), (SGMA SQIK) ,K=1
+l
KOUNT =KOlltH
4),KOU NT
WRIT EI43,6 00) ISGMA XYIK) ,K=1,4 ),CAH EAN(K ),K=1,
KOUNT=KOUNT+l
OUNT
WRITE I43,oO O) (VIOH OMIK ),K=1, 4),(TM EAN,K =1,4),K
KOUNT=KOUNTH
,4) ,KOUNT
WRITE (43,60 0' (DIFEN T IK', K=1t 4', (OIFAV E CK) ,K=l
KOUNT=KO!JNT+l
,4) ,~OUtIT
WP-ITE 14~,600' ()IFVA RIK' ,K=1, 4', (SUHENT (K) ,K=t
KOUNT=KOlJtH+ 1
=1,4',K OUNT
WP.!T E(43,6 00) ISUMA VECK ),K=1, 4),(SU MVAR CK),K
KOUNT=KOUNT+l
t) .KOUNT
WRIT E(43,6 00) (l"ORINF'CK) ,K=l, 4', CCO~HUTCK) .K=l,l
KnU~IT=Kn!JIH+ 1
WRITE 143, 6nll CORMAX (K) ,K=1 ,4) ,KOUNT
600 FORM AT(lX ,8FQ.5 .17)
601 FOR~AT(tX,4Fq.4.3RX,IS)

on1

c
c

WRITE TEXTURE

~EATUR(S

R!TO~lT

'IF'

GO TO sao
1, N) , 1 ~A TI 0
n t-!1, N1 • NF 'r , CANt; MOti CK' , K= t • N) , (E NT ~ or CK) , K=

IFCPNr.H.Nr..T~PE)

209

8'

IUTOW T8'
R: Tm/T')
R:TOW Tg'
RI TOWTq'
RITOWTg
~I rOWT'):
~lTOWT

TO TAPE FILE

!H

RITOH TS,

FORMATllX,4Fq.S~36X~I7~

c
1

,~nOWT

400 CONTINUE

WR I TE n

,Ji,

RITOIH~::

R:TOIH ClI
RITOH T":

n<) ,

g

R'I TOHT q
RiTOW Tg
Rl.TOWTQ
RITOWTQ
RITOWlO

I

I

II

I


1K=1,N, (SMASQ(K), K=1,N) (SMAXY(K), K=1,N) (AMean(K), K=1,N)
1(VDIMOM(K), K=1,N) (THFAN(K), K=1,N) (DIFENT(K), K=1,N)
1(IFAVEF(K), K=1,N) (DIFVAR(K), K=1,N) (SUMFNT(K), K=1,N)
1(SUMAVE(K), K=1,N) (SUMVAR(K), K=1,N) (CORINF(K), K=1,N) (CORMUT(K), K=1,N) 500 CONTINUE
C
C PRINT TEXTURE FEATURES FOR EACH ANGLE AND TITLE
C
60 FORMAT(' THE SCENE ("I2,""I2,"") HAS BEEN REDUCED BY ",I5) WRITE(6,303)
303 FORMAT(6h ANGLE, 6H ANGmom, 6H ENTROP, 6H RATIO, 6H SGASO, RITOWT13
18H SGAXY, 3H IVDOM, 3H OFFFFT, 3H DIFAVE, 3H DIFVAR, 3H SUMENT, RITOWT14
18H SUMAVE, 3H SUMVAR, 3H CORINF, 3H CORMUT, 6H CORMAX ) WRITE(6,300) (1K), ANGmom(K), ENTROP(K), RATIO(K), SGASO(K), SGAXY(K) RITOWT1A
1, VDIMOM(K), DIFENT(K), DIFAVE(K), DIFVAR(K), SUMENT(K), SUMAVE(K), RITOWT17
?SUMVAR(K), CORINF(K), CORMUT(K), CORMAX(K), K=1,N)
300 FORMAT(1X,F5.1,15F9.4) WRITE(6,600) TMEAN
100 CONTINUE
C
C IF NEITHER PNC4 NOR TAPE, PRINT LEX ARRAYS
C IF((PNC4.EQ.Y) .OR. (PNC4.EQ.TYPE)) RETURN
WRITE(6,31) IN
31 FORMAT(3H IQ/(1X,16I7))
NOB = I (I+1) / 2 + 1
C
C IF MERGE, JUST DO LEX1 AND RETURN
C IF(MERGE) GO TO 54
WRITE(6,566)
566 FORMAT(/10X,9H0 DEGREES) DO 50 I=1,NOB
NS = *(I-1)/2 + 1 NE = *(I+1)*7/2
50 WRITE(6,700) (L= 1X2(J), J=NS, NE)
700 FORMAT(1X,265) WRITE(6,567)
567 FORMAT(10X,10H45 DEGREES) DO 51 I=1,NOB
NS = *(I-1)/2 + 1 NE = *(I+1)*7/2
C
C PRINT LEX2 FOR 0 DEGREES
C
50 WRITE(6,700) (L= 1X2(J), J=NS, NE)
700 FORMAT(1X,265) WRITE(6,567)
567 FORMAT(10X,10H45 DEGREES) DO 51 I=1,NOB
NS = *(I-1)/2 + 1 NE = *(I+1)*7/2
C
C
PRINT LEX4 FOR 45 DEGREES

51 WRITE(6,700) (LEX4(J), J=NS,NE)
99 FORMAT(I+1)
54 WRITE(6,568)
568 FORMAT(/10X,10490 DEGREES)
DO 52 L=1,NOBL
   NS=I*(I-1)/2+1
   NF=(I+1)*I/2
52 WRITE(6,700) (LEX4(J), J=NS,NE)
IF (MERGE) RETURN
WRITE(6,569)
569 FORMAT(/10X,114135 DEGREES)
DO 53 I=1,NOBL
   NS=I*(I-1)/2+1
   NF=(I+1)*I/2
PRINT LEX1 FOR 90 DEGREES

52 WRITE(6,700) (LEX1(J), J=NS,NE)
IF (MERGE) RETURN
WRITE(6,569)
569 FORMAT(/10X,114135 DEGREES)
DO 53 I=1,NOBL
   NS=I*(I-1)/2+1
   NF=(I+1)*I/2
PRINT LEX3 FOR 135 DEGREES

53 WRITE(6,700) (LEX3(J), J=NS,NE)
RETURN

RITOWT FOR THE MERGE OPTION
CHECK TO SEE IF A PICTURE HAS BEEN PRINTED

IF (PICTURE) GO TO 23
INCREMENT PAGE COUNT
IMT=IMT+1
IF PAGE IS FULL GO TO TOP OF NEXT PAGE AND WRITE TITLE

IF (IMT.E.1) WRITE(6,662)
IF (IMT.GE.14) IMT=0
662 FORMAT(1X,40X,*ERTS TEXTURE ANALYSIS/*
  1X,*ANGLE, ANGMOH ENTROP, RATIO, SGMAS2, SGMAS1, IVDMA, RITOWT9
1M DIFVAVG DIFVAR SUMENT SUMAVE SUMVAR COPINF CORMUT RITOWT9
1 CORMAX */
CHECK FOR PUNCH

IF (PUNCH,NE,Y) GO TO 40
PUNCH THE MERGED TEXTURE FEATURES
C
WRITE(43,663) M1,N1,NPT,ANGMOM(M),ENTROP(M),RATIO(M),SOGMASQ(M),
1 SGMAXY(M),AHEAN(M),VROMOM(M),KOUNT
KOUNT=KOUNT+1
WRITE(43,664) THEAN,DIFENT(M),DIFAVE(M),DIFVAR(M),SMENT(M),
1 SUMAVE(M),SUMVAR(M),KOUNT
KOUNT=KOUNT+1
WRITE(47,665) CORINF(M),CORMUT(M),CORMAX(M),KOUNT
KOUNT=KOUNT+1
663 FORMAT(12,1X,I7,I2,1X,7F9.5,9)
664 FORMAT(3X,7F9.5,9)
665 FORMAT(3X,3F9.5,36X,19)
40 CONTINUE
C CHECK FOR TAPE OUTPUT
IF(PNCH.NE.TAPE) GO TO 41
C WRITE OUT ON FILE 'IF' THE MERGED TEXTURE FEATURES
WRITE(IF) M1,N1,NPT,RATIO(M),SOGMASQ(M),
1 SGMAXY(M),AHEAN(M),VROMOM(M),THEAN,DIFENT(M),DIFAVE(M),
2 DIFVAR(M),SMENT(M),SUMAVE(M),SUMVAR(M),CORINF(M),CORMUT(M),
3 CORMAX(M)
C IN ANY CASE, PRINT THE MERGED TEXTURE FEATURES
WRITE(6,60) M1,N1,NPT
WRITE(6,666) ANGMOM(M),ENTROP(M),RATIO(M),SOGMASQ(M),SGMAXY(M)
1, VROMOM(M),DIFENT(M),DIFAVE(M),DIFVAR(M),SMENT(M),SUMAVE(M),
2 SUMVAR(M),CORINF(M),CORMUT(M),CORMAX(M)
666 FORMAT(1X,'MERGE',1SF8.4)
667 FORMAT(1X,'MERGE',1SF9.5)
C NOW GO PRINT OUT THE MERGED .EX1 ARRAY AND RETURN
GO TO 100
C END

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
IV.2-c Cross-Band Texture Analysis Program Listings

SPECTR
GETIM / GETIT
ERTS (see IV.2-a)
DIFFER
COVAR
MNCVIN / MNCV
CORREL
DESCRIPTION OF PROGRAM

THIS PROGRAM IS THE MAINLINE OF PROGRAMS WHICH OBTAIN A
NUMLIN X NUMPPL X NDIM SUBIMAGE FROM THE ERTS INPUT TAPE AND FOR
EACH SUBIMAGE CALCULATES THE COVARIANCE MATRIX AND THE
CORRELLATION MATRIX. THIS MATRIX IS WRITTEN TO FILE IFIL FOR
FURTHER ANALYSIS.

THE ERTS TAPE IS PROCESSED IN HORIZONTAL ROWS OF SUBIMAGES.
SUBIMAGES MAY OVERLAP HORIZONTALLY AND VERTICALLY, AND THE
DISTANCE BETWEEN NEIGHBORING CELLS USED IN THE DIFFERENCE ARRAY
IS VARIABLE.

NOTE---

ERTS INPUT TAPE MUST BE ON FILE ES.

INTERNAL PARAMETERS

NUMBER OF GREY TONE N-TUPLE COMPONENT.
NUMBER OF LINES IN A SUBIMAGE
NUMBER OF COLUMNS IN A SUBIMAGE
ROW COORD FOR THE SUBIMAGE
COLUMN COORD FOR THE SUBIMAGE
DISTANCE BETWEEN NEIGHBORING RASTER CELLS FOR THE DIFFERENCE IMAGE
STARTING ROW FOR THIS RUN
STOPPING ROW FOR THIS RUN
SET=0, THE STRIP WILL BE PROCESSED TO ITS END OF FILE
HORIZONTAL OVERLAP OF SUBIMAGES
VERTICAL OVERLAP OF SUBIMAGES
NUMBER OF OVERLAPPING HORIZONTAL SUBIMAGES IN A ROW
FINAL ROW OF SUBIMAGES
CORRELATION COVARIANCE MATRICES
TITLE FOR THE MATRIX
TITLE FOR THE MATRIX
FORMAT FOR PRINTING OUT MATRIX TERMS
TRUE TO PRINT OUT COVARIANCE MATRIX
DETERMINES OUTPUT FILE FOR FEATURES
TRUE FOR FILE 43, PUNCHED BARS
FALSE FOR FILE 01, TAPE OR DISC
OUTPUT FILE FOR FEATURES
ROW AND COLUMN NAMES FOR MATRIX PRINTOUT
DETERMINANT OF THE CORRELATION MATRIX
ENTROPY MEASURE
CARD COUNTER

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
INPUT ARGUMENTS:
IMAGE,X 
ILINE 
YDIM 
IYDIM 
NDIM 
ENTRY POINT:
CALL SPECTR(Image,X,ILINE,IXDIM,IYDIM,NDIM)
EXAMPLE OF DRIVER:
DIMENSION IMAGE(16,17,4),X(16,17,4),ILINE(330))
EQUIVALENCE (IMAGE(1,1,1),X(1,1,1),ILINE(130))
XDIM=17 
YDIM=16 
NDIM=4 
CALL SPECTR(Image,X,ILINE,IXDIM,IYDIM,NDIM)
STOP
END
THISS PROGRAM WILL SET UP THE TEXTURE RUN FOR 16 X 16 SUB-IMAGES
OVER ALL FOUR BANDS, WITH IDIST=1. ***NOTE*** IYDIM MUST INCLUDE
NUMPLU PLUS IDIST, AND ARRAY ILINE MUST HAVE AT LEAST NUMPLU*NDIM
POINTS OUTSIDE OF ANY OTHER ARRAY. THESE POINTS FORM ARRAY XLINE
WHICH IS USED IN COVAR TO SEND ONE LINE OF DATA TO MNQV.
SUBPROGRAMS REQUIRED.
DRIVER
SPECTR
GETIM
SENDIM
GETIT
ERTS (WITH EREWM)
DIFER
COVAR
MNQVIN
MNQV
CORREL
SFA07F
HEMDET
SUBROUTINE SPECTR(Image,X,ILINE,IXDIM,IYDIM,NDIM)
DIMENSION IMAGE(IYDIM,IXDIM,NDIM),ILINE(630),TITLE(14),ARR(5).
1 LABEL(8),OR(8,8),JOV(8,8),X(IYDIM,IXDIM,NDIM)
CHARACTER ROW*12,COLO*12,TTL*6(14)
EQUIVALENCE (ROW,TTL(13)),COL,TTL(11))
LOGICAL OPT,PNC
NAMELIST /PARAM/ NDIM,NOMLIN,NUMPLU,FMT,TITLE,OPT,IIIDIST,IRSTR,
1 IRSTOP,LAPHR,LAPVER,PNC
DATA TTL(1)/'COVARIANCE OVER SUBIMAGES '/TTL(6)/'CORRELATION

