1. Publication Nº	2. Version	3. Date	5. Distribution
INPE-4169-PRE/1066		*May 1987*	☐ Internal ☒ External
4. Origin *DSR*	Program *FOREST*		☐ Restricted

6. Key words - selected by the author(s)
IMAGE CLASSIFICATION
IMAGE TEXTURE
LANDSAT MSS AND TM

7. U.D.C.: *621.376.5*

8. Title *INPE-4169-PRE/1066*

TEXTURAL FEATURES FOR IMAGE CLASSIFICATION IN REMOTE SENSING

10. Nº of pages: *48*

11. Last page: *A.24*

12. Revised by

Nelson D. A. Mascarenhas

9. Authorship *Vitor Haertel**
Yosio Edemir Shimabukuro

Responsible author Vitor Haertel

13. Authorized by

Marco Antônio Raupp
Director General

14. Abstract/Notes

Texture is an important characteristic in identifying regions of interest in an image. Several methods to quantify image texture have been reported in the literature. This paper describes experiments aimed to extract textural features from digital images by calculating statistical properties in and around each pixel. The moving window concept is implemented, and tests using LANDSAT MSS and TM imagery are presented.

15. Remarks *Presented at the IV Simpósio Brasileiro de Sensoriamento Remoto, August 10-15, 1986, Gramado, RS, Brazil.*

** Instituto de Pesquisas Hidráulicas - UFRS, Porto Alegre, RS, Brazil.*

TEXTURAL FEATURES FOR IMAGE CLASSIFICATION IN REMOTE SENSING

Vitor Haertel

Instituto de Pesquisas Hidráulicas
Universidade Federal do Rio Grande do Sul
C.P. 530 - Porto Alegre - RS - Brasil

Yosio Shimabukuro

Instituto de Pesquisas Espaciais - INPE
C. P. 515 - São José dos Campos - SP - Brasil

ABSTRACT

Texture is an important characteristic in identifying regions of interest in an image. Several methods to quantify image texture have been reported in the literature. This paper describes experiments aimed to extract textural features from digital images by calculating statistical properties in and around each pixel. The moving window concept is implemented, and tests using LANDSAT MSS and TM imagery are presented.

Key Words
Image Classification, Image Texture, LANDSAT MSS and TM

AKNOWLEDGMENTS

The authors are grateful to the Brazilian Space Agency (Instituto de Pesquisas Espaciais - INPE) which provided the LANDSAT CCTs for the two scenes in Brazil, and to Ms Claire Hay, a graduate student at Colorado State University, who provided the North Dakota image used in this study.

Mr Vitor Haertel, is presently a Ph D candidate at the Department of Civil Engineering, Colorado State University, Fort Collins, USA. He is sponsored by CAPES, the Brazilian Post-Graduate Education Federal Agency.

CONTENTS

LIST OF FIGURES

Page

1. INTRODUCTION

Digital image processing is the numerical manipulation of digital images. It includes preprocessing, enhancement and classification [1]. Proprocessing involves operations on the raw image such as minimization of systematic errors. Prior to analysis, the image should be corrected for elements such as differences in terrain ilumination due to terrain relief. Sun azimuth and elevation angle, view angle and terrain topography are some of the elements to be considered here.

Enhancement involves the use of techniques such that the processed image is more suitable than the original one for a specific application such as visual interpretation. In many situations, image enhancement is best done in the frequency domain by making use of the Fast Fourier Transform [2].

Classification involves the automatic interpretation of an image, producing a thematic map on which each pixel has been assigned to one of the several possible classes. Pattern Recognition methods are used at this stage.

Spectral, textural and contextual features are three fundamental pattern elements used by a human being to interpret images. Spectral features describe the band to band tonal variations in a multiband image set (e.g., MSS imagery).

Texture contains information with regard to the spatial distribution of tonal values within a spectral band. Image texture is characterized by a repetitive structure or pattern across regions of an image. A more detailed definition for texture is provided in [3]: "the notion of texture appears to depend on three ingredients: (1) some local order is repeated over a region which is large in comparison to the order's size; (2) the order consists in the nonrandom arrangement of parts; (3) the parts are roughly uniform entities having approximately the same dimensions everywhere within the textured region".

Contextual features contain information derived from areas surrounding the image region being analyzed.

Most of the methods currently used in digital image classification are based almost exclusively upon spectral pattern recognition techniques.

The aim of this paper is to report experiments which were carried out using the "texture transforms" approach as an additional source of information in image classification.

2. INSTRUMENTATION USED

This work was develop at Colorado State University (CSU).

A DEC Vax 11/750 computer and a RIPS (Remote Information Processing System) image display system were used.

The RIPS system is made up of a micro processor with memory to support programs, data storage and image display refresh; also, a dual floppy disk drive for auxiliary programs and data storage, a joystick for graphical interaction, a terminal console for control of the system and a color television monitor for image display.

The microprocessor hardware comes with Cromenco supplied software. This software allows the user to execute programs, inspect diskette directories, erase or remove files, list files on the console, etc. In addition to the Cromenco supplied software, a group of RIPS programs are provided for the purpose of image manipulation and analysis. The RIPS application function categories are as follows:

1. The Image Store and Recall category enables an image, either color or black and white, to be saved on a disk file or to be displayed on the image monitor from previously saved disk files.

2. The Image Analysis and Density Slicing category allows the user to select ranges of radiometric values for enhancement, to determine data distributions and to derive quantitative measurements of this data. Many of these routines provide near instantaneous interaction with the displayed image.
3. The Display Generation and Annotation category includes ancillary functions which can be used to augment existing displays or generate new ones. These include inserting text into the display, drafting multicolor line drawings, producing images of constant value and linear and nonlinear pattern displays.
4. The Image Transformation and Enhancement category provides functions that can either alter the display images or generate a new image from stored data files. It perfoms various types of logical and mathematical operations, which measure, alter or otherwise describe spatial or radiometric properties of the data.
5. The Classification Category provides procedures for classifying data sets from the display or from stored data files. It also includes routines for enhancing or altering classification results, or deriving quantitative assessments.

In addition to these capabilities, the RIPS at CSU is connected to the DEC-VAX 11/750 computer. This allows the manipulation of much larger amounts of data (including full LANDSAT scenes) and the development and implementation of larger computer programs for image processing.

3. TEXTURE TRANSFORMS

The spectral features of a digital image are defined by the tone of the pixels on each spectral band. They are directly acquired by multispectral scanners and other remote sensing instruments.

The spatial features like texture refers to the spatial distribution of tonal variations within a spectral band.

In spite of being one of the basic elements used by a human being to perceive the world around, image texture has neither a precise definition nor a formal mathematical approach to its quantification. Several approaches to extract textural characteristics from an image have been proposed.

Some methods attempt to extract textural information from an entire image region or block of contiguous pixels [4], [5]. Fourier Power Spectrum, Gray-Level Run Length, and Gray-Level Co-occurrence Matrices are among the methods most commonly used.

A different approach was proposed by Hsu [6] and Irons and Petersen [7]. In this case, one "texture value" is assigned to each individual pixel by making use of some local property involving a group of pixels in the vicinity of the one under consideration. In this way "texture channels" can be generated and used along with the conventional spectral channels in image classification.

Texture channels can be constructed by implementing the "moving window" concept.

Consider a multispectral image and represent by I (r,c,b) the gray-level of a pixel located at image row "r" and column "c" in spectral band "b". The spatial characteristics or local image properties in and around this pixel can thus be calculated and its numerical value assigned to the pixel.

In this way a new channel called "texture channel" can be constructed which is also referred to as a texture transform of the image.

The approach used is this study is based on the work published by Hsu [6], who implemented the "moving window" concept. Following this approach, a square (or rectangular) array is placed over the image in such a way that each element on this array coincides with one pixel on the digital image. A statistical property is then calculed using the gray-levels within the window and its value is assigned to the central pixel in the window.

Moving the window such that each image pixel serves as a central point at a time, the "texture channel" can be constructed.

Examples of local properties are: the mean, the variance, the skewness and the kurtosis of gray-levels within the window.

4. EXPERIMENTS

Some initial experiments were performed using LANDSAT MSS and TM imagery to gain some experience about the potential usefulness of the "texture channels".

Computer program "texture.f" was developed to perform the following tasks:

- read in digital imagery in the usual compacted storage form (Band Interleaved format),
- separate the multispectral channels to process each one individually,
- calculate "texture channels" using a window size and a local property as specified by the user,
- reformat the computed "texture channel" in the compact Band Interleaved format compatible with the RIPS image display system.

The following local properties are available in the "texture.f" computer program:

Mean - MNL -

$$MNL = \frac{\sum x_{ij}}{n}$$

where x_{ij} represents the gray level of the pixel located at row i and column j within the window and M represents the total number of pixels in the window. The sum extends over all pixels within the window.

Variance - VNL -

$$VNL = \frac{\sum(x_{ij} - MNL)^2}{(n-1)}$$

Skewness - SKEW -

$$SKEN = \frac{| \sum(x_{ij} - MNL)^3|}{(n-1)\ (VNL)^{3/2}}$$

Kurtosis - KURT -

$$KURT = \frac{\sum(x_{ij} - MNL)^4}{(n-1)\ VNL^2}$$

Range - RNL -

$$RNL = \max\ (x_{ij}) - \min\ (x_{ij})$$

Pearson's second coefficient of skewness - PSKEW -

$$PSKEW = \frac{| MNL - x_m |}{VNL^{1/2}}$$

where x_m represents the median norm length in a window.

Absolute value of mean norm length differences - MDIF -

$$MDIF = \frac{| \sum x_{ij} - x_c|}{(n-1)}$$

where x_c representes the norm length of the gray level vector representing a window's central pixel.

Mean of squared norm length differences - MSQ -

$$MSQ = \frac{\sum (x_{ij} - x_c)^2}{(n-1)}$$

Maximum of squared norm lenght differences - MAXSQ -

$$MAXSQ = \max\ (x_{ij} - x_c)^2$$

Tests were carried out using LANDSAT MSS and TM imagery. The size of the moving window used in this experiment was 3 by 3 and the size of the MSS and TM images was 240 by 256 pixels, which corresponds to the full screen of the RIPS system.

The processing was performed using the VAX 11/750 computer and the resulting "texture channels" were transferred back to the RIPS system for display and further analysis.

