STATISTICAL IMAGE PROCESSING

Submitted to:
Statistics and Probability Program
Office of Naval Research, Code 436
Arlington, VA 22217

Prepared by:
Professor C. H. Chen (Principal Investigator)
Dept. of Electrical and Computer Engineering
Southeastern Massachusetts University
North Dartmouth, MA 02747

Unclassified
Key words: image models, image segmentation, maximum entropy spectral analysis, texture image analysis and classification, image software package, automatic spatial clustering.

This document has been approved for public release and sales; its distribution is unlimited.
<table>
<thead>
<tr>
<th>REPORT DOCUMENTATION PAGE</th>
<th>READ INSTRUCTIONS BEFORE COMPLETING FORM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. REPORT NUMBER</strong></td>
<td></td>
</tr>
<tr>
<td><strong>2. GOVT ACCESSION NO.</strong></td>
<td>AP-A/12/1999</td>
</tr>
<tr>
<td><strong>3. RECIPIENT'S Catalog Number</strong></td>
<td></td>
</tr>
<tr>
<td><strong>4. TITLE (and Subtitle)</strong></td>
<td>Statistical Image Processing -- Final Report</td>
</tr>
<tr>
<td><strong>5. TYPE OF REPORT &amp; PERIOD COVERED</strong></td>
<td>Final Report</td>
</tr>
<tr>
<td><strong>6. PERFORMING ORG. REPORT NUMBER</strong></td>
<td></td>
</tr>
<tr>
<td><strong>7. AUTHOR(s)</strong></td>
<td>C. H. CHEN</td>
</tr>
<tr>
<td><strong>8. CONTRACT OR GRANT NUMBER(s)</strong></td>
<td>N00014-79-C-0494</td>
</tr>
<tr>
<td><strong>9. PERFORMING ORGANIZATION NAME AND ADDRESS</strong></td>
<td>Electrical and Computer Engineering Dept., Southeastern Massachusetts University, N. Dartmouth, Mass. 02747</td>
</tr>
<tr>
<td><strong>10. PROGRAM ELEMENT, PROJECT, TASK AREA &amp; WORK UNIT NUMBERS</strong></td>
<td>N042-422</td>
</tr>
<tr>
<td><strong>11. CONTROLLING OFFICE NAME AND ADDRESS</strong></td>
<td>Statistics and Probability Program, Office of Naval Research, Arlington, VA 22217</td>
</tr>
<tr>
<td><strong>12. REPORT DATE</strong></td>
<td>Nov. 16, 1982</td>
</tr>
<tr>
<td><strong>13. NUMBER OF PAGES</strong></td>
<td></td>
</tr>
<tr>
<td><strong>14. MONITORING AGENCY NAME &amp; ADDRESS (if different from Controlling Office)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>15. SECURITY CLASS. (of this report)</strong></td>
<td>Unclassified</td>
</tr>
<tr>
<td><strong>16. DISTRIBUTION STATEMENT (of this Report)</strong></td>
<td>Approved for Release; distribution unlimited.</td>
</tr>
<tr>
<td><strong>17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>18. SUPPLEMENTARY NOTES</strong></td>
<td></td>
</tr>
<tr>
<td><strong>19. KEY WORDS (Continue on reverse side if necessary and identify by block number)</strong></td>
<td>see cover page</td>
</tr>
<tr>
<td><strong>20. ABSTRACT (Continue on reverse side if necessary and identify by block number)</strong></td>
<td>see cover page</td>
</tr>
</tbody>
</table>
Abstract

This report summarizes the major research work accomplished under the contract including the topics of statistical image models, comparative evaluation of image processing techniques, image segmentation algorithms, two-dimensional maximum entropy spectral analysis, and spatial clustering algorithms with applications to artificial and remotely sensed images. Detailed list of publications available in open literature is provided. A list of software package generated is included in the Appendices.

I. Introduction

This report is organized according to the topics we have worked under this Contract. A brief summary of each is presented. Detailed list of publications is provided in References. Over 40 technical reports prepared under the Contract are not listed however as most results documented in reports were published as listed in References. Copies of two papers are included in the Appendices. A detailed list of image processing software package generated is also included in the Appendices. The program listings in magnetic tapes were delivered to Dr. Doug DePriest in October 1981.

II. Major Research Results

1. Statistical image processing techniques for additive noise case were compared. Median filtering followed by Kalman adaptive filtering is most effective. For Seasat images the multivariate noise removal is considered by using local statistics. (Refs. 16, 20)

2. Statistical image segmentation studies include the extensive comparative evaluation of supervised and unsupervised segmentation techniques for texture and infrared images. The segmentation is performed by pixel classification. Both Fisher's linear discriminant and the maximum a posteriori estimation procedures are found to be very effective. Statistical techniques however are limited to pixel based segmentations. (Refs. 4, 9, 10, 12, 13, 14)

3. Statistical image modeling study is concerned with the auto-regressive models and low order ARMA models. Such modeling leads to image enhancement, segmentation and classification. These models provide a nice way to take into account the contextual dependence
among the nearest neighbors. The question remains whether the object boundary should have a separate model from the remaining homogeneous parts of the image. (Refs. 15,18)

4. An automatic spatial clustering algorithm has been developed for image segmentation and compression. The algorithm can determine the minimum number of clusters and can also work with a specified number of clusters. The algorithm has been successfully tested with various images including USC image data base, Seasat images, and U.S. Army topographic images. (Refs. 3,4)

5. A two-dimensional maximum entropy spectral analysis algorithm was thoroughly developed and tested for texture image analysis, classification, segmentation as well as general purpose spectral computation based on limited number of data points. (Refs. 1,2,4,5, 6,8)

6. An initial effort of tracking image sequence was made by using pixel classification for object extraction. Further study to model the statistics of image variation is much needed. (Ref. 4)

III References

The following is a list of publications by C.H. Chen with full or partial support of Contract N00014-79-C-0494.


6. "On a two-dimensional maximum entropy spectral estimation method for the texture-image analysis", presented at the Computer Science and Technology Conference, June 4-6, 1982 in Newton, MA.


Appendix A Master's Theses

The following is a list of master's theses completed under full or partial support of the Contract, under Prof. Chen's supervision.


Appendix B

A STUDY OF TEXTURE CLASSIFICATION USING SPECTRAL FEATURES

C. H. Chen

Electrical Engineering Department
Southeastern Massachusetts University
N. Dartmouth, Mass. 02747 U.S.A.

Abstract

Effort in the past on the use of spectral features for classification of texture images has had limited success and the feature measures computed from the co-occurrence matrix are preferred. In this paper the superiority of spectral features for texture classification is demonstrated. A new two-dimensional maximum entropy spectral analysis is developed which provides superior resolution capability. Thus accurate power spectrum can be determined from which various ring and wedge spectral features are computed. Extensive computer results reported indicate that the spectral features so computed provide not only a good measure of texture coarseness and directional- ity, but also comparable or better classification performance than that reported earlier. A typical performance of over 80 percent correct classification is available from the extracted spectral features by using the Fisher's linear discriminant for classification. A set of normalized features which use both ring and wedge features is particularly recommended. Computationally the method described in this paper is far better than the use of co-occurrence matrix as the iterative algorithm used for spectrum estimation is very fast, even with the use of a minicomputer.

I. Introduction

Although it is generally recognized that texture images contain statistical, spectral and structural domain information, the use of spectral information alone can be quite effective in the texture-image analysis studies such as texture discrimination and segmentation. Bajcsy and Liberman [1] expressed the power spectrum in polar coordinates, then integrate over r and $\phi$ to obtain the two-dimensional functions. The location of peaks in these functions indicates prominent texture coarseness and directionality. Weszka et al. [2] integrated the power spectrum within 16 spatial frequency sectors which were combinations of four 1-octave frequency ranges and four $90^\circ$ orientation sectors. They also computed eight "contrast" measures based on the cooccurrence matrix, and obtained better discrimination than with the power spectrum measures. Laws [3] computed a number of energy measures by filtering the texture with sets of small linear operators, then squaring and summing the output of each filter. He reported better discrimination with the energy than with the cooccurrence measures.

A fundamental problem with the power spectrum analysis is the computational accuracy and computational complexity. For texture study, accurate power spectrum must be computed from the small image segments. In this case, the two-dimensional Fourier analysis cannot provide sufficient accuracy as the Fourier analysis is more accurate with a large number of pixels. The two-dimensional maximum entropy spectral analysis, however, is very suitable for a small number of pixels. The computational complexity has been a drawback in using the two-dimensional maximum entropy spectral estimation methods. Recently, Lim and Malik [4] have proposed an efficient iterative algorithm for the two-dimensional maximum entropy power spectrum estimation which can obtain good resolution and sufficient accuracy for the finite sample two-dimensional data. A study of the spectral estimation of texture image has been proved to be successful [5] by using a minicomputer. In this paper, this method is used for the calculation of spectral features of texture image. In section II, the two-dimensional maximum entropy power spectrum estimation is briefly discussed. The method of selection of features will be described in section III while section IV provides some experimental results of the textural classification.

II. Two-Dimensional Maximum Entropy Power Spectrum Estimation

The basic concept of the maximum entropy method (MEM) of spectral estimation is to extrapolate the autocorrelation function of a random process by maximising the entropy $\mathbb{H}$ of the corresponding probability density function

$$\mathbb{H} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \log \hat{p}_x(w_1, w_2) dw_1 dw_2$$

(1)

where $\hat{p}_x(w_1, w_2)$ is the power spectrum estimate of the random process $x(n_2, n_3)$. The characteristics of this method are equivalent to the autoregressive signal modeling and the power spectrum is calculated by
where \( \lambda(a, b) \) is the autocorrelation whose power spectrum is \( 1/\lambda(w_1, w_2) \) and \( A \) is a set of points \((a, b)\) where the autocorrelation is known.

Since the filter coefficients cannot be obtained directly by solving the normal equation as in the one-dimensional case, Lim and Malik developed a new iterative algorithm, using adaptive filtering concepts. The basic idea of this algorithm is on the notion that the given correlation point in region \( A \) is consistent and the corresponding coefficient should be zero outside region \( A \) and proceed this iteration repeatedly until an optimal solution is obtained.

A simple flowchart is shown in Fig. 1. We begin with some initial estimate of \( \lambda(a, b) \), obtain the corresponding correlation function, correct the resulting correlation for \((a, b)\) with the known \( R(n_1, n_2) \), determine \( P_x(w_1, w_2) \) such that \( P_x(w_1, w_2) \) satisfy (2) and \( R_x(n_1, n_2) = P_x^{-1}[R_x(w_1, w_2)] \) for \((n_1, n_2)\) in \( A \).

In Lim and Malik's paper, the calculation of the autocorrelation \( R_x(n_1, n_2) \) is limited to the closed analytic form especially for the two-dimensional sinusoids. A generalization of this method and the application to a two-dimensional real data have been discussed by Chen and Young[5]. Even for a small number of missing correlation points, the algorithm can still provide an accurate spectrum. Figure 2a shows the spectrum of two sinusoids \((0.1256, 0.3456), (0.1255, 0.200) \) in white noise, based on a 5x5 correlation data set, with signal-to-noise ratio of 0.67. With the correlation points \((-1, -1) \) and \((1, 1) \) missing, Fig. 2b shows the resulting spectrum which is nearly identical to Fig. 2a.

III. Feature Selection and Classification Method

We use two features to classify the texture images. It is generally recognized that a coarse texture will have a high value of power spectrum near the origin while in a fine texture, the value will be more spread out. Thus, if one wishes to analyze texture coarseness, a set of features that should be useful are the averages of the power-spectrum values taken over a ring-shaped region centered at the origin. In this paper, we consider only the first quadrant of the power spectrum, then

\[
\hat{P}_x(w_1, w_2) = \frac{1}{\lambda(w_1, w_2)} \sum_{n_1, n_2 \in A} \lambda(n_1, n_2) e^{-j\omega_1 n_1} e^{-j\omega_2 n_2}
\]

(2)

for various values of \( r \), the ring radius.

For the discrete case, this can be written as

\[
\phi_{r_1, r_2} = \int P_x(r, \theta) d\theta
\]

(3)

for the various values of \( \theta \), the wedge slope.