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
CROSS-BAND TEXTURE ANALYSIS

10 OVER SUBIMAGE --- */
DATA LABEL(1)/*BAND 2 2X2 BAND 3 3X3 2X3 BAND 4 4X4 2X4 */
DATA FORMAT/*E11.4*/, TITLE(1)/0.0*/, .FM/"F9.5/"

***** SECTION I --- SET UP PARAMETERS FOR PROCESSING *****

OPT=.FALSE.
PNCH=.TRUE.
IDIST=1
ISTRT=1
ISTOP=0
LAPHR=0
LAPVE=0
IXOIM=IXOIN
IYOIM=IYOIN
NOIM=NOIN
MCN=MDIN
NIMINOUMLIN
NUMPPL=NUMLIN
FMT=FORMAT
KT=0

READ(5,PAPAM)
WRITE OUT PARAMETERS

READ IN PARAMETERS

WRITE(6,PAPAM)
WRITE OUT PARAMETERS

FORMAT(20X,'SPECTRAL-TEXTURAL ANALYSIS PROGRAM')
IF(INSTOP.EQ.0) IRSTOP=999999
WRITE(6,2) NUMPPL, NUMLIN, NOIM, FMT, IDIST, LAPHR, LAPVE, ISTRT, ISTOP
FORMAT(1X, 'NUMBER OF COLUMNS IN SUBIMAGE IS ',13/
1 'NUMBER OF BANDS IN SUBIMAGE IS ',13/
1 'FORMAT USED TO OUTPUT MATRICES IS ',A5/
1 'DISTANCE BETWEEN CELLS FOR DIFFERENCE IMAGE IS ',I2/
1 'HORIZONTAL SUBIMAGE OVERLAP IS ',I3, ' POINTS'/
1 'VERTICAL SUBIMAGE OVERLAP IS ',I3, ' POINTS'/
1 'STARTING ROW IN STRIP IS ',I6/
1 'FINAL ROW IN STRIP IS ',I6)

IF (.NOT.OPT) WRITE(6,11)

FORMAT('ONLY THE CORRELATION MATRICES WILL BE LISTED')
IF (NUMPPL.LE.IXOIM) GO TO 5
WRITE(6,4)
FORMAT('ERROR -- NUMBER OF POINTS PER LINE EXCEEDS ',I4)
STOP
IF (NOIM.LE.NOIN) GO TO 7
WRITE(6,6) NOIN
FORMAT('ERROR -- NUMBER OF BANDS, NOIM, MUST NOT EXCEED ',I3)
STOP
K9=IXOIM-NUMPPL
IF (IDIST.LE.K9) GO TO 12
WRITE(6,32) K9
32 FORMAT(* ERROR---IDIST MUST NOT EXCEED *",I4) STOP
12 IF (L Aphor,Lt,NUMPPL) GO TO 15
13 WRITE(6,14) STOP
14 FORMAT(* FATAL ERROR---OVERLAP EXCEEDS SIZE OF SUBIMAGE")
15 IF(LAPVEP,GE,NUMLIN) GO TO 13
16 IF(NUMLIN.LE.IYDIM) GO TO 17
17 WRITE(6,16) IYDIM
18 FORMAT(* NUMLIN EXCEEDS*",J4* LINES---EXECUTION TERMINATED") STOP

***** SECTION II --- PROCESS THE SUBIMAGES *****

17 CALL GETIM(LINE,IDIST,NDIM,IRSTR,ISTRTMP,NUMLIN,NUMPPL,LAPVER,
17   LAPHOR,NHOR,INCR,PEND)
17 CALL SETDIM(COV,NHOR,NDIM) SET THE OUTPUT FILE FOR TAPE OR PUNCH
17 KDIM=NDIM-1 IF=01
17 IF (PEND) IF=43
17 NROW=IRSTR/(NUMLIN-LAPVER) FIND THE ROW COORD AND THE LAST ROW
17 LASTIM=((IRSTRP-IRSTR+1)/(NUMLIN-LAPVER)+1.
17 KPE=NUMPPL+IDIST SET UP FOR TITLE OF MATRICES
18 DO 100 M1=1,LASTIM PUT THE COORDINATE INTO HOLLERITH
18 NROW=NROW+1 LITERAL FOR MATRIX PRINTING
18 ENCODE (ROW,8) NROW
18 FORMAT(16,6X) GET A ROW OF HORIZ OVERLAPPING
18 CALL GETIT SUBIMAGES FROM ERTS AND PUT THEM ON
18 GET 90 NCO=1,NHOR
18 KS=INCR DISC FILES 11,12,13,14 FOR 4 BANDS
18 DO 90 KPE=1,NKPE PULL OFF EACH SUBIMAGE, GOING ACROSS
18 DO 90 NCO=1,NHOR THE ENTIRE ROW
18 DO 90 NCO=1,NHOR SET START AND END PTS FOR SUBIMAGE
18 KPE=KS+INCR GO THRU NUMLIN LINES
18 DO 90 50 LLN=1,NUMLIN READ A LINE OF THE SUBIMAGE
18 JP=IPEND*(LLN-1)
18 DO 90 KOL=1,KPE READ IN THE RESOLUTION CELL
18 IP=JP+KS+KOL

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
CROSS-BAND FEATURE ANALYSIS

READ(11,IP) (IMAGE(LMN,KOL,19),IB=1,NDIM)

CONTINUE

CALL DIFFER(IMAGE,X,IXDIM,IXDIM,NOIM,DIIS,NUPPL,NUMLIV)
GET THE DIFFERENCE IMAGE

ENCODE (COL,8) NCOL
PUT THE COORDINATE INTO HOLERITH

CALL COVAR(ILINF,NDIM,NUMPL,X,IXDIM,IXDIM,NUMLIN,NDIM,COV)
GET THE COVARIANCE MATRIX FOR THIS

IF(OPT) CALL SFA07F(COV,NOIM,NDIM,NOIM,NDIM,2,1,2,FMT,TITLE,TTL(1),
SUBIMAGE

1 TTL(11),TTL(13),LABEL,LABE)
WRITE OUT THE COVARIANCE MATRIX

CALL CORREL(COV,NDIM,COR)
GET THE CORRELATION MATRIX

CALL SFA07F(COR,NOIM,NOIM,NDIM,NDIM,2,1,2,FMT,TITLE,TTL(6),TTL(11),
WHOSE COORDINATES ARE (NCOL,NROW)

1 TTL(13),LABEL,LABE)

WRITE OUT THE CORRELATION MATRIX

DO 65 IS=1,NDIM

DO 65 JS=1,NDIM

COV(IS,JS) = COR(IS,JS)

CON TINUE

GET THE DETERMINANT FOR THE

CALL HEMDFT(COV,NOIM,DET)
CORRELATION MATRIX

ENTROP=(-1.)*ALOG(DET)
SAVE THE CORRELATION COEFFICIENTS AS

WRITE(6,91) ENTROP

VECTORS FOR PRINCIPLE COMPONENTS

WRITE(6,91) ENTROP

CONTINUE

SAVE THE CORRELATION COEFFICIENTS AS

FORMAT(*ENTROP MEASURE IS *,F15.9)

VECTORS FOR PRINCIPLE COMPONENTS

CONTINUE

FORMAT(*ENTROP MEASURE IS *,F15.9)

CONTINUE

FORMAT(*ENTROP MEASURE IS *,F15.9)

STOP

7 MEMORY EXPANDED. USE $LIMITS OR CORE= OPTION FOR NEXT RUN

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
GET THE IMAGE FROM ERTS

written by RJ Bosley

January 1974

Description of Program

This program is initialized by calling GETIN. All following
calls must be to GETIT which gets the subimage from the ERTS
tape on file code 'ES' and outputs a row of overlapping
subimages to disc file 11.

Internal Parameters:

- LENGTH: Length of one ERTS line of data
- NOSK: Number of lines to skip after
- REWINING ERTS INPUT TAPE
- LNSTRT: Starting line in ERTS data file
- LNINCR: Vertical increment for the next row
- F: Array containing crossed band n-tuple components

Entry Point:

CALL GETIT(ILINE, IDIST, NOSK, LNSTRT, LSTOP, NUMLIN, NUMPPL,
LAPVER, LAPHOR, NHOR, INCR, IPEND)

Input Arguments:

- ILINE: Array where ERTS line is read into
- IDIST: Distance between neighboring cells
to form difference array
- NOSK: Number of components of each res cell
- LNSTRT: Starting line of ERTS data file
- LSTOP: Ending line of ERTS data file
- NUMLIN: Number of lines in subimage
- NUMPPL: Number of columns in subimage
- LAPVER: Number of lines subimages overlap
- LAPHOR: Number of columns subimages overlap

Output Arguments:

- NHOR: Number of horizontal overlapping
  subimages per row
- INCR: Horizontal increment for first col of
  the next subimage in the row
- IPEND: Last point in row

Subroutine GETIT(ILINE, IDIST, NOSK, LNSTRT, LSTOP, NUMLIN, NUMPPL,
LAPVER, LAPHOR, NHOR, INCR, IPEND)

Dimension ILINE(1), F(8)

Initialize ERTS input tape

Call EINIT(LENGTH)
Write(6, 6) LENGTH
Format(' LENGTH OF ONE ERTS LINE IS', LENGTH)
Check that no. of words per record is 8

If(LENGTH LE, 3300) Go to 5

Reproducibility of the original page is poor.
2-12-74  18.776

GET THE IMAGE FROM ERTS

WRITE(6,1) LENGTH
1 FORMAT(*'LENGTH OF ERTS LINE EXCEEDS 3300--LENGTH=',I6)
STOP

CALL EREWIND

C

IRSTR=IRSTR-1
LNSTR=IRSTR
LNINCR=NUMLIN-LAPVER
ASSIGN 951 TO JP
ASSIGN 971 TO KP

C

IF(LAPVER.NE.0) GO TO 940
ASSIGN 950 TO JP
ASSIGN 970 TO KP

C

940 LEN= (LENGTH-32)/4
NHOR=0
IPEND=8
INCR=NUMPPL-LAPHOR

1991 IPEND=IPEND+INCR
NHOR=NHOR+1
K=IPEND+NUMPPL+IDIST

C

IF(K.LE.LEN) GO TO 1991
IPEND=IPEND+INCR+NUMPPL+IDIST

C

IS=32

C

IF(IPEND*4+32-1)
WRITE(6,955) LEN,IPEND,NHOR

955 FORMAT(*'TOTAL POINTS PER ROW IS *',I6/* FINAL POINT IS *',I6/*
* NUMBER OF OVERLAPPING HORIZONTAL SUBIMAGES IN THE ROW IS *',I5)

C

CALL RANSIZ(11,NOIM,1)

C

IF(IRSTR.EQ.0) RETURN
CALL ESkip(IRSTR)
RETURN

C

ENTRY GETIT

C

GO TO JP,(950,951)

C

951 IF(LNSTR.EQ.IRSTR) GO TO 950
NOSK=LNSTR
GET THE IMAGE FROM ERTS

K=NOSK+NUMLIN
C IF(K.LE.:STOP) GO TO 901
WRITE(6,900) IRSTOP,NOSK
900 FORMAT(' PROCESSING TERMINATED--NEXT ROW WOULD EXTEND PAST LAST LINE','9,E16.4',' LAST LINE COMPLETED WAS ',E16.4)
STOP
C 901 CALL ESKIP(NOSK)
950 CONTINUE
C LNSTRTI=LNSTR+LNINCR
C KT=0
DO 903 I=1,NUMLIN
CALL EREAD(ILINE,N)
.C IF(LN.NE.0) GO TO 902
WRITE(6,905)
905 FORMAT(' EOF DETECTED ON ERTS INPUT TAPE--PROCESSING TERMINATED*)
STOP
C 902 DO 960 IP=IS,IE,4
K=KT+1
C F(1)=ILINE(IP+2)
F(2)=(ILINE(IP+2))*ILINE(IP+2)
F(3)=ILINF(IP+3)
F(4)=(ILINE(IP+3))*ILINE(IP+3)
F(5)=ILINE(IP+3)
F(6)=ILINE(IP+4)
F(7)=(ILINE(IP+4))*ILINE(IP+4)
F(8)=(ILINE(IP+2))*ILINE(IP+4)
C WRITE(11'KT) (F(I),I=1,NDIM)
960 CONTINUE
C 903 CONTINUE
GO TO KP,(970,971)
C 971 CALL ERWIND
970 RETURN
END

7 MEMORY EXPANDED. USE $LIMITS OR CORE= OPTION FOR NEXT RUN
GET THE DIFFERENCE ARRAY

COIFFER

WRITTEN BY RJ BOSLEY

DESCRIPTION OF PROGRAM

This program replaces the original image IA with the nearest neighbor horizontal difference, (II-J1, 12-J2, ..., IN-JN) where I and J are N-DIMENSIONAL HORIZONTALLY NEIGHBORING RESOLUTION CELLS OF DISTANCE IDIST. NOTE THAT THE ABSOLUTE VALUE IS USED GIVING ONLY THE POSITIVE HALF OF THE DISTRIBUTION OF DIFFERENCES I-J AND J-I.

ENTRY POINT.

CALL DIFFER(IA,X,IXDIM,IYDIM,NDIM,IDIST,NUMPPL,NUMLIN)

ARGUMENTS.