The image in these experiments are:

1. LANDSAT MSS image (Figure 1), acquired on 22nd of August 1980. The image covers an area in North Dakota, bordering Manitoba in Canada. The scene was chosen because it has distinct land use practices such as agriculture, forestry, numerous lakes, swamps and open glades, which provide excellent distinction of features.
2. LANDSAT MSS image (Figure 5) acquired on July 27th, 1978. The image covers an area in the State of Mato Grosso do Sul, Brazil. This scene shows a timber management area and provides distinct types of forested areas [8].
3. LANDSAT TM image (Figure 9) acquired on January 19th, 1985. The image covers an area in the State of Paranã, in southern Brazil. The area selected covers part of the city of Maringã and shows urban areas and agricultural fields.

Since the spatial resolution in the TM imagery is much higher than in MSS imagery, it was expected the former to be much richer in spatial information than the latter.

5. RESULTS

The conclusions that could be drawn from the three sets of experiments were essentially similar to the ones presented in [7].

Three "texture channels", the mean, the variance and the mean squared norm length presented useful features for digital image analysis applications. The remaining channels have shown little or no usefulness in this context.

The "mean channel" proved to be useful for image smoothing when the noise is excessive. Its effect with respect to the original image is similar to a low-pass filter. It was used successfully in preprocessing LANDSAT MSS imagery: the reduction in the level of noise helped in many cases the classification process.

The "variance channel" as shown in Figures 2,3,4,6,7,8 and 10 proved to be an efficient edge detector. If applied to the adequate MSS or TM channel, the desired edges which are present in the digital image can be efficiently detected. Figure 2 shows the "variance transform" when applied to MSS channel 6 of the image shown in Figure 1. The channel 6 lies in the near infrared portion of the electromagnetic spectrum (0.7 - 0.8 micrometers). In this region of the spectrum, the healthy vegetation presents a high response whereas water generates a very low one.

As a result, the several water bodies present in this image and surrounded by vegetated areas generate strong well-defined edges.

Figure 3 shows the "variance tranform" when applied to MSS channel 5, which lies on the visible red portion of the electromagnetic spectrum (0.6 - 0.7 micrometres). This band corresponds to the clorophyl absorption region which results in a low reflectance for vegetation. Bare soil, however, reflects well within this portion of the spectrum.

As a result, edges corresponding to areas of bare soil (like roads) surrounded by vegetated areas show up very well.

Figure 4 shows a superposition of the "variance transform" when applied to MSS channels 4,5 and 6. The resulting edges are shown in the colors blue, green and red respectively.

Similar results were obtained when the "variance transform" was applied to the image in Figure 5 (Mato Grosso do Sul). Different types of vegetation also reflect in a distinct way, causing the edges detected by the transform (Figure 6 and 7).

The same tranform was also applied to channels 2,3 and 7 of the LANDSAT TM image (Figure 9). The channel 7 (2.08 - 2.35 micrometers) lies on a local peak for vegetation reflectance which however is lower than the one present in the spectral region corresponding to MSS channel 6. Also in this region, bare soil reflects more than vegetation (the opposie is true for MSS channel 6) and water is again a very poor reflector in this region. The edges resulting from these properties show up very well on the corresponding "variance transform" (Figure 10). The edge around the small lake, surrounded by vegetation (urban park) presents high intensity. Edges between vegetated areas and urban areas show lower intensity. This difference is due to the fact that the difference in reflectance between urban area and vegetation in this region of the spectrum is lower than the difference between water and vegetation. Also the edges around the agricultural fields appear very well on this transform, due to the contrast between roads (bare soil) and vegetation.

Figure 11 shows a superposition of texture transforms for TM channel 2, 3 and 7 on the colors green, blue and red respectively. TM channel 2 senses wavelengths between 0.45 and 0.53 micrometers, and TM channel 3 between 0.52 and 0.60 micrometers. Hence, TM channel 2 and 3 are roughly equivalent to MSS channels 4 and 5 respectively. Edges due to the heterogeneity of the urban area are visible here.

The mean square norm length presented results which are quite similar to the variance transform and will not be discussed here.

6. CONCLUSIONS

The "mean", "variance" and "square norm length" transforms proved to be useful in digital image analysis. The "mean transform" works well as a low pass filter and was successfuly used to reduce image noise.

The "variance transform" works well as an edge detector. It can be useful for image classification. If applied to the adequate channel this transform can help the image segmentation process.

Incorporation of these transforms into the more conventional algorithms for image classification will probably help to increase the accuracy of the classification procedure.

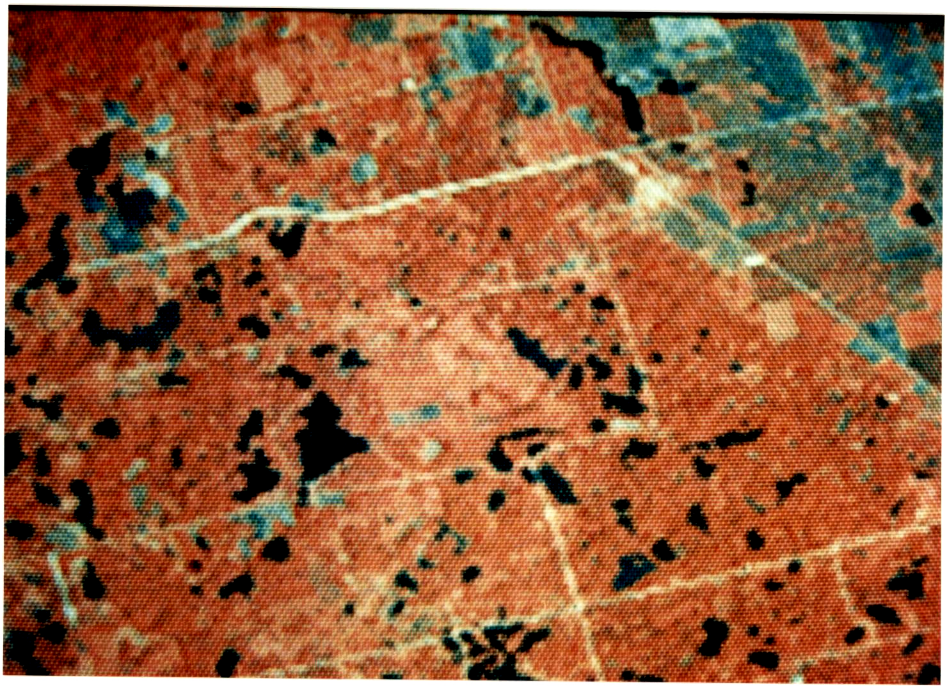

Fig. 1 - LANDSAT MSS sub-scene.
North Dakota.

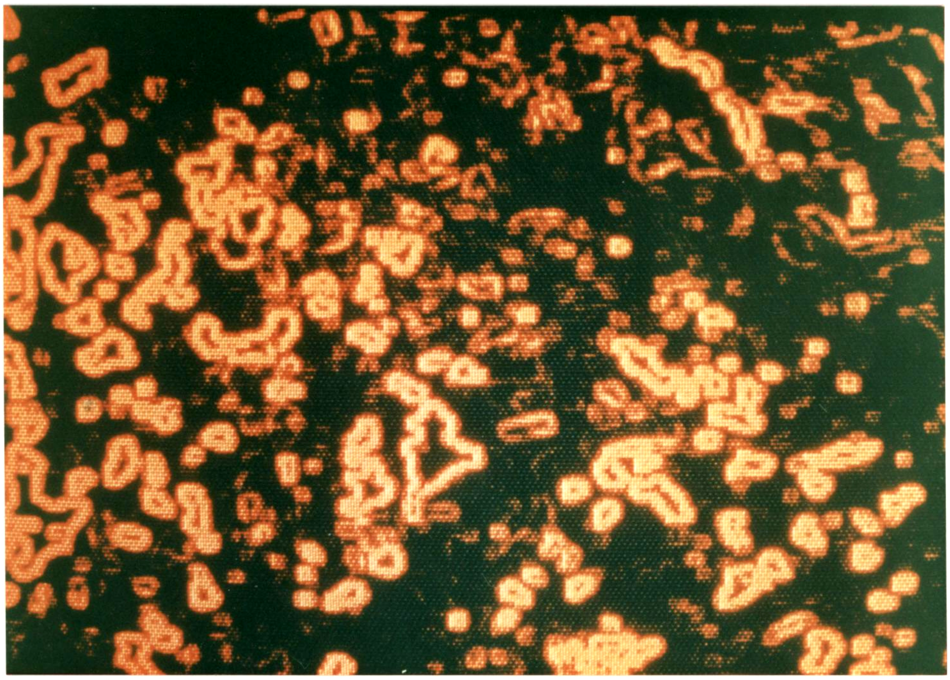

Fig. 2 - Variance Transform. North Dakota sub-scene.
MSS channel 6.

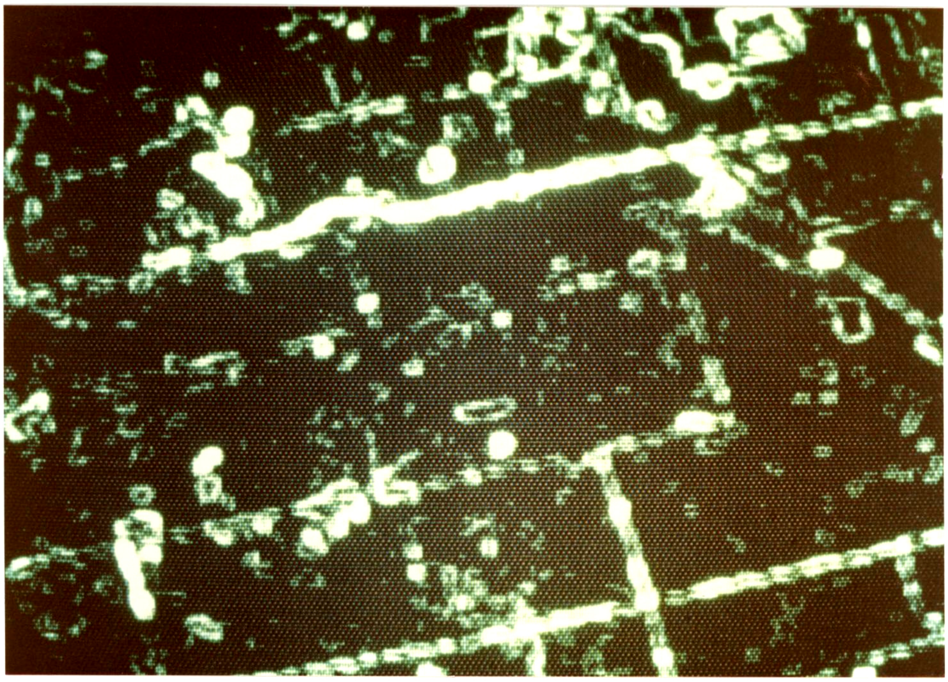

Fig. 3 - Variance Transform. North Dakota sub-scene. MSS channel 5.

