FRIS

FOREST RESOURCE INFORMATION SYSTEM

NASA ST. REGIS LARS

LARSFRIS PREPROCESSING USER'S GUIDE

Purdue University
Laboratory for Applications
of Remote Sensing

LARS Contract
Report No. 100680
October 1, 1980

Star Information Form

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle LARSFRIS Preprocessing User's Guide		5. Report Date 6 October, 1980
		6. Performing Organization Code
7. Author(s) LARS Staff; Catherine Kozlowski Editor		8. Performing Organization Report No. 100680
9. Performing Organization Name and Address Laboratory for Applications of Remote Sensing Purdue University 1220 Potter Drive West Lafayette, IN 47906		10. Work Unit No.
		11. Contract or Grant No. NAS 9-15325
12. Sponsoring Agency Name and Address R. E. Joosten/SF5 NASA/Johnson Space Center Houston, Texas 77058		13. Type of Report and Period Covered
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This document contains user instructions for the proper use and application of the Software which comprises the LARSFRIS Preprocessing package. LARSFRIS represents a compilation of software developed over a number of years by the staff at Purdue University's Laboratory for Applications of Remote Sensing (LARS). The preprocessing software package is designed to help the user prepare Landsat Multispectral scanner data for analysis.

17. Key Words (Suggested by Author(s)) Reformatting; Preprocessing, Multispectral; Geometric Correction, Documentation; Software	18. Distribution Statement

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price

LARSFRIS PREPROCESSING

USER'S GUIDE

Acknowledgements

Many people contributed to the implementation and documentation of the LARSFRIS PREPROCESSING software. A major portion of the Landsat processor was written by Ken Dickman. The geometric correction processor was originally written by Jim Kast. Other contributors were Catherine Kozlowski, Mark Penningroth, Charles Smith, and Walter Van Dyke. Organization of the User's Guide was coordinated by David Freeman and Catherine Kozlowski. The manuscript was prepared by Bonnie Phibbs, Ruth Jarret, Dee Dee Dexter, Pat Whatley, and Darlys McDonald. Sue Ferringer provided graphic inputs and editorial assistance.

Preparation of this document was supported by NASA Contract NAS 9-15325.

TABLE OF CONTENTS

SECTION 1

Introduction

The LARSFRIS PREPROCESSING software package is designed to provide basic image preprocessing capabilities in the context of solving forest information questions through the use of remote sensing digital data analysis. The steps in this process are similar to the scientific method and may be adapted to the remote sensing environment.

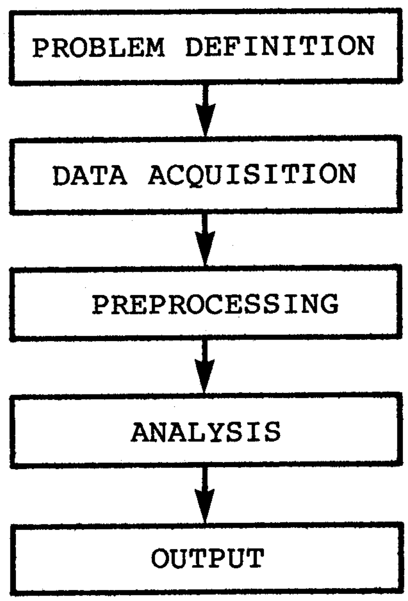


Figure 1. Overview: Steps of remote sensing analysis

Several assumptions have been made for this documentation. The definition of the problem has been completed. The digital remote sensing data is assumed to be primarily satellite data from Landsat or subsequent scanners such as the Thematic Mapper or French SPOT. It is further assumed that this data will have been processed through the Master Data Processor (MDP) at Goddard Space Flight Center for correction to Universal Transverse Mercator (UTM) grid and placed on computer compatible tapes (CCT).

Such data is available from the Eros Data Center (EDC) at Sioux Falls, South Dakota. Order information is available from EDC or in the Landsat Data Users Handbook, revised edition, USGS, 1979. Search listings, computer compatible digital tapes, and film products representing the digital data are available from EDC. Other supporting remote sensing data sources include aircraft data images also available from EDC. Non-image products such as USGS maps, soil surveys, and some forest surveys may be obtained through the U.S.

Government Printing Office or Information Centers.

After acquiring data the next step is data preprocessing. Digital image data preprocessing is required in order to format the data, geometricly or radiometricly enhance data, and to overlay or register one Landsat scene to another. Data to be analyzed by the LARSFRIS ANALYSIS software must be in LARSYS format.

The first processor in the LARSFRIS PREPROCESSING software reformats Landsat or other digital image data in EDIPS P format to LARSYS format. Refer to Holkenbrink, Patrick R. "Manual on Characteristics on Landsat Computer Compatible Tapes Produced by the EROS Data Center Digital Image Processing System," USGS, 1978, for a description of the P format. Although this process outputs corrected P Landsat data in UTM grid, it does not rotate the data to true North. Therefore a geometric enhancement is needed to rotate the data.

The second processor provided in LARSFRIS PREPROCESSING software does this task. The angle of rotation is supplied to the processor automatically through internal calculations. This is a systematic geometric enhancement not directly tied to ground control information. Optionally a LARSYS Landsat non-corrected data scene generated before the Goddard MDP became operational, or X format data, may also be systematicly geometricly corrected. Corrections for data aspect, skew, rotation, and output aspect/scale are included. Radiometric enhancement and precision registration are not directly provided for in the LARSFRIS PREPROCESSING software.

Following data preprocessing, the data is ready for analysis by the LARSFRIS ANALYSIS software. This step employs the assembling of all data sources, and the skilled use of LARSYS software. The ability to extract accurate information has been enhanced through preprocessing. Processors which perform classification of data are described in LARSFRIS ANALYSIS software documentation.

The last step in the process of remote sensing analysis is the output of results. Tables, graphs, and images from before, during, and after analysis are an excellent aid to the understanding of these results. Output products can be generated by such devices as digital image displays, film writers, and pen plotters.

Operation of LARSFRIS PREPROCESSORS

The major LARSFRIS PREPROCESSORS represent two image processing functions. These functions, Landsat-to-LARSYS reformatting, and geometric enhancement, were designed to operate in a batch environment on an IBM 3031 computer with a VM/CMS operating system. Each processor operates independently.

INPUT to each preprocessor is in two forms: tape and card image. Input tapes containing image data are generally in LARSYS format with the exception of "raw" data in EDIPS format. Two EDIPS format tapes are input for the Landsat reformatting program. Control of each processor is accomplished by control language specified in appropriate processor abstracts and inputted via card images.

PROCESSING provided by each function is specified by the control cards. Because of large array requirments, a virtual memory capability is necessary. The IBM 3031 operating under VM/CMS has this capability.

Each of the preprocessing functions has its own primary software processor. Primary processor for reformatting is named *LANDSAT. The primary processor for geometric enhancement is *GEOMETRIC CORRECTION.

OUTPUT from each process comes in three forms: printer, punch, and tape.

The printer output contains housekeeping information and documentation of the processing task that was accomplished.

Punch output is a single card containing the run number, tape and file numbers of the image data that has been created.

Tape outputs contain reformatted or enhanced data. All data is in LARSYS image data tape format per the LARSYS User's Manual. Generally computer compatible tapes are produced on 9-track, 1600 BPI format.

Landsat Reformatting

Landsat reformatting is the process of producing from a Landsat data tape one frame, or portion of one frame, of data acceptable as input to the LARSYS analysis software. The output is produced on a digital magnetic tape in the LARSYS format.