For the discrete case, this is (the wedge between \( \theta_1 \) and \( \theta_2 \) given by

\[
\phi_{\theta_1, \theta_2} = \sum_{0<x<y<0} \hat{P}_x(x, y)
\]

(4)

The features calculated by (4) and (6) are sensitive to size and orientation respectively, but not to both. In order to obtain the comparable feature sets, we obtain a set of equalized features by taking the average over the intersection area of rings and wedges. These equalized features are also studied in section IV. After the calculation of features, the Fisher discriminant technique is used for classification [6].

IV. Experimental Results

Because of the computational requirements of the method and the limited memory capacity of the PDP 11/45, all test samples are stored in our DEC 20 system and sent through a communication line to the PDP 11/45 for the spectrum computation. The test samples are the texture images taken from the USC data base. To verify the sensitivity both in coarseness and directionality, we select some textures that contain such informations. The test samples contain six classes of texture (each one has four samples) and are shown in Fig. 3. Each data is a 32x32 array of gray level 0-255. The pictures of class 1 reappear but are two times larger in Fig. 4(a). Figure 4(b) is the corresponding estimated power-spectrum display of the upper left data in each class. The spectra of all classes are different either in radial or angular distribution [7].
The feature sets used in this paper are:

ring: \( \phi_{r_1, r_2} \) for \((r_1, r_2) = (1,3), (3,6), (6,12), (12,24), (24,48)\)

wedge: \( \phi_{\theta_1, \theta_2} \) for \((\theta_1, \theta_2) = (0,15), (15,30), (30,45), (45,60), (60,75), (75,90)\)

The maximum ring radius used is 48 since it already covers most part of a 64x64 array power spectrum.

A combination feature of ring and wedge has been tested for 30 pairs of feature values. Table I shows part of features which did higher than 19 out of 28 correct, i.e., more than 80% correct. Table II shows the best performing pairs using the same kind of features (ring and ring, wedge and wedge): there are 6 out of 25 pairs which classified correctly higher than 75%. Other pairs' results are concentrated near 12-17, i.e., more than 50% correct recognition. For the pairs that contain the wedge near the edges, the results are very good since the test samples give some directional information. Also for the rings a little farther from the origin, the results are better since it shows a large difference in the spectrum value there.

Equalized features are also tested: we used five rings intersected with three wedges (ring: \((1,3), (3,6), (6,12), (12,24), (24,48)\) and wedge: \((0,30), (30,60), (60,90)\)). 105 pairs of features have been tested. Table III shows the best performing pairs of which the best score, 23 out of 24, is 95% correct. From the results, we can see that the ring feature \((24,48)\) gives very useful classification information indicating that there exists a large textural variation in that region as the texture coarseness plays an important role in the pair.

V. Discussion

In this paper, we have observed that equalized features did better than unequalized ones for this test samples. It is shown that both the coarseness and the directionality are important factors in texture discrimination. For the consideration of practical use in automatic classification, various kinds of textures must be tested and compared with other methods using the non-spectral features. Another important factor which may influence the results is that if we increase the autocorrelation function and the discrete Fourier transform length while estimating the power spectrum, the accuracy and the resolution will be better. But there is a tradeoff between the accuracy and the computational time. In this paper, these parameters (i.e., autocorrelation function: 7x7, discrete Fourier transform length: 32) are chosen for the real-time processing purpose. Also the locations of the main and second components of frequencies can serve as another important features because they vary among different textures.

Acknowledgement: This work was supported by the ONR Statistics and Probability Program Contract No. N00014-79-C-0494. The programming assistance of Mr. Gia-Kinh Young is gratefully acknowledged.

References


Initial estimate of $\lambda(n_1,n_2)$

$$R_y(n_1,n_2) = y^{-1} \left( \frac{1}{F[\lambda(n_1,n_2)]} \right)$$

Correct $R_y(n_1,n_2)$ with $R_x(n_1,n_2)$ for $(n_1,n_2) \in A$

$$\lambda(n_1,n_2) = y^{-1} \left( \frac{1}{F[R_y(n_1,n_2)]} \right)$$

$\lambda(n_1,n_2) = 0$ for $(n_1,n_2) \in A$

$$P_x(n_1,n_2) = F[R_y(n_1,n_2)]$$

**Figure 1**

<table>
<thead>
<tr>
<th>Features</th>
<th>Number correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ring</strong></td>
<td></td>
</tr>
<tr>
<td>(24,48)</td>
<td>22</td>
</tr>
<tr>
<td>(3,6)</td>
<td>20</td>
</tr>
<tr>
<td>(6,12)</td>
<td>19</td>
</tr>
<tr>
<td>(12,24)</td>
<td>19</td>
</tr>
<tr>
<td>(3,6)</td>
<td>19</td>
</tr>
<tr>
<td>(6,12)</td>
<td>19</td>
</tr>
</tbody>
</table>

**Table I**: Best performing pairs using the combination feature of ring and wedge for those with more than 80% correct classification.

<table>
<thead>
<tr>
<th>Features</th>
<th>Number correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ring</strong></td>
<td></td>
</tr>
<tr>
<td>(6,12)</td>
<td>20</td>
</tr>
<tr>
<td>(6,12)</td>
<td>19</td>
</tr>
<tr>
<td><strong>Wedge</strong></td>
<td></td>
</tr>
<tr>
<td>(30,45)</td>
<td>20</td>
</tr>
<tr>
<td>(15,30)</td>
<td>18</td>
</tr>
<tr>
<td>(30,45)</td>
<td>18</td>
</tr>
<tr>
<td>(45,60)</td>
<td>18</td>
</tr>
</tbody>
</table>

**Table II**: Best performing pairs using same kind of features, for those with more than 75% correct classification.

<table>
<thead>
<tr>
<th>Features</th>
<th>Number correctly classified</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ring</strong></td>
<td></td>
</tr>
<tr>
<td>(3,6)</td>
<td>23</td>
</tr>
<tr>
<td>(12,24)</td>
<td>22</td>
</tr>
<tr>
<td>(24,48)</td>
<td>22</td>
</tr>
<tr>
<td>(1,3)</td>
<td>22</td>
</tr>
<tr>
<td>(1,3)</td>
<td>22</td>
</tr>
<tr>
<td>(1,3)</td>
<td>22</td>
</tr>
<tr>
<td>(1,3)</td>
<td>22</td>
</tr>
<tr>
<td>(6,12)</td>
<td>21</td>
</tr>
<tr>
<td>(12,24)</td>
<td>21</td>
</tr>
<tr>
<td>(1,3)</td>
<td>20</td>
</tr>
<tr>
<td>(1,3)</td>
<td>20</td>
</tr>
<tr>
<td>(1,3)</td>
<td>20</td>
</tr>
<tr>
<td>(1,3)</td>
<td>20</td>
</tr>
<tr>
<td>(1,3)</td>
<td>20</td>
</tr>
<tr>
<td>(1,3)</td>
<td>20</td>
</tr>
<tr>
<td>(1,3)</td>
<td>20</td>
</tr>
<tr>
<td>(6,12)</td>
<td>19</td>
</tr>
<tr>
<td>(12,24)</td>
<td>19</td>
</tr>
<tr>
<td>(1,3)</td>
<td>19</td>
</tr>
<tr>
<td>(1,3)</td>
<td>19</td>
</tr>
<tr>
<td>(1,3)</td>
<td>19</td>
</tr>
<tr>
<td>(1,3)</td>
<td>19</td>
</tr>
<tr>
<td>(6,12)</td>
<td>19</td>
</tr>
</tbody>
</table>

**Table III**: Best performing pairs using equalized features for those with more than 80% correct classification.
ON A SPATIAL CLUSTERING ALGORITHM FOR IMAGE ANALYSIS

C. H. Chen
Electrical Engineering Department
Southeastern Massachusetts University
North Dartmouth, MA 02747 U.S.A.

Abstract

A computationally efficient spatial clustering algorithm is presented for image segmentation and compression. The algorithm can automatically determine the minimum number of clusters and can also work on a specified number of clusters. Examples are given on the processing of Seasat images using the algorithm.

Introduction

Besides the use of acoustic sensors, remotely sensed images can provide essential information on the object extraction and tracking in the ocean environment. Seasat SAR images are a good example. Many automatic image analysis algorithms have been developed. Such algorithms are generally application dependent. For remote sensing images, cluster analysis is important as it reveals the structure of the data from which useful information can be derived. Conventional clustering methods do not preserve the spatial relations in a image. Spatial clustering for image analysis has been considered [1][2]. However feature extraction was not taken into account. Furthermore the computation involved is quite extensive. A more efficient spatial clustering algorithm is developed for minicomputer processing, that employs properly selected features. The clustered image shows various regions (segments) from which desired objects may be extracted. Furthermore considerable image data compression is accomplished with essentially no loss of information. Examples are based exclusively on Seasat images dealing with the ocean environment.

Algorithm for Spatial Clustering

The algorithm proceeds as follows:

(1) Form a feature set, for each pixel, consisting of local mean and gradient. Other features may also be used.

(2) For each 2x2 subarea, measure the mean vector and dispersion.

(3) Determine the critical dispersion, and calculate the merging distance d.

(4) Merge adjacent subareas with distance less than d to form subregions. Calculate the mean vectors of subregions.

(5) Group these mean vectors into clusters using K-mean algorithm which converges to several cluster centers representing the mean vectors of regions.

For a given inter-region threshold distance, the algorithm can automatically adjust to an appropriate number of clusters. If the number of clusters is specified as in conventional cluster analysis, the algorithm will provide the specified number of clusters.
The image speckles in synthetic aperture are multiplicative in nature. Without removing such noise, the cluster results may still be very noisy. A simple pre-processing method is to use the Sigma filter suggested by J.S. Lee [3]. For each 5x5 (or 3x3) subarea the average gray level value is compared with the three-standard deviation (3σ) of the normalized image histogram (first order probability density). If the value is within 3σ of the center pixel is replaced by the average value. If the value exceeds 3σ, then it is an indication of edges or object boundary and the original gray level is retained. The procedure thus provides a compromise between noise filtering and edge preserving and adds only slight amount of computation to the clustering algorithm.

Computer Results

The original Seasat images are all of 256 gray levels. For convenience with minicomputer processing the digitized pictures are all reduced to 256x256 pixels even though the original images are much larger in size. The cluster algorithm is set such that a maximum of 7 clusters is selected. Figure 1 is for the scene of a ship off Chesapeake Bay. Figure 1a is the original. Figure 1b is the spatially clustered image. Figure 1c has the histograms of original (left) and clustered (right) images. Figure 1d uses the Sigma filter preprocessing with 3x3 subarea while Fig. 1e is the corresponding result with 5x5 subarea. Preprocessing with 5x5 subarea followed by spatial clustering appears to be the best. Figure 2 is the scene of Anchorage, Alaska with Fig. 2a for original and Fig. 2b for 5x5 preprocessing followed by spatial clustering. Figure 3 is the scene of Nantucket Shoals with Fig. 3a for original, Fig. 3b for 5x5 median filtering and Fig. 3c for 5x5 preprocessing followed by spatial clustering. Computer results all show that the "natural" clusters of the original images are very much preserved while noises are considerably reduced, and at the same time the contrast is enhanced, with the use of the spatial clustering algorithm.

Acknowledgement

This work was supported by ONR Contract N00014-79-C-0492. The digital Seasat SAR images were kindly provided by Dr. Jong-Sen Lee of Naval Research Laboratory who also brought to our attention of his work on filtering of multiplicative noise.