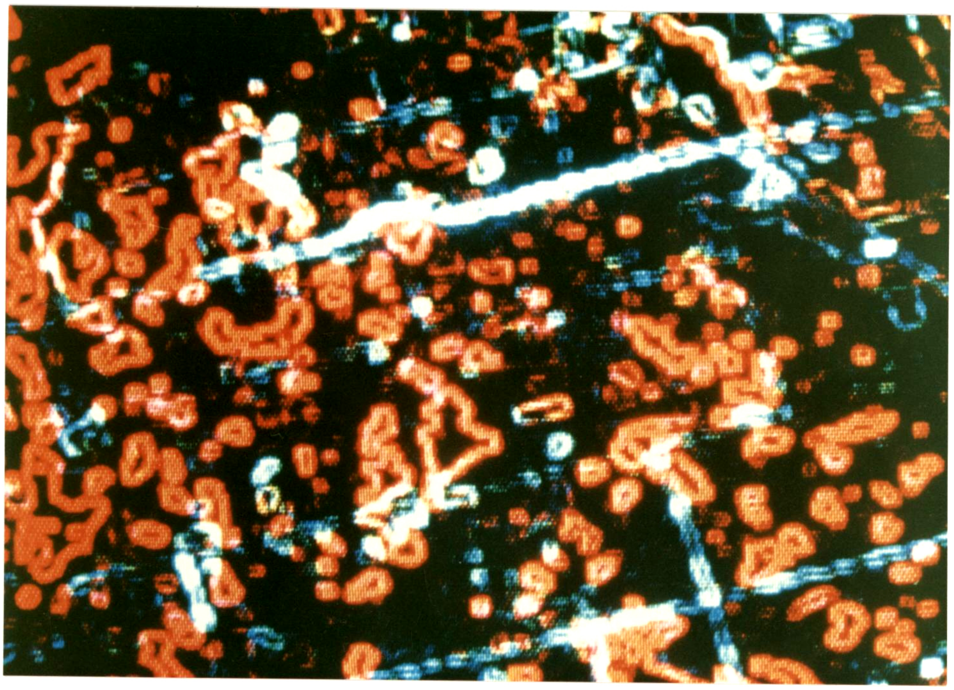

Fig. 4 - Variance Transforms. North Dakota sub-scene. MSS channel 4 (blue), channel 5 (green) and channel 6 (red).

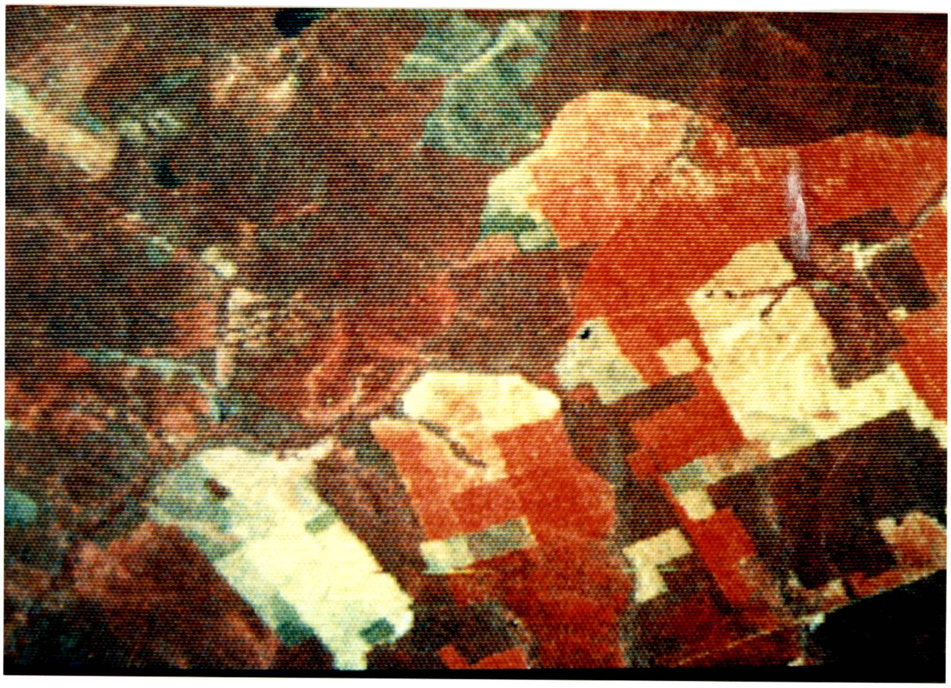

Fig. 5 - LANDSAT MSS sub-scene.
Mato Grosso do Sul.

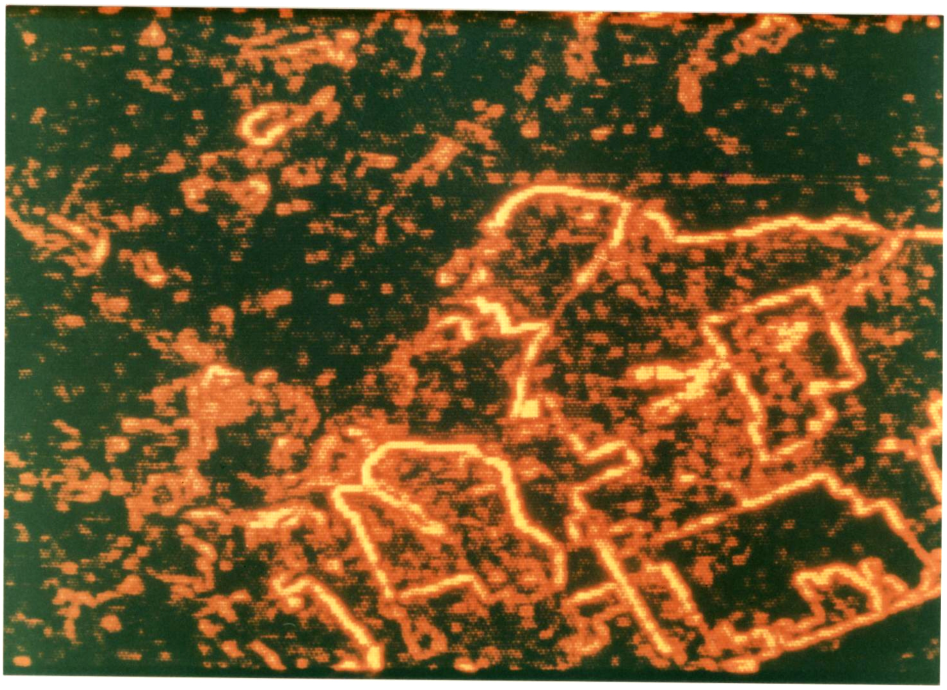

Fig. 6 - Variance Transform. Mato Grosso
do Sul sub-scene. MSS channel 6.

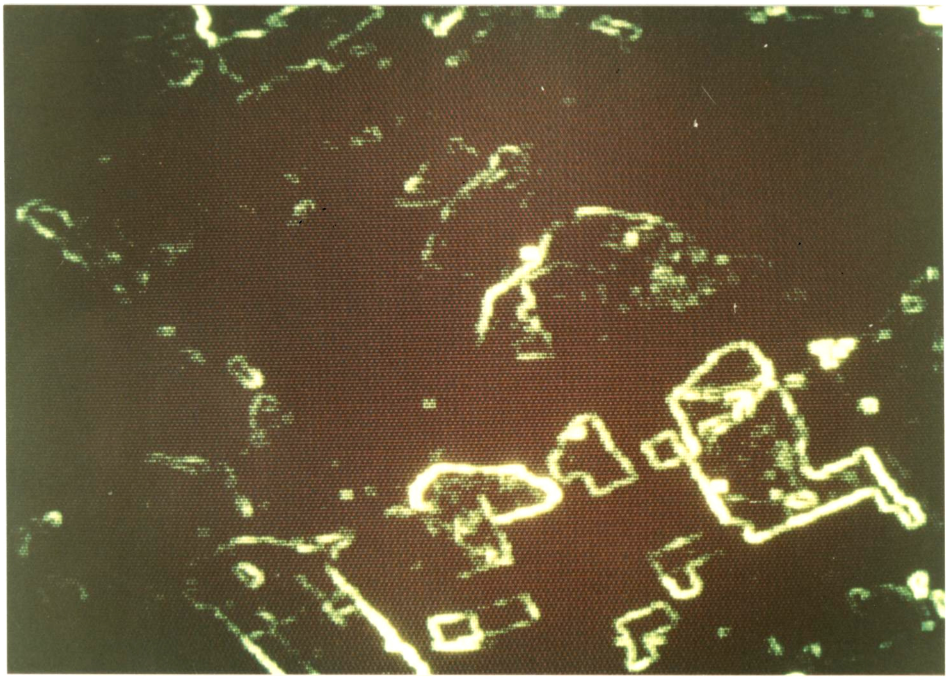

Fig. 7 - Variance Transform. Mato Grosso do Sul
sub-scene. MSS channel 5.

