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.

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

* Instituto de Pesquisas Hidráulicas - UFRS, Porto Alegre, RS, Brasil.
TEXTURAL FEATURES FOR IMAGE CLASSIFICATION IN REMOTE SENSING

Vitor Haertel
Instituto de Pesquisas Hidrâulicas
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
ACKNOWLEDGMENTS

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

1. INTRODUCTION ................................................. 1  
2. INSTRUMENTATION USED ......................................... 2  
3. TEXTURE TRANSFORMS ........................................... 3  
4. EXPERIMENTS .................................................. 5  
5. RESULTS ...................................................... 8  
6. CONCLUSIONS .................................................. 10  
7. REFERENCES ................................................... 17
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LANDSAT MSS sub-scene. North Dakota</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>Variance Transform. North Dakota sub-scene. MSS channel 6</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Variance Transform. North Dakota sub-scene. MSS channel 5</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>Variance Transform. North Dakota sub-scene. MSS channel 4 (blue), channel 5 (green) and channel 6 (red)</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>LANDSAT MSS sub-scene. Mato Grosso do Sul</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Variance Transform. Mato Grosso do Sul sub-scene. MSS channel 6</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>Variance Transform. Mato Grosso do Sul sub-scene. MSS channel 5</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Variance Transforms. Mato Grosso do Sul sub-scene. MSS channel 4 (blue), channel 5 (green) and channel 6 (red)</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>LANDSAT TM sub-scene. Maringá</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>Variance Transform. Maringá sub-scene. TM channel 7</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>Variance Transforms. Maringá sub-scene. TM channel e (blue), channel 3 (green) and channel 7 (red)</td>
<td>16</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

Digital image processing is the numerical manipulation of digital images. It includes preprocessing, enhancement and classification [1]. Preprocessing 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 illumination 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 performs 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 in 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 calculated.
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:

\[ \text{Mean} - MNL - \quad MNL = \frac{\sum x_{ij}}{n} \]

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 -

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

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 -

\[ MDIF = \frac{\left| \sum x_{ij} - x_c \right|}{(n-1)} \]

where \( x_c \) represents 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 length 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 Paraná, in southern Brazil. The area selected covers part of the city of Maringá 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 transform" 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 chlorophyll 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 transform 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 opposite 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 successfully 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


PROGRAM texture.f READS A 4 CHANNEL LANDSAT DATA FILE AND GENERATES 4 CHANNEL TEXTURE DATA FILE BY MAKING USE OF ONE OF THE FOLLOWING STATISTICAL PROPERTIES WITHIN A GIVEN WINDOW:

- MNL MEAN
- VNL VARIANCE
- SKEW SKEWNESS
- KURT KURTOSIS
- RNL RANGE
- PSKEW PEARSON'S SECOND ORDER COEFFICIENT OF SKEWNESS
- MDIF ABSOLUTE VALUE OF MEAN NORM LENGTH DIFFERENCES
- MSQ MEAN OF SQUARED NORM LENGTH DIFFERENCES
- MAXSQ MAXIMUM OF SQUARED NORM LENGTH DIFFERENCES

INPUT *****************************************

PARAMETERS
- NTOTAL NUMBER OF PIXELS IN ONE CHANNEL ON INPUT FILE 'landsat'
- IROW MAXIMUM NUMBER OF LINES THAT CAN BE DISPLAYED (240 on the RIPS)
- ICOLUMN MAXIMUM NUMBER OF COLUMNS THAT CAN BE DISPLAYED (256 on the RIPS)