References
Fig. 1d

Fig. 1e
**APPENDIX D**

**APPENDIX PRESPA.FOR**

```
C **********************************************************************
C PRESPA.FOR         ( WAS PREA4.FOR )
C 30-MAY-82         ( 300 SECTION : READ DATA FILE FROM TAPE )
C 27-APR-82
C PROGRAM TO READ A DISK FILE 256X256 PIXELS
C OUTPUT ITS FEATURE VECTORS, MEAN, GRADIENT
C OR ORIGINAL INTENSITY, GRADIENT
C CHOICE 3: READ DATA FROM TAPE
C TO FORM NOF1, NOF2 OUTPUT FILES
C FEATURE 1 AND FEATURE 2 RESPECTIVELY
C NOF1: OUTPUT DATA, MEAN OR ORIGINAL GRAY LEVEL, COMPONENT 1
C NOF2: OUTPUT DATA, GRADIENT, COMPONENT 2
C IF CHOICE 1 OR 2: READ DISK FILE NOF4
C NOF4: INPUT DATA
C **********************************************************************

INTEGER P(1024),F2(1024),CHOICE,IMEAN(256),IGRAD(256)
REAL DMEAN(256),DGRAD(256),WS1(256),WS2(256)

DATA NOFINOF2,NOF4/*I2,4*/
1001 FORMAT(" PROGRAM PRESPA.FOR"/
2 " PREPROCESSING IMAGE DATA TO FORM FEATURE VECTORS"/
3 " FOR AUTO SPATIAL CLUSTERING"/
4 " INPUT: FTN4.DAT OR TAPE - ORIGINAL GRAY LEVEL"/
5 " OUTPUT: FTN1.DAT FEATURE COMPONENT 1"/
6 " FTN2.DAT, FEATURE COMPONENT 2"/
7 " CHOICE:"/
8 " 1: LOCAL MEAN, LOCAL GRADIEN"/
9 " 2: ORIGINAL INTENSITY, LOCAL GRADIENT"/
10 " 3: COMPRESS A 2 BY 2 SUBIMAGE INTO 1 PIXEL"/
20 " 4: READ TAPE DATA (NORGL*NORGP) TO DISK(NOL*NOP)"
1002 FORMAT(15)
1003 FORMAT(" ENTER INPUT AND OUTPUT FILES SIZE"/
1 " NOLIN,NOPIN,NOL,NOP: FORMAT(4F5)"
1004 FORMAT(4F5)
1 CONTINUE
WRITE(7,1001)
READ(5,1002)CHOICE
IF (CHOICE.LE.0.OR.CHOICE.GT.4) GOTO 1
GOTO (50,50,300,400),CHOICE
50 CONTINUE
WRITE(7,1003)
READ(5,1004)NOLIN,NOPIN,NOL,NOP
C ORIGINAL INTEGER DATA FILE OF AN IMAGE
DEFINE FILE NOF4(NOLIN,NOPIN,U,INDX4)
NOL=NOL-1
NOP=NOP-1
C OUTPUT FILES, NOF1, NOF2, INTEGER NUMBERS
DEFINE FILE NOF1(NOL,NOP,U,INDX1)
DEFINE FILE NOF2(NOL,NOP,U,INDX2)
GOTO (100,200),CHOICE
100 CONTINUE
```
APPENDIX PRESPA*FOR

DO 90 I=1,NOLI
INDEX4=I
READ(NOF4*INDEX4)(F(K),K=1,NOPIN)
READ(NOF4*INDEX4)(F2(K),K=1,NOPIN)
DO 70 J=1,NOP1
IMEAN(J)=(F(J)+F(J+1)+F2(J)+F2(J+1))/4
IGRAD(J)=((IABS(F(J))-F(J+1))+IABS(F2(J)-F(J)))/2
70 CONTINUE
IMEAN(NOP)=IMEAN(NOP1)
IGRAD(NOP)=IGRAD(NOP1)
INDEX1=I
INDEX2=I
WRITE(NOF1*INDEX1)(IMEAN(K),K=1,NOP)
WRITE(NOF2*INDEX2)(IGRAD(K),K=1,NOP)
90 CONTINUE
WRITE(NOF1*INDEX1)(IMEAN(K),K=1,NOP)
WRITE(NOF2*INDEX2)(IGRAD(K),K=1,NOP)
GOTO 900
200 CONTINUE
C CHOICE 2: SECTION
C FEATURE: ORIGINAL INTENSITY, LOCAL GRADIENT
DO 290 I=1,NOLI
INDEX4=I
READ(NOF4*INDEX4)(F(K),K=1,NOPIN)
READ(NOF4*INDEX4)(F2(K),K=1,NOPIN)
DO 270 J=1,NOP1
IGRAD(J)=((IABS(F(J))-F(J+1))+IABS(F2(J)-F(J)))/2
270 CONTINUE
IGRAD(NOP)=IGRAD(NOP1)
INDEX1=I
INDEX2=I
WRITE(NOF1*INDEX1)(F(K),K=1,NOP)
WRITE(NOF2*INDEX2)(IGRAD(K),K=1,NOP)
290 CONTINUE
WRITE(NOF1*INDEX1)(F(K),K=1,NOP)
WRITE(NOF2*INDEX2)(IGRAD(K),K=1,NOP)
GOTO 900
300 CONTINUE
C READ TAPE DATA FILE SECTION
C NOLI,NOLF,NOPI,NOPF: COVER THE AREA INTERESTED
C NOL, NOP: SIZE OF THE OUTPUT DATA IN FEATURE SPACE
C EACH RECORD REPRESENTS 2 BY 2 ORIGINAL PIXELS
WRITE(7,1031)
1031 FORMAT(" ENTER ORIGINAL TAPE FILE SIZE AND WHICH FILE IN TAPE"/
   " NOLI,NOLF,NOPI,NOPF: ( FORMAT(3I6) )":")
READ(5,1032)NOLI,NOLF,NOPI,NOPF
1032 FORMAT(3I6)
WRITE(7,1033)
1033 FORMAT(" WHICH PART OF IMAGE TO BE PROCESSED?":/
   " NOLI,NOLF,NOPI,NOPF: ( FORMAT(4I6) )":")
READ(5,1034)NOLI,NOLF,NOPI,NOPF
1034 FORMAT(4I6)
NOL=(NOLF-NOLI+1)/2
NOP=(NOPF-NOPI+1)/2
APPENDIX PRESPA.FOR

WRITE(7,1035)NOLI,NOLF,NOP1,NOP2,NOL,NOP
1035 FORMAT(" CHECK: NOLI,NOLF,NOP1,NOP2,NOL,NOP")
 1 616
 2 /* NOL X NOP WILL BE THE OUTPUT SIZE */
DEFINE FILE NOP1(NOL,NOP,U,INDEX1)
DEFINE FILE NOP2(NOL,NOP,U,INDEX2)
REWIND NOP1
REWIND NOP2
C SKIP THE PART NOT TO BE PROCESSED
NSKIP=NOLI-1
C NOLI IS THE FIRST LINE TO BE PROCESSED
IF (NSKIP.LT.1) GOTO 312
DO 310 I=1,NSKIP
310 CALL READUM(F1,I,NTH)
CONTINUE
DO 310 I=1,NSKIP
  DO 315 J=1,NOP
    IMEAN(J)=0
    IGRAD(J)=0
  CONTINUE
  CALL READUM(F1,I,NTH)
  CALL READUM(F2,I,NTH)
  DO 320 J=1,NOP
    IMEAN(J)=(F(NOP1+J+J-2)+F(NOP1+J+J-1)+F2(NOP1+J+J-2)+
      1 F2(NOP1+J+J-1))/4
    IGRAD(J)=(IABS(F(NOP1+J+J-2)-F(NOP1+J+J-1))+
      1 IABS(F(NOP1+J+J-2)-F2(NOP1+J+J-2)))/2
320 CONTINUE
INDX1=1
INDX2=1
WRITE(NOF1*INDEX1)(IMEAN(K),K=1,NOP)
WRITE(NOF2*INDEX2)(IGRAD(K),K=1,NOP)
350 CONTINUE
GOTO 900
400 CONTINUE
C READ TAPE DATA FILE TO FORM A COMPRESSED DISK DATA FILE
C INPUT SIZE : NORGL BY NORGP PIXELS
C OUTPUT SIZE: NOL BY NOP
WRITE(7,1041)
1041 FORMAT(" ENTER NORGL,NORGP,NOL,NOP,NOP,NTH"
     1 E.G., 256,256,64,64,64,64")
READ(5,1042)NORGL,NORGP,NOL,NOP,NOP,NTH
1042 FORMAT(616)
DEFINE FILE NOP(NOL,NOP,U,INDEX)
ITERL=NORGL/NOL
ITERP=NORGP/NOP
DO 450 I=1,NOL
  DO 420 J=1,ITERL
    CALL READUM(F1,I,NTH)
    WRITE(NOF*INDEX)(F(ITERP*K),K=1,NOP)
450 CONTINUE
GOTO 900
900 CONTINUE
CALL EXIT
END
C -------------------------------------------------------------
C 9A1234567898B123456789C123456789D123456789E123456789F123456789
C FILE NAME: SPABW.FOR
C AUTO SPATIAL CLUSTERING FOR BLACK/WHITE IMAGE DATA
C
C REFERENCE FUKADA, "SPATIAL CLUSTERING PROCEDURES FOR REGION
C ANALYSIS", PATTER RECOGNITION, 12, 395-403, (1980).
C
C INPUTS: NOF1, FTN1.DAT, FEATURE 1
C NOF2, FTN2.DAT, FEATURE 2
C OUTPUTS: NOFC, FTN12.DAT, CLUSTERING RESULT FOR COLOR
C DISPLAY
C NOFB, FTN11.DAT, CLUSTERING RESULT FOR BLACK/WHITE
C DISPLAY
C NOFF, FTN15.DAT, SOME PARAMETERS DURING PROCESSING
C SIZE OF IMAGE IS RESTRICTED TO 256 BY 256 IN FEATURE SPACE
C -------------------------------------------------------------

LOGICAL*1 DDMMY(9)
REAL PREY(60*2),PRENO(60)
COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,2),CMO(60),FKV(30,2),FNO(30)
COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
COMMON /BLOCK5/ICOLOR(256),IBN(256)
DATA NOF1,NOF2,NOFB,NOFC,NOFF/1,2,11,12,15/

1701 FORMAT("RUNNING PROGRAM SPABW.FOR AUTO SPATIAL
CLUSTERING"/
1   "INPUTS , OUTPUTS: NOF1,NOF2,NOFB,NOFC,NOFF"/
6   "FOR CHECK:"/515/
2   "NOFC, NOFB STORE TEMPORARY DATA DURING
PROCESSING"/
3   "OUTPUTS: NOFC(REWRITTEN) 7 OR LESS COLORS"/
4   "NOFB(REWRITTEN) BLACK AND WHITE
DISPLAY"/
5   "NOFF, INFORMATIONS DURING PROCESSING")

1515 FORMAT("MERGE ITERATION",I5)
1520 FORMAT("THE "/I5,"-TH ITERATION REACHES MAXIMUM NO.
CLUSTERS")/
1511 FORMAT("KERNEL CANDIDATE VECTORS FOR",I4," STARTING ",
1 "CLUSTER CENTERS")/
1512 FORMAT("ENTER IMAGE DATA FILE SIZE (FEATURE SPACE )"/
1 "NOL, NOP: FORMAT(2I5)")
1513 FORMAT(2I5)
1514 FORMAT("CHECK:NOL,NOP,NOL2,NOP2,NOLH,NOPH"/6I5)
1516 FORMAT("ENTER OPTIONS: (IOP(K),K=1,10)"/
1 "IOP(1): CONTROLS PRINTER, 1: MEANS, TRACE
MATRIX")/
2 "IOP(2): K-MEANS ALGORITHM DETAILS ON SCREEN"/
3 "IOP(3): MERGE DETAILS, LABEL(2,K) ARRAY")/
APPENDIX  SPABW.FOR