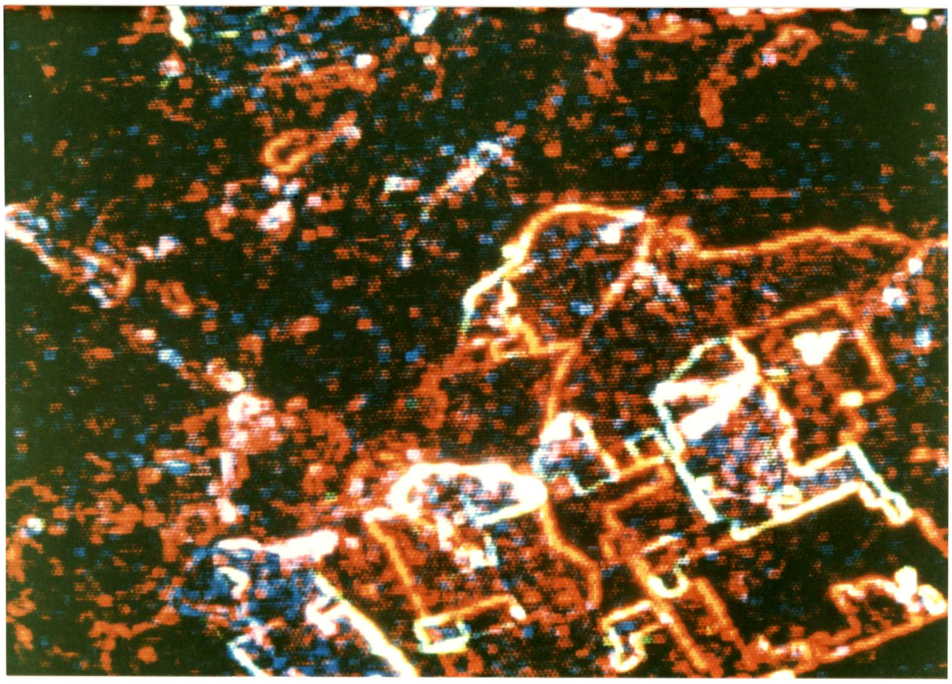

Fig. 8 - Variance Transforms. Mato Grosso do Sul sub-scene.
MSS channel 4 (blue), channel 5 (green) and channel
6 (red).

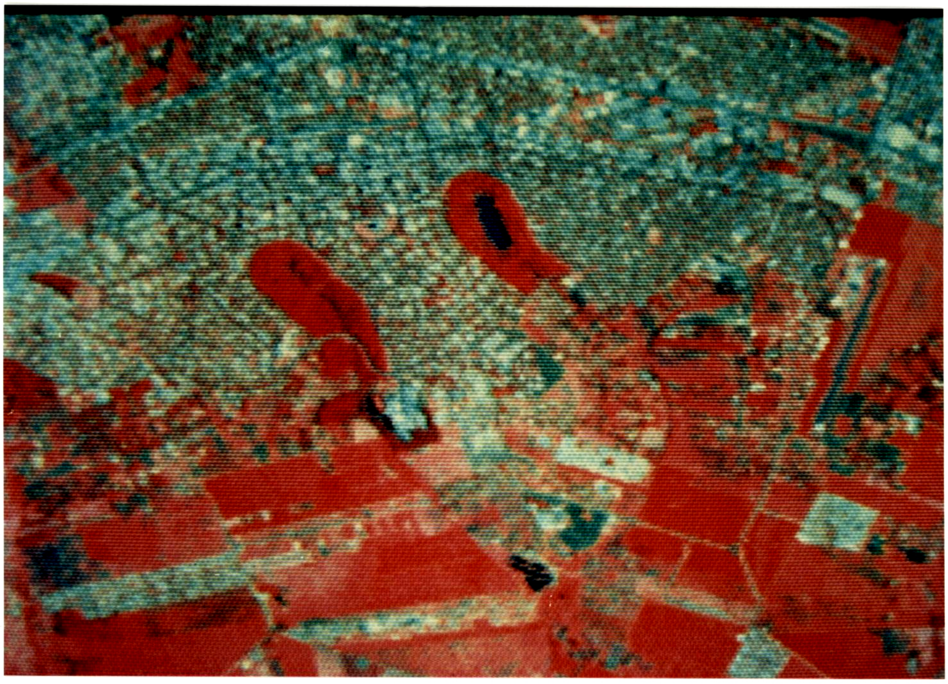

Fig. 9 - LANDSAT TM sub-scene. Maringá.

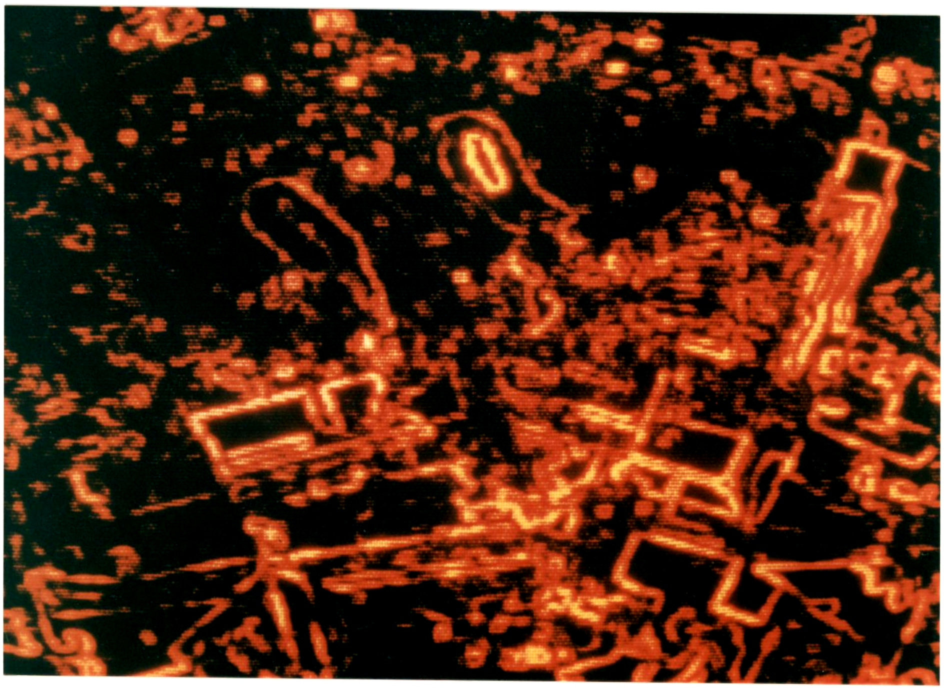

Fig. 10 - Variance Transform. Maringá sub-scene. TM channel 7.

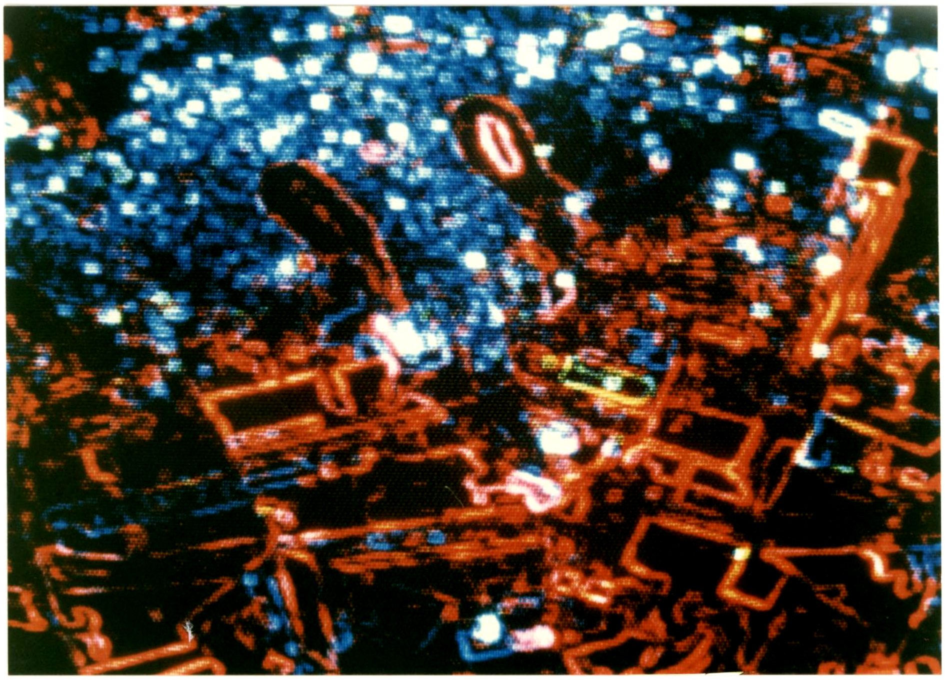

Fig. 11 - Variance Transforms. Maringá sub-scene.
TM channel 2 (blue), channel 3 (green)
and channel 7 (red).

7. REFERENCES

[1] SHOWENGERDT, R. Techniques for image processing and classification in remote sensing. New York, NY, Academic Press, 1983.

[2] GONZALEZ, R. and WINTZ, P. Digital image processing. Reading, Massachusetts, Addison Wesley, 1979.

[3] HAWKINS, J.K. Textural properties for pattern recognition, in picture processing and psycopictorics. B.S. LIPKIN and A. ROSENFELD editors, New York, NY, Academic Press, 1970.

[4] HARALICK, R.; SHANMUNGAN, K. and DIENSTEIN, I. Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics, SMC-3 (6), p. 610-621, Nov. 1973.

[5] HARALICK, R. Statistical and structural approaches to texture. Proceedings of the IEEE, 67(5), p. 786-804, May 1979.

[6] HSU, S. Texture-tone analysis for automated land-use mapping. Photogrammetric Engineering and Remote Sensing, 44(11), p. 1393-1404, Nov. 1978.

[7] IRONS, J. and PETERSEN, G. Texture transforms of remote sensing data. Remote Sensing of Environment, (11), p. 359-370, 1981.

[8] HERNANDEZ FILHO, P.; SHIMABUKURO, Y.E.; MEDEIROS, J.S.; SANTANA, C.C.; ALVES, E.C.M. Relatório das atividades do projeto IBDF/ INPE - Subprojeto Reflorestamento - Durante o ano de 1979. São José dos Campos, INPE, Jan. 1980. (INPE-1664-RPE/104).