VARIABLES
- NTOTAL TOTAL NUMBER OF PIXELS ON EACH CHANNEL
- NLINE
- NCOLUMN LINE AND COLUMN NUMBER CORRESPONDING TO THE UPPER LEFT CORNER OF THE SUB IMAGE TO BE PROCESSED
- NCHNL NUMBER OF CHANNELS TO BE PROCESSED
- HALFWW ((SIZE OF THE WINDOW TO BE USED) -1)/2
- NSIZE DIMENSION OF THE IMAGE TO BE PROCESSED

column 0  column 255

row 0  

row 239

RIPS IMAGE CHARACTERISTICS
A.3

CC WORK AREA

INTEGER LSAT(NTOTAL) INTEGER ARRAY TO STORE DATA FROM FILE 'landsat' IN THE PACKED RIPS FORMAT
INTEGER IA(256,256) INTEGER ARRAY TO STORE DATA FROM SUB IMAGE TO BE PROCESSED
INTEGER TEXTURE(256,256) INTEGER ARRAY TO STORE THE TEXTURAL DATA
INTEGER CHANL1(NTOTAL)
INTEGER CHANL2(NTOTAL)
INTEGER CHANL3(NTOTAL)
INTEGER CHANL4(NTOTAL) INTEGER ARRAYS TO STORE DATA FROM INDIVIDUAL CHANNELS

OUTPUT

FILE 'texture' WHICH CONTAINS THE 4 TEXTURE CHANNELS IN THE PACKED RIPS FORMAT

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

COMMON/AREA1/IA, TEXTURE

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.....READ LANDSAT DATA - 4 CHANNELS -
READ(UNIT=0, REC=1)(LSAT(I), I=1, NTOTAL)

C.....UNPACK LANDSAT DATA - SEPARATE CHANNELS -
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........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.NCOLUMN) JLIMIT = NCOLUMN
  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 NCHNL = 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........CALL THE REQUIRED SUBROUTINE
C
IF (PROPERTY.EQ. 'MNL ') THEN
   CALL MNL(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'VNL ') THEN
   CALL VNL(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'SKEW ') THEN
   CALL SKEW(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'KURT ') THEN
   CALL KURT(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'RNL ') THEN
   CALL RNL(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'PSKEW') THEN
   CALL PSKEW(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'MDIF ') THEN
   CALL MDIF(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'MSQ ') THEN
   CALL MSQ(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
ELSE IF (PROPERTY.EQ. 'MAXSQ') THEN
   CALL MAXSQ(ILIMIT, JLIMIT, IROW, COLUMN, HALFWNW)
END IF

C........SUBSTITUTION OF TRANSFORMED IMAGE INTO THE CORRESPONDENT CHANNEL
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
PACK THE FOUR CHANNELS IN A FORM SUITABLE FOR THE RIPS

DO 24 I = 1, NTOTAL
    CHANL2(I) = LSHIFT(CHANL2(I),8)
    CHANL3(I) = LSHIFT(CHANL3(I),16)
    CHANL4(I) = LSHIFT(CHANL4(I),24)
    CHANL1(I) = OR(CHANL1(I),CHANL2(I))
    CHANL1(I) = OR(CHANL1(I),CHANL3(I))
    CHANL1(I) = OR(CHANL1(I),CHANL4(I))
24 CONTINUE

WRITE OUTPUT FILE 'texture'

WRITE(UNIT=2,REC=1) (CHANL1(I),I=1,NTOTAL)
ENDFILE (UNIT=2)
CLOSE (UNIT=2,STATUS='KEEP')
STOP
END

SUBROUTINE MNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)

INTEGER HALFWNW
INTEGER IA(256,256), TEXTURE(256,256)
COMMON/AREA/.IA,TEXTURE

DO 16 I = 1,ILIMIT
    DO 14 J = 1,JLIMIT
        NPIXEL = 0
        DO 12 K = I-HALFWNW,I+HALFWNW
            IF(K.GE.I.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
        12 CONTINUE
    14 CONTINUE
    TEXTURE(I,J) = TEXTURE(I,J)/NPIXEL
16 CONTINUE
RETURN
END
SUBROUTINE VNL(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)