IA THE SUBIMAGE BEING PROCESSED
X DIFFERENCE ARRAY
IXDIM COL DIMENSION OF X
IYDIM ROW DIMENSION OF X
NDIM THE DIMENSION OF EACH RESOLUTION CELL
IDIST DISTANCE BETWEEN RESOLUTION CELLS USED FOR THE DIFFERENCE ARRAY
NUMPPL NUMBER OF COLUMNS IN SUBIMAGE
NUMLIN NUMBER OF LINES IN SUBIMAGE

SUBROUTINE DIFFER(IA,X,IXDIM,IYDIM,NDIM,IDIST,NUMPPL,NUMLIN)

DIMENSION IA(IYDIM,IXDIM,1),X(IYDIM,IXDIM,1)

DO 5 IBAND=1,NDIM GO THRU EACH DIMENSION OR 3AND

DO 4 LINE=1,NUMLIN GO THRU EACH LINE

DO 3 KOL=1,NUMPPL GO THRU ALL BUT THE LAST COLUMN

X(LINE,KOL,IBAND)=IABS(IA(LINE,KOL,IBAND)-IA(LINE,KKOL,IBAND)) REPLACE EACH RESOLUTION CELL COMPONENT BY THE DIFFERENCE

3 CONTINUE
4 CONTINUE
5 CONTINUE
RETURN
END
FIND THE COVARIANCE MATRIX FOR THE SUBIMAGE

CCOVAR

WRITTEN BY RJ BOSLEY
JAN 1974

DESCRIPTION OF PROGRAM

This program takes the difference subimage and calculates the covariance matrix for it.

DESCRIPTION OF PARAMETERS

PERCNT

PERCENTAGE OF TOTAL VECTORS X FROM WHICH COVARIANCE IS CALCULATED.

SCR, XLINE, SAM

SCRATCH ARRAYS.

IER

ERROR FLAG FROM MNCVIN.

ERROR FLAG FROM MNCV.

ENTRY POINT:

CALL COVAR(XLINE, NUMPL, XIDIM, YIDIM, NUMLIN, NOIN, COV)

INPUT ARGUMENTS.

XLINE

ARRAY USED TO SEND ONE LINE TO MNCV.

DIMENSION OF VECTORS X, NO OF BANDS.

NUMPL

NUMBER OF COLUMNS IN SUBIMAGE.

X

FLOATING POINT DIFFERENCE ARRAY.

IXDIM, IYDIM

DIMENSION OF X.

NUMLIN

NUMBER OF LINES IN SUBIMAGE.

NOIN

DIMENSION OF COV ARRAY.

OUTPUT ARGUMENTS.

COV

COVARIANCE MATRIX FOR THE SUBIMAGE.

SUBROUTINE COVAR(XLINE, NUMPL, XIDIM, YIDIM, NUMLIN, NOIN, COV)

DIMENSION XIDIM, YIDIM, XLINE(NUMPL, XIDIM, YIDIM), NUMLIN, NOIN

PERCNT=100.0

CALL MNCVIN(NUMPL, NOIN, NUMLIN, PERCNT, XLINE, XMEAN, SCR, SAM, IER, JER)

IF (IER .EQ. 0) GO TO 1

WRITE (6, 2) IER

FOMAT (1X, ERROR IN MNCVIN, IERROR IS '13)

STOP

CONTINUE

DO 10 LINE=1, NUMLIN

GO THRU EACH LINE OF SUBIMAGES

10 CONTINUE

223
GET ONE LINE OF DATA

INCREMENT COVARIANCE CALCULATIONS

CHECK FOR GROUND TRUTH ERROR

NOTE ***ONLY THE POSITIVE DIFFERENCES WERE USED IN THE CALCULATIONS---THE TRUE MEAN MUST BE ZERO, SO WE MUST ADD XMEAN**2 TO EACH ELEMENT

END
IDENTIFICATION

PROGRAM NAME
MNCVIN
OTHER ENTRY POINT
MNCV
SYSTEM
POP-15
SOURCE LANGUAGE
FORTRAN IV
AUTHOR
JAMES D. YOUNG
DATE
8/18/73

PURPOSE

TO CALCULATE THE MEAN VECTOR AND COVARIANCE MATRIX FOR EACH CATEGORY OF A SET OF VECTORS. THE CALCULATIONS WILL BE BASED ON A SPECIFIED PERCENTAGE OF THE VECTORS RANDOMLY CHOSEN WITHIN THE SET.

ENTRY POINT - MNCVIN(NVPCAL, N0CUR, NCALL, PERCENT, NCAT, X, NTRUTH, COV, XMEAN, SC, TMEN, SAMSZ, IERROR, IERROR)

THIS INITIALIZES THE ROUTINE. AFTER MNCVIN HAS BEEN CALLED CHECK IERROR TO SEE IF IT IS NONZERO WHICH INDICATES THAT AN ERROR HAS OCCURRED.

ENTRY POINT - MNCV

THE CALL TO MNCV SHOULD BE PERFORMED NCALL TIMES, EACH TIME WITH THE NEXT GROUP OF VECTORS IN X. IF MORE THAN ONE CATEGORY IS BEING CONSIDERED, THE GROUND TRUTH INTEGERS IN NTRUEH ASSOCIATED WITH THE VECTORS SHOULD ALSO BE UPDATED EACH TIME MNCV IS CALLED, AND IERROR SHOULD BE CHECKED AFTER EACH CALL TO SEE IF IT IS NONZERO WHICH INDICATES THAT AN ILLEGAL GROUND TRUTH INTEGER HAS BEEN FOUND. IF ONLY ONE CATEGORY IS BEING CONSIDERED HOWEVER, THE VALUES IN NTRUEH ARE NOT USED IN THE ROUTINE.

AFTER MNCV HAS BEEN CALLED NCALL TIMES THE MEAN VECTOR AND COVARIANCE MATRIX FOR EACH CATEGORY IS COMPLETED. IN ADDITION, THE NUMBER OF VECTORS USED FOR THE CALCULATIONS FOR EACH CATEGORY IS OUTPUT. THESE NUMBERS SHOULD BE CHECKED WHEN APPROPRIATE TO SEE IF ENOUGH VECTORS WERE USED FROM EACH CATEGORY TO GIVE A REASONABLE ESTIMATE OF THE DESIRED STATISTICS.

ARGUMENTS

INPUT -

NVPCAL  NUMBER OF VECTORS PER CALL
DIMENSION OF DATA VECTORS

NUMBER OF CALLS

PERCENTAGE OF TOTAL NUMBER OF VECTORS FROM WHICH MEAN AND COVARIANCE MATRICES WILL BE CALCULATED

NUMBER OF CATEGORIES CONSIDERED

NUMBER OF CATEGORIES IN DATA IF ONE SET OF STATISTICS WILL BE CALCULATED FOR EACH CATEGORY

MATRIX CONTAINING INPUT DATA VECTORS IN ITS COLUMNS

VECTOR CONTAINING THE GROUND TRUTH INTEGERS, 1 THROUGH NCAT, ASSOCIATED WITH THE DATA VECTORS OF X. IF NCAT EQUALS 1 THIS VECTOR WILL NOT BE USED.

MATRIX CONTAINING COVARIANCE MATRICES OF THE DATA

MATRIX CONTAINING MEAN VECTORS OF THE DATA

SCRATCH MATRIX CONTAINING AN ESTIMATE OF THE MEAN VECTOR/VECTORS

VECTOR CONTAINING NUMBER OF VECTORS USED TO CALCULATE THE STATISTICS FOR EACH CATEGORY

ERROR INDICATED IF RETURNED NONZERO

=1 IF NVPCAL .LE. 0

=2 IF NDIM .LE. 0

=3 IF NCAL .LE. 0

=4 IF PERCNT .GT. 100., OR PERCNT IS SO SMALL THAT FEWER THAN 2 VECTORS WILL BE USED TO CALCULATE ALL THE STATISTICS

=5 IF NCAT .LE. 0

(IF MORE THAN ONE OF THESE ERRORS OCCURS, THE HIGHER VALUE WILL BE RETURNED)

ERROR INDICATED IF RETURNED NONZERO

=1 IF ILLEGAL GROUND TRUTH LABEL IS FOUND

INTERNAL PARAMETERS

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

226
ICALL NUMBER OF TIMES MNCV HAS BEEN CALLED
VECTS TOTAL NUMBER OF VECTORS WHICH WILL BE INPUT TO MNCV
VTBU TOTAL NUMBER OF VECTORS TO BE USED IN CALCULATION
OF STATISTICS
VLTBU NUMBER OF VECTORS LEFT TO BE USED IN CALCULATION
OF STATISTICS
VLEFT NUMBER OF VECTORS LEFT TO BE CONSIDERED
INTU INTEGER DENOTING GROUND TRUTH CATEGORY
IP ARGUMENT TO RCM
IQ ARGUMENT TO RCM

INITIALIZER ENTRY POINT
SUBROUTINE MNCVIN(NVPCAL,NOIM,NCALL,PERCNT,NCAT,X,NTRUTH,COV,
XMEAN,SCOTMEN,SAMSZ,IERROR,JERROR)

DIMENSION X(NOIM, 1),NTRUTH(1),COV(NOIM,NOIM,NCAT),XMEAN(NCAT,1),
1 SCOTMEN(NOIM,1),SAMSZ(NCAT)

ICALL=0
VECTS=NVPCAL*NCALL
IERROR=0
NVBU=VECTS*PERCNT/100.+0.49999
VTBU=NVBU
VLTBU=VTBU
VLEFT=VECTS
INTU=1
IP=5555
IQ=5555
JERROR=0

CHECK LEGALITY OF SOME NUMBERS:
IF(NVPCAL.LE.0) IERROR=1
IF(NOIM.LE.0) IERROR=2
IF(NCALL.LE.0) IERROR=3
IF(PERCNT.GT.100. OR.VTBU.LT.2.) IERROR=4
IF(NCAT.LE.0) IERROR=5

ZERO OUT A FEW ARRAYS
DO 14 K=1,NCAT
SAMSZ(K)=0
DO 14 J=1,NOIM
XMEAN(J,K)=0
SCOTMEN(J,K)=0
DO 14 I=1,NOIM
14 COV(I,J,K)=0

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
02-09-74  20.440

MNCVIN - MNCV

SET TRANSFER TO REFLECT NUMBER OF CATEGORIES

ASSIGN 5 TO IGO
IF (NCAT GT 1) ASSIGN 15 TO IGO
RETURN

ENTRY POINT FOR ALL CALLS AFTER THE FIRST

ENTRY MNCV

ICALL=ICALL+1
IF( (ICALL/50)*50 .NE. ICALL. AND. ICALL .NE. 10) GO TO 1

UPDATE ESTIMATE OF THE MEANS AND MODIFY COVARIANCE CALCULATIONS TO REFLECT THIS UPDATE

DO 10 K=1,NCAT
   IF(SAMSZ(K).EQ.0.) GO TO 10
   DO 2 J=1,NDIM
   DO 2 I=1,J
      COV(I,J,K)=COV(I,J,K)-SAMSZ(K)*(XMEAN(I,K)/SAMSZ(K)-SCHME(I,K))*
      (XMEAN(J,K)/SAMSZ(K)-SCHME(J,K))
   2 CONTINUE
   DO 3 J=1,NDIM
      SCHME(J,K)=XMEAN(J,K)/SAMSZ(K)
   3 CONTINUE
10 CONTINUE

CONTINUE TO CALCULATE MEAN AND UPPER TRIANGLE OF COVARIANCE MATRICES

1 DO 8 I=1,NVPCAL

DETERMINE WHETHER TO SKIP THIS VECTOR

IF (RCM(IP,IP).GT.VLTBU/VLEFT) GO TO 11

INCUCE THIS VECTOR IN CALCULATIONS

VLTBU=VLTBU-1.
GO TO IGO,(15,5)

WE ARE TO CONSIDER MORE THAN ONE CATEGORY

15 INTRU=INTRUTH(I)

CHECK LEGALITY OF GROUND TRUTH LABEL

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR
SUM FOR MEAN
XMEAN(J,K) = SUM(J,K,INTRU)/SAMSZ(K)

SUM FOR COVARIANCE
COV(J,K) = SUM(J,K,INTRU+XMEAN(J,K)-SCTHRN(J,K), XMEAN(J,K)/SAMSZ(K)-SCTRN(J,K))/SAMSZ(K)

All vectors have been input
Finish calculation of statistics

RETURN
CONTINUE

CONTINUE
GET THE CORRELATION MATRIX FOR THE SUBIMAGE

CCORPEL

GET THE CORRELATION MATRIX FOR THE SUBIMAGE

CC

CC

CC

CC

CC

DESCRIPTION OF PROGRAM

THIS PROGRAM CALCULATES THE CORRELATION MATRIX GIVEN THE

COVARIANCE MATRIX AND ITS ORDER.

CC

ENTRY POINT.

CALL CORREL(COV,NOM,COR)

CC

ARGUMENTS.

COV

COVARIANCE MATRIX ARRAY

NOM

ORDER OF MATRICES

COR

CORRELATION MATRIX ARRAY

CC

SUBROUTINE CORREL(COV,NOM,COR)

DIMENSION COV(NOM,NOM),COR(NOM,NOM)

DO 10 LINE=1,NOM

COVL=ABS(COV(LINE,LINE))

GO THRU UPPER DIAGONAL

DO 9 KOL=LINE,NOM

COVC=ABS(COV(KOL,KOL))

COR(LINE,KOL)=COV(LINE,KOL)/SQRT(COV(LINE,LINE)*COVC)

9 COR(KOL,LINE)=COR(LINE,KOL)

10 CONTINUE

RETURN

END

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR
IV.2-d Piecewise Linear Classification Program Listings

RCLASS
XIN
LINEAR
WEIGHT
DESCRIPTION OF PROGRAM.