4   " IOP(4): NSTEP, NO. OF STEPS"/
5   " IOP(5): 1, SKIP K-MEAN ITERATION"/

1517 FORMAT(1016)
1518 FORMAT(" TODAY IS ",9A1)
REWWWIND NOFF
WRITE(7,1501)
WRITE(NOFP,1501)
CALL DATE(DDMMY)
WRITE(NOFP,1518)(DDMMY(K),K=1,9)
WRITE(7,1518) (DDMMY(K),K=1,9)
WRITE(NOFP,1701)NOFP,NOFB,NOFC,NOFF
WRITE(7,1701)NOFP,NOFB,NOFC,NOFF
WRITE(7,1512)
WRITE(NOFP,1512)
READ(5,1513)NOL,NOP
WRITE(7,1516)
WRITE(NOFP,1516)
READ(5,1517)(IOP(K),K=1,10)
WRITE(NOFP,1517)(IOP(K),K=1,10)
NOL2=NOL+NOL
NOP2=NOP+NOP
NOLH=NOL/2
NOPH=NOP/2
WRITE(NOFP,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
WRITE(7,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
DEFINE FILE NOFP1(NOL,NOP,U,INDX1)
DEFINE FILE NOFP2(NOL,NOP,U,INDX2)
DEFINE FILE NOFB(NOL,NOP,U,INDX1)
DEFINE FILE NOFC(NOL,NOP,U,INDX2)
C
C NOFF=FTM15.DAT UNFORMATTED
C
C REWWWIND NOP1
C REWWWIND NOP2
C REWWWIND NOFB
C REWWWIND NOFC

1501 FORMAT(" THIS IS THE LOG FILE OF EXECUTING SPABW.FOR")
C
C CALCULATE MEANS OF FEATURE VECTORS OF 2 BY 2 SUBIMAGE
C STORE IN NOFB: FIRST HALF -- FEATURE 1 MEANS OF 128X128
C SECOND HALF -- FEATURE 2 MEANS
C
C 128 X 128 SUBIMAGES EACH
C
C IN NOFC: FIRST AND SECOND HALF ARE THE SAME, TRACES
C
C CALL DISPER
C
C FIND MAX, MIN OF TRACE MATRIX
C CALL MAXMIN(DMAX,DMIN)
C
C MERGING SECTION
C
C NSTEP=IOP(4)
C STEP=(DMAX-DMIN)/FLOAT(NSTEP)
C WRITE(7,1522)DMAX,DMIN,STEP
C WRITE(NOFP,1522)DMAX,DMIN,STEP

1522 FORMAT(" DMAX="E20.8," DMIN="E20.8," STEP="E20.8)
C IPREV=0
C
C ITERATIONS TO FIND MAXIMUM NO. OF CLUSTERS
APPENDIX SPABW.FOR

DO 300 I=1,NSTEP
IN=I
WRITE(NOFF,1515)IN
CALL MERGE(IN,STEP,DMIN,NCLSR)
C ACCEPTED NO. OF CLUSTERS: 7 OR LESS
IF (IPREV.LE.7.AND.IPREV.GT.1) GOTO 333
IPREV=NCLSR
C SAVE CURRENT NUMBER OF CLUSTERS AND KERNEL VECTORS
DO 200 J=1,IPREV
PRENO(J)=CNO(J)
PREV(J,1)=CKV(J,1)
PREV(J,2)=CKV(J,2)
200 CONTINUE
300 CONTINUE
333 NI=IN-1.
WRITE(NOFF,1520)NI,IPREV
DO 350 J=1,IPREV
CNO(J)=PRENO(J)
CKV(J,1)=PREV(J,1)
CKV(J,2)=PREV(J,2)
350 CONTINUE
WRITE(NOFF,1561)
1561 FORMAT(" MERGE ENDED WITH MAXIMUM NO. CLUSTERS")
WRITE(NOFF,1562)((CKV(N,L),L=1,2),N=1,IPREV)
1562 FORMAT(" BEFORE SORTING/
1 " KERNEL CANDIDATE VECTORS/
2 30((5X,2E20.8)/))
C SORT THE CANDIDATE VECTORS
NC=IPREV
C ------- ----------------
C SORT THE CANDIDATE KERNEL VECTORS
CALL SORT(NC)
WRITE(NOFF,1563)((CKV(N,L),L=1,2),N=1,NC)
1563 FORMAT(" SORTED KERNEL CANDIDATE VECTORS/
1 30((5X,2E20.8)/))
IF (NC.GT.7) NC=7
C FOR THE PURPOSE OF AED-512 PSEUDO COLOR DISPLAY
WRITE(NOFF,1511)NC
C IF (IOP(5).EQ.1) SKIP THE K-MEAN ITERATIONS
C DIRECTLY USE MERGING RESULT CANDIDATE KERNEL VECTORS
C TO CLASSIFY THE IMAGE
WRITE(7,1568)IOP(5)
1568 FORMAT(" IOP(5)=",I5)
IF (IOP(5).NE.1) GOTO 700
WRITE(7,1570)
1570 FORMAT(" SKIP K-MEAN ITERATION")
DO 650 N=1,NC
DO 640 L=1,2
FKV(N,L)=CKV(N,L)
640 CONTINUE
650 CONTINUE
GOTO 800
700 CONTINUE
WRITE(7,1580)
1580 FORMAT(" CALLING KERVEC: K-MEAN ITERATION")
C ITERATIONS TO FIND MORE ACCURATE KERNEL VECTORS
C CALLED FINAL KERNEL VECTORS
CALL KERVEC(NC,KK,DD)
WRITE(NOFF,1500)KK,DD
1500 FORMAT(1X,"CLUSTERING REPEATS",1X,I3,1X,"TIMES",/1X,
1"THE FINAL WITHIN-CLASS DISTANCE IS",1X,E20.8/)
800 CONTINUE
C CLASSIFICATION SECTION
C OUTPUTS: NOFC, COLOR DISPLAY RESULT
C NOFD, BLACK/WHITE DISPLAY RESULT
CALL CLASS(NC)
WRITE(NOFF,1523)
1523 FORMAT(IOX,"III COMPLETE EXECUTION OF PROGRAM SPABW III")
CALL EXIT
END

C SUBPROGRAMS

C SUBROUTINE TO CALCULATE TRACE MATRICES OF FEATURE MATRICES STORED IN NOFD
SUBROUTINE DISPER
COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
COMMON /BLOCK1/NOF,NOFB,NOF2,NOFB,NOFB,NOFF
COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
REWIND NOF1
REWIND NOF2
REWIND NOFB
REWIND NOFC
C PROCESS THROUGH ROWS OF DATA MATRIX
DO 100 I=1,NOLH
I2=I+I-1
INDEX1=I2
INDEX2=I2
DO 40 JJ=1,2
C READ 2 LINES OF EACH FILE
READ(NOF1,INDEX1)(IA(K),K=1,NOP)
DO 10 J=1,NOP
10 A(J,JJ)=FLOAT(IA(J))
READ(NOF2,INDEX2)(IB(K),K=1,NOP)
DO 20 J=1,NOP
20 B(J,JJ)=FLOAT(IB(J))
40 CONTINUE
C CALCULATION THROUGH EACH SUBIMAGE
DO 80 K=1,NOPH
K1=K+K-1
K2=K1+1
S1=0.
S2=0.
DO 62 M=K1,K2
DO 60 L=1,2

Si = S1 + A(M, L)  
S2 = S2 + B(M, L)

60 CONTINUE
62 CONTINUE
ABM(2, K, 1) = S1 * 0.25
ABM(2, K, 2) = S2 * 0.25
S1 = 0.
S2 = 0.
DO 72 M = K1, K2
DO 70 L = 1, 2
S1 = S1 + (A(M, L) - ABN(2, K, 1)) ** 2
S2 = S2 + (B(M, L) - ABM(2, K, 2)) ** 2
70 CONTINUE
72 CONTINUE
TRACE(K) = (S1 + S2) * 0.25
80 CONTINUE

C ---------------------------------------------
C READ TRACE MATRIX TO FIND MAX, MIN
C
SUBROUTINE MAXMIN(DMAX, DMIN)
COMMON /BLOCKO/IOP(10), NOB1, NOB2, NOFB, NOFC, NOFF
COMMON /BLOCK1/NOL, NOB, NOL2, NOPF, NOLH, NOPH
COMMON /BLOCK3/IA(256), IB(256), A(256, 2), B(256, 2)
COMMON /BLOCK4/AB(256, 2), ABM(2, 128, 2), TRACE(128)
REWIND NOFC
INDX = 1
READ(NOFC*INDX)(TRACE(K), K = 1, NOPH)
DMAX = TRACE(1)
DMIN = TRACE(1)
DO 10 J = 2, NOPH
IF (TRACE(J) .LT. DMIN) DMIN = TRACE(J)
IF (TRACE(J) .GT. DMAX) DMAX = TRACE(J)
10 CONTINUE
DO 100 I = 2, NOLH
INDX = I
READ(NOFC*INDX)(TRACE(K), K = 1, NOPH)
DO 30 J = 1, NOPH
IF (TRACE(J) .LT. DMIN) DMIN = TRACE(J)
30 CONTINUE
APPENDIX SPABW.FOR

IF (TRACE(J).GT.DMAX) DMAX=TRACE(J)
30 CONTINUE
100 CONTINUE
WRITE(7,1001)DMAX,DMIN
1001 FORMAT(" DMAX="F12.4," DMIN="F12.4)
RETURN
END

C
C ----------------------------------------
C MERGE AND DECIDE KERNEL CANDIDATE VECTORS
C
SUBROUTINE MERGE(IMRGE,DSTEP,DMIN,LLBS)
REAL FIRST(60)
COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1502 FORMAT(30(" CH0(N):"F12.1))
1503 FORMAT(" *FMIN, DMRGE:"E20.8)
1504 FORMAT(" IMRGE:"I5)
1505 FORMAT(" FIRST SUBIMAGE:")
1506 FORMAT(" JOIN CLUSTER NL:")
1507 FORMAT(" NEW CLUSTER:")
1508 FORMAT(" LLBS:*GE.60")
1532 FORMAT(15X,"THETA",7X,"SIGMA-SQUARE",3X,
MERGING-DISTANCE:"/
1 3E20.8)
1533 FORMAT(" K,LLBS,TRACE(K),THETA,DMRGE:"/215,3E20.8)
REWIND NOFB
REWIND NOFC
C FOR EACH ITERATION: ZERO OUT THE VARIABLES
DO 20 J=1,60
CNO(J)=0.
DO 10 L=1,2
CKV(J,L)=0.
10 CONTINUE
20 CONTINUE
LLBS=0
C DMIN: SIGMA-SQUARE
C THETA: SOME THETA
C DMRGE: MERGING DISTANCE
THETA=DMIN+DSTEP*FLOAT(IMRGE)
DMRGE=SQRT(4./3.*(THETA-DMIN))
WRITE(NOF1,1532)THETA,DMIN,DMRGE
WRITE(7,1504)IMRGE
C GO THROUGH SUBIMAGES AND LABEL THEM WITH CLUSTERS
DO 300 J=1,NOLH
IF (J.GT.1) GOTO 35
C J=1: CASE OF FIRST LINE OF SUBIMAGES
DO 30 K=1,NOFF
LABEL(2,K)=0
DO 30 L=1,2
30 ABM(1,K,L)=0.
APPENDIX SPABW.FOR

GOTO 45
35 CONTINUE
C GET PREVIOUS LINE OF SUBIMAGES IN RGBM ARRAY
DO 40 K=1,NOPH
DO 40 L=1,2
40 ABM(1,K,L)=ABM(2,K,L)
45 CONTINUE
C INITIAL LABEL FOR EACH SUBIMAGE
INDXB=J
READ(NOFB*INDXB)(ABM(2,K,1),K=1,NOPH)
INDXB=J+NOLH
READ(NOFB*INDXB)(ABM(2,K,2),K=1,NOPH)
C GO THROUGH IMAGES ONE BY ONE
DO 50 K=1,NOPH
LABEL(1,K)=LABEL(2,K)
LABEL(2,K)=0
50 CONTINUE
INDXC=J
READ(NOFC*INDXC)(TRACE(K),K=1,NOPH)
C CHECK IF THE TRACE OF CURRENT SUBIMAGE > THETA
IF (TRACE(K).GT.THETA) GOTO 200
C GO THROUGH IMAGES ONE BY ONE
DO 201 K=1,NOPH
IF (L2EQ.9) WRITE(7,1533)K,LLBS,TRACE(K),THETA,DMRGE
IF (LLBS.GE.60) WRITE(7,1508)
IF (LLBS.GE.60) GOTO 900
C CHECK IF THE TRACE OF CURRENT SUBIMAGE > THETA
IF (TRACE(K).GT.THETA) GOTO 200
C SKIP
IF (J.GT.1) GOTO 52
C J=1: FIRST LINE OF SUBIMAGES
C THE FIRST LINE SECTION: CONSIDERING THE NEIGHBOR
M1=2
M2=2
K1=K-1
K2=K
GOTO 54
52 CONTINUE
C NOT THE FIRST LINE; SO PREVIOUS LINE EXISTS
M1=1
M2=2
K1=K
K2=K
54 CONTINUE
C CHECK IF FIRST SUBIMAGE OR NOT
IF (LLBS.EQ.0) GOTO 90
IF (LABEL(M1,K1).EQ.0) GOTO 55
C POTENTIAL NEIGHBOR NOT LABELLED; DIRECTLY CHECK CLUSTERS
C LABEL(M1,K1) NEIGHBOR HAS BEEN LABELLED
C AND SPATIAL CLUSTERING SHOULD BE APPLIED
DIFF=0.
DO 62 L=1,2
62 DIFF=DIFF+(ABM(M1,K1,L)-ABM(M2,K2,L))**2
DIFF=SQRT(DIFF)
IF (DIFF.GT.DMRGE) GOTO 55
C WITHIN MERGING DISTANCE?
LABEL(M2,K2)=LABEL(M1,K1)
NL=LABEL(M1,K1)
APPENDIX SPABW.POR