Background

In July 1972, the National Aeronautics and Space Administration launched the first Earth Resources Technology Satellite, originally called ERTS-1, to collect information about the surface of the earth from hundreds of miles above it. In February 1975, a second satellite in the series was launched to continue this data collection task. At about the same time, the satellites were renamed Landsat I and Landsat II. Landsat III was launched in March 1978, while Landsat I's data operations were terminated as of the winter 1978. Although there were plans to receive thermal data from Landsat III, no thermal channel data is available. The MSS (multispectral scanner) data is divided into frames dimensioned 185 by 185 km (115 by 115 miles) which corresponds to the frame size of the RBV (return beam vidicon) data. It takes just 25 seconds to collect the data in one frame. After some processing by NASA, the data is made available to users through the data distribution centers.

Landsat data is being applied to many resource management problems, including agricultural crop surveys, land use mapping, geological studies of structures and rock types, mineral exploration, determining soil type and moisture content, assessing water resources including surface water distribution, surveying coastal and marine resources, collecting water pollution data, evaluating forest and range resources, and detecting air pollution.

Procedure

Digital Landsat data as released from the EROS Data Center is on CCT's (computer compatible tapes) in one of three formats, X, A or P.

X format - Old format; sample interleaved in four strips.

A format - EDIPS (EROS Data Center Digital Image Processing System) format with partially processed data (without geometric corrections applied); can be in either BIL (band interleaved) or BSQ (band sequential) form.

P format - EDIPS format with fully processed data (with geometric corrections applied and resampled to a map projection); can be in either BIL (band interleaved) or BSQ (band sequential) form.

For additional details on CCT-X format, refer to "Physical Characteristics of the Landsat 1 and 2 MSS Computer Compatible Tapes", Valerie L. Thomas, Goddard Space Flight Center, November 1975 (GSFC Document X-563-75-223). For additional details on CCT-A and CCT-P formats, refer to "Manual on Characteristics of Landsat Computer Compatible Tapes Produced by the EROS Data Center Digital Image Processing System", Patrick F. Holkenbrink, U.S. Geological Survey, 1978.

The Landsat processor included in the LARSFRIS PREPROCESSOR software reformats CCT's in P format, BIL form to LARSYS format. The P format data can be either MSS data (also referred to as CCT-PM) or RBV data (also referred to as CCT-PR). The CCT's are normally ordered from EROS Data Center. The LNDRDR abstract in Section 2 contains a detailed description of the control cards that can be used in the Landsat processor. An example output is shown in Section 3.

Geometric Correction

A geometric correction can be performed on any portion of a Landsat scene in LARSYS format. The output produced is a LARSYS formatted data tape which is geometricly corrected for image distortions due to the earth's rotation and the sampling aspect. The image is also rotated to North orientation and rescaled for line printer or image display aspect with a scale selected by the requestor.

Background

The geometric correction process applies corrections that can be reasonably well specified independently of the particular data set. Scaling and skew correction, for example, can be performed open loop, i.e., without feedback from ground control points, to approximately correct the data. This approach makes improved data available to users rapidly and at relatively low cost. The initial correction is useful to those wishing to visually relate points on maps and Landsat data, and especially to those studying the millions of rectangular North South-oriented agricultural fields which exist in certain areas.

Landsat Data

Before it is sent out by EROS Data Center, Landsat data in CCT-X format has been calibrated and line length adjustments made. Landsat data in CCT-PM and CCT-PR formats have also been geometricly corrected for distortions due to band-to-band offsets, earth rotation, and detector-to-detector sampling delay. It is highly desirable for many research applications if all Landsat data is geometrically corrected, North-oriented, and scaled to the researcher's needs.

Procedure

The geometric correction process consists of up to five linear transformations which act on an entire image block or specified portion of a LARSYS run. This is contrasted to nonlinear transformations which could compensate for randomly varying scale, skew, and other distortions. A Landsat image can be thought of as a three-dimensional array containing lines of data points, columns across the image, and spectral bands or channels. Since the four channels are assumed to be in registration in this discussion, the problem can be studied as a two-dimensional single channel image problem. Transformation of this array into another array which, when displayed on a certain output device, has given characteristics, is the geometric correction problem.

Five corrections are made on the data, as applicable. The first correction adjusts for the Landsat sampling ration of approximately 3 horizontal samples to every 2 vertical samples. The second correction adjusts for skew in the Landsat frame due to the earth's rotation during data collection. The third correction modifies the data from 191 orientation to 180 North South orientation. Output scaling and aspect corrections may be performed as an option on P format data. The fourth correction transforms the data to the output scale requested. The final correction accounts for the output device aspect ratio. If the output will be represented pictorially with line printer output, the aspect should be 10:8 to compensate for the printer limitation of 10 columns per inch across the page by 8 lines per inch down the page. To achieve the desired rescaling and geometric correction, sample interpolation is done using the nearest neighbor rule. All corrections are applied to data originating from CCT-X formatted Landsat, but only corrections 3, 4, and 5 are applied to data from CCT-PM and CCT-PR formats. For a detailed explanation of all corrections, see LARS Information Note 103073, "Geometric Correction of ERTS-1 Digital Multispectral Scanner Data" by Paul E. Anuta.

Accuracy

The word "approximate" was used through the discussion and it should be emphasized that most of the parameters used are not known accurately; thus, these corrections are not exact. The sensor and satellite induced errors vary randomly over the frame; thus the rigid body assumption implicit in the use of the linear transformation is also invalid. The accuracy of the correction is, therefore, unknown; however, measurements made using topographic maps indicate about a 1 to 2 scale error. This means that if a point in the data is exactly lined up with a known ground point, that, in say 1000 meters, the image would be 10 to 20 meters in error from the true ground point.

The GCTROL abstract in Section 2 contains a detailed description of the control cards that can be used in the Geometric Correction procesor. An example output is shown in Section 3.

SECTION 2

LARSFRIS PREPROCESSING

Abstracts

Section 2 contains the abstracts for the Landsat and Geometric Correction processors. The abstracts are in alphabetical order by routine name. For the Landsat processor, the LNDSUP abstract is a guide to the other abstracts and should be consulted first. For the Geometric Correction processor, the GEMCOR abstract is the main abstract.

MODULE IDENTIFICATION

Module Name: BINSRH Function Name: General Utility

Purpose: Perform a binary search on an ordered array

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 2/27/70

Latest Revisor: Date:

MODULE ABSTRACT

To perform a linear search on an unordered list takes at most the number of comparisons equal to the number of items in the list (LISTSZ). The unordered linear search can be improved by putting the most frequently needed items at the beginning of the list. Another way to improve the linear search is to order the list (but one must also consider the cost of sorting the list into the proper order), reducing the maximum number of comparisons by about half. A binary search on an ordered list reduces the number of comparisons further to LOG(2) (LISTSZ). Many other searching algorithms are available and careful consideration should be used when choosing an algorithm.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Perform a binary search on the one dimensional array LIST for the value corresponding to KEY. If KEY exists within LIST, FOUND is set to TRUE and MIDDLE is set to the ordinal index of the element in LIST that matches KEY, otherwise FOUND is set to FALSE and MIDDLE is undefined. The statement to invoke BINSRH is:

CALL BINSRH(LIST, LISTSZ, KEY, MIDDLE, FOUND)

Input Arguments:

LIST - I*4. The array of items in lexigraphical order.

LISTSZ - I*4. The number of elements in LIST.

KEY - I*4. The item to attempt to match in LIST.

Output Arguments:

MIDDLE - I*4. An integer between 1 and LISTSZ that points to the elements in LIST that matches KEY (if no match is found then MIDDLE is undefined).

FOUND - L*1. .TRUE. if a match of KEY and LIST was found, otherwise .FALSE.

Lexicographical order is machine dependent and can be determined from the character set on each machine. The character set on the IBM computer at LARS is EBCDIC.