THIS SUBROUTINE CLASSIFIES A GIVEN SET OF PATTERN VECTORS INTO ONE OF MANY POSSIBLE CATEGORIES USING THE INFORMATION CONTAINED IN A SET OF TRAINING PATTERNS WHOSE CATEGORIES ARE KNOWN.

ENTRY POINT.

CALL RCLASS(WORK,ISIZE)

INPUT ARGUMENTS.

WORK - SCRATCH ARRAY USED BY RCLASS
ISIZE - SIZE OF THE SCRATCH ARRAY WORK. THE ARRAY WORK MUST BE DIMENSIONED IN THE CALLING PROGRAM, WITH A DIMENSION GREATER THAN OR EQUAL TO ISIZE. THE VALUE OF ISIZE IS GIVEN BY

ISIZE=NOIM*(NTRAIN+10)+1000+(NC*(NC+1)/2)*ND+PAI/ND*ND

WHERE NO=NO:M-1

INPUT PARAMETER CARDS.

CARD 1. SHOULD CONTAIN THE PROGRAM OPTION PARAMETERS NOPT1, NOPT2,NOPT3,NOPT4, IN FORMAT(4I1).

IF(NOPT1.NE.O) THE TRAINING PATTERNS ARE PRINTED OUT
IF(NOPT2.NE.O) THE TEST PATTERNS ARE PRINTED OUT
IF(NOPT3.NE.O) THE CLASSIFICATION OF EACH TRAINING PATTERN IS LISTED. OTHERWISE ONLY THE CONTINGENCY TABLE IS PRINTED OUT
IF(NOPT4.NE.O) THE CLASSIFICATION OF EACH TEST PATTERN IS LISTED. OTHERWISE ONLY THE CONTINGENCY TABLE IS PRINTED OUT.

CARD 2. SHOULD CONTAIN PARAMETERS NNOT, NPART, NC, NO, NPAIR IN FORMAT (5I5).

NNOT -- TOTAL NUMBER OF PATTERN VECTORS IN THE DATA SET
NPART -- NPART OUT OF EVERY 10 PATTERN VECTORS IN THE DATA SET WILL BE USED FOR TRAINING. THE REMAINING PATTERNS WILL BE USED FOR TESTING THE CLASSIFIER
NOIM -- NUMBER OF MEASUREMENTS PER VECTOR PLUS 2
NC -- NUMBER OF GROUND TRUTH CATEGORIES
NPAIR -- 2*XMAXIMUM NUMBER OF TRAINING PATTERNS IN
ANY ONE CATEGORY)

INPUT DATA SET.
INPUT DATA VECTORS ARE READ IN BINARY FROM FILE CODE 01
ACCORDING TO THE FOLLOWING READ---
READ(01) IGT,M1,NFT,(FEAT(i),i=1,NMEAS)
WHERE IGT IS THE GROUND TRUTH CATEGORY
M1,N1,NFT ARE NOT USED (MAY BE USE) AS TO TAG R
FEAT(i) IS THE FEATURE VECTOR i-TH COMPONENT
NMEAS IS THE NUMBER OF COMPONENTS OR MEASUREMENTS IN
EACH FEATURE VECTOR=NOIM-2.

THE DATA FILE SHOULD HAVE A TOTAL OF NMT LOGICAL RECORDS IN
BINARY FORM. EACH LOGICAL RECORD MUST BE OF LENGTH (NOIM+2)
WORDS. WORD 1 IS THE GROUND TRUTH CATEGORY AND 5 THRU NOIM+2 ARE
THE MEASUREMENT VALUES. THE PATTERN VECTORS MUST BE SORTED BY
CATEGORY. THE DATA SET IS SORTED INTO TRAINING AND TEST SETS
ACCORDING TO THE USER SPECIFIED RATIO IN NPART. TRAINING PATTERNS
ARE COPIED TO CORE AND TEST PATTERNS ARE COPIED TO DISC ON FILE
CODE 02.

LIMITATIONS.
1. MAXIMUM NUMBER OF CATEGORIES IS 15.
2. MAXIMUM VALUE OF NOIM IS 100.
3. SCRATCH DISC FILE MUST BE ON 02 AND INPUT DATA ON 01.
4. PATTERN VECTORS MUST BE SORTED BY CATEGORY.

THEORY.
USING A REGRESSION TYPE ALGORITHM, THE PROGRAM
OBTAINS A SET OF HYPERPLANES FOR SEPARATING THE
TRAINING PATTERNS. THE SEPARATION IS DONE FOR
DIFFERENT CATEGORY PAIRS. A TOTAL OF NMT(NMT-1)/2
HYPER PLANES ARE DETERMINED. TEST PATTERNS ARE
CLASSIFIED ON THE BASIS OF A MAJORITY VOTE ON THESE
HYPER PLANES.

BIBLIOGRAPHY.
FOR A COMPLETE DISCUSSION ON DETAILS, SEE "INTRODUCTION TO
STATISTICAL PATTERN RECOGNITION" BY FUKUNAGA, ACADEMIC PRESS,
NEW YORK, 1972.

SUBPROGRAMS REQUIRED.
DRIVER
RCLASS
XIN
LINEAR
WEIGHT
MINV -- A FORTRAN CALLABLE ROUTINE FROM IBM SSP

EXAMPLE.
IN ORDER TO PROCESS A DATA SET REQUIRING A SCRATCH ARRAY OF

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR.
SIZE 15000, THE DRIVER SET UP IS AS FOLLOWS ---

**SAMPLE DRIVER FOR CLASSIFICATION PROG**

DIMENSION WORK(15000)
ISIZE = 15000
CALL RCLASS(WORK,ISIZE)
STOP
END

SUBROUTINE RCLASS(WORK,ISIZE)

DIMENSION WORK(ISIZE),DD(50)
COMMON NTRAIN,TEST,NOH,NO,NC,NPAIR,NH,NOPT1,NOPT2,NOPT3,NOPT4,
1 NTOT,NPART

CALL XIN(WORK(1),DD(1))

READ IN PARAMETER CARDS AND DATA SET

LOCATE THE BEGINNING ADDRESS OF THE VARIOUS ARRAYS FOR THE LINEAR PROG

I1=1
I2=I1+NTRAIN*NOH
I3=I2+NOH
I4=1+NH+NO
I5=I4+NPAIR*NO+1500

CHECK THAT THE DIMENSION OF THE WORK ARRAY IS ADEQUATE

IF(I5.LE.ISIZE) GO TO 53
WRITE(6,82)
FORMAT(1H1,5X,"PROBLEM WONT FIT THE CORE ALLOTED FOR THIS PROG")
RETURN
53 CONTINUE

NOW GO DO THE CLASSIFICATION

CALL LINEAR(WORK(I1),WORK(I2),WORK(I3),WORK(I4),WORK(I5))
RETURN
END
READ IN THE DATA

ENTRY POINT.
CALL XIN(WORK,U)

INPUT ARGUMENTS.
WORK
U

INTERNAL PARAMETERS.
NTRAIN
NTEST
NDIM
NO
NC
NPAIR
NH
NOPT1-4
NSKIP
NTIMES
ILEFT

DIMENSION WORK(l),U(l)
COMMON NTRAIN,NTEST,NDIM,NO,NC,NPAIR,NH,NOPT1,NOPT2,NOPT3,NOPT4

READ IN PARAMETER CARD 1.
READ(5,100) NOPT1,NOPT2,NOPT3,NOPT4
READ(5,101) NTOl,NPART,NDIM,NC,NPAIR

READ THRU THE INPUT FILE TWO TIMES --
ONCE TO GET THE TRAINING PATTERNS AND
ONCE TO GET THE TEST PATTERNS

REIN 01

INITIALIZE INTERNAL PARAMETERS
READ IN THE DATA

NTRAIN=0
NDO=NDIM+2
NTIMES=NTOT/10

NSKIP=10-NPART

IF(NOPT1 .NE. 0) WRITE(6,201)
FORMAT(1H1,10X,'TRAINING PATTERNS')

DO 20 MM=1,NTIMES
IF(NSKIP.EQ.0) GO TO 16
DO 15 JJ=1,NSKIP
READ(1) (U(K),K=1,NDO)
CONTINUE
DO 10 JJ=1,NPART
NTRAIN=NTRAIN+1
IBEGIN=(NTRAIN-1)*NDIM+1
IEND=NTRAIN*NDIM
IB1=IBEGIN+1
IB2=IB1+1
WORK(IB1)=1.0
READ(1) WORK(IBEGIN),NM1,NM2,NM3,(WORK(K),K=IB2,IEND)
IF(NOPT1 .NE. 0) WRITE(6,102)
FORMAT(1X,I3,3X,6(F10.4))
CONTINUE

READ IN THE DATA

NTRAIN=0
NDO=NDIM+2
NTIMES=NTOT/10

NSKIP=10-NPART

OUT OF EACH OF THE NTIMES BLOCKS OF
TEN INPUT PATTERNS PICK THE FIRST
NPART PATTERNS AS TRAINING PATTERNS

WRITE HEADING

GO THRU EACH BLOCK OF TEN
PATTERN VECTORS

SKIP OVER THE TEST VECTORS
READ PAST THIS VECTOR

READ NPART TRAINING PATTERN VECTORS
COUNT THE NO. OF TRAINING VECTORS
INDEX THIS VECTOR INTO WORK

SET THE FIRST COMPONENT TO 1.0
READ THE TRAINING PATTERN VECTORS
WRITE OUT THE TRAINING PATTERNS

READ IN THE LEFTOVER PATTERNS
INCREMENT TRAINING VECTOR COUNT
INDEX THE VECTOR INTO WORK
READ IN THE DATA

I92=IN1+1
WORK(IN1) = 1.0
C
C READ A TRAINING PATTERN
C READ (1) WORK (BEGIN), NM1, NM2, NM3,... (WORK(K), K = I92, IEND)
C WRITE OUT TRAINING PATTERNS
C IF (NLOPT .NE. 0) WRITE (6, 102) NTRAIN, WORK (BEGIN), NM1, NM2,... (WORK(K), K = I91, IEND)

C
C CONTINUE
C 25 CONTINUE
C 26 CONTINUE
C 27 REWIND 01
C 28 REWIND 02
C
C COPY TEST PATTERNS TO DISC FILE 02
C
C WRITE HEADING
C 103 FORMAT (1H1, 10X, 'TEST PATTERNS')
C NTST = 0
C
C SKIP IF THERE ARE NO TEST PATTERNS
C IF (NLOPT .NE. 0) GO TO 99
C DO 45 NN = 1, NTIMES
C DO 43 JJ = 1, NSKIP
C NTEST = NTEST + 1
C C
C READ (1) (U(KK), KK = 1, NCD)
C C U(4) = 1.0
C C IF (NLOPT2 .NE. 0) WRITE (6, 107)
C 43 WRITE (2), (U(KK), KK = 1, NDD)
C C DO 44 JJ = 1, NPAR
C 44 READ (1) (U(KK), KK = 1, NDD)
C 45 CONTINUE
C END FILE 02
C REWIND 02
C C
C SKIP OVER THE TRAINING VECTORS
C 99 CONTINUE
C REWIND 01
C C
C WRITE OUT A PROGRESS NOTE
C WRITF (6, 702)
C 702 FORMAT ('DATA HAS BEEN COPIED ON TO DISK 02')
C C NH = NC*(NC+11)/2
C NG = NDD-1
C RETURN
C END

237
LINEAR DISCRIMINANT FUNCTION

DESCRIPTION OF PROGRAM.

This subroutine calls the regression routine weight to get the linear discriminant functions, combines these to obtain a piecewise linear discriminant function for each category, and classifies the training and test patterns. This routine also outputs the results.

THEORY.

The algorithm for determining the weight vector for separating the i and j-th categories is

\[ w = (U^T U)^{-1} U^T T \]

where \( T \) is a vector of length \( n \) with components of values equal to 1. For details see **Introduction to Statistical Pattern Recognition** by Fukunaga, Academic Press, 1972.

The matrices (dimensioned as one dimensional in pg 4) have the following structures:

- **XTRAIN** --- XTRAIN is an \( N \times N \text{TRAIN} \) matrix. Each column vector in \( X \text{TRAIN} \) represents one training pattern. The first entry in each column is the category name. The second entry is the constant 1. The entries \( -1 \) to \( 0 \) are the values of the components of the pattern vector \( x(j) \). \( X \text{TRAIN} \) has \( \text{NTRAIN} \) columns and \( n+1 = \text{NOIM} \) rows.

- **XTEST** --- XTEST has the same configuration as \( X \text{TRAIN} \).