LABEL(M2,K2)=NL
CNO(NL)=CNO(NL)+1.
DO 64 L=1,2
64 CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+ABM(M2,K2,L))/CNO(NL)
GOTO 200
55 CONTINUE
DO 58 N=1,LLBS
DIFF=0.
DO 56 L=1,2
DIFF=DIFF+(CKV(N,L)-ABM(2,K2,L))**2
56 CONTINUE
FIRST(N)=SQR(DIFF)
58 CONTINUE
CALL DISMIN(FIRST,FMIN,NL,LLBS)
IF (FMIN.GT.014RGE) GOTO 90
IF (IOP(1).EQ.9) WRITE(7,1503)FMIN,DMRGE
C LABEL CURRENT SUBIMAGE WITH CLOSEST CENTER
LABEL(M2,K2)=NL
C UPDATE NO. OF SUBIMAGES OF CURRENT CLUSTER
CNO(NL)=CNO(NL)+1.
C UPDATE MEAN VECTOR OF THIS CLUSTER
DO 60 L=1,2
CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+ABM(2,K2,L))/CNO(NL)
60 CONTINUE
IF (IOP(1).EQ.9) WRITE(7,1506)
GOTO 200
C NEW CLUSTER SECTION
90 CONTINUE
LLBS=LLBS+1
C UPDATE OF SUBIMAGES OF THIS CLUSTER
CNO(LLBS)=CNO(LLBS)+1.
C UPDATE NEW CLUSTER VECTOR
DO 92 L=1,2
92 CKV(LLBS,L)=ABM(M2,K2,L)
200 CONTINUE
201 CONTINUE
250 CONTINUE
C CHECK CURRENT LINE'S LABELS
IF (IOP(3).EQ.1) WRITE(7,1545)(LABEL(2,K),K=1,32)
300 CONTINUE
900 CONTINUE
IF (IOP(1).EQ.9) WRITE(7,1502)(CNO(N),N=1,LLBS)
IF (IOP(1).EQ.9) WRITE(7,1501)IMRGE,LLBS
WRITE(1545)" LABEL",3212
1501 FORMAT(1H MERGE ITERATION:"I5," END WITH LLBS: "I5/
RETURN
END
C
C -----------------------------------------------
C SORTING THE KERNEL VECTORS
C
SUBROUTINE SORT(NCLRS)
REAL TEMP(2)
COMMON /BLOCKO/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
APPENDIX SPABM.FOR

COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
DO 30 I=2,NCLRS
I2=NCLRS+1-I
DO 20 J=1,I2
IF (CKV(J+1,1).GE.CKV(J,1)) GOTO 20
DO 10 L=1,2
TEMP(L)=CKV(J+1,L)
CKV(J+1,L)=CKV(J,L)
CKV(J,L)=TEMP(L)
10 CONTINUE
20 CONTINUE
30 CONTINUE
RITE(7,NOFF,1533)
FORMAT(/" SORTING CANDIDATE KERNEL VECTORS"/)
RETURN
END

C
TO FIND FINAL KERNEL VECTORS
C LIMIT TO 10 ITERATIONS

C SUBROUTINE KERVEC(NC,KK,DIS)
C DIST ARRAY STORES THE TOTAL DISTANCES OF ITERATIONS
C C ARRAY STORES NUMBER OF PIXELS FOR EACH CLUSTER
C D ARRAY STORES TEMPORARY DISTANCES TO CLUSTER CENTERS
C FOR CURRENT PIXEL BEING PROCESSED
REAL DIST(10),FIRST(60)
COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOF,C,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,2),CNO(60),FKV(30,2),FNO(30)
COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1020 FORMAT(/" IN KERVEC, KK: ",15)
C FINAL KERNEL VECTORS SAVED IN FKV ARRAY
C FNO STORE NO. OF PIXELS IN EACH CLUSTER
C K-MEANS ALGORITHM
C 10-JUN-82 CORRECT IMPLEMENTATION
C REFERENCE: TOU AND GONZALEZ, "PATTERN RECOGNITION
DO 12 N=1,NC
FNO(N)=CNO(N)
CNO(N)=0.
DO 10 L=1,2
FKV(N,L)=CKV(N,L)
CKV(N,L)=0.
10 CONTINUE
12 CONTINUE
C NO. OF ITERATIONS LIMIT TO 10
DO 500 KK=1,10
WRITE(7,NOFF,1020)KK
WRITE(NOFF,1020)KK
DO 15 N=1,NC
15 FNO(N)=0.
REWIND NOF1
APPENDIX SPABW.FOR

REWIND NOF2
REWIND NOFB
C FOR EACH ITERATION:
C REWIND THE FEATURES FILES: A , B COMPONENTS
C REWIND THE TEMPORARY CLASSIFIED RESULT FILE, NOFB
C CLASSIFYING STANDARDS IN FKV ARRAY
C AT THE SAME TIME, COLLECTING THE NEW CENTERS IN CKV ARRAY
C I.E., UPDATING THE KERNEL VECTORS BY CURRENT CLUSTERING
C ICOLOR ARRAY STORES CLASSIFIED RESULT OF CURRENT LINE
DO 200 I=1,NOL
INDX1=1
READ(NOF1*INDX1)(IA(K),K=1,NOP)
INDX2=1
READ(NOF2*INDX2)(IB(K),K=1,NOP)
DO 20 J=1,NOP
AB(J,1)=FLOAT(IA(J))
AB(J,2)=FLOAT(IB(J))
20 CONTINUE
C GO THROUGH PIXELS TO LABEL THEM WITH CLUSTERS
DO 100 J=1,NOP
DO 40 N=1,NC
SUM=0.
DO 30 L=1,2
SUM=SUM+(AB(J,L)-FKV(N,L))**2
30 SUM=SUM+(AB(J,L)-FKV(N,L))**2
40 FIRST(N)=SQRT(SUM)
CALL DISPLAY(FIRST,FMIN,NN,NC)
ICOLOR(J)=NN
CNO(NN)=CNO(NN)+1.
C CURRENT PIXEL WAS FOUND CLOSER TO NN-TH CLUSTER
C THE FEATURES SHOULD BE INCLUDED TO UPDATE THE NN-TH
C KERNEL VECTOR
DO 70 L=1,2
70 CKV(NN,L)=(CKV(NN,L)*(CNO(NN)-1.))\AB(J,L))/CNO(NN)
100 CONTINUE
C CURRENT IN CKV; STORE IN FKV TO BE USED TO CLASSIFY
C IN ROUTINE DISTANCE, AND WILL GIVE TOTAL DISTANCE
DO 260 N=1,NC
FNO(N)=CNO(N)
CNO(N)=0.
DO 250 L=1,2
FKV(N,L)=CKV(N,L)
250 CONTINUE
260 CONTINUE
WRITE(NOFF,1503)((FKV(N,L),L=1,2),N=1,NC)
1503 FORMAT(10X,'CHECK FKV: '/
130((5X,2E20.8))/)
WRITE(NOFF,1506)(FNO(N),N=1,NC)
1506 FORMAT(* OF PIXELS IN EACH CLUSTER:*/
130(/F12.1))
CALL DISTAN(NC,DIST)

********
DIST(KK)=DIS
WRITE(7,1504)DIST(KK),KK
WRITE(NOFF,1504)DIST(KK),KK

1504 FORMAT(/' IN KERVEC: DIST(KK)=',E20.8,' KK=' ,I2/)  
IF(KK.EQ.1)GO TO 500
RATIO=(DIST(KK)-DIST(KK-1))/DIST(KK-1)
WRITE(NOFF,1005)RATIO
IF(ABS(RATIO).LT.0.001) GOTO 900

500 CONTINUE
900 CONTINUE
1005 FORMAT( ' RATIO IN KERVEC: ',E20.8)
1505 FORMAT( ' COLOR:',64I1)
RETURN
END

---------------------
---------------------

USE CURRENT KERNEL VECTORS TO CALCULATE THE TOTAL
DISTANCE OF THE IMAGE
NC: NUMBER OF CLUSTERS
DISTOT: (RESULT) TOTAL DISTANCE

SUBROUTINE DISTAN(NC,DISTOT)
COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOCF,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,2),CMO(60),FKV(30,2),FMO(30)
COMMON /BLOCK3/IA(256),IB(256),A(256,2),B(256,2)
COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRACE(128)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)

DISTOT=0.
REWIND NOF1
REWIND NOF2
REWIND NOFB
DO 200 I=1,NOL
INDXB=I
READ(NOFB,INDXB)(ICOLOR(K),K=1,NOP)
INDX1=I
READ(NOF1,INDX1)(IA(K),K=1,NOP)
INDX2=I
READ(NOF2,INDX2)(IB(K),K=1,NOP)

STORE FEATURE VECTOR IN WORKING VARIABLE X
DO 10 K=1,NOP
AB(K,1)=FLOAT(IA(K))
AB(K,2)=FLOAT(IB(K))
10 CONTINUE
DO 100 J=1,NOP
NCLSR=ICOLOR(J)
SUM=0.
DO 30 L=1,2
SUM=SUM+(AB(J,L)-FKV(NCLSR,L))**2
30 CONTINUE
DISTOT=DISTOT+SQRT(SUM)
100 CONTINUE
200 CONTINUE
SUBROUTINE CLASS(NC)
REAL FIRST(60)
COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOFB,NOFC,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/FKV(60,2),CMO(60),FKV(30,2),PNO(30)
COMMON /BLOCK3/I(256),i(256),A(256,2),B(256,2)
COMMON /BLOCK4/AB(256,2),ABM(2,128,2),TRAC(128)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
WRITE(7,1505)
1505 FORMAT(1X:"IN CLASS:
")
WRITE(7,1501)((FKV(N,L),L=1,2),N=1,NC)
WRITE(7,1501)((FKV(N,L),L=1,2),N=1,NC)
1501 FORMAT(10X,″FINAL KERNEL VECTORS :″/(30(5X1,2920.8/)))
1509 FORMAT(1X,6411)
C USE FINAL KERNEL VECTORS TO CLASSIFY THE PICTURE
C NOFC: INTEGER OUTPUT; NOFD: REAL OUTPUT
REWIND NOF1
REWIND NOF2
REWIND NOFB
REWIND NOFC
C RANGE OF FINAL RESULT 0..180
FACT=180./FLOAT(NC)
DO 200 I=1,NOL
INDX1=I
INDX2=I
READ(NOF1"INDX1)(IA(K),K=1,NOP)
READ(NOF2"INDX2)(IB(K),K=1,NOP)
DO 10 J=1,NOP
AB(J,1)=FLOAT(IA(J))
AB(J,2)=FLOAT(IB(J))
10 CONTINUE
DO 100 J=1,NOP
DO 40 N=1,NC
SUM=0.
DO 30 L=1,2
SUM=SUM+(AB(J,L)-FKV(N,L))**2
30 CONTINUE
FIRST(N)=SORT(SUM)
40 CONTINUE
C DISTANCES TO FINAL KERNEL VECTORS OF NC CLUSTERS
C FROM CURRENT PIXEL ARE STORED IN DIS ARRAY,
C CALLING SUBROUTINE DISMIN TO FIND TO WHICH CLUSTER
C THE CURRENT PIXEL IS CLOSER ( MINIMUM DISTANCE ).
C THE RESULT IS KMIN-TH CLUSTER
CALL DISMIN(FIRST,SMIN,NMIN,NC)
C BLACK AND WHITE DISPLAY PURPOSE: NEEDS TO MULTIPLY A FACTOR
C TO BE IN REASONABLE GRAY LEVEL RANGE
IBW(J)=INT(FLOAT(MIN)*FACT)
C COLOR DISPLAY PURPOSE: AN INTEGER IN RANGE 1 TO 7
ICOLOR(J)=MIN
100 CONTINUE
INDXB=I
WRITE(NOFB,INDXB)(IBW(K),K=1,NOP)
INDXC=I
WRITE(NOFC,INDXC)(ICOLOR(K),K=1,NOP)
IF (I.LE.64) WRITE(7,1509)(ICOLOR(K),K=1,64)
IF (I.LE.64) WRITE(NOFP,1509)(ICOLOR(K),K=1,64)
200 CONTINUE
RETURN
END