APPENDIX A

"PROGRAM LISTINGS"

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                              CC
CC                                                                              CC
CC                                                                              CC
CC     PROGRAM texture.f READS A 4 CHANNEL LANDSAT DATA FILE AND GENERATES      CC
CC     4 CHANNEL TEXTURE DATA FILE  BY MAKING USE OF  ONE OF THE FOLLOWING      CC
CC     STATISTICAL PROPERTIES WITHIN A GIVEN WINDOW:                            CC
CC         MNL      MEAN                                                        CC
CC         VNL      VARIANCE                                                    CC
CC         SKEW     SKEWNESS                                                    CC
CC         KURT     KURTOSIS                                                    CC
CC         RNL      RANGE                                                       CC
CC         PSKEW    PEARSON'S SECOND ORDER COEFFICIENT OF SKEWNESS              CC
CC         MDIF     ABSOLUTE VALUE OF MEAN NORM LENGTH DIFFERENCES              CC
CC         MSQ      MEAN OF SQUARED NORM LENGTH DIFFERENCES                     CC
CC         MAXSQ    MAXIMUM OF SQUARED NORM LENGTH DIFFERENCES                  CC
CC                                                                              CC
CC         INPUT ***********************************************************CC
CC                                                                              CC
CC             PARAMETERS       NTOTAL  NUMBER OF PIXELS IN ONE CHANNEL ON      CC
CC                                      INPUT FILE 'landsat'                    CC
CC                              IROW    MAXIMUM NUMBER OF LINES THAT CAN BE     CC
CC                                      DISPLAYED (240 on the RIPS)             CC
CC                              ICOLUMN MAXIMUN NUMBER OF COLUMNS THAT CAN BE   CC
CC                                      DISPLAYED (256 on the RIPS)             CC
CC                                                                              CC
CC             VARIABLES        NTOTAL  TOTAL NUMBER OF PIXELS ON EACH CHANNEL  CC
CC                              NLINE                                           CC
CC                                                                              CC
CC                              NCOLUMN LINE AND COLUMN NUMBER CORRESPONDING TO CC
CC                                      THE UPPER LEFT CORNER OF THE  SUB IMAGE CC
CC                                      TO BE PROCESSED                         CC
CC                              NCHNL   NUMBER OF CHANNELS TO BE PROCESSED      CC
CC                              HALFWNW ((SIZE OF THE WINDOW TO BE USED) -1)/2  CC
CC                              NSIZE   DIMENSION OF THE IMAGE TO BE PROCESSED  CC
CC                                                                              CC
CC                  column 0                           column 255               CC
CC           row 0  _____________________________________                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC                  !                                   !                       CC
CC           row 239 !___________________________________!                      CC
CC                                                                              CC
CC                       RIPS IMAGE CHARACTERISTICS                             CC
CC                                                                              CC
CC                                                                              CC
```

```
CC    WORK AREA ***************************************************************CC
CC                                                                            CC
CC            LSAT(NTOTAL)      INTEGER ARRAY TO STORE DATA FROM FILE 'landsat'  CC
CC                              IN THE PACKED RIPS FORMAT                       CC
CC            IA(256,256)       INTEGER ARRAY TO STORE DATA FROM SUB IMAGE TO BE CC
CC                              PROCESSED                                       CC
CC            TEXTURE(256,256)  INTEGER ARRAY TO STORE THE TEXTURAL DATA        CC
CC            CHANL1(NTOTAL)                                                    CC
CC            CHANL2(NTOTAL)                                                    CC
CC            CHANL3(NTOTAL)                                                    CC
CC            CHANL4(NTOTAL)    INTEGER ARRAYS TO STORE DATA FROM INDIVIDUAL    CC
CC                              CHANNELS                                        CC
CC                                                                            CC
CC    OUTPUT   *************************************************************CC
CC                                                                            CC
CC          FILE 'texture' WHICH CONTAINS THE 4 TEXTURE CHANNELS IN THE PACKED CC
CC          RIPS FORMAT                                                       CC
CC                                                                            CC
CC    FOR MORE DETAILS PLEASE REFER TO "texture transforms of Remote Sensing"   CC
CC    by James R. Irons & Gary W. Petersen in Remote Sensing of Environment     CC
CC    1981 pp 359,370                                                         CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC
CC

      PARAMETER (NTOTAL=61440,IROW=240,ICOLUMN=256)
      PARAMETER (MASK1= 255, MASK2= 65280, MASK3= 16711680)
      PARAMETER (MASK4= 4278190080)
      INTEGER IA(256,256),TEXTURE(256,256)
      INTEGER CHANL1(NTOTAL), CHANL2(NTOTAL), CHANL3(NTOTAL)
      INTEGER CHANL4(NTOTAL)
      INTEGER LSAT(NTOTAL)
      INTEGER CHANNEL, HALFWNW
      CHARACTER*5 PROPRTY
C
      COMMON/AREA1/IA, TEXTURE
C
      OPEN(UNIT=0,FILE='lsat',ACCESS='DIRECT',FORM='UNFORMATTED',
     *     RECL=1,STATUS='OLD')
      OPEN(UNIT=1,FILE='data',ACCESS='SEQUENTIAL',FORM='FORMATTED',
     *     STATUS='OLD')
      OPEN(UNIT=2,FILE='texture',ACCESS='DIRECT',FORM='UNFORMATTED',
     *     RECL=1,STATUS='NEW')
c
c
C.....READ LANDSAT DATA - 4 CHANNELS -
c
      READ(UNIT=0,REC=1)(LSAT(I),I=1,NTOTAL)
C
C
C.....UNPACK LANDSAT DATA - SEPARATE CHANNELS -
C
C
```

```
      DO 10 I=1,NTOTAL
         CHANL1(I) = AND(MASK1,LSAT(I))
         CHANL2(I) = AND(MASK2,LSAT(I))
         CHANL3(I) = AND(MASK3,LSAT(I))
         CHANL4(I) = AND(MASK4,LSAT(I))
C
C........SHIFT THE BITS TO THE CORRECT POSITION
C
         CHANL2(I) = RSHIFT(CHANL2(I),8)
         CHANL3(I) = RSHIFT(CHANL3(I),16)
         CHANL4(I) = RSHIFT(CHANL4(I),24)
   10 CONTINUE
C
C.....READ THE LINE NUMBER (NLINE) AND COLUMN NUMBER (NCOLUMN) CORRESPONDING
C.....TO THE UPPER LEFT CORNER OF THE IMAGE TO BE PROCESSED, THE SIZE OF THE
C.....IMAGE (NSIZE) , AND THE NUMBER OF CHANNELS TO BE PROCESSED (NCHNL)
C
      READ(1,100) NLINE, NCOLUMN, NSIZE, NCHNL, HALFWNW
  100 FORMAT(5I10)
C
C.....INITIALIZE MATRIX IA
C
      DO 12 I = 1,NSIZE
         DO 11 J =1,NSIZE
            IA(I,J) = 0
   11    CONTINUE
   12 CONTINUE
C
C.....SET THE WORKING AREA IN ARRAY IA
C
      JLIMIT = NSIZE
      ILIMIT = NSIZE
      IF(JLIMIT.GT.ICOLUMN) JLIMIT = ICOLUMN
      IF(ILIMIT.GT.IROW) ILIMIT = IROW
C
C.....GO THROUGH ALL CHANNELS TO BE PROCESSED
C.....READ CHANNEL NUMBER TO BE PROCESSED AND PROPERTY TO BE IMPLEMENTED
C
      DO 22 NCHNEL = 1,NCHNL
         READ(1,102) CHANNEL, PROPRTY
  102    FORMAT(I10,A5)
C
C........INITIALIZE MATRIX TEXTURE
C
         DO 16 I= 1,NSIZE
            DO 14 J= 1,NSIZE
               TEXTURE(I,J) = 0
   14       CONTINUE
   16    CONTINUE
C
C........FORM  ARRAY IA(NSIZE,NSIZE) WHICH CONTAINS THE PART OF THE IMAGE
C........THAT IS GOING TO BE PROCESSED
C
```

```
      K = NLINE *256 + NCOLUMN
      DO 17 I=1,ILIMIT
         DO 15 J=1,JLIMIT
            K = K + 1
            IF(CHANNEL.EQ.1) IA(I,J) = CHANL1(K)
            IF(CHANNEL.EQ.2) IA(I,J) = CHANL2(K)
            IF(CHANNEL.EQ.3) IA(I,J) = CHANL3(K)
            IF(CHANNEL.EQ.4) IA(I,J) = CHANL4(K)
   15    CONTINUE
         K = K + 256 - JLIMIT
   17 CONTINUE
C
C
C........CALL THE REQUIRED SUBROUTINE
C
      IF(PROPRTY.EQ.'MNL  ') THEN
         CALL MNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'VNL  ') THEN
         CALL VNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'SKEW ') THEN
         CALL SKEW(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'KURT ') THEN
         CALL KURT(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'RNL  ') THEN
         CALL RNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'PSKEW') THEN
         CALL PSKEW(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'MDIF ') THEN
         CALL MDIF(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'MSQ  ') THEN
         CALL MSQ(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      ELSE IF(PROPRTY.EQ.'MAXSQ') THEN
         CALL MAXSQ(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
      END IF
C
C
C
C........SUBSTITUTION OF TRANSFORMED IMAGE INTO THE CORRESPONDENT CHANNEL
C
C
      K = NLINE*256 + NCOLUMN
      DO 20 I= 1, ILIMIT
         DO 18 J= 1, JLIMIT
            K = K+1
            IF(CHANNEL.EQ.1) CHANL1(K) = TEXTURE(I,J)
            IF(CHANNEL.EQ.2) CHANL2(k) = TEXTURE(I,J)
            IF(CHANNEL.EQ.3) CHANL3(K) = TEXTURE(I,J)
            IF(CHANNEL.EQ.4) CHANL4(K) = TEXTURE(I,J)
   18    CONTINUE
         K = K +256 - JLIMIT
   20 CONTINUE
   22 CONTINUE
```