CC SUBROUTINE FUNCTION:
CC SUBROUTINE VNL CALCULATES THE VARIANCE OF PIXELS WITHIN THE WINDOW
CC AND ASSIGNES THIS VALUE TO THE CENTRAL PIXEL
CC
CC        \        \    __ ( x(k,l) - mean)**2 / (n-1)
CC        /        /    
CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
REAL MAX, MEAN, AUX(256,256)
COMMON/AREA1/IA, TEXTURE

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.I.AND.K.LE.IROW.AND.
               L.GE.I.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.I.AND.K.LE.IROW.AND.
               L.GE.I.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 SCALE TEXTURE

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
SUBROUTINE SKEW(ILIMIT, JLIMIT, IROW, ICOLUMN, HALFWNW)

SUBROUTINE SKEW CALCULATES THE SKEWNESS WITHIN THE WINDOW AND ASSIGNS ITS VALUE TO THE CENTRAL PIXEL

ABS \[ \frac{(x(i,j) - \text{mean})^3}{(n-1) \times (\text{variance})^{1.5}} \]

INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
REAL MEAN, VAR, AUX(256,256), MAX
COMMON/AREA1/IA, TEXTURE

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.I.AND.K.LE.IROW.AND.L.GE.I.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.I.AND.K.LE.IROW.AND.L.GE.I.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.I.AND.K.LE.IROW.AND.L.GE.I.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  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
SUBROUTINE KURT(ILIMIT, JLIMIT, IROW, ICOLUMN, HALFWNW)

SUBROUTINE KURT CALCULATES THE VALUE FOR THE KURTOSIS WITHIN THE WINDOW AND ASSIGN ITS VALUE TO THE CENTRAL PIXEL

\[
\frac{\sum (x(i,j) - \text{mean})^4}{(n-1)(\text{variance})^2}
\]

INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
REAL AUX(256,256), MEAN, VAR, MAX
COMMON/AREA1/IA, TEXTURE

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.I.AND.K.LE.IROW.AND.
               L.GE.I.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.I.AND.K.LE.IROW.AND.
               L.GE.I.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.I.AND.K.LE.IROW.AND.
               L.GE.I.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
SCALE TEXTURE

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

SUBROUTINE RNL(ILIMIT, JLIMIT, IROW, ICOLUMN, HALFWNW)

INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
REAL MAXTOTL
COMMON/AREA1/IA, TEXTURE

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.I.AND.K.LE.IROW.AND.
               L.GE.I.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

SCALE TEXTURE

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
SUBROUTINE PSKEW(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)

SUBROUTINE PSKEW CALCULATES THE PEARSON'S SECOND COEFFICIENT OF SKEWNESS AND ASSIGNES ITS VALUE TO THE CENTRAL PIXEL

ABS( mean - median ) / (variance)**0.5

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.I.AND.K.LE.IROW.AND.
               L.GE.I.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.I.AND.K.LE.IROW.AND.
               L.GE.I.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
      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)/SORT(VAR)
      IF(AUX(I,J).GT.MAX) MAX = AUX(I,J)
   22 CONTINUE
24 CONTINUE
SUBROUTINE MDIF(ILIMIT, JLIMIT, IROW, ICOLUMN, HALFWNW)

INTEGER IA(256, 256), TEXTURE(256, 256), HALFWNW
REAL AUX(256, 256), MAX
COMMON/AREA1/IA, TEXTURE

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. I .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
         CONTINUE
      CONTINUE
      AUX(I, J) = ABS(AUX(I, J) / (NPIXEL - 1))
      IF (AUX(I, J) .GT. MAX) MAX = AUX(I, J)
   CONTINUE
14 CONTINUE
16 CONTINUE

SCALE TEXTURE(I, J)

MAX = 255. / MAX
DO 20 I = 1, ILIMIT
   DO 18 J = 1, JLIMIT
      TEXTURE(I, J) = INT(AUX(I, J) * MAX)
   CONTINUE
20 CONTINUE
RETURN
END
SUBROUTINE MSQ(ILIMIT, JLIMIT, IROW, ICOLUMN, HALFWNW)