- **U** --- DATA MATRIX \( U \) is used for calculating the boundary between the categories \( i \) and \( j \). \( U \) has \( NOIM \times N \text{TRAIN} \) columns, as follows:

  - **PATTERNS FROM CATEGORY I**
  - **PATTERNS FROM CATEGORY J**

- **ENTRY POINT**

  CALL LINEAR(XTRAIN, XTEST, W, U, DUMMY)

INPUT ARGUMENTS:

- XTRAIN
- XTEST
- W
- U

M - matrix containing training patterns

W - matrix containing test patterns

LINEAR W - weight vectors
SUBROUTINE LINEAR(XTRAIN, XTEST, W, U, DUMMY)
C
DIMENSION XTRAIN(1), XTEST(1), W(1), U(1), DUMMY(1)
DIMENSION IERROR(15, 15), NTR(15), NAME(15)
DIMENSION L(100), M(100)
DIMENSION WT(100)
COMMON NTRAIN, NTEST, NDIM, NO, NC, NPAIR, NH, NOPT1, NOPT2, NOPT3, NOPT4,
1 NTOT, NPART
REAL NAME
C
WRITE(6, 1401)
1401 FORMAT(1H1, 10X, ’INPUT DATA SUPPLIED BY YOU’)
WRITE(6, 1402) NTOT, NPART, NMJ
1402 FORMAT(”NO. PATTERNS =”, I4/10X, ”TOTAL NUMBER OF PATTERNS =”, I4/10X,
1 ’TERMS WERE USED IN TRAINING’/10X, ’OUT OF 10 TERMS’)
C
DO 1 J=1, NC
1 NTR(J)=0
C
DO 3 I=1, NTRAIN
C
NAME(1)=XTRAIN(I)
GO THRU ALL THE TRAINING PATTERNS
LINAR DISCRIMINANT FUNCTION

\[ l_1 = (I - 1) \cdot N_{\text{DIM}} + 1 \]

C IF NAME(J) .EQ. XTRAIN(I1), GO TO 2
C J = J + 1
C NAME(J) = XTRAIN(I1)
C COUNT THE NUMBER IN THIS CATEGORY
C CONTINUE
3 WRITE OUT NO. OF TRAINING AND TEST PATTERN AND THE NO. OF CATEGORIES FOUND
C WRITE(6,1403) NTRAIN, NTEST, J
1403 FORMAT(10X,'NUMBER OF TRAINING PATTERNS USED= ',14/I3), 'NUMBER OF FIRST PATTERNS USED= ',14/I3), 'NUMBER OF GROUND TRUTH LABELS FOUND= ',14/I3)
C WRITE(6,1404) NC
1404 FORMAT(10X,'NUMBER OF GROUND TRUTH LABELS SPECIFIED = ',13)
C WRITE(6,1411)
1411 FORMAT(1H1,5X,'SUMMARY OF WEIGHT VECTORS'//)
C FIND THE HYPERPLANES WHICH SEPARATE CATEGORIES I AND J, WHERE I GOES FROM 1 TO NC AND J GOES FROM 1+1 TO NC.
C INCOL=0
DO 10 I=1,NC
IPLUS1=I+1
DO 10 J=IPLUS1,NC
C IF (J .GT. NC) GO TO 868
C INCOL=INCOL+1
ISUM=0
DO 9 K=1,1
ISUM=ISUM+NTP(K)
9 ISUM=ISUM+NTP(K)
C INBEG=ISUM-NTR(I)
ISUM=0
DO 8 K=1,J
ISUM=ISUM+NTP(K)
JWBEG=ISUM-NTR(J)
C GET ENTRIES FROM XTRAIN AND PUT INTO THE MATRIX U
GO THRU EACH TRAINING PATTERN IN CAT I
INDEX COL FOR XTRAIN
GO THRU EACH COMP IN VECTOR
SET UP INDEXES
TRANSFER PATTERN TO U FROM XTRAIN
DO THE SAME FOR CATEGORY J
GO THRU EACH PATTERN IN CAT J
SET UP COL INDEX
TRANSFER THE VECTOR IN XTRAIN TO U
AND CHANGE THE SIGN
CALL THE REGRESSION ROUTINE
GET WEIGHT VECTOR FOR CATS I AND J
INDEX THE VECTOR INTO W ARRAY
WRITE OUT THE WEIGHT VECTOR
PUNCH OUT THE WEIGHT VECTOR
LINEAR DISCRIMINANT FUNCTION

1761 FORMAT((4X,6F12.5))
1868 CONTINUE
10 CONTINUE

BEGIN CLASSIFICATION
GO THRU FIRST FOR TRAINING PATTERNS
THEN FOR TEST PATTERNS

DO 67 INDEX=1,?  
WRITE HEADING
IF((INDEX.EQ.1).AND.(N0PT.4.NE.0)) WRITE(6,1498)
IF((INDEX.EQ.2).AND.(N0PT.4.NE.0)) WRITE(6,1499)
1498 FORMAT(1H1,15X,"CLASSIFICATION OF TRAINING PATTERNS")
1499 FORMAT(1H1,10X,"CLASSIFICATION OF TEST PATTERNS")

DO 78 IMN=1,15  
WRITE HEADING
IF((INDEX.EQ.1).AND.(N0PT.4.NE.0)) WRITE(6,1500)
IF((INDEX.EQ.2).AND.(N0PT.4.NE.0)) WRITE(6,1500)

C

IF(INDEX.EQ.1) NPT=NPTAIN  
IF(INDEX.EQ.2) NPT=NTEST
RETURN IF NO TEST SET

C

DO 66 III=1,NPT  
GO THRU EACH PATTERN TO BE CLASSIFIED

C

DO 55 NOUM=1,ND  
CONTINUE

C

IF(INDEX.EQ.1) U(NOUN)=XRAIN(IIXIII)
CONTINUE

C

IF(INDEX.EQ.2) READ(2) DUMMY1,DUMMY2,DUMMY3,(U(NOUN),NOUN=1,ND)
KMAX=-20

C

DO 65 IIIC=1,NC  
CONTINUE

C

OTHERWISE, READ TEST VECTORS FROM 02
CONTINUE

C

GO THRU EACH CAT FOR VOTING
LOCATE THE HYPERPLANE THAT SEPARATES
Cats IIIC and JJC. THE SIGN OF THE
WEIGHT VECTORS IS CHANGED DEPENDING
UPON WHETHER OR NOT JJC.GT.IIC
DO 64 JJC=1,NC
IF(JJC.EQ.IIC) GO TO 63
IF(JJC.GT.IIC) GO TO 50
IF(JJC.LT.IIC) GO TO 41
53 CONTINUE

C
SIGN=1.0
ISUM=0
IS=IIC-1
IF(IS.EQ.0) GO TO 53
DO 51 KS=1,IS
ITERM=NC-KS
51 ISUM=ISUM+ITERM
53 CONTINUE
NCOLW=ISUM+JJC-IIC
C
GO TO 45
41 CONTINUE

C
ISUM=0
SIGN=-1.0
IS=JJC-1
IF(IS.EQ.0) GO TO 43
DO 42 KS=1,IS
ITERM=NC-KS
42 ISUM=ISUM+ITERM
43 CONTINUE
NCOLW=ISUM+IIC-JJC
45 CONTINUE

C
SUM=0.0
DO 52 IROW=1,NO
IWRCH=(NCOLW-1)*NO+IROW
SUM=SUM+(I(IROW))*W(IWRCH)**(SIGN)
52 CONTINUE
IVOTE=0
C
IF(SUM.GT.0.0) IVOTE=1
ELSE VOTE NEGATIVE
C
IF(SUM.LT.0.0) IVOTE=-1
CONTINUE
C
IF(JJC.EQ.IIC) IVOTE=0
IF SAME CATEGORY, DONT VOTE
C
KOUNT=KOUNT+IVOTE
TALLY THE COUNT
C
IF(KOUNT.GT.KMAX) ICLASS=IIC
DETERMINE THE CAT FOR WHICH VOTE IS MAXIMUM
RESET MAX COUNT

65 CONTINUE

GET THE RESULTS AND OUTPUT THEM

C
C III1=(III-1)*NO1+1
IF(INDEX.EQ.1) TCAT=TRAIN(III1)
IF(INDEX.EQ.2) TCAT=Dummy
C WRITE RESULTS FOR THIS PATTERN VECTOR
C IF((INDEX.EQ.1).AND.(NOPT3.NE.0).AND.(KMAX.EQ.0)) WRITE(6,1501)III LINEAR
1,TCAT
IF((INDEX.EQ.2).AND.(NOPT4.NE.0).AND.(KMAX.EQ.0)) WRITE(6,1502)III LINEAR
1,TCAT
IF((INDEX.EQ.1).AND.(NOPT3.NE.0).AND.(KMAX.GT.0)) WRITE(6,1502)III LINEAR
1,TCAT,NAME(1CLASS),KMAX
IF((INDEX.EQ.2).AND.(NOPT4.NE.0).AND.(KMAX.GT.0)) WRITE(6,1502)III LINEAR
1,TCAT,NAME(1CLASS),KMAX
C 1501 FORMAT(6X,15X,6X,15X,14X,14X,'NOT CLASSIFIED')
1502 FORMAT(6X,15X,6X,15X,14X,15X,14X,'CLASSIFIED')
C DO 71 ICAT=1,NC
C IF(INDEX.EQ.1) WRITE(6,721)
C IF(INDEX.EQ.2) WRITE(6,722)
C CONTINUE

71 ICAT=JCAT
C UPDATE THE CONTINGENCY TABLE
C IF(KMAX.GT.0) IERROR(ICAT,1CLASS)=IERROR(ICAT,1CLASS)+1
C CONTINUE
C WRITE OUT THE HEADING
C IF(INDEX.EQ.1) WRITE(6,721)
C IF(INDEX.EQ.2) WRITE(6,722)
C FORMAT(15X,'CONTINGENCY TABLE FOR TRAINING PATTERNS')
C WRITE(6,723)
C FORMAT(15X,'CONTINGENCY TABLE FOR TEST PATTERNS')
C WRITE(6,724)
C FORMAT(15X,'TRUE',15X,'CATEGORY')
C WRITE(6,725)
C FORMAT(15X,'ASSIGNED CATEGORY')
C CONTINUE
C RETURN
END
GET THE WEIGHT VECTOR

WEIGHT

GET THE WEIGHT VECTOR

written by sam shanmugam

sept 1972

documented by rj bOSLEY

dec 1973

DESCRIPTION OF PROGRAM.

given a matrix u containing patterns of category i and j,

this routine finds a hyperplane to separate these patterns. the

weight vector defining the hyperplane is given by

wt = ((uu')**-1)*(u*t)

where wt is the weight vector

t is a column vector whose components are +1's

nd is the length of the pattern vectors, and

ni j is the number of patterns from cat i + number

of patterns from category j

the data matrix u, used for calculating the boundary between the

categories i and j, is formed as follows ---

*****patterns from category i**********patterns from category j -***

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>-x</td>
<td>-x</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

***** u has no rows and ni j columns. *************

ENTRY POINT.

call weight(u,dummy,wt,nd,ni j)

INPUT ARGUMENTS.

u matrix containing patterns of
categories i and j

dummy dummy array used for minv

nd length of pattern vectors in u

ni j number of patterns from cat i and j

OUTPUT ARGUMENT.

wt weight vector defining the hyperplane

separating patterns of cat i and j

subprograms required.

minv matrix inversion pgm from ibm ssp

internal parameters.

l,m scratch arrays used by minv

d determinant of dummy, from minv

irown row or component index, for wt

idjd index for a vector in dummy

irowkk index for a vector in dummy after it
SUBROUTINE WEIGHT(U, DUMMY, WT, ND, NIJ)

DIMENSION U(100), DUMMY(100), WT(100)
DIMENSION L(100), M(100)

DO 10 ID = 1, ND
  DO 15 JD = 1, ND
    SUM = 0.0
    DO 14 K = 1, NIJ
      INJ = (K-1)*ND + TO
      IJK = (K-1)*ND + JD
      SUM = SUM + U(IJK)*U(JKD)
      IOJ = (JO-1)*NO + ID
    14  CONTINUE
    DUMMY(IOJ) = SUM
    CALL MINV(DUMMY, ND, ID, L, M)
    DO 15 K = 1, ND
      SUM = 0.0
      DO 25 KK = 1, NIJ
        U(KK) = SUM
    25      CONTINUE
        DO 35 K = 1, ND
          IROW = (K-1)*NO + IROW
        35      SUM = SUM + DUMMY(IROW)*U(K)
        WT(IROW) = SUM
      RETURN
    10  CONTINUE
Glossary and Index to Remotely Sensed Image Pattern Recognition Concepts*

ROBERT M. HARALICK†
Remote Sensing Laboratory, University of Kansas, Lawrence, Kansas 66045, U.S.A.

(Received 29 March 1973 and in revised form 21 June 1973)

Abstract—The purpose of the glossary is to state in the simplest possible way the general meaning or word usage for many of the terms in image pattern recognition. There is no intent to provide definitive statements for terms such as "resolution" but rather only statements about the general nature of what resolution is. There is no intent to provide mathematical formulas involving integrals or derivatives in any of the statements. Those who need the mathematics can get it from technical papers or texts.

The glossary is designed to be read by those generally unfamiliar with the area and provide for them an overall perspective. The organization approaches that of programmed learning material and can be smoothly (I hope) read from beginning to end. Those needing to look up a specific term can do so via the index.

There is some overlap of terms in this glossary with those glossaries or definitions in radiometry and aerial photography. There is no intent that the way the terms are described here replace the way they are described in those glossaries and definitions. The overlap is provided here so that the reader can get a perspective of a cluster of terms frequently used in our field. The perspective is intended to start from what the image concept is through the recording of an image by some sensor, the possible conversion of image format and the simple analog or more complex digital processing which must be done on the imagery. In short, the perspective is one of image pattern recognition.

1. An Image is a spatial representation of an object, scene, or another image. It can be real or virtual as in optics. In pattern recognition, image usually means a recorded image such as a photograph, map, or picture. It may be abstractly thought of as a continuous function $I$ of two variables defined on some bounded region of a plane. When the image is a photograph, the range of the function $I$ is the set of grey shades usually considered to be normalized to the interval $[0, 1]$. The grey shade located at spatial coordinate $(x, y)$ is denoted by $I(x, y)$ and is usually proportional to the radiant energy in the electromagnetic band to which the photographic sensor is sensitive. When the image is a map, the range of the function $I$ is a set of symbols or colors, and the symbol or color located at spatial coordinate $(x, y)$ is denoted by $I(x, y)$. A recorded image may be in photographic, video signal, or digital format.