```
C
C.....PACK THE FOUR CHANNELS IN A FORM SUITABLE FOR THE RIPS
C
      DO 24 I = 1,NTOTAL
         CHANL2(I) = LSHIFT(CHANL2(I),8)
         CHANL3(I) = LSHIFT(CHANL3(I),16)
         CHANL4(I) = LSHIFT(CHANL4(I),24)
C
         CHANL1(I) = OR(CHANL1(I),CHANL2(I))
         CHANL1(I) = OR(CHANL1(I),CHANL3(I))
         CHANL1(I) = OR(CHANL1(I),CHANL4(I))
   24 CONTINUE
C
C.....WRITE OUTPUT FILE 'texture'
C
      WRITE(UNIT=2,REC=1) (CHANL1(I),I=1,NTOTAL)
C
      ENDFILE (UNIT=2)
      CLOSE (UNIT=2,STATUS='KEEP')
c
      STOP
      END
C
C
      SUBROUTINE MNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                          CC
CC    SUBROUTINE FUNCTION:                                                  CC
CC    SUBROUTINE MEAN ASSIGNS THE MEAN VALUE OF PIXELS WITHIN WINDOW TO THE CC
CC    CENTRAL PIXEL                                                         CC
CC                          --                                              CC
CC                          \     X(k,l) / n                                CC
CC                          /                                               CC
CC                          --                                              CC
CC                                                                          CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INTEGER HALFWNW
      INTEGER IA(256,256), TEXTURE(256,256)
      COMMON/AREA1/IA,TEXTURE
C
      DO 16 I= 1,ILIMIT
         DO 14 J= 1,JLIMIT
            NPIXEL = 0
            DO 12 K= I-HALFWNW,I+HALFWNW
               DO 10 L= J-HALFWNW,J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     NPIXEL = NPIXEL + 1
                     TEXTURE(I,J) = TEXTURE(I,J) + IA(K,L)
                  END IF
   10          CONTINUE
   12       CONTINUE
            TEXTURE(I,J) = TEXTURE(I,J)/NPIXEL
   14    CONTINUE
   16 CONTINUE
C
      RETURN
      END
C
```

```
C
      SUBROUTINE VNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                             CC
CC    SUBROUTINE FUNCTION:                                                     CC
CC    SUBROUTINE VNL CALCULATES THE VARIANCE OF PIXELS WHITHIN THE WINDOW      CC
CC    AND ASSIGNES THIS VALUE TO THE CENTRAL PIXEL                             CC
CC                                                                             CC
CC                          __                                                 CC
CC                          \  ( x(k,l) - mean)**2 / (n-1)                     CC
CC                          /                                                  CC
CC                          --                                                 CC
CC                                                                             CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
      REAL MAX, MEAN, AUX(256,256)
      COMMON/AREA1/IA, TEXTURE
C
      MAX = 0.
      DO 20 I= 1, ILIMIT
         DO 18 J= 1, JLIMIT
            NPIXEL = 0
            MEAN = 0.
            AUX(I,J) = 0.
            DO 12 K= I-HALFWNW,I+HALFWNW
               DO 10 L= J-HALFWNW,J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     NPIXEL = NPIXEL+1
                     MEAN = MEAN + IA(K,L)
                  END IF
   10          CONTINUE
   12       CONTINUE
            MEAN = MEAN/FLOAT(NPIXEL)
            DO 16 K = I-HALFWNW,I+HALFWNW
               DO 14 L = J-HALFWNW,J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     AUX(I,J) = AUX(I,J)+(IA(K,L)-MEAN)**2
                  END IF
   14          CONTINUE
   16       CONTINUE
            AUX(I,J) = AUX(I,J)/(NPIXEL-1)
            IF(AUX(I,J).GT.MAX) MAX = AUX(I,J)
   18    CONTINUE
   20 CONTINUE
C
C     SCALE TEXTURE
C
      MAX = 255./MAX
      DO 24 I=1,ILIMIT
         DO 22 J=1, JLIMIT
            TEXTURE(I,J) = INT(AUX(I,J) * MAX)
   22    CONTINUE
   24 CONTINUE
      RETURN
      END
C
```

```
C
      SUBROUTINE SKEW(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                          CC
CC    SUBROUTINE SKEW CALCULATES THE SKEWNESS WITHIN THE WINDOW AND ASSIGNS CC
CC    ITS VALUE TO THE CENTRAL PIXEL                                        CC
CC                                                                          CC
CC                    ___                                                   CC
CC            ABS     \   (x(i,j) - mean)**3  / (n-1) * (variance)**1.5    CC
CC                    /                                                     CC
CC                    --                                                    CC
CC                                                                          CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
      REAL MEAN, VAR, AUX(256,256), MAX
      COMMON/AREA1/IA, TEXTURE
C
      MAX = 0.
      DO 24 I= 1, ILIMIT
         DO 22 J= 1, JLIMIT
            AUX(I,J) = 0.
            NPIXEL = 0
            MEAN = 0.
            VAR  = 0.
            DO 12 K= I-HALFWNW, I+HALFWNW
               DO 10 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     NPIXEL = NPIXEL+1
                     MEAN = MEAN + IA(K,L)
                  END IF
   10          CONTINUE
   12       CONTINUE
            MEAN = MEAN/NPIXEL
            DO 16 K= I-HALFWNW, I+HALFWNW
               DO 14 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     VAR = VAR + (IA(K,L) - MEAN)**2
                  END IF
   14          CONTINUE
   16       CONTINUE
            VAR = VAR/(NPIXEL-1)
            IF(VAR.EQ.0.) VAR = 1.E-10
C
            DO 20 K= I-HALFWNW, I+HALFWNW
               DO 18 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     AUX(I,J) = AUX(I,J) + (IA(K,L) - MEAN)**3
                  END IF
   18          CONTINUE
   20       CONTINUE
            AUX(I,J) = ABS(AUX(I,J))/((NPIXEL-1)*VAR**1.5)
            IF(AUX(I,J).GT.MAX) MAX = AUX(I,J)
   22    CONTINUE
   24 CONTINUE
```

```
C
C     SCALE TEXTURE(I,J)
C
      MAX = 255./MAX
      DO 28 I= 1, ILIMIT
         DO 26 J= 1, JLIMIT
            TEXTURE(I,J) = INT(AUX(I,J) * MAX)
   26    CONTINUE
   28 CONTINUE
C
      RETURN
      END
C
```

```
C
      SUBROUTINE KURT(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                            CC
CC    SUBROUTINE KURT CALCULATES THE VALUE FOR THE KURTOSIS WITHIN THE WINDOW  CC
CC    AND ASSIGNS ITS VALUE TO THE CENTRAL PIXEL                              CC
CC                                                                            CC
CC                      __                                                    CC
CC                      \    ( x(i,j) - mean)**4 / (n-1)*(variance)**2        CC
CC                      /                                                     CC
CC                      --                                                    CC
CC                                                                            CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
      REAL AUX(256,256), MEAN, VAR, MAX
      COMMON/AREA1/IA, TEXTURE
C
      DO 24 I= 1, ILIMIT
         DO 22 J= 1, JLIMIT
            AUX(I,J) = 0.
            NPIXEL = 0
            MEAN = 0.
            VAR  = 0.
            DO 12 K= I-HALFWNW, I+HALFWNW
               DO 10 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     NPIXEL = NPIXEL + 1
                     MEAN = MEAN + IA(K,L)
                  END IF
   10          CONTINUE
   12       CONTINUE
            MEAN = MEAN/NPIXEL
            DO 16 K= I-HALFWNW, I+HALFWNW
               DO 14 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     VAR = VAR + (IA(K,L) - MEAN)**2
                  END IF
   14          CONTINUE
   16       CONTINUE
            VAR = VAR/(NPIXEL-1)
            IF(VAR.EQ.0.) VAR = 1.E-10
            DO 20 K= I-HALFWNW, I+HALFWNW
               DO 18 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     AUX(I,J)=AUX(I,J) + (IA(K,L)-MEAN)**4
                  END IF
   18          CONTINUE
   20       CONTINUE
            AUX(I,J)=AUX(I,J)/((NPIXEL-1)* VAR**2)
            IF(AUX(I,J).GT.MAX) MAX = AUX(I,J)
   22    CONTINUE
   24 CONTINUE
```

```
C
C     SCALE TEXTURE
C
      MAX = 255./MAX
      DO 28 I=1, ILIMIT
         DO 26 J=1, JLIMIT
            TEXTURE(I,J) = INT(AUX(I,J) * MAX)
   26    CONTINUE
   28 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE RNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                       CC
CC    SUBROUTINE RNL CALCULATES THE RANGE WITHIN THE WINDOW AND ASSIGNS ITS CC
CC    VALUE TO THE CENTRAL PIXEL                                         CC
CC                                                                       CC
CC                      max( x(i,j) ) - min( x(i,j) )                    CC
CC                                                                       CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
      REAL MAXTOTL
      COMMON/AREA1/IA, TEXTURE
C
      DO 16 I= 1, ILIMIT
         DO 14 J= 1, JLIMIT
            MAXTOTL= 0.
            MAX = 0
            MIN = 256
            DO 12 K= I-HALFWNW, I+HALFWNW
               DO 10 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     IF(IA(K,L).GT.MAX) MAX = IA(K,L)
                     IF(IA(K,L).LT.MIN) MIN = IA(K,L)
                  END IF
   10          CONTINUE
   12       CONTINUE
            TEXTURE(I,J) = MAX - MIN
            IF(TEXTURE(I,J).GT.MAXTOTL) MAXTOTL = TEXTURE(I,J)
   14    CONTINUE
   16 CONTINUE
C
C     SCALE TEXTURE
C
      MAXTOTL = 255./MAXTOTL
      DO 20 I=1, ILIMIT
         DO 18 J=1, JLIMIT
            TEXTURE(I,J) = TEXTURE(I,J) * INT(MAXTOTL)
   18    CONTINUE
   20 CONTINUE
      RETURN
      END
C
```

```
C
      SUBROUTINE PSKEW(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                              CC
CC    SUBROUTINE PSKEW CALCULATES THE PEARSON'S SECOND COEFFICIENT OF SKEWNESS  CC
CC    AND ASSIGNES ITS VALUE TO THE CENTRAL PIXEL                               CC
CC                                                                              CC
CC                 ABS( mean - median ) / (variance)**0.5                       CC
CC                                                                              CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256),SORT(30), HALFWNW, FLAG
      REAL  AUX(256,256),MAX, MEAN
      COMMON/AREA1/IA, TEXTURE
C
      MAX = 0.
      DO 24 I= 1, ILIMIT
         DO 22 J= 1, JLIMIT
            NPIXEL = 0
            MEAN = 0.
            VAR  = 0.
            M=1
            DO 12 K= I-HALFWNW, I+HALFWNW
               DO 10 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     NPIXEL = NPIXEL + 1
                     MEAN = MEAN + IA(K,L)
                     SORT(M) = IA(K,L)
                     M = M + 1
                  END IF
   10          CONTINUE
   12       CONTINUE
            MEAN = MEAN/NPIXEL
            DO 16 K= I-HALFWNW, I+HALFWNW
               DO 14 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     VAR = VAR + (IA(K,L) - MEAN)**2
                  END IF
   14          CONTINUE
   16       CONTINUE
            VAR = VAR/(NPIXEL-1)
            IF(VAR.EQ.0.) VAR = 1. E-10
   18       FLAG = 0
            DO 20 K= 1,NPIXEL-1
               IF(SORT(K).GT.SORT(K+1)) THEN
                  TEMP= SORT(K)
                  SORT(K) = SORT(K+1)
                  SORT(K+1)=TEMP
                  FLAG=1
               END IF
   20       CONTINUE
            IF(FLAG.NE.0) GO TO 18
            MEDIAN = SORT((NPIXEL-1)/2)
            AUX(I,J) = ABS(MEAN-MEDIAN)/SQRT(VAR)
            IF(AUX(I,J).GT.MAX) MAX = AUX(I,J)
   22    CONTINUE
   24 CONTINUE
```