Character set are usually represented in hexadecimal but BINSRH compares characters in the integer frame of reference. For example, when comparing the character ´ ´, ´=´, ´A´, and ´9´, the corresponding integer values are 1,007,952,576, 2,118,139,968 -1,052,753,856, and -113,229,760. So the characters in increasing integer order are ´A´, ´9´, ´ ´, and ´=´, (the above characters are assumed to be four-byte, left-justified, blank filled words). The following subroutine determines whether an array is in increasing integer order so BINSRH will work properly.

```
      SUBROUTINE SORTED(LIST<, LISTSZ)
      IMPLICIT INTEGER *4 (A-Z)
      INTEGER*4 LIST(LISTSZ), LISTSZ
      INTEGER*4, IJ
      I=1
      DO 200 J=2, LISTSZ
          IF (LIST(I) .LT. LIST(J)) GO TO 200
              WRITE (6,9100) LIST(I), LIST(J)
9100          FORMAT(´ERROR, THE WORD ´´´, A4,
      $              ´´´IS NOT LESS THAN THE WORD´´´, A4, ´´´´)
200   CONTINUE
      RETURN
      END
```

2. Internal Description

BINSRH compares items one word (four bytes) in length. It would be easy to expand BINSRH to handle items of multiple words in length.

A binary search works by repeatedly looking at the middle of the list, where the middle of the list is determined by the extreme points LOWER and UPPER. If a comparison determines that KEY is less than the middle of the list, only the items below MIDDLE need to be searched. If a comparison determines that KEY is greater than the middle of the list, only the items above MIDDLE need to be searched. If LOWER becomes greater than UPPER (i.e., they cross paths), either KEY does not exist within LIST or LIST is not properly ordered.

3. Input Description

Not Applicable.

4. Output Description

Not Applicable.

5. Supplemental Information

This algorithm was taken from Donald E. Knuth's "The Art of Computer Programming" Series, volume 3 entitled "Searching and Sorting", chapter 6, section 2. The binary search technique can also be found in any introductory Computer Science book.

6. Program Algorithm

```
SUBROUTINE binsrh(list, listsz, key, middle, found)
     (* search list for key assuming list is in lexicographical order *)

DECLARATIONS
     INTEGER*4 key, list(listsz), listsz, middle
     LOGICAL*1 found

BEGIN
     found  := FALSE;
     upper  := 1;
     lower  := listsz;
     WHILE lower ≤ upper AND NOT found DO
         middle  := (lower + upper) / 2;
         If key < list(middle) THEN
             upper  := middle -1   (* too big, go down *)
         ELSE
             IF key > list(middle) THEN
                 lower  := middle +1   (* too small, go up *)
             ELSE
                 found  := TRUE    (* just right! *)
             ENDIF
         ENDIF
     ENDWHILE
END
```

MODULE IDENTIFICATION

Module Name: CASCII Function Name: General Utility

Purpose: Convert ASCII characters to EBCDIC characters

System/Language: Data Reformatting/IBM CMS 370 FORTRAN-IV

Author: Ken Dickman Date: 2/19/80

Latest Revisor: Date:

MODULE ABSTRACT

Convert American Standard Code for Information Interchange (ASCII) characters to EBCDIC characters, where one character occupies one byte. EBCDIC is the character set that is presently in use at LARS with the IBM CMS 370 operating system.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL CASCII(INBUF, INSZ, FIRST, NOBYTS, OUTBUF, OUTSZ)

Input Arguments:

INBUF - L*1. The input buffer containing a string (four characters per word) of one or more characters in the ASCII characters set. The integer representation of the ASCII characters set ranges from 0 to 127. Any values outside that range are undefined and are con verted to an EBCDIC question mark ('?').

INSZ - I*4. The length of the INBUF array in bytes. The value of INSZ must be greater than zero.

FIRST - I*4. The beginning byte in INBUF to convert from ASCII to EBCDIC. First may take on values from 1 to INSZ.

NOBYTS - I*4. The number of bytes (beginning at FIRST) to convert. The following restriction is placed on NOBYTS: (FIRST + NOBYTS - 1) .LE. INSZ

OUTSZ - I*4. The length of the OUTBUF array in bytes. OUTSZ should be greater than zero and no greater than the size of OUTBUF.

Output Arguments:

NOBYTS - If NOBYTS is less than 1 or NOBYTS is greater than OUTSZ, NOBYTS will be set to 1 or OUTSZ, respectively.

OUTBUF - The output buffer containing the converted EBCDIC characters. If OUTSZ is greater than NOBYTS, the remaining right-most bytes of OUTBUF are filled with EBCDIC blanks (',')

2. Internal Description

The conversion is performed using an array with the integer representation of the ASCII character as the index into the array. The contents of the array is the corresponding EBCDIC character.

3. Input Description

Not applicable.

4. Output Description

If the value of NOBYTS is out-of-range (1 to OUTSZ), an error message is written to the line printer and the user's terminal.

5. Supplemental Information

Any table that contains an ASCII or EBCDIC character set.

6. Program Algorithm

```
SUBROUTINE cascii (inbuf, insz, first, nobyts, outbuf, outsz)
     (* convert ASCII characters to EBCDIC characters *)

DECLARATIONS
     INTEGER*4 first, insz, nobyts, outsz
     LOGICAL*1 inbuf(insz). outbuf(outsz)

BEGIN
     (* check output array length *)

     IF nobyts < 1 THEN
         ERROR - must have one or more bytes to convert to EBCDIC;
         nobyts := 1
     ENDIF;
     IF nobyts > outsz THEN
         ERROR - bytes to convert exceeds output buffersize;
         nobyts ;= outsz
     ENDIF;

     (* convert ASCII to ebcdic *)

     FOR i := first TO first + nobyts - 1 DO
         IF inbuf (i) < 0 OR inbuf > 127 THEN
             (* error-character out of range for ASCII character set *)
             outbuf (i) := '?'
         ELSE
             outbuf(i) := ascii-to-ebcdic-array (inbuf(i))
         ENDIF
     ENDFOR;

     (* check for output buffer full *)

     IF nobyts < outsz THEN
         FOR i := nobyts +1 TO outsz DO
             outbuf(i) := ' '              (* fill blanks *)
         ENDFOR
     ENDIF
END
```

MODULE IDENTIFICATION

Module Name: CHAR Function Name: General Utility

Purpose: Assign a literal character string to a variable

System/Language: Data Reformatting/IBM CMS 370 FORTRAN-IV

Author: Ken Dickman Date: 2/21/80

Latest Revisor: Date:

MODULE ABSTRACT

Four characters, enclosed in single quotes, are assigned to an integer variable. Something like the following statement may be desirable:

```
IWORD = 'AB3+'
```

IBM FORTRAN flags the above statement as a syntax error, but does not consider the following statement in error:

```
IWORD = CHAR('AB3+')
```

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

To use CHAR, the calling routine must contain either ´IMPLICIT INTEGER *4 (A-Z), or ´INTEGER *4 CHAR´. The function CHAR is invoked with:

```
I = CHAR (STRING)
```

STRING - I*4. A four character literal string. If fewer than four characters are desired, blanks should follow the string to total four, otherwise the IBM FORTRAN compiler may insert unwanted "garbage" characters (eg. if the string ´16´ is desired CHAR(´16 ´) should be used). If more than four characters are present, it will be truncated to four.

Output Arguments:

CHAR - I*4. The integer variable on the left-hand side of the assignment statement will be set to the EBCDIC value of the first four characters of STRING.

2. Internal Description

Perhaps the best way to explain the simple operation is to show the actual code.

```
      INTEGER FUNCTION CHAR (STRING)
      INTEGER *4 STRING
      CHAR = STRING
      RETURN
      END
```

If the following code was executed and the first four characters on the logical file unit 5 was ´ABCD´, the IF-statement would branch to statement label 200.

```
      IMPLICIT INTEGER *4 (A-Z)
      DATA I/´ABCD´/
      READ (5,9100) J
9100  FORMAT(A4)
      K= CHAR (´ABCD´)
      IF (I .EQ. J .AND. J .EQ. K) GOTO 200
                  .
                  .
                  .
```

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

None

6. Program Algorithm

Not Applicable.