2. The grey shade or grey tone is a number or value assigned to a position $(x, y)$ on an image. The number is proportional to the integrated output, reflectance, or transmittance of a small area, usually called a resolution cell or pixel, centered on the position $(x, y)$. The grey shade can be measured as or expressed in any one of the following ways:

   (1) transmittance
   (2) reflectance
   (3) a coordinate of the ICI color coordinate system

* The glossary was prepared as the report from the definitions and standards subcommittee, Automatic Image Pattern Recognition Committee of Electronic Industries Association.
† I would like to acknowledge the helpful suggestions, comments and corrections which many of my colleagues made. Particular thanks are due to R. Asendorf, B. Bulluck, L. Kirvida, P. Pryor, A. Rosenfeld, B. Scheps, R. Swonger, S. Viglione and R. Zaputowycz.
(4) a coordinate of the tristimulus value color coordinate system 
(5) brightness 
(6) radiance 
(7) luminance 
(8) density 
(9) voltage 
(10) current.

3. A *photograph* is a "hard copy" pictorial record of an image formed by a sensor. The photograph is usually recorded on some type of photosensitive emulsion. It can be either reflective, as is a paper print, or transmissive, as is a transparency. It is usually two-dimensional and its reflectance or transmittance, (either monochromatic or polychromatic) varies as a function of position. If it is a multi-colored image (polychromatic), it can be either natural color where the colors are similar to the original, or false color where the colors of the photograph are radically different from the original. The sensor used to form the image may be any type such as an optical camera with or without spectral filtration, infrared optical–mechanical scanners, TV systems, radars, or sonar sensors, etc. The type of sensor recording the image and spectral region the sensor is sensitive to, should always be indicated when referring to a photograph.

4. A *map* is a representation, of physical and/or cultural features (natural, artificial or both) of a region (such as the sky) or a surface such as that of the earth or a planet. It indicates by a combination of symbols and colors those regions having designated category identifications. Very often ground truth and/or decision rule category assignments are displayed by maps. A photograph with limited symbolism and annotation is often called a *photo-map*.

5. The *radiant intensity* of a point object is a measure of the radiant power per steradian radiated or reflected by an object. In general, radiant intensity is a function of the nature of the object, the viewing angle, spectral wavelength and band-width.

6. The *reflectance or reflection coefficient* is the ratio of the energy per unit time per unit area (radiant power density) reflected by the object to the energy per unit time per unit area incident on the object. In general, reflectance is a function of the incident angle of the energy, viewing angle of the sensor, spectral wavelength and bandwidth, and the nature of the object.

7. The *transmittance or transmittance coefficient* is the ratio of the energy per unit time per unit area (radiant power density) transmitted through the object to the energy per unit time per unit area incident on the object. In general, transmittance is a function of the incident angle of the energy, viewing angle of the sensor, spectral wavelength and bandwidth, and the nature of the object.

8. The *density* of an \((x, y)\) position on a photograph is a measure of the light absorbing capability of the silver or dye deposited on that position. It is defined by the logarithm of the position’s reciprocal transmittance. The density measured should be specified as to whether it is specular or diffuse.

9. *Densitometry* is the field devoted to the measurement of optical image densities on film or print grey shades usually caused by the absorption or reflection of light by developed photographic emulsion.

10. A *densitometer* is a device used to measure the average image density of a small area of specified size on a photographic transparency or print. The measurement may be a
meter reading or an electronic signal. When the small area is smaller than a few hundred microns square, the instrument is called a micro-densitometer.

11. The contrast for a point object against its background can be measured by: (1) its contrast ratio, which is the ratio between the higher of object transmittance or background transmittance to the lower of object transmittance or background transmittance; (2) its contrast difference, which is the difference between the higher density of object or background to the lower density of object or background; (3) its contrast modulation, which is the difference between the darker of object or background grey shade and the lighter of object or background grey shade divided by the sum of object grey shade and background grey shade.

12. Resolution is a generic term which describes how well a system, process, component or material, or image can reproduce an isolated object or separate closely spaced objects or lines. The limiting resolution, resolution limit or spatial resolution is described in terms of the smallest dimension of the target or object that can just be discriminated or observed. Resolution may be a function of object contrast, spatial position as well as element shape (single point, number of points in a cluster, continuum, or line etc.).

13. The resolving power of an imaging system, process, component or material is a measure of its ability to image closely spaced objects. The most common practice in measuring resolving power is to image a resolving power target composed of lines and spaces of equal width. Resolving power is usually measured at the image plane in line pairs per millimeter, i.e. the greatest number of lines and spaces per millimeter that can just be recognized. This threshold is usually determined by using a series of targets of decreasing size and basing the measurement on the smallest one in which all lines can be counted. In measuring resolving power the nature of the target (number of lines and their aspect ratio), its contrast and the criteria for determining the limiting resolving power must be specified.

14. Acuteness is a measure of the sharpness of edges in a photograph or image. It is defined for any edge by the average squared rate of change of the density across the edge divided by the total density difference from one side of the edge to the other side of the edge.

15. The spread function of an image system, process, component, or material describes the resulting spatial distribution of grey shade when the input to the system is some well defined object much smaller than the width of the spread function. If the input to the system is a line, the spread function is called the line spread function. If the input to the system is a point, the spread function is called the point spread function.

16. The Modulation Transfer Function of an imaging system or component measures the spatial frequency modulation response of the system or component. As an imaging system or component processes or records an image, the contrast modulation of the processed or recorded image is different from the input image. In fact, there is always a spatial frequency beyond which the contrast modulation of the processed or recorded (output) image is smaller (worse) than the contrast modulation of the input image. The modulation transfer function can be thought of as a curve indicating, for each spatial frequency, the ratio of the contrast modulation of the output image to the contrast modulation of the input image. It is formally defined as the magnitude of the Fourier transform of the line spread function of the imaging system or component.

17. A resolution cell is the smallest most elementary areal constituent of grey shades considered by an investigator in an image. A resolution cell is referenced by its spatial coordinates. The resolution cell or formations of resolution cells can sometimes constitute the basic unit for pattern recognition of image format data.
18. A digital image, or digitized image, or digital picture function of an image is an image in digital format and is obtained by partitioning the area of the image into a finite two-dimensional array of small uniformly shaped mutually exclusive regions, called resolution cells, and assigning a "representative" grey shade to each such spatial region. A digital image may be abstractly thought of as a function whose domain is the finite two-dimensional set of resolution cells and whose range is the set of grey shades.

19. A picture element or pixel or pel is a pair whose first member is a resolution cell and whose second member is the grey shade assigned by the digital image to that resolution cell. Sometimes picture element, pixel, or pel refer only to the grey shade or grey shade n-tuple in a resolution cell.

20. A multi-image is a set of images, each taken of the same subject at different times, or from different positions, or with different sensors, or at different electromagnetic frequencies, or with different polarizations. Although there is a high degree of information redundancy between images in a multi-image set, each image usually has information not available in any one of or combinations of the other images in the set.

21. A multi-digital image is a multi-image in digital form. It can be, for example, a set of digital images obtained from the images in a multi-image. A multi-digital image is often called a multi-image for short when it is understood from context that digital images are involved.

22. A flying spot scanner is a device used to rapidly convert image data from photographic format to electronic video signal format. Normally, the scanner directs an electron beam across the face of a cathode ray tube (CRT) in a TV-like raster. The photographic transparency is placed in front of the CRT (either directly or through some optics) and the light coming from the CRT is passed through it. The modulated light beam is detected by a photomultiplier or other photo detector and amplified to a usable video signal level.

23. A scanning densitometer is a device used to convert image data from transparency photographic format to electronic video signal format. Usually, the photographic transparency is placed on a glass cylinder which rotates and slowly translates. A fine beam of light is focused on the transparency, passed through it, and is detected by a photo-multiplier where it is amplified to a usable video signal. The scanning densitometer is a much slower conversion device than the flying spot scanner. However, this disadvantage is compensated by its fine resolution capability of a few microns.

24. The vidicon is an imaging vacuum tube having a photosensitive surface and is a means of converting image data from instantaneous radiance format to electronic video signal format. The scene being viewed is imaged on the photosensitive surface which can be scanned by an electron beam generating a signal whose amplitude corresponds to the radiant intensity focused on the surface at each point. This signal is called a video signal and may be amplified to any desired level.

25. A video image is an image in electronic signal format capable of being displayed on a cathode ray tube screen. The video signal is generated from devices like a vidicon or flying spot scanner which converts an image from photographic form to video signal form by scanning it line by line. The video signal itself is a sequence of signals, the ith signal representing the ith line of the scanned image.

26. Registering is the translation–rotation alignment process by which two images of like geometries and of the same set of objects are positioned coincident with respect to one another so that corresponding elements of the same ground area appear in the same place on the registered images. In this manner, the corresponding grey shades of the two images...
at any \((x, y)\) coordinate or resolution cell will represent the sensor output for the same object over the full image frame being registered.

27. Congruencing is the process by which two images of a multi-image set are transformed so that the size and shape of any object on one image is the same as the size and shape of that object on the other image. In other words, when two images are congruenced, their geometries are the same and they coincide exactly.

28. Rectifying is a process by which the geometry of an image area is made planimetric. For example, if the image is taken of an equally spaced rectangular grid pattern, then the rectified image will be an image of an equally spaced rectangular grid pattern. Rectification does not remove relief distortion.

29. Change detection is the process by which two images may be compared, resolution cell by resolution cell, and an output generated whenever corresponding resolution cells have different enough grey shades or grey shade \(n\)-tuples.

30. An optical color combiner is an instrument which produces "false" or "true" color images by linearly combining a few black and white transparencies of the same scene. The transparencies are usually obtained from multi-spectral, multi-band, or time-sequential photography. The transparencies are placed in projectors which are all focused and registered on the same screen and which have various color filters placed in front of their lenses. The viewing brightness of the projector's lamp in each projector can be changed independently thereby changing chromaticity balance. An optical color combiner is sometimes called an additive color display.

31. An electronic color combiner is an instrument which produces a "false" color image by linearly combining video signals of images of the same scenes. The images are usually obtained from multi-spectral, multi-band, or time-sequential photography. If the original image format is photographic, then the image format is changed from photographic to video signal format by synchronized vidicons or flying spot scanners. The resulting video signals are linearly combined through a matrix multiplier circuit, and the three linearly combined signals then drive the color gun of a color TV tube. An electronic color combiner usually has greater versatility for congruencing or registering than an optical color combiner.

32. Level slicing or density slicing or thresholding is an operation performed by an instrument (usually electronic) called a level slicer to change one or more a grey scale images to one binary image.

33. The level slicer, density slicer or thresholder is an instrument (usually electronic) which takes a single or multi-image as an input and produces a binary image for an output. A binary "one" is produced on the output image whenever the grey shades on each of the input images lie within the independently set minimum and maximum thresholds. A set of \(N\) input images would, therefore, require a setting \(N\) minimum and \(N\) maximum levels.

34. A figure \(F\), or a subimage \(F\) in a continuous or digital image \(I\) is any function \(F\) whose domain is some subset \(A\) of the set of spatial coordinates or resolution cells, whose range is the set \(G\) of grey shades, and which is defined by \(F(x, y) = I(x, y)\) for any \((x, y)\) belonging to \(A\).

35. A figure \(F\) is connected if there is a path between any two spatial coordinates or resolution cells contained in the domain of \(F\). More precisely, \(F\) is connected if for each pair of spatial coordinates \((x, y)\) and \((u, v)\) belonging to the domain of \(F\), there exists some sequence \((a_1, b_1), (a_2, b_2), \ldots, (a_m, b_m)\) of spatial coordinates belonging to the domain.
of $F$ such that $(x, y) = (a_i, b_i), (u, v) = (a_m, b_m)$, and $(a_i, b_i)$ and $(a_i+1, b_i+1)$ are sufficiently close neighboring coordinates, $i = 1, 2, \ldots, m-1$.

36. A figure $F$ is convex if the domain of $F$ contains the line segment which joins any pair of spatial coordinates in the domain of $F$.

37. A discrete tonal feature on a continuous or digital image is a connected set of spatial coordinates or resolution cells all of which have the same or almost the same grey shade.

38. Texture is concerned with the spatial distribution of the grey shades and discrete tonal features. When a small area of the image has little variation of discrete tonal features, the dominant property of that area is grey shade. When a small area has wide variation of discrete tonal features, the dominant property of that area is texture. There are three things crucial in this distinction: (1) the size of the small areas, (2) the relative sizes of the discrete tonal features, and (3) the number of distinguishable discrete tonal features.

39. Quantizing is the process by which each grey shade in an image of photographic, video, or digital format is assigned a new value from a given finite set of grey shade values. There are three often used methods of quantizing:

(1) in equal interval quantizing or linear quantizing, the range of grey shades from maximum grey shade to minimum grey shade is divided into contiguous intervals each of equal length, and each grey shade is assigned to the quantized class which corresponds to the interval within which it lies;

(2) in equal probability quantizing, the range of grey shades is divided into contiguous intervals such that after the grey shades are assigned to their quantized class there is an equal frequency of occurrence for each quantized grey shade in the quantized digital image or photograph; equal probability quantizing is sometimes called central stretching;

(3) in minimum variance quantizing, the range of grey shades is divided into contiguous intervals such that the weighted sum of the variance of the quantized intervals is minimized. The weights are usually chosen to be the grey shade interval probabilities which are computed as the proportional area on the photograph or digital image which have grey shades in the given interval.