```
C
C     SCALE TEXTURE(I,J)
C
      MAX = 255./MAX
      DO 28 I=1, ILIMIT
         DO 26 J=1, JLIMIT
            TEXTURE(I,J) = INT(AUX(I,J) * MAX)
   26    CONTINUE
   28 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE MDIF(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                        CC
CC    SUBROUTINE MDIF CALCULATES THE ABSOLUTE VALUE OF MEAN NORM LENGTH   CC
CC    DIFFERENCES AND ASSIGNS THIS VALUE TO THE CENTRAL PIXEL             CC
CC                                                                        CC
CC                    __                                                  CC
CC            ABS     \   ( x(i,j) - x (central) ) / (n-1)                CC
CC                    /                                                   CC
CC                    --                                                  CC
CC                                                                        CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
      REAL AUX(256,256), MAX
      COMMON/AREA1/IA, TEXTURE
C
      MAX = 0.
      DO 16 I= 1, ILIMIT
         DO 14 J= 1, JLIMIT
            NPIXEL=0
            AUX(I,J) = 0.
            DO 12 K= I-HALFWNW, I+HALFWNW
               DO 10 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     NPIXEL = NPIXEL + 1
                     AUX(I,J) = AUX(I,J) + IA(K,L)-IA(I,J)
                  END IF
   10          CONTINUE
   12       CONTINUE
            AUX(I,J)= ABS(AUX(I,J)/(NPIXEL-1))
            IF(AUX(I,J).GT.MAX) MAX = AUX(I,J)
   14    CONTINUE
   16 CONTINUE
C
C     SCALE TEXTURE(I,J)
C
      MAX = 255./MAX
      DO 20 I= 1, ILIMIT
         DO 18 J= 1, JLIMIT
            TEXTURE(I,J) = INT(AUX(I,J) * MAX)
   18    CONTINUE
   20 CONTINUE
      RETURN
      END
C
```

```
      SUBROUTINE MSQ(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                             CC
CC    SUBROUTINE MSQ  CALCULATES THE MEAN OF SQUARED NORM LENGTH DIFFERENCES    CC
CC    AND ASSIGNS THIS VALUE TO THE CENTRAL PIXEL                               CC
CC                                                                             CC
CC                 __                                                          CC
CC                 \   ( x(i,j) - x(central) )**2 / (n-1)                      CC
CC                 /                                                           CC
CC                 --                                                          CC
CC                                                                             CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
      REAL AUX(256,256),MAX
      COMMON/AREA1/IA, TEXTURE
C
      MAX = 0.
      DO 16 I= 1, ILIMIT
         DO 14 J= 1, JLIMIT
            NPIXEL = 0
            DO 12 K= I-HALFWNW, I+HALFWNW
               DO 10 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     NPIXEL = NPIXEL + 1
                     AUX(I,J) = AUX(I,J) + (IA(K,L)-IA(I,J))**2
                  END IF
   10          CONTINUE
   12       CONTINUE
            AUX(I,J) = AUX(I,J)/(NPIXEL-1)
            IF(AUX(I,J).GT.MAX) MAX = AUX(I,J)
   14    CONTINUE
C
C     SCALE TEXTURE(I,J)
C
   16 CONTINUE
      MAX = 255./MAX
      DO 20 I=1,ILIMIT
         DO 18 J=1,JLIMIT
            TEXTURE(I,J) = INT(AUX(I,J) * MAX)
   18    CONTINUE
   20 CONTINUE
      RETURN
      END
C
C
C
```

```
      SUBROUTINE MAXSQ(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                                          CC
CC    SUBROUTINE MAXSQ CALCULATES THE MAXIMUM OF SQUARED NORM LENGTH         CC
CC    DIFFERENCES AND ASSIGNS THIS VALUE TO THE CENTRAL PIXEL               CC
CC                                                                          CC
CC                max( x(i,j) - x(central) )**2                             CC
CC                                                                          CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
      INTEGER MAX2, AUX
      REAL MAX1
      COMMON/AREA1/IA, TEXTURE
C
C
      MAX1 = 0.
      DO 16 I= 1, ILIMIT
         DO 14 J= 1, JLIMIT
            MAX2 = 0
            DO 12 K= I-HALFWNW, I+HALFWNW
               DO 10 L= J-HALFWNW, J+HALFWNW
                  IF(K.GE.1.AND.K.LE.IROW.AND.
     *               L.GE.1.AND.L.LE.ICOLUMN) THEN
                     AUX = (IA(K,L) - IA(I,J))**2
                     IF(AUX.GT.MAX2) MAX2 = AUX
                  END IF
   10          CONTINUE
   12       CONTINUE
            TEXTURE(I,J) = MAX2
            IF(TEXTURE(I,J).GT.MAX1) MAX1 = TEXTURE(I,J)
   14    CONTINUE
   16 CONTINUE
C
C     SCALE TEXTURE(I,J)
C
      MAX1 = 255./MAX1
      DO 20 I= 1,ILIMIT
         DO 18 J= 1,JLIMIT
            TEXTURE(I,J) = TEXTURE(I,J) * MAX1
   18    CONTINUE
   20 CONTINUE
      RETURN
      END
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                             c
c     program function:                                                       c
c                                                                             c
c     program inpe.f reads landsat data (INPE I-100 format) and changes to    c
c     RIPS and TEKTRONICS format                                              c
c                                                                             c
c     input  *********************************************************        c
c                                                                             c
c            binary file 'itapeva' which contains 5 channels:  the four       c
c            LANDSAT channels and a fifth channel which contains the          c
c            image classification                                             c
c            the size of the image is (512,512) and the channels are stored   c
c            in a single file, sequentially                                   c
c                                                                             c
c            variables:                                                       c
c                                                                             c
c            nline, ncolumn    line and column number of the upper left       c
c                              corner of part of (512,512) image to be        c
c                              displayed at the RIPS (256,249)                c
c            lines columns     size of the image to be displayed at the RIPS  c
c                              maximum values:  lines = 240 - columns= 256    c
c                                                                             c
c                                                                             c
c     output *********************************************************        c
c                                                                             c
c            binary files:     'mtgrosso' which contains the four LANDSAT     c
c                              channels in the RIPS (.DAT) format             c
c                              'class' which contains the fifth channel       c
c                              (classification) on the RIPS (.IMG) format     c
c                                                                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
      parameter(limitl=327680, limit2=65536, limit3=15360, limit4=61440)
      parameter(mask1=255, mask2=65280, mask3=16711680)
      parameter(mask4=4278190080)
c
      integer lsat(limitl), lsatl(limit3), lsat2(limit3), lsat3(limit3)
      integer lsat4(limit3), lsat5(limit3), chanll(limit4)
      integer chanl2(limit4), chanl3(limit4), chanl4(limit4)
      integer lines, columns
c
      open(unit=0,file='itapeva',access='direct', form='unformatted',
     *     recl=1,status='old')
      open(unit=1,file='mtgrosso',access='direct',form='unformatted',
     *     recl=1,status='new')
      open(unit=2,file='class',access='direct',form='unformatted',
     *     recl=1,status='new')
      open(unit=3,file='datI100',access='sequential',form='formatted',
     *     status='old')
c
c.....read input file 'itapeva'  I-100  INPE
c
      read(unit=0,rec=1) (lsat(i),i=1,limitl)
c
```

```
c.....read the line number(nline) and the column number (ncolumn) of the
c.....upper left corner of the image
c.....read the size of the image 'lines' and 'columns'.  note that on the
c.....RIPS lines=240  and columns=256
c
      read(3,100) nline, ncolumn, lines, columns
  100 format(4i10)
c
c.....separate the five channels
c
      ncolumn = ncolumn/4
      columns = columns/4
      k1 = nline * 128 + ncolumn
      k2 = 0
c
      do 12 i=1,lines
         do 10 j=1,columns
            k1=k1+1
            k2=k2+1
            lsat1(k2) = lsat(k1)
            lsat2(k2) = lsat(k1+   limit2)
            lsat3(k2) = lsat(k1+ 2*limit2)
            lsat4(k2) = lsat(k1+ 3*limit2)
            lsat5(k2) = lsat(k1+ 4*limit2)
   10    continue
         k1 = k1+ 128 - columns
   12 continue
c
c.....unpack channels 1 through 4
c
      k = 0
      do 14 i=1,limit3
         k=k+1
         chanl1(k) = and(mask1,lsat1(i))
         chanl2(k) = and(mask1,lsat2(i))
         chanl3(k) = and(mask1,lsat3(i))
         chanl4(k) = and(mask1,lsat4(i))
c
         k=k+1
c
         chanl1(k) = and(mask2,lsat1(i))
         chanl2(k) = and(mask2,lsat2(i))
         chanl3(k) = and(mask2,lsat3(i))
         chanl4(k) = and(mask2,lsat4(i))
c
         k=k+1
c
         chanl1(k) = and(mask3,lsat1(i))
         chanl2(k) = and(mask3,lsat2(i))
         chanl3(k) = and(mask3,lsat3(i))
         chanl4(k) = and(mask3,lsat4(i))
c
         k=k+1
c
         chanl1(k) = and(mask4,lsat1(i))
         chanl2(k) = and(mask4,lsat2(i))
         chanl3(k) = and(mask4,lsat3(i))
         chanl4(k) = and(mask4,lsat4(i))
   14 continue
```