MODULE IDENTIFICATION

Module Name: CONDMP Function Name: General Utility

Purpose: Print information about the contents of an array

System/Language: Data Reformatting/IBM CMS 370 FORTRAN-IV

Author: Ken Dickman Date: 2/23/80

Latest Revisor: Date:

MODULE ABSTRACT

Print a special character or a blank, the bytes where the data can be found, a description of the meaning of the data, and the value of the data. CONDMP provides a simple and concise way to present the contents of an array.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL CONDMP(HILITE, BYTES, MEAN, MEANSZ, VALU, VALUSZ, TYPE)

Input Arguments:

- HILITE - I*4. Any single, printable character for highlighting or a blank for no highlighting.
- BYTES - I*4. Any 9 characters describing the location of the data.
- MEAN - L*1. Any 1 to 50 characters describing the meaning of the data.
- MEANSZ - I*4. The size of MEAN. MEANSZ must be greater than zero. If MEANSZ is greater than 50, MEAN is truncated to 50.
- VALU - L*1. The contents of the data. VALU may be one word (INTEGER*4) of integer or 11 words (44 bytes of characters.
- VALUSZ - I*4. The size of VALU in bytes.
- TYPE - I*4. If TYPE contains the character string 'INTE', then VALU is printed as an integer, otherwise VALU is printed as characters.

2. Internal Description

The information message in MEAN is copied to DESCRP according to MEANSZ. From MEANSZ + 1 to 50, DESCRP is filled with dots. For character data, VALU is copied to IVALU from four bytes per word (A4 to A1 format). For integer data, the magnitude of MEAN is determined so VALU (now copied from logical to integer in I4) can be left-justified.

3. Input Description

Not Applicable.

4. Output Description

One line of line printer output is written for each call of CONDMP.

5. Supplemental Information

See the LNDDIR, LNDHED, and LNDANN module abstracts for routines using CONDMP.

6. Program Algorithm

```
SUBROUTINE condmp (hilite, meansz, valusz, type)
     (* dump contents of records and give corresponding info *)

DECLARATIONS

     INTEGER*4 bytes(3), hilite, meansz, valusz, type
     LOGICAL*1 mean(meansz), valu(valusz)

BEGIN
     IF type = 'INTEGER' THEN
         (* print valu in integer format *)

         PRINT hilite, bytes, mean, valu
     ELSE
         (* print valu in character format *)

         PRINT hilite, bytes, mean, valu
     ENDIF
END
```

MODULE IDENTIFICATION

Module Name: CTLCBC Function Name: General Utility

Purpose: Insert commas into a control card for compatibility with CTLPRM

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 3/07/80

Latest Revisor: Date:

MODULE ABSTRACT

Entry point CTLPRM of module CTLWRD expects a comma between every control parameter. This restriction is too limiting when a control card designer wishes to make the control card syntax more meaningful. CTLCBC removes this restriction by inserting commas, where necessary, after the control card is read and before CTLPRM is invoked.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

After calling the main entry point CTLWRD, use CTLCBC as follows:

```
CALL CTLCBC(CARD, COL)
```

Input Arguments:

CARD - I*4. A twenty element array of characters in A4 format as read by CTLWRD.

COL - I*4. Index into CARD as set by CTLWRD.

Output Arguments:

CARD - I*4. Will have blank characters converted to commas where needed for compatibility with CTLPRM.

2. Internal Description

Blanks will not be converted to commas within matched pairs of parentheses. To convert a blank to a comma, the character before the series of blanks must be a letter, number, right parenthesis, or a slash, and the character after the series of blanks must be a letter, number or a right parenthesis.

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

CTLCBC is used in the UNIFORM and LANDSAT data reformatting programs.

6. Program Algorithm

```
SUBROUTINE ctlcbc(card, col);
     (* convert blanks to commas for subroutine ctlprm *)

DECLARATIONS
     legal  = ´A´ through ´Z´, ´0´ through ´9´, ´)´;
     plegal = legal, ´/´;

BEGIN
     IF col < 72 THEN

         (* convert 4 bytes per word to 1 byte per word *)
         CALL MOVBYT(card→xcard);

         (* scan from right to left until non-blank character is found *)
         first  := col +1;
         last   := 72;
         WHILE last > first AND xcard = ´b´ DO
             last  := last -1;

         (* insert commas between words when necessary*)

         WHILE first < last DO
             (* search left to right for the first blank *)
             WHILE first < last AND xcard(first) = ´b´ DO
                 IF xcard(first = ´(´ THEN
                     (* find a corresponding ´)´ *)
                     CALL ctlspn (xcard, first, last) (* skip characters *)
                 ELSE
                     first  := first +1
                 ENDIF
             ENDWHILE;

             (* search left for the first non-blank after a blank *)
             end-of-data  :+ first +1
             WHILE first < last AND xcard(end-of-data) = ´b´ DO
                 end-of-data  := end-of-data +1
             ENDWHILE;

             (* determine whether a comma should be inserted *)

             IF xcard(end-if-data) is within legal THEN
                 begin-of-data  := first -1;
                 IF xcard(begin-of-data)is within plegal THEN
                     (* convert a blank to a comma *)
                     xcard(first)  := ´,´
                 ENDIF
             ENDIF;

             first  := end-of-data
```

```
        ENDWHILE;

        (* convert 1 byte per word back to 4 bytes per word *)
        CALL MOVBYT (xcard→card)
    END
END
```

MODULE IDENTIFICATION

Module Name: CTLSPN Function Name: General Utility

Purpose: Span over pairs of matched parentheses on a control card

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 3/07/80

Latest Revisor: Date:

MODULE ABSTRACT

Columns on a control card are skipped until the number of left parentheses equals the number of right parentheses.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usge

CALL CTLSPN(XCARD, FIRST, LAST)

Input Arguments:

XCARD - I*4. An array of 80 words, in A1 format, containing one control card.

FIRST - I*4;. An Index pointing to a left parenthesis.

LAST - I*4. An Index pointing to the end of the control card.

Output Arguments:

FIRST - I*4. Will be set to point after the beginning left parenthesis. If there is not a right parenthesis corresponding to each left parenthesis, FIRST will be set to LAST.

2. Internal Description

Not Applicable

3. Input Description

Not applicable

4. Output Description

Not Applicable

5. Supplemental Information

Subroutine CTLCBC call CTLSPN in the UNIFORM and LANDSAT data reformatting programs.

6. Program Algorithm

```
SUBROUTINE ctlspn(xcard,first, last);
     (* Skip over balanced parentheses *)

BEGIN
     paren   := 1; (* pairs of parentheses to span *)
     first   := first +1;
     WHILE first < last OR
         CASE xcard (first) OF
             ´(´: paren  := paren +1
             ´)´: paren  := paren -1;
             otherwise: (* do nothing *)
         ENDCASE;
         first  := first +1
     ENDWHILE
END
```

MODULE IDENTIFICATION

Module Name: CTOIA1 Function Name: General Utility

Purpose: Convert a character in A1 format to an integer

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/22/80

Latest Revisor: Date:

MODULE ABSTRACT

A left-justified, blank filled, four byte word containing one character between ´0´ and ´9´ is converted to an integer between 0 and 9.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

I = CTOIA1(C,ERROR)

Input Arguments:

C - I*4. The character to be converted. The range of C must be between hexadecimal F0404040 and hexadecimal F9404040, When FO = ´0´, F9 = ´9´, and 40 = ´ ´.

ERROR - L*4. If C is between ´0´ and ´9´, ERROR will be set to .FALSE., otherwise ERROR will be set to .TRUE..