C
C SUBROUTINE DISMIN(DARRY,DATMIN,MIN,NCSTR)
C PASS DARRY ARRAY WITH NCSTR ELEMENTS MEANINGFUL
C SEARCH FOR THE MINIMUM ELEMENT, NOMIN-TH ELEMENT,
C WITH VALUE DATMIN; PASS BACK DATMIN AND NOMIN BACK
C CALLING ROUTINE
REAL DARRY(60)
DATMIN=DARRY(1)
MIN=1
C ASSUME THE FIRST ELEMENT IS THE MINIMUM
C THEN GO THROUGH THE REST OF THE ARRAY TO FIND ANY SMALLER
IF (NCSTR.EQ.1) GOTO 900
DO 100 I=2,NCSTR
IF (DARRY(I).GE.DATMIN) GOTO 100
DATMIN=DARRY(I)
MIN=I
100 CONTINUE
900 CONTINUE
RETURN
END
APPENDIX SPACLR.FOR

C --------------------------------------------------------
C 9A123456789B123456789C123456789D123456789E123456789F123456789
9G12
C FILE NAME: SPACLR.FOR
C AUTO SPATIAL CLUSTERING FOR COLOR IMAGE DATA
C
C REFERENCE FUKADA, "SPATIAL CLUSTERING PROCEDURES FOR REGION
ANALYSIS", PATTERN RECOGNITION, 12, 395-403 (1980).
C
C INPUTS: NOF1, FTN1.DAT, FEATURE 1
C NOF2, FTN2.DAT, FEATURE 2
C NOF3, FTN3.DAT, FEATURE 3
C OUTPUTS: NOFC, FTN12.DAT, CLUSTERING RESULT FOR COLOR
DISPLAY
C NOFD, FTN11.DAT, CLUSTERING RESULT FOR BLACK/WHITE
DISPLAY
C NOFF, FTN15.DAT, SOME PARAMETERS DURING PROCESSING
C SIZE OF IMAGE IS RESTRICTED TO 256 BY 256 IN FEATURE SPACE
C --------------------------------------------------------

LOGICAL*1 DDMMY(9)
REAL PREV(60,3),PRENO(60)
COMMON /BLOCKO/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
COMMON /BLOCK4/RGBM(2,128,3),TRACE(128)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
COMMON /BLOCK6/RI(256,2),G(256,2),B(256,2)
DATA NOF1,NOF2,NOF3,NOFC,NOFD,NOFF/1.2.3,1/2,1/5/
*1701 FORMAT("RUNNING PROGRAM SPACLR.FOR AUTO SPATIAL
CLUSTERING")/
1   "INPUTS, OUTPUTS: NOF1,NOF2,NOF3,NOFC,NOFD,NOFF/
6   "FOR CHECK:"/615/
2   "NOFC, NOFD STORE TEMPORARY DATA DURING
PROCESSING"/
3   "OUTPUTS: NOFC(REWRITTEN) 7 OR LESS COLORS"/
4   "NOFD(REWRITTEN) BLACK AND WHITE
DISPLAY"/
5   "NOFF, INFORMATIONS DURING PROCESSING")
1515 FORMAT("MERGE ITERATION"/15)
1520 FORMAT("THE ",15,"-TH ITERATION REACHES MAXIMUM NO.
CLUSTERS")/
1   "NO. OF CLUSTERS:"/15)
1511 FORMAT("KERNEL CANDIDATE VECTORS FOR"/14," STARTING ",
1   "CLUSTER CENTERS")/
1512 FORMAT("ENTER IMAGE DATA FILE SIZE (FEATURE SPACE ")/
1   "NOL, NOP: FORMAT(215))")
1513 FORMAT(215)
1514 FORMAT("CHECK=NOL,NOP,NOL2,NOP2,NOLH,NOPH*/615)
1516 FORMAT("ENTER OPTIONS: (IOP(K),K=1,10)/")
1   "IOP(1): CONTROLS PRINTER, 1: MEANS, TRACE
MATRIX")/
APPENDIX SPACLR.FOR

2 * IOP(2): K-MEANS ALGORITHM DETAILS ON SCREEN*/
3 * IOP(3): MERGE DETAILS, LABEL(2,K) ARRAY*/
4 * IOP(4): NSTEP, NO. OF STEPS*/
5 * IOP(5): 1, SKIP K-MEAN ITERATION*/

1517 FORMAT(1016)
1518 FORMAT(" TODAY IS ",9A1)
REWIND NOFF
WRITE(7,1501)
WRITE(NOFF,1501)
CALL DATE(DDMMY)
WRITE(NOFF,1518)(DDMMY(K),K=1,9)
WRITE(7,1518) (DDMMY(K),K=1,9)
WRITE(NOFF,1701)NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
WRITE(7,1701)NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
WRITE(7,1512)
WRITE(NOFF,1512)
READ(5,1513)NOL,NOP
WRITE(7,1516)
WRITE(NOFF,1516)
READ(5,1517)(IOP(K),K=1,10)
WRITE(NOFF,1517)(IOP(K),K=1,10)
WRITE(7,1517)(IOP(K),K=1,10)
NOL2=NOF+NOL
NOP2=NOF+NOP
NOLH=NOL/2
NOPH=NOP/2
WRITE(NOFF,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
WRITE(7,1514)NOL,NOP,NOL2,NOP2,NOLH,NOPH
DEFINE FILE NOF1(NOL,NOP,U,INDX1)
DEFINE FILE NOF2(NOL2,NOP2,U,INDX2)
DEFINE FILE NOF3(NOL,NOP3,U,INDX3)
DEFINE FILE NOFC(NOL,NOP,X,INDXC)
DEFINE FILE NOFD(NOL2,NOP2,U,INDXD)

C NOFF=FTN15.DAT UNFORMATTED
REWIND NOF1
REWIND NOF2
REWIND NOF3
REWIND NOFC
REWIND NOFD

1501 FORMAT(" THIS IS THE LOG FILE OF EXECUTING SPACLR.FOR")
C---------------------------------------------------------------
C CALCULATE MEANS OF FEATURE VECTORS OF 2 BY 2 SUBIMAGE
C STORE IN NOFC AND HALF NOFD 128 BY 128 SUBIMAGES
C IN NOFC: FIRST HALF IS R MEANS, SECOND HALF IS G MEANS
C IN NOFD: FIRST HALF IS B MEANS, SECOND HALF IS TRACES
C CALL DISPER
C---------------------------------------------------------------
C FIND MAX, MIN OF TRACE MATRIX
C CALL MAXMIN(DMAX,DMIN)
C---------------------------------------------------------------
C MERGING SECTION
NSTEP=IOP(4)
STEP=(DMAX-DMIN)/FLOAT(NSTEP)
WRITE(7,1522)DMAX,DMIN,STEP
WRITE(NOFF,1522)DMAX,DMIN,STEP
APPENDIX SPACLR.FOR

IPREV=0  
C ITERATIONS TO FIND MAXIMUM NO. OF CLUSTERS  
DO 300 I=1,NSTEP  
IM=I  
WRITE(NOFF,1515)IM  
CALL MERGE(IM,STEP,DMIN,NCLR)  
C ACCEPTED NO. OF CLUSTERS: 7 OR LESS  
IF (IPREV.LE.7.AND.IPREV.GT.1) GOTO 333  
IPREV=NCLR  
C SAVE CURRENT NUMBER OF CLUSTERS AND KERNEL VECTORS  
DO 200 J=1,IPREV  
PRENO(J)=CNI(J)  
PREVC(J,1)=CKV(J,1)  
PREVC(J,2)=CKV(J,2)  
PREVC(J,3)=CKV(J,3)  
200 CONTINUE  
300 CONTINUE  
333 NI=IM-1  
WRITE(NOFF,1520)NI,IPREV  
DO 350 J=1,IPREV  
CNO(J)=PRENO(J)  
CKV(J,1)=PREVC(J,1)  
CKV(J,2)=PREVC(J,2)  
CKV(J,3)=PREVC(J,3)  
350 CONTINUE  
WRITE(NOFF,1561)  
1561 FORMAT(" MERGE ENDED WITH MAXIMUM NO. CLUSTERS")  
WRITE(NOFF,1562)((CKV(N,L),L=1,3),N=1,IPREV)  
1562 FORMAT(" BEFORE SORTING")  
1 30((5X,E20.8)/))  
C SORT THE CANDIDATE VECTORS  
NC=IPREV  
C SORT THE CANDIDATE KERNEL VECTORS  
CALL SORT(NC)  
WRITE(NOFF,1563)((CKV(N,L),L=1,3),N=1,NC)  
1563 FORMAT(" SORTED KERNEL CANDIDATE VECTORS")  
1 30((5X,E20.8)/))  
IF (NC.GT.7) NC=7  
C FOR THE PURPOSE OF AED-512 PSEUDO COLOR DISPLAY  
WRITE(NOFF,1511)NC  
C IF (IOP(5).EQ.1) SKIP THE K-MEAN ITERATIONS  
C DIRECTLY USE MERGING RESULT CANDIDATE KERNEL VECTORS  
C TO CLASSIFY THE IMAGE  
WRITE(7,1568)IOP(5)  
1568 FORMAT(" IOP(5)=",I5)  
IF (IOP(5).NE.1) GOTO 700  
WRITE(7,1570)  
1570 FORMAT(" SKIP K-MEAN ITERATION")  
DO 650 N=1,NC  
DO 640 L=1,3  
FKV(N,L)=CKV(N,L)  
640 CONTINUE
APPENDIX SPACLR.FOR

650 CONTINUE
GOTO 800
700 CONTINUE
WRITE(7,1580)
1580 FORMAT(" CALLING KERVEC: K-MEAN ITERATION")
C  ITERATIONS TO FIND MORE ACCURATE KERNEL VECTORS
C  CALLED FINAL KERNEL VECTORS
CALL KERVEC(NC, KK, DD)
WRITE(NOFF,1500) KK, DD
1500 FORMAT(1X,"CLUSTERING REPEATS",1X, 13, 1X, "TIMES"/1X,
1 "THE FINAL WITHIN-CLASS DISTANCE IS",1X,E20.8/)
800 CONTINUE
C  CLASSIFICATION SECTION
C  OUTPUTS: NOFC, COLOR DISPLAY RESULT
C            NOFD, BLACK/WHITE DISPLAY RESULT
CALL CLASS(NC)
WRITE(NOFF,,1523)
1523 FORMAT(IOX,-Ill
      COMPLETE EXECUTION OF PROGRAM SPACLR III")
CALL EXIT
END
C  --------- SUBPROGRAMS  ---------
C  SUBROUTINE TO CALCULATE TRACE MATRICES OF FEATURE MATRICES
C  STORED IN NOFD
SUBROUTINE DISPER
COMMON /BLOCK0/IOP(10), NOF1, NOF2, NOF3, NOFC, NOFD, NOFF
COMMON /BLOCK1/NOL, NOP, NOL2, NOP2, NOLH, NOPH
COMMON /BLOCK3/IR(256), IG(256), IB(256), RGB(256,3)
COMMON /BLOCK4/RGBK(2,128,3), TRACE(128)
COMMON /BLOCK6/R(256,2), G(256,2), B(256,2)
REWIND NOF1
REWIND NOF2
REWIND NOF3
REWIND NOFC
REWIND NOFD
C  PROCESS THROUGH ROWS OF DATA MATRIX
DO 100 I=INOLH
  I2=I+I-1
  INDX1=I2
  INDX2=12
  INDX3=I2
  DO 40 JJ=1,2
  C  READ 2 LINES OF EACH FILE
  READ(NOF1*INDX1)(IR(K), K=1, NOP)
  DO 10 J=1, NOP
  10  R(J,JJ)=FLOAT(IR(J))
  READ(NOF2*INDX2)(IG(K), K=1, NOP)
  DO 20 J=1, NOP
  20  G(J,JJ)=FLOAT(IG(J))
  READ(NOF3*INDX3)(IB(K), K=1, NOP)
  DO 100 J=I+1, NOLH
APPENDIX SPACLR.FOR