40. A quantizer is an instrument which does quantizing. The quantizer has three functional parts. The first part allows the determining and/or setting of the quantizing intervals, the second part is a level slicer which indicates when a signal is in any quantizing interval, and the third part takes the binary output from the level slicers and either codes it to some binary code or converts it to some analog signal representing quantizing interval centers or means.

41. The simplest and most practical unit to observe and measure in the pattern recognition of image data is often the basic picture element (the grey shade or the grey shade $n$-tuple in its particular resolution cell). This is what makes pattern recognition so hard sometimes for the objects requiring analysis or identification are not simple picture elements but are often complex spatial formations of picture elements such as houses, roads, forest, etc.

42. A measurement $n$-tuple or measurement pattern or pattern or measurement vector is the ordered $n$-tuple of measurements obtained of a unit under observation. Each component of the $n$-tuple is a measurement of a particular quality, property, feature, or characteristic of the unit. In image pattern recognition, the units are usually picture elements or simple formations of picture elements and the measurement $n$-tuples are the corresponding grey shades, grey shade $n$-tuples, or formations of grey shade $n$-tuples.
43. The range set $R_i$ for the $i$th sensor which produces the $i$th image in the multiimage set, is the set of all measurements which can be produced by the $i$th sensor. Simply, it is the set of all grey shades which could possibly exist on the $i$th image.

44. The Cartesian product of two sets $A$ and $B$, denoted by $A \times B$, is the set of all ordered pairs where the first component of the pair is some element from the first set and the second component of the pair is some element from the second set. The Cartesian product of $N$ sets can be inductively defined in the usual fashion.

45. Measurement space is a set large enough to include in it the set of all possible measurement $n$-tuples which could be obtained by observing physical attributes of some set of units. When the units are single resolution cells or picture elements, measurement space $M$ is the Cartesian product of the range sets of the sensors; $M = R_1 \times R_2 \times \cdots \times R_n$.

46. Each unit is assumed to be of one and only one given type. The set of types is called the set of pattern classes or categories $C$, each type being a particular category. The categories are chosen specifically by the investigator as being the ones of interest to him.

47. A feature or feature pattern or feature $n$-tuple or pattern feature is a $n$-tuple or vector with (a small number of) components which are functions of the initial measurement pattern variables or some subsequence of the measurement $n$-tuples. Feature $n$-tuples or vectors are designed to contain a high amount of information relative to the discrimination between units of the types of categories in the given category set. Sometimes the features are predetermined and other times they are determined at the time the pattern discrimination problem is being solved. In image pattern recognition, features often contain information relative to grey shade, texture, shape or context.

48. Feature space is the set of all possible feature $n$-tuples.

49. Feature selection is the process by which the features to be used in the pattern recognition problem are determined. Sometimes feature selection is called property selection.

50. Feature extraction is the process in which an initial measurement pattern or some subsequence of measurement patterns is transformed to a new pattern feature. Sometimes feature extraction is called property extraction.

51. The word pattern can be used in three distinct senses:
(1) as measurement pattern;
(2) as feature pattern; and
(3) as the dependency pattern or patterns of relationships among the components of any measurement $n$-tuple or feature $n$-tuple derived from units of a particular category and which are unique to those $n$-tuples, that is, they are dependencies which do not occur in any other category.

52. A signature is the observable or characteristic measurement or feature pattern derived from units of a particular category. A category is said to have a signature only if the characteristic pattern is highly representative of the $n$-tuples obtained from units of that category. Sometimes a signature is called a prototype pattern.

53. A data sequence $S_d = \langle d_1, d_2, \ldots, d_J \rangle$ is a sequence of patterns derived from the measurement patterns or features of some sequence of observed units. $d_1$ is the pattern associated with the first unit; $d_2$ is the pattern associated with the second unit; and $d_J$ is the pattern associated with the $J$th unit.

54. A decision rule $f$ usually assigns one and only one category to each observed unit on the basis of the sequence of measurement patterns in the data sequence $S_d$ or in the corresponding sequence of feature patterns.
55. A simple decision rule is a decision rule which assigns a unit to a category solely on the basis of the measurements or features associated with the unit. Hence, the units are treated independently and the decision rule may be thought of as a function which assigns one and only one category to each pattern in measurement space or to each feature in feature space.

56. A compound decision rule is a decision rule which assigns a unit to a category on the basis of some non-trivial subsequence of measurement patterns in the data sequence or in the corresponding sequence of feature patterns. A compound decision rule is not a simple decision rule.

57. Provision can be made for the decision rule to reserve judgement or to defer assignment if the pattern is too close to the category boundary in measurement or feature space. With this provision, a deferred assignment is an assignment to the category of "reserved judgement."

58. A category identification sequence or ground truth \( S = \langle c_1, c_2, \ldots, c_J \rangle \) is a sequence of category identifications obtained from some sequence of observed units. \( c_1 \) is the category identification of the first unit; \( c_2 \) is the category identification of the second unit; and \( c_J \) is the category identification of the \( J \)th unit.

59. A training sequence is a set of two sequences: (1) the data sequence and (2) a corresponding category identification sequence (sometimes called ground truth). The training sequence is used to estimate the category conditional probability distributions from which the decision rule is constructed.

60. The conditional probability of a measurement or feature \( n \)-tuple \( d \) given category \( c \) is denoted by \( P(d|c) \), and is defined as the relative frequency or proportion of times the \( n \)-tuple \( d \) is derived from a unit whose true category identification is \( c \).

61. A distribution-free or non-parametric decision rule is one which makes no assumptions about the functional form of the conditional probability distribution of the patterns given the categories.

62. A simple maximum likelihood decision rule is one which treats the units independently and assigns a unit \( u \) having pattern measurement or features \( d \) to that category \( c \) whose units are most probable to have given rise to pattern or feature vector \( d \), that is, such that the conditional probability of \( d \) given \( c \), \( P(d|c) \), is highest.

63. A simple Bayes decision rule is one which treats the units independently and assigns a unit \( u \) having pattern measurements or features \( d \) to the category \( c \) whose conditional probability, \( P(d|c) \), given measurement \( d \), is highest.

64. Let \( \langle u_1, u_2, \ldots, u_J \rangle \) be a sequence of units with corresponding data sequence \( \langle d_1, d_2, \ldots, d_I \rangle \) and known category identification sequence \( \langle c_1, c_2, \ldots, c_J \rangle \). A simple nearest neighbor decision rule is one which treats the units independently and assigns a unit \( u \) of unknown identification and with pattern measurements or features \( d \) to category \( c_j \) where \( d_j \) is that pattern closest to \( d \) by some given metric or distance function.

65. A discriminant function \( f(d) \) is a scalar function, whose domain is usually measurement space and whose range is usually the real numbers. When \( f(d) \geq f_k(d) \), \( k = 1, 2, \ldots, K \), then the decision rule assigns the \( k \)th category to the unit giving rise to pattern \( d \).

66. A linear discriminant function \( f \) is a discriminant function of the form

\[
f(d) = b + \sum_{j=1}^{n} a_j \delta_j \quad \text{where} \quad d = (\delta_1, \delta_2, \ldots, \delta_n).
\]
67. A decision boundary between the \( i \)th and \( k \)th categories is a subset \( H \) of patterns in measurement space \( M \) defined by

\[
H = \{d \in M | f_i(d) = f_k(d)\},
\]

where \( f_i \) and \( f_k \) are the discriminant functions for the \( i \)th and \( k \)th categories.

68. A hyperplane decision boundary is the special name given to decision boundaries arising from the use of linear discriminant functions.

69. A linear decision rule is a simple decision rule which usually treats the units independently and makes the category assignments using linear discriminant functions. The decision boundaries obtained from linear decision rules are hyperplanes.

70. The pattern discrimination problem is concerned with how to construct the decision rule which assigns a unit to a particular category on the basis of the measurement pattern(s) in the data sequence or on the basis of the category pattern(s) in the data sequence.

71. Pattern identification is the process in which a decision rule is applied. If \( S_u = \langle u_1, u_2, \ldots, u_J \rangle \) is the sequence of units to be observed and identified, and if \( S_d = \langle d_1, d_2, \ldots, d_J \rangle \) is the corresponding data sequence of patterns, then the pattern identification process produces a category identification sequence \( S_c = \langle c_1, c_2, \ldots, c_J \rangle \) where \( c_j \) is the category in \( C \) to which the decision rule assigns unit \( u_j \) on the basis of the \( J \) patterns in \( S_d \). In general, each category in \( S_c \) can be assigned by the decision rule as a function of all the patterns in \( S_d \). Sometimes pattern identification is called "pattern classification" or "classification".

72. A cluster is a homogeneous group of units which are very "like" one another. "Likeness" between units is usually determined by the association, similarity, or distance between the measurement patterns associated with the units.

73. A cluster assignment function is a function which assigns each observed unit to a cluster on the basis of the measurement pattern(s) in the data sequence or on the basis of their corresponding features. Sometimes the units are treated independently; in this case the clustering assignment function can be considered as a transformation from measurement space to the set of clusters.

74. The pattern classification problem is concerned with constructing the cluster assignment function which groups similar units. Pattern classification is synonymous with numerical taxonomy or clustering.

75. The cluster identification process is the process in which the cluster assignment function is applied to the sequence of observed units thereby yielding a cluster identification sequence.

76. A misidentification, or misdetection, or type I error occurs for category \( c_i \) if a unit whose true category identification is \( c_i \) is assigned by the decision rule to category \( c_k \), \( k \neq i \). A misidentification error is often called an error of omission.

77. A false identification, or false alarm, or type II error occurs for category \( c_i \) if a unit whose true category identification is \( c_k \), \( k \neq i \), is assigned by the decision rule to category \( c_i \). A false identification error is often called an error of commission.

78. A prediction sequence, or test sequence, or a generalization sequence is a set of two sequences: (1) a data sequence (whose corresponding true category identification sequence may be considered to be unknown to the decision rule) and (2) a corresponding category identification sequence determined by the decision rule assignment. By comparing the category identification sequence determined by the decision rule assignment with the
category identification sequence determined by the ground truth, the misidentification rate and the false identification rate for each category may be estimated.

79. A confusion matrix or contingency table is an array of probabilities whose rows and columns are both similarly designated by category label and which indicates the probability of correct identification for each category as well as the probability of type I and type II errors. The \( P_{jk} \) is the probability that a unit has true category identification \( c_j \); and is assigned by the decision rule to category \( c_k \).

80. A unit is said to be detected if the decision rule is able to assign it as belonging only to some given subset \( A \) of categories from the set \( C \) of categories. To detect a unit does not imply that the decision rule is able to identify the unit as specifically belonging to one particular category.

81. A unit is said to be recognized, identified, classified, categorized or sorted if the decision rule is able to assign it to some category from the set of given categories. In military applications, there is a definite distinction between recognize and identify. Here, for a unit to be recognized, the decision rule must be able to assign it to a type of category, the type having included within it many subcategories. For a unit to be identified, the decision rule must be able to assign it not only to a type of category but also to the subcategory of the category type. For example, a small area ground patch may be recognized as containing trees, which may be specifically identified as apple trees.

82. A unit is said to be located if specific coordinates can be given for the units physical location.

83. A unit is said to be acquired if it can be located and recognized.

84. A target is one type of category used in the pattern recognition of image data. It usually occupies some relatively small area on the image and has a unique or characteristic set of attributes. It has a high a priori interest to the investigator.

85. Target discrimination is the process by which decision rules for targets (small area extensive categories) are constructed.

86. Target identification or target recognition is the process by which targets contained within image data are identified by means of a decision rule.

87. An image transformation is a function or operator which takes an image for its input and produces an image for its output. The domain of the transform operator is often called the spatial domain. The range of the transform operator is often called the transformed domain. Some transformations have spatial and transform domains of entirely different character. For these transforms, the image in the spatial domain may appear entirely different from and have a different interpretation from the image in the transformed domain. Specific examples of these kinds of transformations are the Fourier, Hadamard, and Karhunen-Loève transformations. Other transformations have spatial and transform domain of similar character. For these transformations, the image in the transformed domain may appear similar to the image of the spatial domain. These types of transformations are often called spatial filters.

88. A spatial filter is an image transformation, usually a one-one operator used to lessen noise or enhance certain characteristics of the image. For any particular \( (x, y) \) coordinate on the transformed image, the spatial filter assigns a grey shade on the basis of the grey shades of a particular spatial pattern near the coordinates \( (x, y) \).

89. A linear spatial filter is a spatial filter for which the grey shade assignment at coordinates \( (x, y) \) in the transformed image is made by some weighted average (linear combination) of grey shades located in a particular spatial pattern around coordinates \( (x, y) \) of the domain image. The linear spatial filter is often used to change the spatial
frequency characteristics of the image. For example, a linear spatial filter which emphasizes high spatial frequencies will tend to sharpen the edges in an image. A linear spatial filter which emphasizes the low spatial frequencies will tend to blur the image and reduce salt and pepper noise.

90. **Template matching** is an operation which can be used to find out how well two photographs or images match one another. The degree of matching is often determined by cross-correlating the two images or by evaluating the sum of the squared corresponding grey shade differences. Template matching can also be used to best match a measurement pattern with a prototype pattern.

91. **Matched filtering** is a template matching operation done by using the magnitude of the cross-correlation function to measure the degree of matching.

92. In pattern recognition problems such as target discrimination, for which the category of interest is some specific formation of resolution cells with characteristic shape or tone-texture composition, the problem of pattern segmentation may occur. **Pattern segmentation** is the problem of determining which regions or areas in the image constitute the patterns of interest, i.e. which resolution cells should be included and which excluded from the pattern measurements.

93. **Screening** is the operation of separating the uninteresting photographs or images from those photographs containing areas of potential interest.