Output Arguments:

CTOIA1 - I*4. The integer representtion of C.

The following code will assign the number 5 to the variables I, J, and K assuming the first character on logical unit 10 is 5:

```
        LOGICAL*4 ERROR
        INTEGER*4 A, B, C, CTOIA1, I, J, K
        DATA A/´5
        READ(10,9100) C
9100    FORMAT(A,1)
        I = CTOIA1(A, ERROR)
        J = CTOIA1(B, ERROR)
        K = CTOIA1(C, ERROR)
```

2. Internal Description

Not Applicable.

3. Input Description

Not Applicable.

4. Output Description

If C is not in the range from ´0´ to ´9´, an error message is written to the line printer.

5. Supplemental Information

None

6. Program Algorithm

```
INTEGER FUNCTION ctoial (c,error);
    (* convert character to integer, where the character is in Al format *)

DECLARATION
     character-set = ('0', '1', '2', '3', '4', '5', '6', '7', '8', '9' ),

BEGIN
          error    := TRUE;
          ctoill   := 0;
          found    := FALSE;
          i        := 1;
          WHILE i< = 10 AND NOT found DO
               IF c = character-set(i) THEN
                   found   := TRUE:
                   error   := FALSE;
                   ctoial  := i - 1
            ENDIF
         ENDWHILE;

         IF NOT found THEN
             PRINT ('digit expected in Al format')
         ENDIF
END
```

MODULE IDENTIFICATION

Module Name: DISMAT Function Name: GEMCOR

Purpose: Derive Distortion Matrix for Geometric Correction

System/Language: IBM VM 370/Fortran IV

Author: Mark Penningroth Date: 9/11/80

Latest Revisor: Date:

MODULE ABSTRACT

Subroutine DISMAT derives the distortion msytic for the Geometric Correction Routine. This is derived by setting up correction matrices for: 1) Scale correction matrix; 2) Rotation-to-north correction matrix; 3) Skew correction matrix; 4) Output device correction matrix; and 5) Rescaling correction matrix. These correction matrices are then multiplied to get the distortion matrix. It is important that matrices are multipled the order given above. The correction matrices which are to be used, are set up by the control cards read in by GCTROL

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

DISMAT

Called by the program GEMCOR after the control cards have been read. Dismat is set up to correctly handle LANDSAT data originally from 2 different sources: NASA X-format CCT´s (sometimes called sample interleaved) and EDIPS BIL CCT-PM (band interleaved with geometric corrections applied and resampled to a map projection).

2. Internal Description

```
SUBROUTINE dismat
     (* set up distortion matrix *)

Begin
     If Instrument type is in X-format then
         set up correction matrix to change
         pixel dimensions from 57 x 57 to 79 x 79

     ELSE
         set up correction matrix to change
         pixel dimensions from 57 x 79 to 79 x 79
     ENDIF

     IF data is to be rotated and deskewed THEN
         IF angle = 0.0 THEN
             get angle (in radians) from GEMANG
         ELSE

         set up rotation and
         deskew correction matrices;
     ELSE
         IF data is to be rotated only THEN
             IF angle = 0.0 THEN
                 get angle (in radinas) from GEMANG
             ELSE be rotated and deskewed THEN
                 convert angle to radians
             ENDIF

             set up rotation correction matrix
         ENDIF

         IF data is to be deskewed only THEN
             get angle (in radians) from GEMANG;
             set up skew correction matrix
         ENDIF
     ENDIF

     IF output device is the printer THEN
         set up correction matrix for printer output
     ENDIF
```

```
     multiply all correction matrices to
     get distortion matrix;
     output all intermediate matrices with
     headings;
     output final distortion matrix;
END
```

3. Input Description

All needed information for DISMAT is in the GEMCOM common block.

4. Output Descriptions

The distortion matrix will be in MAT (2,2)

5. Supplemental Information

Anuta, Paul E. "Geometric correction of ERTS-1 Digital Multispectral Scanner Data" LARS Information Note 103073 LARS, Purdue University, 1973

Holkenbrink, Patrick F. "Manual on Characteristics of LANDSAT Computer Compatible Tapes Produced by the EROS Data Center Digital Image Processing System" USGS, 1978

Thomas, Valerie L. "Generation and Physical Characteristics of the LANDSAT 1 and 2 MSS Computer Compatible Tapes" (X-563-75-223) Goddard Space Flight Center November 1975

ERTS Data Users Handbook

Document number 715D4249,
Goddard Space Flight Center

MODULE IDENTIFICATION

Module Name: ERRPTR Function Name: General Utility

Purpose: Print a dollar sign beneath an error in the current control card.

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 2/21/80

Latest Revisor: Date:

MODULE ABSTRACT

When an error is detected on a control card by CTLWRD, BCDVAL, or any other routine, ERRPTR can be used to mark the column in the card where the error occurred. Without ERRPTR or something similar, the user may not have any idea where the trouble lies on the card.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

```
CALL ERRPTR(COLO)
```

Input Argument:

COL - I*4 the current column-pointer position

Output Argument:

None

2. Internal Description

ERRPTR initializes a 73 element array to blank and then set one element to ´$´ with the statement

```
LINE(COL) = ´$´
```

The line is then printed.

3. Input Description

Not Applicable.

4. Output Description

One line is written to the line printer and the user´s terminal with a dollar sign (´$´) in the COL column. Five spaces are skipped before the line written (FORMAT(5X)) to be aligned with lines written by CTLWRD.

5. Supplemental Information

Routines LNDRDR, LNDPRM, and LNDVAL use ERRPTR. For more information see the LNDRDR, LNDPRM, LNDVAL, CTLWRD, and BCDVAL module abstracts and program code.

6. Program Algorithm

Not Applicable

MODULE IDENTIFICATION

Module Name: FILSRH Function Name: General utility

Purpose: Search a tape for an empty file (two consecutive end-offiles)

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 2/25/80

Latest Revisor: Date:

MODULE ABSTRACT

There may be times when it is desired to position a tape at the first empty file. At this position, a new file may be appended to the previous files.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Before invoking FILSRH, a tape must be mounted on the tape drive specified by UNIT. FILSRH is then invoked by:

CALL FILSRH(UNIT, ERROR)

Input Agruments:

UNIT - I*4. The logical unit number (11 through 15) of the tape drive to search on.

Output Arguments:

ERROR - L*4. .TRUE. if an error was detected by the TOPRD entry point of the TAPPOP module, otherwise .FALSE.

FILSRH may be used to archive data. After positioning the tape with FILSRH, a file of data may be written followed by two end-of-files.

2. Internal Description

The two consecutive end-of-files are found by reading the tape with TOPRD. The maximum record size that FILSRN may read is set to 5400 bytes (this may be changed by editing and recompiling FILSRH). For proper operation, TOPRD must not detect an error before reporting two consecutive end-of-files. After the two consecutive end-of-files are found, entry point TOPBF (also of module TAPOP) is called to position the tape between the end-of -files. Regardless of the success of FILSRH, a message is printed telling the tape file positioned to.

3. Input Description

Reading (scanning) of the tape is done by TOPRD. Currently FILSRH is set to read records of up to 5400 bytes long.

4. Output Description

Commands are written to the tape drive to control tape positioning. An error message is written to the line printer and user's terminal if TOPRD detects an error and just before FILSRH returns control to the calling routine.

5. Supplemental Information

For more information about TOPRD and TOPBF, see the TAPOP module abstract and for more information about how to use FILSRH see the LNDDIR module abstract.