SUBROUTINE MSQ calculates the mean of squared norm length differences and assigns this value to the central pixel.

\[
\frac{1}{n-1} \sum \left( x(i,j) - x(\text{central}) \right)^2
\]

INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
REAL AUX(256,256), MAX
COMMON/AREA1/IA, TEXTURE

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.I.AND.K.LE.IROW.AND.L.GE.J.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
SCALE TEXTURE(I,J)
MAX = 255./MAX
DO 20 I = 1, ILIMIT
   DO 18 J = 1, JLIMIT
      TEXTURE(I,J) = INT(AUX(I,J) * MAX)
   18 CONTINUE
RETURN
END
SUBROUTINE MAXSQ(ILIMIT,JLIMIT,IROW,ICOLUMN,HALFWNW)

    INTEGER IA(256,256), TEXTURE(256,256), HALFWNW
    INTEGER MAX2, AUX
    REAL MAX1
    COMMON/AREA1/IA, TEXTURE

    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.I.AND.K.LE.IROW.AND.
                L.GE.J.AND.L.LE.ICOLUMN) THEN
                AUX = (IA(K,L) - IA(I,J))**2
                IF(AUX.GT.MAX2) MAX2 = AUX
            END IF
          END DO 10
        END DO 12
        TEXTURE(I,J) = MAX2
        IF(TEXTURE(I,J).GT.MAX1) MAXI = TEXTURE(I,J)
      END DO 14
    END DO 16

    MAX1 = 255./MAX1
    DO 20 I = 1, ILIMIT
      DO 18 J = 1, JLIMIT
        TEXTURE(I,J) = TEXTURE(I,J) * MAXI
      END DO 18
    END DO 20
    RETURN
END
program inpe.f reads landsat data (INPE I-100 format) and changes to RIPS and TEKTRONICS format

Input: 
- binary file 'itapeva' which contains 5 channels: the four LANDSAT channels and a fifth channel which contains the image classification
- the size of the image is (512,512) and the channels are stored in a single file, sequentially

Variables:
- nline, ncolumn: line and column number of the upper left corner of part of (512,512) image to be displayed at the RIPS (256,249)
- lines, columns: size of the image to be displayed at the RIPS
- maximum values: lines = 240 - columns= 256

Output: 
- binary files: 'mtgrosso' which contains the four LANDSAT channels in the RIPS (.DAT) format
- 'class' which contains the fifth channel (classification) on the RIPS (.IMG) format

Parameter:
- (limit1=327680, limit2=65536, limit3=15360, limit4=61440)
- (mask1=255, mask2=65280, mask3=16711680)
- (mask4=4278190080)

Integer:
- lsat(limit1), lsatl(limit3), lsat2(limit3), lsat3(limit3)
- lsat4(limit3), lsat5(limit3), chan1(limit4)
- chan2(limit4), chan3(limit4), chan4(limit4)
- nlines, columns

Open:
- unit=0, file='itapeva', access='direct', form='unformatted',
- unit=1, file='mtgrosso', access='direct', form='unformatted',
- unit=2, file='class', access='direct', form='unformatted',
- unit=3, file='datI100', access='sequential', form='formatted',
- status='old'