DO 30 J=1,NOP
30 B(J,J)=FLOAT(IB(J))
40 CONTINUE

C CALCULATION THROUGH EACH SUBIMAGE
DO 80 K=1,NOPH
K1=K+K-1
K2=K1+1
S1=0.
S2=0.
S3=0.
DO 62 M=K1,K2
DO 60 L=1,2
S1=S1+R(M,L)
S2=S2+G(M,L)
S3=S3+B(M,L)
60 CONTINUE
62 CONTINUE
RGBM(2,K,1)=S1*0.25
RGBM(2,K,2)=S2*0.25
RGBM(2,K,3)=S3*0.25
S1=0.
S2=0.
S3=0.
DO 72 M=K1,K2
DO 70 L=1,2
S1=S1+(R(M,L)-RGBM(2,K,1))**2
S2=S2+(G(M,L)-RGBM(2,K,2))**2
S3=S3+(B(M,L)-RGBM(2,K,3))**2
70 CONTINUE
72 CONTINUE
TRACE(K)=(S1+S2+S3)*0.25
80 CONTINUE

INDXCI=I
IF (IOP(1).EQ.1) WRITE(6,1001)(RGBM(2,K,1),K=1,32)
WRITE(NOFC"INDXC)(RGBM(2,K,1),K=1,NOPH)
INDXCI=I+NOPH
IF (IOP(1).EQ.1) WRITE(6,1002)(RGBM(2,K,2),K=1,32)
WRITE(NOFD"INDXD)(RGBM(2,K,2),K=1,NOPH)
INDXD=I
IF (IOP(1).EQ.1) WRITE(6,1003)(RGBM(2,K,3),K=1,32)
WRITE(NOFD"INDX)(RGBM(2,K,3),K=1,NOPH)
INDXD=I+NOPH
IF (IOP(1).EQ.1) WRITE(6,1004)(TRACE(K),K=1,32)
WRITE(NOFD"INDX)(TRACE(K),K=1,NOPH)
100 CONTINUE
1001 FORMAT(" RM",32F4.0)
1002 FORMAT(" GM",32F4.0)
1003 FORMAT(" BM",32F4.0)
1004 FORMAT(" TR",32F4.0)
RETURN
END

C -------------------------------
C READ TRACE MATRIX TO FIND MAX, MIN
C
SUBROUTINE MAXMIN(DMAX,DMIN)
APPENDIX SPACL.R FOR

COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOPD,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
COMMON /BLOCK4/RGBM(2,128,3),TRACE(128)
RESEND NOFD
INDXD=I+1+NOLH
READ(NOFP*INDXD)(TRACE(K),K=1,NOPH)
DMAX=TRACE(1)
DMIN=TRACE(1)
DO 10 J=2,NOPH
IF (TRACE(J).LT.DMIN) DMIN=TRACE(J)
IF (TRACE(J).GT.DMAX) DMAX=TRACE(J)
10 CONTINUE
DO 100 I=2,NOLH
INDXD=I+NOLH
READ(NOFP*INDXD)(TRACE(K),K=1,NOPH)
DO 30 J=1,NOPH
IF (TRACE(J).LT.DMIN) DMIN=TRACE(J)
IF (TRACE(J).GT.DMAX) DMAX=TRACE(J)
30 CONTINUE
100 CONTINUE
WRITE(7,1001)DMAX,DMIN
1001 FORMAT(/ DMAX=",F12.4,"
DMIN="",F12.4)
RETURN
END

C
C-- - - - - - - - - - - - - - - - - - - - - -
C
MERGE AND DECIDE KERNEL CANDIDATE VECTORS
C
SUBROUTINE MERGE(IMRGEDSTEP,DSTEP,DMIN,LLBS)
REAL FIRST(60)
COMMON /BLOCK0/IOP(10),NOF1,NOF2,NOF3,NOFC,NOPD,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
COMMON /BLOCK4/RGBM(2,128,3),TRACE(128)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1502 FORMAT(30(" CNO(N)":"",F12.1))
1503 FORMAT(" *FMIN, DMERGE="",2E20.8)
1504 FORMAT("/ IMRGE="",I5)
1505 FORMAT("/ FIRST SUBIMAGE:"
1506 FORMAT("/ JOIN CLUSTER NL:"
1507 FORMAT("/ NEW CLUSTER:"
1508 FORMAT("/ LLBS.GE.60")
1532 FORMAT(15X,"THETA",7X,"SIGMA-SQUARE",3X,"MERGING-DISTANCE:"/
1 3E20.8)
1533 FORMAT(" K,LLBS,TRACE(K),THETA,DMERGE="",2I5,3E20.8)
REWIND NOFC
REWIND NOFD
C FOR EACH ITERATION: ZERO OUT THE VARIABLES
DO 20 J=1,60
CNO(J)=0.
DO 10 L=1,3
CKV(J,L)=0.
10 CONTINUE
20 CONTINUE
APPENDIX SPACLR.FOR

LLBS=0
C DMIN: SIGMA-SQUARE
C THETA: SOME THETA
C DMRGE: MERGING DISTANCE
THETA=DMIN+DSTEP*FLOAT(IMRGE)
DMRGE=SQRT(4./3.*(THETA-DMIN))
WRITE(NOFC,1532)THETA,DMIN,DMRGE
WRITE(7,1504)IMRGE
C GO THROUGH SUBIMAGES AND LABEL THEM WITH CLUSTERS
DO 300 J=1,NOLH
IF (J.GT.1) GOTO 35
C J=1: CASE OF FIRST LINE OF SUBIMAGES
DO 30 K=1,NOPH
LABEL(2,K)=0
DO 30 L=1,3
30 RGBM(1,K,L)=0.
GOTO 45
35 CONTINUE
C GET PREVIOUS LINE OF SUBIMAGES IN RGBM ARRAY
DO 40 K=1,NOPH
DO 40 L=1,3
40 RGBM(1,K,L)=RGBM(2,K,L)
45 CONTINUE
INDEX=J
READ(NOFC*INDEX)(RGBM(2,K,1),K=1,NOPH)
INDEX=J+NOLH
READ(NOFC*INDEX)(RGBM(2,K,2),K=1,NOPH)
INDEX=J
READ(NOFC*INDEX)(RGBM(2,K,3),K=1,NOPH)
C INITIAL LABEL FOR EACH SUBIMAGE
DO 50 K=1,NOPH
LABEL(1,K)=LABEL(2,K)
LABEL(2,K)=0
50 CONTINUE
INDEX=J+NOLH
READ(NOFC*INDEX)(TRACE(K),K=1,NOPH)
C GO THROUGH IMAGES ONE BY ONE
DO 201 K=1,NOPH
IF (IIP(1).EQ.9) WRITE(7,1533)K,LLBS,TRACE(K),THETA,DMRGE
IF (LLBS.GE.60) WRITE(7,1508)
IF (LLBS.GE.60) GOTO 900
C CHECK IF THE TRACE OF CURRENT SUBIMAGE > THETA
IF (TRACE(K).GT.THETA) GOTO 200
C SKIP
IF (J.GT.1) GOTO 52
C J=1: FIRST LINE OF SUBIMAGES
C THE FIRST LINE SECTION: CONSIDERING THE NEIGHBOR
M1=2
M2=2
K1=K-1
K2=K
GOTO 54
52 CONTINUE
C NOT THE FIRST LINE; SO PREVIOUS LINE EXISTS
M1=1
APPENDIX SPACLR.FOR

M2=2
K1=K
K2=K

CONTINUE
C CHECK IF FIRST SUBIMAGE OR NOT
IF (LLBS.EQ.0) GOTO 90
IF (LABEL(M1,K1).EQ.0) GOTO 55
POTENTIAL NEIGHBOR NOT LABELLED, DIRECTLY CHECK CLUSTERS
C LABEL(M1,K1) NEIGHBOR HAS BEEN LABELLED
C AND SPATIAL CLUSTERING SHOULD BE APPLIED
DIFF=0.
DO 62 L=1,3
62 DIFF=DIFF+(RGBM(M1,K1,L)-RGBM(M2,K2,L))**2
DIFF=SQRT(DIFF)
IF (DIFF.GT.DMRGE) GOTO 55
C WITHIN MERGING DISTANCE?
LABEL(M2,K2)=LABEL(M1,K1)
NL=LABEL(M1,K1)
LABEL(M2,K2)=NL
CNO(NL)=CNO(NL)+1.
DO 64 L=1,3
64 CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+RGBM(M2,K2,L))/CNO(NL)
GOTO 200
55 CONTINUE
DO 58 N=1,LLBS
DIFF=0.
DO 56 L=1,3
DIFF=DIFF+(CKV(N,L)-RGBM(2,K2,L))**2
56 CONTINUE
FIRST(N)=SQRT(DIFF)
58 CONTINUE
CALL DISMIN(FIRST,FMIN,NL,LLBS)
IF (FMIN.GT.DMRGE) GOTO 90
IF (IDP(1).EQ.9) WRITE(7,1503)FMIN,DMRGE
C LABEL CURRENT SUBIMAGE WITH CLOSEST CENTER
LABEL(M2,K2)=NL
C UPDATE NO. OF SUBIMAGES OF CURRENT CLUSTER
CNO(NL)=CNO(NL)+1.
C UPDATE MEAN VECTOR OF THIS CLUSTER
DO 60 L=1,3
60 CKV(NL,L)=(CKV(NL,L)*(CNO(NL)-1.)+RGBM(2,K2,L))/CNO(NL)
CONTINUE
IF (IDP(1).EQ.9) WRITE(7,1506)
GOTO 200
C NEW CLUSTER SECTION
90 CONTINUE
LLBS=LLBS+1
C UPDATE NO. OF SUBIMAGES OF THIS CLUSTER
CNO(LLBS)=CNO(LLBS)+1.
C UPDATE NEW CLUSTER VECTOR
DO 92 L=1,3
92 CKV(LLBS,L)=RGBM(M2,K,L)
200 CONTINUE
201 CONTINUE
250 CONTINUE
APPENDIX SPACLR.FOR

C CHECK CURRENT LINE'S LABELS
IF (IOP(3).EQ.1) WRITE(7,1545)(LABEL(2,K),K=1,32)
300 CONTINUE
900 CONTINUE
IF (IOP(1).EQ.9) WRITE(7,1502)(CNO(N),N=1,LLBS)
IF (IOP(1).EQ.9) WRITE(7,1501)IMRGE,LLBS
WRITE(NOFF,1501)IMRGE, LLBS
1545 FORMAT( "LABEL",32I2)
1501 FORMAT( /* MERGE ITERATION: "'IT" END WITH LLBS: "'IT/)
RETURN
END

----------------------------------
C SORTING THE KERNEL VECTORS
C
SUBROUTINE SORT(NCLRS)
REAL TEMP(3)
COMMON /BLOCK0/IOP(10),NOFXNOF2,NOF3,NOFC,NOFD,NOFF
COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
DO 30 I=2,NCLRS
I2=NCLRS+1-I
DO 20 J=1,12
IF (CKV(J+1,1).GE.CKV(J,1)) GOTO 20
DO 10 L=1,3
TEMP(L)=CKV(J+1,L)
CKV(J+1,L)=CKV(J,L)
CKV(J,L)=TEMP(L)
10 CONTINUE
20 CONTINUE
30 CONTINUE
WRITE(NOFF,1533)
1533 FORMAT(1X"SORTING CANDIDATE KERNEL VECTORS")
RETURN
END