6. Program Algorithm

```
SUBROUTINE filsrh (unit,error)
     (* scan a tape for two consecutive end-of-files *)

DECLARATIONS
     INTEGER *4 file, unit
     LOGICAL *4 error, first-eof-found, second-eof-found, info-on-tape

BEGIN
     file   := 0;
     error  := FALSE;
     first-eof-found  := FALSE;
     second-eof-found := FALSE;
     info-on-tape     := FALSE;
     WHILE (NOT error) AND NOT (first-eof-found AND second-eof-found) DO
         READ-TAPE (unit);
         IF tape-eof THEN
             IF first-eof-found THEN
                 second-eof-found :=TRUE
             ENDIF;
             first-eof-found := TRUE;
             file  :=file +1
         ELSE
             if tape-error THEN
                 error  := TRUE;
                 PRINT ('tape read error')
             ELSE (* data record read *)
                 first-eof-found  := FALSE;
                 second-eof-found := FALSE;
                 info-on-tape     :=TRUE
             ENDIF
         ENDIF
     ENDWHILE;

(* back-up to the empty file *)
IF NOT ERROR THEN
     BACK-FILE(unit);
IF info-on-tape THEN

     (* there is no data before encountering two consecutive eof's*)
     (* so need to position to the beginning of the tape *)

         BACK-FILE(unit);
         file  :=file -1
     ENDIF
ENDIF;

PRINT ('tape on unit', unit, 'has been positioned to file', file)

END
```

MODULE IDENTIFICATION

Module Name: GCTROL Function Name GEMCOR

Purpose: Read the control cards for the geometric correction routine (GEMCOR) and mount the input tape.

System/Language: IBM VM 370/FORTRAN-IV

Author: Mark Penningroth Date: 09/11/80

Latest Revisor: Date:

MODULE ABSTRACT

GCTROL first sets up initial defaults then it reads all control cards for the geometric correction routine (GEMCOR). After all control cards are read, processing of the control cards begins. The input tape is mounted and control is returned to GEMCOR.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Abstract:

A. Input:

Input is expected from the card reader (device 5). Note: The filedef in Gemcor´s Exec allows input to be on disk.

2. Internal Description:

A. Set up initial Defaults.

B. Read in all control cards and print them out. Also check for errors and print appropriate error message.

C. When "End" control card is read, all processing begins.

D. Mount the input tape and return.

3. Subroutines Called:

BCDVAL
CTLWRD
GADRUN
MOVBYT

4. Input Description:

R E Q	KEY WORD (COL 1)	CONTROL PARAMETER	FUNCTION	DEFAULT
	-RUNTABLE	DATA		
		RUN (xxxxxxxx)	Specify input run number	
		TAPE (xxxx)	Tape that contains run	Look up input run in the runtable
		FILE (xx)	File that contains run	
		END		
*	*GEOMETRICORRECTION		Specify geometric correction	None

R E Q	KEY WORD (COL 1)	CONTROL PARAMETER	FUNCTION	DEFAULT
*	GCORRECTION	RUN (xxxxxxxx)	Specify input run number	None
	NEWRUN	NEWRUN (xxxxxxxx)	New output run	None
		NEWTAPE (xxxx)	New output tape	Determined by tape handler
		NEWFILE (xx)	New output file	Determined by tape handler
	* OUTPUT	RUN (xxxxxxxx)	Output run	None
		TAPE (xxxx)	Output tape	Determined by tape handler
		FILE (xx)	Output file	Determined by tape handler
	AREA	COLUMNS (xx,xx)	Specify input columns	All columns
		LINES (xx,xx)	Specify input lines	All lines
	ASPECT	SCALE (xxxxx.xx)	Specify scale	24000.0
		PRINTER	Output device will be the printer	PRINTER
		DISPLAY	Output device will be the display	PRINTER
	FLIGHTLINE	AAAAAAAAAA	Specify flightline	get flight line from ID record
	NLA1	AAAAAA	The number of input lines to add to work buffer at one time	15

R E Q	KEY WORD (COL 1)	CONTROL PARAMETER	FUNCTION	DEFAULT
	SENSOR	LANDSAT (x)	Specify landsat Satelite number	Derived from ID record
		EDIPS	Data in EDIPS format	EDIPS
		X	Data in X-format	EDIPS
	OPTION	CALIBRATION	Add calibration bytes	CALIBRATION
		DESKEW	Deskew data	NODESKEW
		LATITUDE (xxx.xx)	Specify latitude	Derived from ID record
		NOCALIBRATION	Do not add cal ibration bytes	CALIBRATION
		NODESKEW	Do not deskew data	NODESKEW
		NOROTATE	Do not rotate data	ROTATE
		ROTATE	Rotate data using an angle based on latitude.	ROTATE
		ROTATE (xxx.xx)	Rotate data using angle given here. (Angle in degrees).	ROTATE
	-COMMENT		Comment cards	None
*	END		Begin processing	None

5. Output description:

Control cards are printed as read. Also all options are echoed after processing has been completed.

Error Messages:

ERROR	ACTION
EOF reached before "End" card read -	Job Halted
First card no - runtable or *GEO card -	Job Halted
Output run is not 8 digits long -	Job Halted
Error on card -	Card echoed read new card
More than 1 -RUNTABLE card -	Job Halted
*GEO card expected -	Job Halted
RUNNUM and INRUN are different -	Job Halted
No GCORRECTION Card	Job Halted

6. Subroutines and Entry Points Called:

```
BCDVAL
  IVAL
  FVAL
CTLWRD
  CTLPRM
GADRUN
  GETRUN
MOVBYT
```

7. Sample Control Card Deck

```
-Runtable
Data
Run  (79002400), Tape  (44), File  (1)
End
-Comment sample G.C. control card deck
*Geometriccorection
Gcorrection      Run (79002400)
Output           Run (79002401)
Area             Lines (200,500), Columns (1000,1500)
Aspect           Printer, Scale  (20000,0)
End
```

8. Minimal Control Card Deck:

```
*GEO
Gcor      Run  (79002500)
Output    Run  (79002501)
End
```

9. Supplemental Information:

Anuta, Paul E. "Geometric Correction of ERTS-1 Digital Multispectral Scanner Data" LARS Information Note 103073 LARS, Purdue University, 1973.

Holkenbrink, Patrick R. "Manual on Characteristics of LANDSAT Computer Compatible Tapes Produced by the EROS Data Center Digital Image Processing System" USGS, 1978.

Thomas, Valerie L. "Generation and Physical Characteristics of the LANDSAT 1 and 2 MSS Computer Compatible Tapes" Goddard Space Flight Center, November, 1975.

ERTS Data Users Handbook, Document number 715D4249, Goddard Space Flight Center.

Landsat Data Users Handbook, Revised edition, USGS, 1979.

MODULE IDENTIFICATION

Module Name: GEMANG Function Name: GEMCOR

Purpose: Calculate angle of correction

System/Language: IBM VM370 FORTRAN-IV

Author: B. Catherine Kozlowski Date: 06/10/81

Latest Revisor: Date:

MODULE ABSTRACT

GEMANG calculates an angle based on the latitude of the scene. This angle is then used in the subroutine DISMAT to calculate correction matrices for distortions due to rotation and/or skew.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

GEMANG

Called by the subroutine DISMAT. GEMANG returns the angle to be used in the calculation of the correction matrices for distortion due to rotation and/or skew.

Input Parameters:

None

Output Parameters:

THETA	REAL*4
ERROR	LOGICAL*1

2. Internal Description

```
SUBROUTINE gemang
     (* calculate angle for correction calculations *)
BEGIN
     set up constant for conversion from degrees to radians;
     convert angle of inclination to radians;
     IF latitude in ID record is to be used THEN
         convert latitude to radians;
         IF data is originally from EDIPS format
             CCT's THEN
             calculate angle of correction based on
             latitude of frame center
         ELSE
             offset the latitude of frame center
             to the latitude of area to be
             corrected
         ENDIF
     ELSE
         convert user - supplied latitude to radians
     ENDIF
     calculate angle of correction based on
     angle of inclination and adjusted latitude
END
```

3. Input Description

GEMANG uses several variables from the GEMCOM common block as input. It uses the Landsat satellite number to determine the angle of inclination (this angle varies slightly for each satellite). From the common block GEMANG knows whether the latitude is user-supplied or should be taken from the input ID record. Also from the common block GEMANG can determine the original CCT format of the input data set. If the input was originally from an X-format CCT, more adjustments must be made than if the input was originally from an EDIPS format CCT.