```
c
c.....pack the data into the RIPS format
c
      do 16 i=1,limit4,4
         chanl2(i)   = lshift(chanl2(i),8)
         chanl3(i)   = lshift(chanl3(i),16)
         chanl4(i)   = lshift(chanl4(i),24)
c
         chanl1(i+1) = rshift(chanl1(i+1),8)
         chanl3(i+1) = lshift(chanl3(i+1),8)
         chanl4(i+1) = lshift(chanl4(i+1),16)
c
         chanl1(i+2) = rshift(chanl1(i+2),16)
         chanl2(i+2) = rshift(chanl2(i+2),8)
         chanl4(i+2) = lshift(chanl4(i+2),8)
c
         chanl1(i+3) = rshift(chanl1(i+3),24)
         chanl2(i+3) = rshift(chanl2(i+3),16)
         chanl3(i+3) = rshift(chanl3(i+3),8)
c
   16 continue
c
      do 18 i=1,limit4
         chanl1(i) = or(chanl1(i),chanl2(i))
         chanl1(i) = or(chanl1(i),chanl3(i))
         chanl1(i) = or(chanl1(i),chanl4(i))
   18 continue
c
c.....write output file
c
      write(unit=1,rec=1) (chanl1(i),i=1,limit4)
      endfile(unit=1)
      close(unit=1,status='keep')
c
      write(unit=2,rec=1) (lsat5(i),i=1,limit4)
      endfile(unit=2)
      close(unit=2,status='keep')
c
      stop
      end
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
c     program function:                                                 c
c     program inpe.tm.f reads a 256 by 240 pixels area from a LANDSAT-TM CCT  c
c     format INPE-BRAZIL (one channel each time) and outputs two binary files c
c     'channel.img' and 'channel.dat' according to RIPS (.IMG) and (.DAT)  c
c     formats respectively.                                              c
c                                                                        c
c     **** input *******************************************************c
c                                                                        c
c     file 'lsat.tm' which contains one LANDSAT-TM channel formatted as in the c
c     INPE-BRAZIL CCTs.                                                  c
c                                                                        c
c     nline                                                              c
c     ncolumn  line and column number corresponding to the upper left corner  c
c     of the area selected to be displayed. Note that the input file contains c
c     one quadrant of a LANDSAT-TM scene i.e.: 3088 lines with 3600 bytes each c
c                                                                        c
c     **** output ******************************************************c
c                                                                        c
c     binary files 'channel.img' and 'channel.dat' of size 256 by 240 pixels  c
c     to be displayed at the RIPS                                        c
c                                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c

      parameter(nsize1=240*256, nsize2=4*240*256)
      parameter(mask1=255, mask2=65280, mask3=16711680)
      parameter(mask4=4278190080)
c
      integer lsat(nsize1), chnl(nsize2), nline, ncolumn, nrec
c
      open(unit=0,file='lsat.tm',access='direct',form='unformatted',
     *     recl=4,status='old')
      open(unit=1,file='channel.img',access='direct',form='unformatted',
     *     recl=1,status='new')
      open(unit=2,file='channel.dat',access='direct',form='unformatted',
     *     recl=1,status='new')
      open(unit=3,file='data.tm',access='sequential',form='formatted',
     *     status='old')
c
c.....read line number (nline) and column number (ncolumn) corresponding
c.....to the upper left corner of the area to be displayed.  NOTE that
c.....ncolumn must be a multiple of 4.
c
      read(3,100) nline, ncolumn
  100 format(2i10)
c
c.....adjust ncolumn to the record length (recl=4) to access the proper record
c.....in the image file
c
      ncolumn=ncolumn/4
c
c.....add 8 records (32 bytes) which correspond to the prefix data
c
      ncolumn=ncolumn+8
c
c.....adjust nline for the first line (which is the file descriptor record)
c
      nline=nline + 1
c
```

```
c.....calculate the first record to be read in image file
c.....note that each line is 3600 bytes long. since we have specified recl=4
c.....then each line has 900 records
c
      nrec = 900 * (nline-1) + ncolumn
c
      k=0
      nrec=nrec-1
c
      do 12 i=1,240
         do 10 j=1,64
            k = k +1
            nrec = nrec + 1
            read(unit=0,rec=nrec) lsat(k)
   10    continue
         nrec = nrec + 900 - 64
   12 continue
c
c.....write output file 'channel.img' on RIPS (.IMG) format
c
      write(unit=1,rec=1) (lsat(i),i=1,k)
      endfile(unit=1)
      close(unit=1,status='keep')
c
c.....write output file 'channel.dat' on RIPS (.DAT) format
c
      j = 0
      do 14 i=1,4*k,4
         j = j + 1
         chnl(i)   = and(mask1,lsat(j))
         chnl(i+1) = and(mask2,lsat(j))
         chnl(i+2) = and(mask3,lsat(j))
         chnl(i+3) = and(mask4,lsat(j))
c
         chnl(i+1) = rshift(chnl(i+1),8)
         chnl(i+2) = rshift(chnl(i+2),16)
         chnl(i+3) = rshift(chnl(i+3),24)
   14 continue
c
      write(unit=2,rec=1) (chnl(i),i=1,4*k)
      endfile(unit=2)
      close(unit=2,status='keep')
c
      stop
      end
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
c     program function:                                                 c
c     program display.rips.f is a complement to program  ' inpe.tm.f '  c
c     it takes as input four of the output files (LANDSAT-TM channels) from  c
c     'inpe.tm.f' which are on (.DAT) format and pack them into a single file  c
c     in the RIPS (.DAT) format for a multi-channel display             c
c                                                                       c
c     **** input ******************************************************c
c                                                                       c
c     channel1                                                          c
c     channel2                                                          c
c     channel3                                                          c
c     channel4                                                          c
c              are the four selected channels from a single image to be packed  c
c              into a single file on RIPS (.DAT) format                 c
c                                                                       c
c     **** output *****************************************************c
c                                                                       c
c     display                                                           c
c             file containing the four selected channels on RIPS (.DAT) format  c
c                                                                       c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      parameter(limit1=4*256*240, limit2=256*240)
c
      integer lsat(limit1), channel(limit2)
c
      open(unit=1,file='channel1',access='direct',form='unformatted',
     *     recl=4,status='old')
      open(unit=2,file='channel2',access='direct',form='unformatted',
     *     recl=4,status='old')
      open(unit=3,file='channel3',access='direct',form='unformatted',
     *     recl=4,status='old')
      open(unit=4,file='channel4',access='direct',form='unformatted',
     *     recl=4,status='old')
      open(unit=5,file='display', access='direct',form='unformatted',
     *     recl=1,status='new')
c
c.....read the four channels into the array 'lsat' and shift the bits to the
c.....convenient position within the word for packing into the RIPS (.DAT)
c.....format.  Note that this is a 32 bit 4 bytes word
c
      nrec=0
c
      do 10 i=1,limit1,4
         nrec=nrec+1
c
         read(unit=1,rec=nrec) lsat(i)
c
         read(unit=2,rec=nrec) lsat(i+1)
         lsat(i+1)= lshift(lsat(i+1),8)
c
         read(unit=3,rec=nrec) lsat(i+2)
         lsat(i+2)= lshift(lsat(i+2),16)
c
         read(unit=4,rec=nrec) lsat(i+3)
         lsat(i+3)= lshift(lsat(i+3),24)
   10 continue
```

```
c
c.....pack into the RIPS (.DAT) format
c
      k=0
      do 12 i=1,limit1,4
         k=k+1
         channel(k) = lsat(i)
         channel(k) = or(channel(k),lsat(i+1))
         channel(k) = or(channel(k),lsat(i+2))
         channel(k) = or(channel(k),lsat(i+3))
   12 continue
c
c.....write file 'display'
c
      write(unit=5,rec=1) (channel(i),i=1,limit2)
      endfile(unit=5)
      close(unit=5,status='keep')
c
      stop
      end
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                             c
c     program function:                                                       c
c     the aim is to have a general overview of one quadrant of LANDSAT-TM    c
c     imagery (which is 3244 by 3088 pixels) on the screen of the RIPS which c
c     is 256 by 240 pixels in size.  This is acomplished by displaying every c
c     12th pixel of the original image.                                       c
c                                                                             c
c     ****  input  ********************************************************  c
c                                                                             c
c     file which contains one LANDSAT-TM channel formatted as in the          c
c     INPE-BRAZIL CCTs                                                        c
c                                                                             c
c     ****  output  *******************************************************  c
c                                                                             c
c     binary file 'overview' which is 256 by 240 pixels in size and on        c
c     RIPS (.IMG) format                                                      c
c                                                                             c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
      parameter(mask1=255, limitl=61440)
      integer lsat(limitl)
c
      open(unit=0,file='tm.chnl3',access='direct',form='unformatted',
     *     recl=4,status='old')
      open(unit=1,file='overview',access='direct',form='unformatted',
     *     recl=1,status='new')
c
c.....read tm imagery INPE-BRAZIL format every 12th line and every 12th pixel
c.....to get a general overview of the image on RIPS screen
c
      n = 0
      do 12 i=2,2881,12
c........j is the number of records  at the beginning of this image line
c........note that there are 32 bytes or 8 records at the beginning of
c........each line which correspond to the prefix data and should be skipped
c........note that the reason to take record length equal to 4 bytes for the
c........input file (binary file) is that lsat(i) has been defined as an
c........integer and so it will take one word (4 bytes)
c........we will take 2880 lines which, sampled every 12th line will generate
c........240 lines, the size of the RIPS screen.
c........we also take 3072 pixels/line which sampled every 12th pixel will
c........generate 256 pixels/line, the size of RIPS screen.  3072 pixels
c........correspond to 900 records
c
         j= 900 * i + 8
         do 10 k=1,768,3
            nrec= j + k
            n = n + 1
            read(unit=0,rec=nrec) lsat(n)
c...........take only one pixel
            lsat(n) = and(mask1,lsat(n))
   10    continue
   12 continue
```

```
c
c.....pack these pixels into the RIPS (.IMG) format
c
      do 14 i=1,n,4
         lsat(i+1) = lshift(lsat(i+1),8)
         lsat(i+2) = lshift(lsat(i+2),16)
         lsat(i+3) = lshift(lsat(i+3),24)
c
         lsat(i)   = or(lsat(i),lsat(i+1))
         lsat(i)   = or(lsat(i),lsat(i+2))
         lsat(i)   = or(lsat(i),lsat(i+3))
   14 continue
c
c.....write the output file 'overview'
c
      write(unit=1,rec=1) (lsat(i),i=1,n,4)
      endfile(unit=1)
      close(unit=1,status='keep')
c
      stop
      end
```