----------------------------------
C TO FIND FINAL KERNEL VECTORS
C LIMIT TO 10 ITERATIONS
C
SUBROUTINE KERVEC(NC,KK,DIS)
C DIST ARRAY STORES THE TOTAL DISTANCES OF ITERATIONS
C C ARRAY STORES NUMBER OF PIXELS FOR EACH CLUSTER
C D ARRAY STORES TEMPORARY DISTANCES TO CLUSTER CENTERS
C FOR CURRENT PIXEL BEING PROCESSED
REAL DIST(10),FIRST(60)
COMMON /BLOCK0/IOP(10),NOFXNOF2,NOF3,NOFC,NOFD,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
1020 FORMAT( "IN KERVEC, KK: "'IT)
C FINAL KERNEL VECTORS SAVED IN FKV ARRAY
C FNO STORE NO. OF PIXELS IN EACH CLUSTER
C K-MEANS ALGORITHM
C 10-JUN-82 CORRECT IMPLEMENTATION
APPENDIX SPACLR.FOR

C REFERENCE: TOU AND GONZALEZ, "PATTERN RECOGNITION"
C PRINCIPLES", PP. 94-97
DO 12 N=1, NC
FNO(N)=CNO(N)
CNO(N)=0.
DO 10 L=1,3
FKV(N,L)=CKV(N,L)
CKV(N,L)=0.
10 CONTINUE
12 CONTINUE
C NO. OF ITERATIONS LIMIT TO 10
DO 500 KK=1,10
WRITE(7, 1020)KK
WRITE(NOFP, 1020)KK
DO 15 N=1, NC
15 FNO(N)=0.
REWIND NOF1
REWIND NOF2
REWIND NOF3
REWIND NOFD
C FOR EACH ITERATION:
C REWIND THE FEATURES FILES: R, G, B COMPONENTS
C REWIND THE TEMPORARY CLASSIFIED RESULT FILE, NOFD
C CLASSIFYING STANDARDS IN FKV ARRAY
C AT THE SAME TIME, COLLECTING THE NEW CENTERS IN CKV ARRAY
C I.E., UPDATING THE KERNEL VECTORS BY CURRENT CLUSTERING
C ICOLOR ARRAY STORES CLASSIFIED RESULT OF CURRENT LINE
DO 200 I=1, NOL
INDX1=1
READ(NOF1*INDX1)(IR(K), K=1, NOP)
INDX2=1
READ(NOF2*INDX2)(IG(K), K=1, NOP)
INDX3=1
READ(NOF3*INDX3)(IB(K), K=1, NOP)
DO 20 J=1, NOP
RGB(J, 1)=FLOAT(IR(J))
RGB(J, 2)=FLOAT(IG(J))
RGB(J, 3)=FLOAT(IB(J))
20 CONTINUE
C GO THROUGH PIXELS TO LABEL THEM WITH CLUSTERS
DO 100 J=1, NOP
DO 40 N=1, NC
SUM=0.
DO 30 L=1,3
30 SUM=SUM+(RGB(J, L)-FKV(N, L))**2
40 FIRST(N)=SQRT(SUM)
CALL DISMIN(FIRST, FMIN, NN, NC)
ICOLOR(J)=NN
CNO(NN)=CNO(NN)+1.
C CURRENT PIXEL WAS FOUND CLOSER TO NN-TH CLUSTER
C THE FEATURES SHOULD BE INCLUDED TO UPDATE THE NN-TH
C KERNEL VECTOR
DO 70 L=1,3
70 CKV(NN, L)=(CKV(NN, L)*(CNO(NN)-1.)+RGB(J, L))/CNO(NN)
100 CONTINUE
APPENDIX  SPACLR.FOR

INCLUD=I
IF (IOP(2).EQ.1) WRITE(7,1505)(ICOLOR(K),K=1,64)
WRITE(NOFD*INDXO)(ICOLOR(K),K=1,NOF)
200 CONTINUE
C CURRENT IN CKV; STORE IN FKV TO BE USED TO CLASSIFY
C IN ROUTINE DISTAN, AND WILL GIVE TOTAL DISTANCE
DO 260 N=1,NC
FNO(N)=CNO(N)
CNO(N)=0.
DO 250 L=1,3
FKV(N,L)=CKV(N,L)
CKV(N,L)=0.
250 CONTINUE
260 CONTINUE
WRITE(NOFP,1503)((FKV(N,L),L=1,3),N=1,NC)
1503 FORMAT(/10X,"CHECK FKV:"
130((5X,3E20.8)/))]
WRITE(NOFP,1506)(FNO(N),N=1,NC)
1506 FORMAT(130(/F12*1))
C "********
C CALL DISTAN(NC,DIST)
C "********
DIST(KK)=DIST
WRITE(7,1504)DIST(KK),KK
WRITE(NOFP,1504)DIST(KK),KK
1504 FORMAT(130("IN KERVEC: DIST(KK)=",E20.8," KK=",I2/))
IF(KK.EQ.1)GO TO 500
RATIO=(DIST(KK)-DIST(KK-1))/DIST(KK-1)
WRITE(NOFP,1005)RATIO
IF(ABS(RATIO).LT.0.001) GOTO 900
500 CONTINUE
900 CONTINUE
1005 FORMAT("RATIO IN KERVEC:",E20.8)
1505 FORMAT(130("COLOR:11),(64II)
RETURN
END

C USE CURRENT KERNEL VECTORS TO CALCULATE THE TOTAL
C DISTANCE OF THE IMAGE
C NC: NUMBER OF CLUSTERS
C DISTOT: ( RESULT ) TOTAL DISTANCE
C
SUBROUTINE DISTAN(NC,DISTOT)
COMMON /BLOCKO/IOP(10),NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOLH,NOPH
COMMON /BLOCK2/CKV(60,3),CNO(60),FKV(30,3),FNO(30)
COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
COMMON /BLOCK5/LABEL(2,128),ICOLOR(256),IBW(256)
DISTOT=0.
REWIND NOF1
REWIND NOF2
REWIND NOF3
REWIND NOFD
APPENDIX  SPACLR.FOR

DO 200 I=1,NOL
   INDXD=I
   READ(NOFD*INDXD)(ICOLOR(K),K=1,NOP)
   INDX1=I
   READ(NOF1*INDX1)(IR(K),K=1,NOP)
   INDX2=I
   READ(NOF2*INDX2)(IG(K),K=1,NOP)
   INDX3=I
   READ(NOF3*INDX3)(IB(K),K=1,NOP)
C STORE FEATURE VECTOR IN WORKING VARIABLE X
   DO 10 K=1,NOP
      RGB(K,1)=FLOAT(IR(K))
      RGB(K,2)=FLOAT(IG(K))
      RGB(K,3)=FLOAT(IB(K))
   10 CONTINUE
   DO 100 J=1,NOP
      NCLSR=ICOLOR(J)
      SUN=0.
      DO 30 L=1,3
         SUN=SUN+(RGB(J,L)-FKV(NCLSRL))**2
      30 CONTINUE
      DISTOT=DISTOT+SQRT(SUM)
   100 CONTINUE
   200 CONTINUE
   WRITE(NOFF,1501)DISTOT
1501 FORMAT(" IN DIStAN: DISTOT = ",E20.8)
   RETURN
   END

C C C
USE THE FINAL KERNEL VECTORS TO CLASSIFY THE IMAGE
C INTO CLUSTERS ( SUBREGIONS )
C
SUBROUTINE CLASS(NC)
   REAL FIRST(60)
   COMMON /BLOCK0/IOP(10),NOL,NOF1,NOF2,NOF3,NOFC,NOFD,NOFF
   COMMON /BLOCK1/NOL,NOP,NOL2,NOP2,NOL3,NOPH
   COMMON /BLOCK2/CVK(60,3),CHO(60),FKV(30,3),FNO(30)
   COMMON /BLOCK3/IR(256),IG(256),IB(256),RGB(256,3)
   COMMON /BLOCKS/LABEL(2,128),ICOLOR(256),IBW(256)
   WRITE(NOFF,1505)
1505 FORMAT(" IN CLASS:
      WRITE(7,1501)((FKV(N,L),L=1,3),N=1,NC)
      WRITE(NOFP,1501)((FKV(N,L),L=1,3),N=1,NC)
1501 FORMAT(10X," FINAL KERNEL VECTORS : "/(30(5X,3E20.8/)))
1509 FORMAT(1X,6411)
C USE FINAL KERNEL VECTORS TO CLASSIFY THE PICTURE
C NOFC: INTEGER OUTPUT; NOFD: REAL OUTPUT
   REWIND NOFC
   REWIND NOF2
   REWIND NOF3
   REWIND NOFC
   REWIND NOFD
C RANGE OF FINAL RESULT 0..180
FACT=180./FLOAT(NC)
DO 200 I=1,NOL
INDEX1=I
INDEX2=I
INDEX3=I
READ(NOF1*INDEX1)(IR(K),K=1,NOP)
READ(NOF2*INDEX2)(IG(K),K=1,NOP)
READ(NOF3*INDEX3)(IB(K),K=1,NOP)
DO 10 J=1,NOP
RGB(J,1)=FLOAT(IR(J))
RGB(J,2)=FLOAT(IG(J))
RGB(J,3)=FLOAT(IB(J))
10 CONTINUE
DO 100 J=1,NOP
DO 40 N=1,NC
SUM=0.
DO 30 L=1,3
SUM=SUM+(RGB(J,L)-FKV(N,L))**2
30 CONTINUE
FIRST(N)=SQRT(SUM)
40 CONTINUE
C DISTANCES TO FINAL KERNEL VECTORS OF NC CLUSTERS
C FROM CURRENT PIXEL ARE STORED IN DIS ARRAY,
C CALLING SUBROUTINE DISMIN TO FIND TO WHICH CLUSTER
C THE CURRENT PIXEL IS CLOSER (MINIMUM DISTANCE).
C THE RESULT IS KMIN-TH CLUSTER
CALL DISMIN(FIRST,SMIN,NMIN,NC)
C BLACK AND WHITE DISPLAY PURPOSE: NEEDS TO MULTIPLY A FACTOR
C TO BE IN REASONABLE CRAY LEVEL RANGE
IBW(J)=INT(FLOAT(NMIN)*FACT)
C COLOR DISPLAY PURPOSE: AN INTEGER IN RANGE 1 TO 7
ICOLOR(J)=NMIN
100 CONTINUE
INDEXD=I
WRITE(NOFD*INDEXD)(IBW(K),K=1,NOP)
INDEXC=I
WRITE(NOFC*INDEXC)(ICOLOR(K),K=1,NOP)
IF (I.LE.64) WRITE(7,1509)(ICOLOR(K),K=1,64)
IF (I.LE.64) WRITE(NOFF,1509)(ICOLOR(K),K=1,64)
200 CONTINUE
RETURN
END

C------------------------------
C SUBROUTINE DISMIN(DARRY,DATMIN,NOMIN,NCLSTR)
C PASS DARRY ARRAY WITH NCLSTR ELEMENTS MEANINGFUL
C SEARCH FOR THE MINIMUM ELEMENT, NOMIN-TH ELEMENT,
C WITH VALUE DATMIN; PASS BACK DATMIN AND NOMIN BACK
C CALLING ROUTINE
REAL DARRY(60)
DATMIN=DARRY(1)
NOMIN=1
C ASSUME THE FIRST ELEMENT IS THE MINIMUM
C THEN GO THROUGH THE REST OF THE ARRAY TO FIND ANY SMALLER
APPENDIX  SPACLR.FOR

IF (NCLSTR.EQ.1) GOTO 900
DO 100 I=2,NCLSTR
IF (DARRY(I).GE.DATMIN) GOTO 100
DATMIN=DARRY(I)
NOMIN=I
100 CONTINUE
900 CONTINUE
RETURN
END