There are also several constants in the subroutine. The angle of inclination (on each satellite was calculated by substracting the inclination angle given on page 5-4 of the Landsat Data Users Handbook from 180 degrees.

180 - (inclination in handbook)

The number of lines and columns in a full frame of Landsat data in Xformat was derived from past experience with these tapes.

4. Output Description

GEMANG returns two values to the calling routines, the angle of correction in radians and an error flag. At the present time, the error flag is always returned indicating no errors because no error checks are made in GEMANG.

GEMANG also sets the latitude variable in the common block GEMCOM, if it had not been previously set.

5. Subroutines Called

Double precision trigonometric functions:

- DARSIN
- DATAN
- DCOS
- DSIN

6. Supplemental Information

Anuta, Paul E. "Geometric Correction of ERTS-1 Digital Multispectral Scanner Data" LARS Information note 103073, LARS, Purdue University, 1973.

Landsat Data Users Handbook, Revised Edition, USGS, 1979.

MODULE IDENTIFICATION

Module Name: GEMCOR Function Name: GEMCOR

Purpose: Geometric correction of a specified area of an input run

System/Language: CMS/FORTRAN

Author: Mark Penningroth Date: 02/19/81

Latest Revisor: B. Catherine Kozlowski Date: 03/05/81

MODULE ABSTRACT

GEMCOR reads in data from an input run and then geometrically corrects a section of the input run data (specified by user), and then outputs a new run, which is the corrected image. GEMCOR also prints out Form-17's for the new run.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Input Arguments:

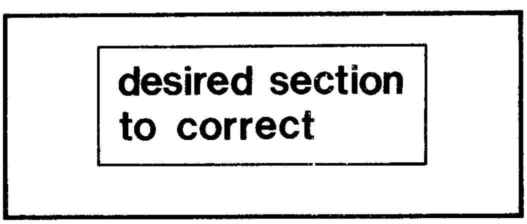
Input is a tape containing data in LARSYS-III format. The tape number and all other pertinent information for GEMCOR is read in by the routine GCTROL, which is the control card reader for GEMCOR. See GCTROL's Abstract for more information.

Output Arguments:

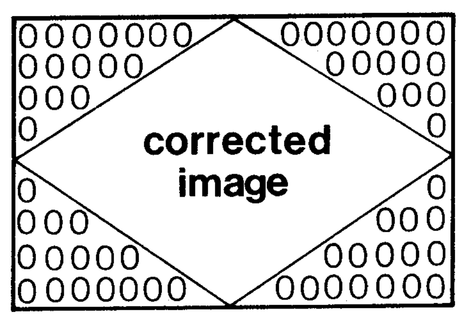
Output is the corrected run (over area specified by control cards), which is put on a new tape which is also specified in GCTROL. Form 17's for the correction are also printed.

2. Description of Problem

We have an input run that contains a section of data that we would like to geometrically correct and then output.



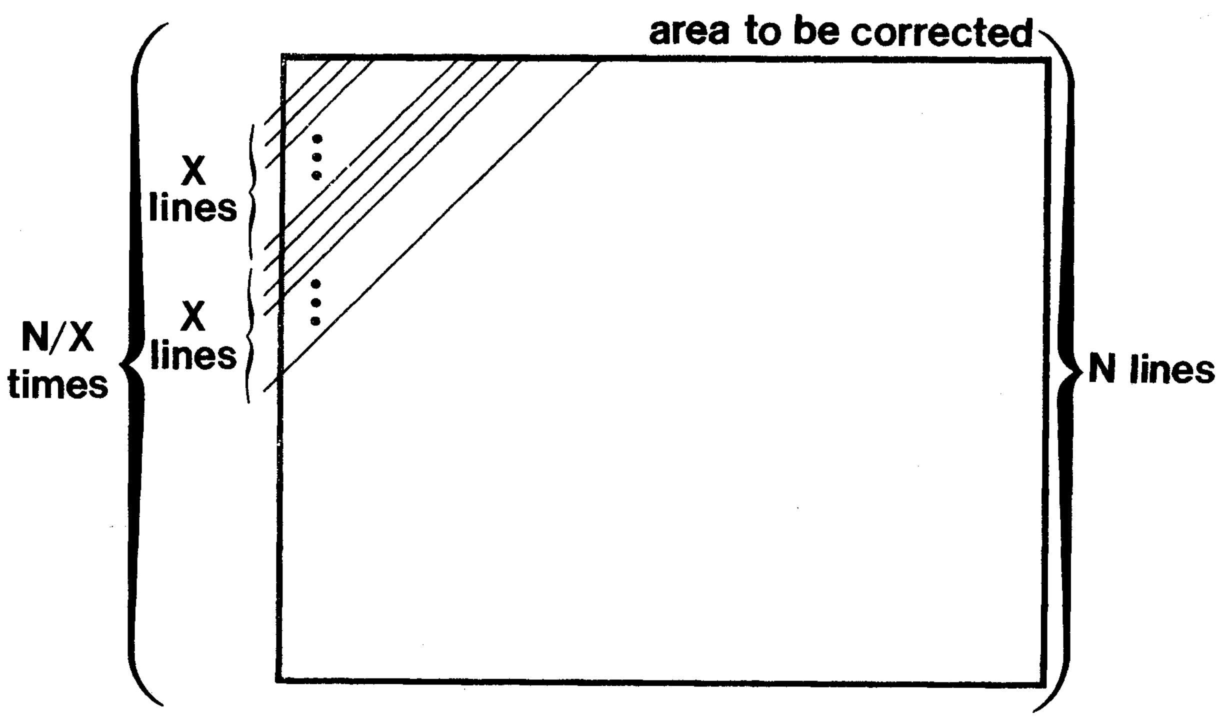
We want output to look like this



The "O"'s in the above diagram will contain spectral values or zero fill depending on the location of the corrected section in relation to the entire input data set. This will be elaborated on later in the abstract. There are many possible choices for correction depending on skewness, rotation, latitude, etc. See GCTROL's abstract for more information on variables used.

The correction is done by filling up a work buffer with X lines of the input run where the lines are sloped according to the corrections to be performed. If you have N lines in the input buffer, then this must be done in a loop which is executed N/X times. Within this loop, corrections are made and the corrected portion is written to the output tape.

The diagram below illustrates how the work buffer is filled.



Note: X is usually 15.

3. Internal Description

GEMCOR may be broken up into 5 main parts, each with a certain number of steps.

PART	STEP	DESCRIPTION
1	1	Initializations.
1	2	Call GCTROL (control card reader).
1	3	Do filing (ID information).
1	4	Call DISMAT (get distortion matrix).
1	5	Calculate distortion matrix's determinate and inverse.
2	1	Input calculations (keypoints).
2	2	Calculate number of lines and samples for correction.
2	3	Write the new ID record.
2	4	Set up line length.
2	5	Convert keypoints to buffered X and Y values.
3	1	Position input tape.
3	2	Read uncorrected image into input buffer.
4	1	Initialize X points and Y points
4	2	If needed, get 15 new lines in work buffer.
4	3	Calculate the loop indices for the 3. Transfer loops and then transfer data.
4	4	Fill remainder of line and calibration bytes with zero.
4	4a	Zero front of line.
4	4b	Zero end of line.
4	5	Transfer line and roll information.
4	6	Write an output line.
4	7	Check for end of tape.
4	8	GOTO part 4 step 2.
5	1	Print form 17's and calculate time.
5	2	Stop.