......read input file 'itapeva' I-100 INPE

read(unit=0, rec=1) (lsat(i), i=1, limit1)
read the line number(nline) and the column number (ncolumn) of the
upper left corner of the image
read the size of the image 'lines' and 'columns'. note that on the
RIPS lines=240 and columns=256

read(3,100) nline, ncolumn, lines, columns
100 format(4i10)

separate the five channels
ncolumn = ncolumn/4
columns = columns/4
k1 = nline * 128 + ncolumn
k2 = 0

do 12 i=1,lines
   do 10 j=1,columns
      kl=kl+1
      k2=k2+1
      lsatl(k2) = lsat(kl)
      lsat2(k2) = lsat(kl+ limit2)
      lsat3(k2) = lsat(kl+ 2*limit2)
      lsat4(k2) = lsat(kl+ 3*limit2)
      lsat5(k2) = lsat(kl+ 4*limit2)
   10 continue
   kl = kl+ 128 - columns
12 continue

unpack channels 1 through 4
k = 0
   do 14 i=1,limit3
      k=k+1
      chanll(k) = and(mask1,lsatl(i))
      chanl2(k) = and(mask1,lsat2(i))
      chanl3(k) = and(mask1,lsat3(i))
      chanl4(k) = and(mask1,lsat4(i))
   14 continue

c......pack the data into the RIPS format

do 16 i=1,limit4,4
   chanl2(i) = lshift(chanl2(i),8)
   chanl3(i) = lshift(chanl3(i),16)
   chanl4(i) = lshift(chanl4(i),24)

   chanl1(i+1) = rshift(chanl1(i+1),8)
   chanl3(i+1) = lshift(chanl3(i+1),8)
   chanl4(i+1) = lshift(chanl4(i+1),16)

   chanl1(i+2) = rshift(chanl1(i+2),16)
   chanl2(i+2) = rshift(chanl2(i+2),8)
   chanl4(i+2) = lshift(chanl4(i+2),8)

   chanl1(i+3) = rshift(chanl1(i+3),24)
   chanl2(i+3) = rshift(chanl2(i+3),16)
   chanl3(i+3) = rshift(chanl3(i+3),8)

16 continue

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

write output file

write(unit=1,rec=1) (chanl1(i),i=1,limit4)
endfile(unit=1)
close(unit=1,status='keep')

write(unit=2,rec=1) (lsat5(i),i=1,limit4)
endfile(unit=2)
close(unit=2,status='keep')

stop
end
A program function:

program inpe.tm.f reads a 256 by 240 pixels area from a LANDSAT-TM CCT format INPE-BRAZIL (one channel each time) and outputs two binary files 'channel.img' and 'channel.dat' according to RIPS (.IMG) and (.DAT) formats respectively.

**** input ***********************************************

file 'lsat.tm' which contains one LANDSAT-TM channel formatted as in the INPE-BRAZIL CCTs.

nline
ncolumn line and column number corresponding to the upper left corner of the area selected to be displayed. Note that the input file contains one quadrant of a LANDSAT-TM scene i.e.: 3088 lines with 3600 bytes each

**** output ***********************************************

binary files 'channel.img' and 'channel.dat' of size 256 by 240 pixels to be displayed at the RIPS

parameter(nsizel=240*256, nsize2=4*240*256)
parameter(mask1=255, mask2=65280, mask3=16711680)
parameter(mask4=4278190080)

integer lsat(nsizel), chnl(nsize2), nline, ncolumn, nrec

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.....read line number (nline) and column number (ncolumn) corresponding to the upper left corner of the area to be displayed. NOTE that ncolumn must be a multiple of 4.

c
read(3,100) nline, ncolumn
100 format(2i10)

c.....adjust ncolumn to the record length (recl=4) to access the proper record in the image file

c
ncolumn=ncolumn/4

c.....add 8 records (32 bytes) which correspond to the prefix data

c
ncolumn=ncolumn+8

c.....adjust nline for the first line (which is the file descriptor record)

c
nline=nline + 1
calculate the first record to be read in image file
note that each line is 3600 bytes long, since we have specified recl=4
then each line has 900 records

\[
nrec = 900 \times (nline-1) + ncolumn
\]

\[
k=0
\]
\[
nrec=nrec-1
\]

\[
do 12 \ i=1,240
\]
\[
do 10 \ j=1,64
\]
\[
k = k +1
\]
\[
nrec = nrec + 1
\]
\[
read(unit=0,rec=nrec) \ lsat(k)
\]
\[
continue
\]
\[
nrec = nrec + 900 - 64
\]
\[
12 continue
\]