94. **Preprocessing** is an operation applied before pattern identification is performed. Preprocessing produces, for the categories of interest, pattern features which tend to be invariant under changes such as translation, rotation, scale, illumination levels, and noise. In essence, preprocessing converts the measurements patterns to a form which allows a simplification in the decision rule. Preprocessing can bring into registration, bring into congruence, remove noise, enhance images, segment target patterns, detect, center, and normalize targets of interest.

95. **Image compression** is an operation which preserves all or most of the information in the image and which reduces the amount of memory needed to store an image or the time needed to transmit an image.

96. **Image restoration** is a process by which a degraded image is restored to its original condition. Image restoration is possible only to the extent that the degradation transform is mathematically invertible.

97. **Image enhancement** is any one of a group of operations which improve the detectability of the targets or categories. These operations include, but are not limited to, contrast improvement, edge enhancement, spatial filtering, noise suppression, image smoothing, and image sharpening.

98. **Image processing** encompasses all the various operations which can be applied to photographic or image data. These include, but are not limited to, image compression, image restoration, image enhancement, preprocessing, quantization, spatial filtering, and other image pattern recognition techniques.

99. **Interactive Image Processing** refers to the use of an operator or analyst at a console with a means of assessing, preprocessing, feature extracting, classifying, identifying and displaying the original imagery or the processed imagery for his subjective evaluation and further interactions.

100. **Pattern recognition** is concerned with, but not limited to, problems of: (1) pattern discrimination, (2) pattern classification, (3) feature selection, (4) pattern identification, (5) cluster identification, (6) feature extraction, (7) preprocessing, (8) filtering, (9) enhancement, (10) pattern segmentation, or (11) screening.
# INDEX TO GLOSSARY OF REMOTELY SENSED IMAGE PATTERN RECOGNITION CONCEPTS

<table>
<thead>
<tr>
<th>Concept</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actuance</td>
<td>14</td>
</tr>
<tr>
<td>Acquire</td>
<td>83</td>
</tr>
<tr>
<td>Additive color display</td>
<td>30</td>
</tr>
<tr>
<td>Bayes decision rule</td>
<td>63</td>
</tr>
<tr>
<td>Category</td>
<td>46</td>
</tr>
<tr>
<td>Categorize</td>
<td>81</td>
</tr>
<tr>
<td>Category identification sequence</td>
<td>58</td>
</tr>
<tr>
<td>Cartesian product</td>
<td>44</td>
</tr>
<tr>
<td>Change detection</td>
<td>29</td>
</tr>
<tr>
<td>Classification</td>
<td>71</td>
</tr>
<tr>
<td>Classify</td>
<td>81</td>
</tr>
<tr>
<td>Cluster</td>
<td>72</td>
</tr>
<tr>
<td>Cluster assignment function</td>
<td>73</td>
</tr>
<tr>
<td>Cluster identification</td>
<td>75</td>
</tr>
<tr>
<td>Clustering</td>
<td>74</td>
</tr>
<tr>
<td>Compound decision rule</td>
<td>56</td>
</tr>
<tr>
<td>Conditional probability</td>
<td>60</td>
</tr>
<tr>
<td>Confusion matrix</td>
<td>79</td>
</tr>
<tr>
<td>Congruencing</td>
<td>27</td>
</tr>
<tr>
<td>Connectedness</td>
<td>35</td>
</tr>
<tr>
<td>Contingency table</td>
<td>79</td>
</tr>
<tr>
<td>Contrast</td>
<td>11</td>
</tr>
<tr>
<td>Contrast difference</td>
<td>11</td>
</tr>
<tr>
<td>Contrast modulation</td>
<td>11</td>
</tr>
<tr>
<td>Contrast ratio</td>
<td>11</td>
</tr>
<tr>
<td>Contrast stretching</td>
<td>39</td>
</tr>
<tr>
<td>Convex</td>
<td>36</td>
</tr>
<tr>
<td>Data sequence</td>
<td>53</td>
</tr>
<tr>
<td>Decision boundary</td>
<td>67</td>
</tr>
<tr>
<td>Decision rule</td>
<td>54</td>
</tr>
<tr>
<td>Defer assignment</td>
<td>57</td>
</tr>
<tr>
<td>Densitometer</td>
<td>10</td>
</tr>
<tr>
<td>Densitometry</td>
<td>9</td>
</tr>
<tr>
<td>Density</td>
<td>8</td>
</tr>
<tr>
<td>Density slicer</td>
<td>33</td>
</tr>
<tr>
<td>Density slicing</td>
<td>32</td>
</tr>
<tr>
<td>Detect</td>
<td>80</td>
</tr>
<tr>
<td>Digital image</td>
<td>18</td>
</tr>
<tr>
<td>Digital picture function</td>
<td>18</td>
</tr>
<tr>
<td>Digitized image</td>
<td>18</td>
</tr>
<tr>
<td>Discrete tonal feature</td>
<td>37</td>
</tr>
<tr>
<td>Discriminant function</td>
<td>65</td>
</tr>
<tr>
<td>Distribution-free decision rule</td>
<td>61</td>
</tr>
<tr>
<td>Electronic color combiner</td>
<td>31</td>
</tr>
<tr>
<td>Equal interval quantizing</td>
<td>39</td>
</tr>
<tr>
<td>Equal probability quantizing</td>
<td>39</td>
</tr>
<tr>
<td>Error of commission</td>
<td>77</td>
</tr>
<tr>
<td>Error of omission</td>
<td>76</td>
</tr>
<tr>
<td>False alarm</td>
<td>77</td>
</tr>
<tr>
<td>False identification</td>
<td>77</td>
</tr>
<tr>
<td>Feature</td>
<td>47</td>
</tr>
<tr>
<td>Feature extraction</td>
<td>50</td>
</tr>
<tr>
<td>Feature n-tuple</td>
<td>47</td>
</tr>
<tr>
<td>Feature pattern</td>
<td>47</td>
</tr>
<tr>
<td>Feature selection</td>
<td>49</td>
</tr>
<tr>
<td>Feature space</td>
<td>48</td>
</tr>
<tr>
<td>Figure</td>
<td>34</td>
</tr>
<tr>
<td>Flying spot scanner</td>
<td>22</td>
</tr>
<tr>
<td>Generalization sequence</td>
<td>78</td>
</tr>
<tr>
<td>Grey shade</td>
<td>2</td>
</tr>
<tr>
<td>Grey tone</td>
<td>2</td>
</tr>
<tr>
<td>Ground truth</td>
<td>58</td>
</tr>
<tr>
<td>Hyperplane decision boundary</td>
<td>68</td>
</tr>
<tr>
<td>Identify</td>
<td>81</td>
</tr>
<tr>
<td>Image</td>
<td>1</td>
</tr>
<tr>
<td>Image compression</td>
<td>95</td>
</tr>
<tr>
<td>Image enhancement</td>
<td>97</td>
</tr>
<tr>
<td>Image processing</td>
<td>98</td>
</tr>
<tr>
<td>Image restoration</td>
<td>96</td>
</tr>
<tr>
<td>Image transformation</td>
<td>87</td>
</tr>
<tr>
<td>Interactive image processing</td>
<td>99</td>
</tr>
<tr>
<td>Level slicer</td>
<td>33</td>
</tr>
<tr>
<td>Level slicing</td>
<td>32</td>
</tr>
<tr>
<td>Limiting resolution</td>
<td>12</td>
</tr>
<tr>
<td>Line spread function</td>
<td>15</td>
</tr>
<tr>
<td>Linear decision rule</td>
<td>69</td>
</tr>
<tr>
<td>Linear discriminant function</td>
<td>66</td>
</tr>
<tr>
<td>Linear spatial filter</td>
<td>89</td>
</tr>
<tr>
<td>Linear quantizing</td>
<td>39</td>
</tr>
<tr>
<td>Locate</td>
<td>82</td>
</tr>
<tr>
<td>Map</td>
<td>4</td>
</tr>
<tr>
<td>Term</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Matched filtering</td>
<td>91</td>
</tr>
<tr>
<td>Maximum likelihood decision rule</td>
<td>62</td>
</tr>
<tr>
<td>Measurement n-tuple</td>
<td>42</td>
</tr>
<tr>
<td>Measurement pattern</td>
<td>42</td>
</tr>
<tr>
<td>Measurement space</td>
<td>45</td>
</tr>
<tr>
<td>Measurement vector</td>
<td>42</td>
</tr>
<tr>
<td>Micro-densitometer</td>
<td>10</td>
</tr>
<tr>
<td>Minimum variance quantizing</td>
<td>39</td>
</tr>
<tr>
<td>Misdetection</td>
<td>76</td>
</tr>
<tr>
<td>Misidentification</td>
<td>76</td>
</tr>
<tr>
<td>Modulation transfer function</td>
<td>16</td>
</tr>
<tr>
<td>Multi-digital image</td>
<td>21</td>
</tr>
<tr>
<td>Multi-image</td>
<td>20</td>
</tr>
<tr>
<td>Nearest neighbor decision rule</td>
<td>64</td>
</tr>
<tr>
<td>Non-parametric decision rule</td>
<td>61</td>
</tr>
<tr>
<td>Numerical taxonomy</td>
<td>74</td>
</tr>
<tr>
<td>Optical color combiner</td>
<td>30</td>
</tr>
<tr>
<td>Pattern</td>
<td>42, 51</td>
</tr>
<tr>
<td>Pattern class</td>
<td>46</td>
</tr>
<tr>
<td>Pattern classification</td>
<td>71, 74</td>
</tr>
<tr>
<td>Pattern discrimination</td>
<td>70</td>
</tr>
<tr>
<td>Pattern feature</td>
<td>47</td>
</tr>
<tr>
<td>Pattern identification</td>
<td>71</td>
</tr>
<tr>
<td>Pattern recognition</td>
<td>100</td>
</tr>
<tr>
<td>Pattern segmentation</td>
<td>92</td>
</tr>
<tr>
<td>Pel</td>
<td>19</td>
</tr>
<tr>
<td>Photograph</td>
<td>3</td>
</tr>
<tr>
<td>Photomap</td>
<td>4</td>
</tr>
<tr>
<td>Picture element</td>
<td>19</td>
</tr>
<tr>
<td>Pixel</td>
<td>19</td>
</tr>
<tr>
<td>Point spread function</td>
<td>15</td>
</tr>
<tr>
<td>Prediction sequence</td>
<td>78</td>
</tr>
<tr>
<td>Preprocessing</td>
<td>94</td>
</tr>
<tr>
<td>Property extraction</td>
<td>50</td>
</tr>
<tr>
<td>Property selection</td>
<td>49</td>
</tr>
<tr>
<td>Prototype pattern</td>
<td>52</td>
</tr>
<tr>
<td>Quantizer</td>
<td>40</td>
</tr>
<tr>
<td>Quantizing</td>
<td>39</td>
</tr>
<tr>
<td>Radiant intensity</td>
<td>5</td>
</tr>
<tr>
<td>Range set</td>
<td>43</td>
</tr>
<tr>
<td>Recognize</td>
<td>81</td>
</tr>
<tr>
<td>Rectifying</td>
<td>28</td>
</tr>
<tr>
<td>Reflectance</td>
<td>6</td>
</tr>
<tr>
<td>Reflection coefficient</td>
<td>6</td>
</tr>
<tr>
<td>Registering</td>
<td>26</td>
</tr>
<tr>
<td>Reserve judgement</td>
<td>57</td>
</tr>
<tr>
<td>Resolution</td>
<td>12</td>
</tr>
<tr>
<td>Resolution cell</td>
<td>17</td>
</tr>
<tr>
<td>Resolution limit</td>
<td>12</td>
</tr>
<tr>
<td>Resolving power</td>
<td>13</td>
</tr>
<tr>
<td>Scanning densitometer</td>
<td>23</td>
</tr>
<tr>
<td>Screening</td>
<td>93</td>
</tr>
<tr>
<td>Segmentation</td>
<td>92</td>
</tr>
<tr>
<td>Signature</td>
<td>52</td>
</tr>
<tr>
<td>Simple decision rule</td>
<td>55</td>
</tr>
<tr>
<td>Sort</td>
<td>81</td>
</tr>
<tr>
<td>Spatial filter</td>
<td>88</td>
</tr>
<tr>
<td>Spatial resolution</td>
<td>12</td>
</tr>
<tr>
<td>Spread function</td>
<td>15</td>
</tr>
<tr>
<td>Subimage</td>
<td>34</td>
</tr>
<tr>
<td>Target</td>
<td>84</td>
</tr>
<tr>
<td>Target discrimination</td>
<td>85</td>
</tr>
<tr>
<td>Target identification</td>
<td>86</td>
</tr>
<tr>
<td>Target recognition</td>
<td>86</td>
</tr>
<tr>
<td>Template matching</td>
<td>90</td>
</tr>
<tr>
<td>Test sequence</td>
<td>78</td>
</tr>
<tr>
<td>Texture</td>
<td>38</td>
</tr>
<tr>
<td>Thresholder</td>
<td>33</td>
</tr>
<tr>
<td>Thresholding</td>
<td>32</td>
</tr>
<tr>
<td>Training sequence</td>
<td>59</td>
</tr>
<tr>
<td>Transmittance</td>
<td>7</td>
</tr>
<tr>
<td>Transmittance coefficient</td>
<td>7</td>
</tr>
<tr>
<td>Type I error</td>
<td>76</td>
</tr>
<tr>
<td>Type II error</td>
<td>77</td>
</tr>
<tr>
<td>Unit</td>
<td>41</td>
</tr>
<tr>
<td>Video image</td>
<td>25</td>
</tr>
<tr>
<td>Video signal</td>
<td>24</td>
</tr>
<tr>
<td>Vidicon</td>
<td>24</td>
</tr>
</tbody>
</table>
REFERENCES