write output file 'channel.img' on RIPS (.IMG) format

\[
write(unit=1,rec=1) (\ lsat(i),i=1,k)
\]
\[
endfile(unit=1)
\]
\[
close(unit=1,status='keep')
\]

write output file 'channel.dat' on RIPS (.DAT) format

\[
j = 0
\]
\[
do 14 \ i=1,4\times k,4
\]
\[
j = j + 1
\]
\[
chnl(i) = and(mask1,lsat(j))
\]
\[
chnl(i+1) = and(mask2,lsat(j))
\]
\[
chnl(i+2) = and(mask3,lsat(j))
\]
\[
chnl(i+3) = and(mask4,lsat(j))
\]
\[
chnl(i+1) = rshift(chnl(i+1),8)
\]
\[
chnl(i+2) = rshift(chnl(i+2),16)
\]
\[
chnl(i+3) = rshift(chnl(i+3),24)
\]
\[
14 continue
\]
\[
write(unit=2,rec=1) (chnl(i),i=1,4\times k)
\]
\[
endfile(unit=2)
\]
\[
close(unit=2,status='keep')
\]

stop
end
program display.rips.f is a complement to program 'inpe.tm.f'
it takes as input four of the output files (LANDSAT-TM channels) from
'inpe.tm.f' which are on (.DAT) format and pack them into a single file
in the RIPS (.DAT) format for a multi-channel display

**** input ********************************************

channel1
channel2
channel3
channel4

are the four selected channels from a single image to be packed
into a single file on RIPS (.DAT) format

**** output ********************************************

display
file containing the four selected channels on RIPS (.DAT) format

parameter(limit1=4*256*240, limit2=256*240)

integer lsat(limit1), channel(limit2)

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')

do 10 i=1,limit1,4
nrec=nrec+1
read(unit=1,rec=nrec) lsat(i)
read(unit=2,rec=nrec) lsat(i+1)
lsat(i+1)= lshift(lsat(i+1),8)
read(unit=3,rec=nrec) lsat(i+2)
lsat(i+2)= lshift(lsat(i+2),16)
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
12 continue

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
program function:
the aim is to have a general overview of one quadrant of LANDSAT-TM
imagery (which is 3244 by 3088 pixels) on the screen of the RIPS which
is 256 by 240 pixels in size. This is accomplished by displaying every
12th pixel of the original image.

**** input *************************************************

file which contains one LANDSAT-TM channel formatted as in the
INPE-BRAZIL CCTs

**** output *************************************************

binary file 'overview' which is 256 by 240 pixels in size and on
RIPS (.IMG) format

parameter(maskl=255, limitl=61440)
integer lsat(limitl)

open(unit=0,file='tm.chnl3',access='direct',form='unformatted',
rec=4,status='old')
onopen(unit=1,file='overview',access='direct',form='unformatted',
rec=1,status='new')

n = 0
do 12 i=2,2881,12
   j is the number of records at the beginning of this image line
   note that there are 32 bytes or 8 records at the beginning of
each line which correspond to the prefix data and should be skipped
   note that the reason to take record length equal to 4 bytes for the
   input file (binary file) is that lsat(i) has been defined as an
   integer and so it will take one word (4 bytes)
   we will take 2880 lines which, sampled every 12th line will generate
   240 lines, the size of the RIPS screen.
   we also take 3072 pixels/line which sampled every 12th pixel will
   generate 256 pixels/line, the size of RIPS screen. 3072 pixels
   correspond to 900 records
   j = 900 * 1 + 8
   do 10 k=1,768,3
      nrec= j + k
      n = n + 1
      read(unit=0,rec=nrec) lsat(n)
      take only one pixel
      lsat(n) = and(maskl,lsat(n))
   10 continue
12 continue
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