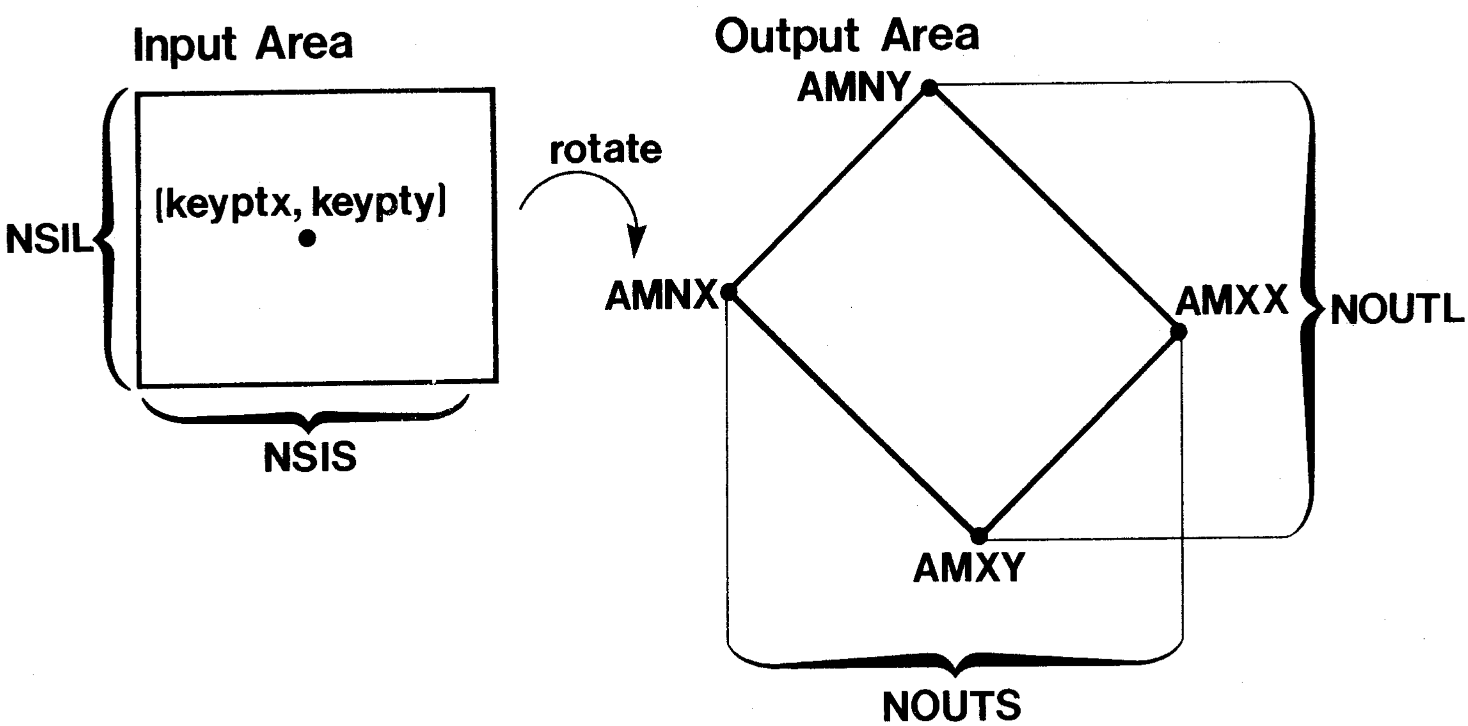
PART 1

In part 1, timing and flag variables are initialized, the control cards are read (GCTROL), and the distortion matrix is obtained from DISMAT. The distortion matrix's inverse and determinate are calculated. Information is stored in the NEWID record (NEWTAP,NEWFIL, NEWRUN, DATE) and NC, NS, NL, OFIL, OTAP (number of input channels, samples, and lines, input file number, input tape number) are obtained from the input ID record.

PART 2

STEP 1

The midpoint of the input frame is found and put into the variables (KEYPTX, KEYPTY). Then NSIL (number of lines in input subarea) and NSIS (number of samples in input subarea) are calculated. Then, using the inverse of the distortion matrix, the four corners for the output area are calculated along with NOUTS (number of output samples) and NOUTL (number of output lines).

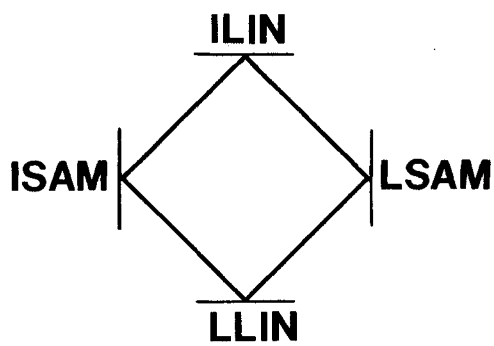


The center of the output frame is then calculated using the distortion matrix. This is put in the variables (XIP, YIP).

STEP 2

Calculates the initial and final numbers for lines and samples in the input space for the correction.

ILINC = ILIN = initial line number in input space
LLINC = LLIN = last line number in input space
ISAMC = ISAM = initial sample number in input space
LSAMC = LSAM = last sample number in input space



Also information needed to print LARS form-17's stored at this time.

STEP 3

Write the new ID record

STEP 4

Set up line length in bytes.

Calculate NLBYT (number of bytes in line) and NAX (total number of bytes needed).

STEP 5

Convert XIP, YIP to buffered X and Y values

XIP and YIP are buffered by the following equations:

XIP = XIP - ISAM + 1
YIP = XIP - ILIN + 1

This point (XIP, YIP) is now the starting location for the work data buffer.

PART 3

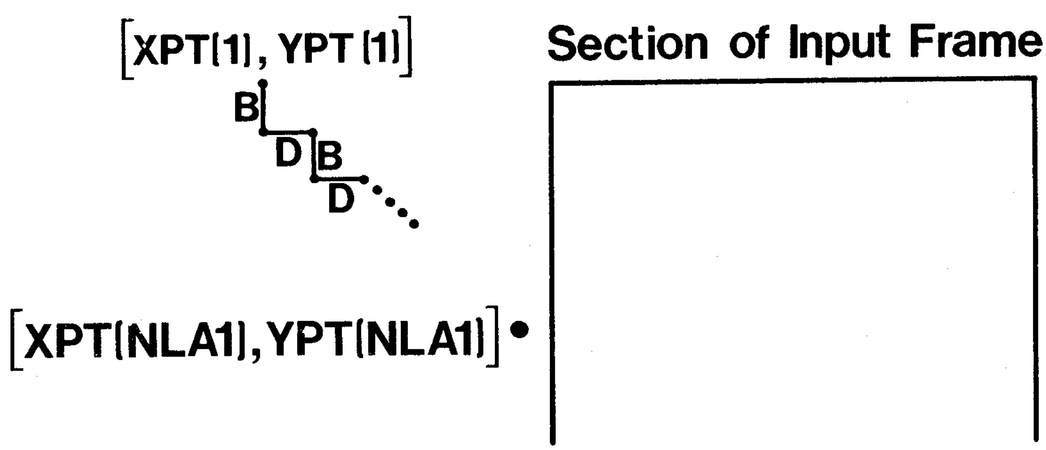
In Part 3 the input tape is positioned and then the input data is read into the input buffer. After this, the work buffer is filled with data. Checks are performed to make sure data will fit in the buffer.

PART 4

STEP 1

Initialize XPT(1...NLA1) and YPT(1...NLA1)

This step initializes XPT (1...NLA1) and YPT (1...NLA1) by using B and D from the distortion matrix. These initial points are not necessarily in the input frame.



STEP 2

Possibly fill up work buffer

If needed, the work buffer is filled with NLA1 lines from the input buffer.

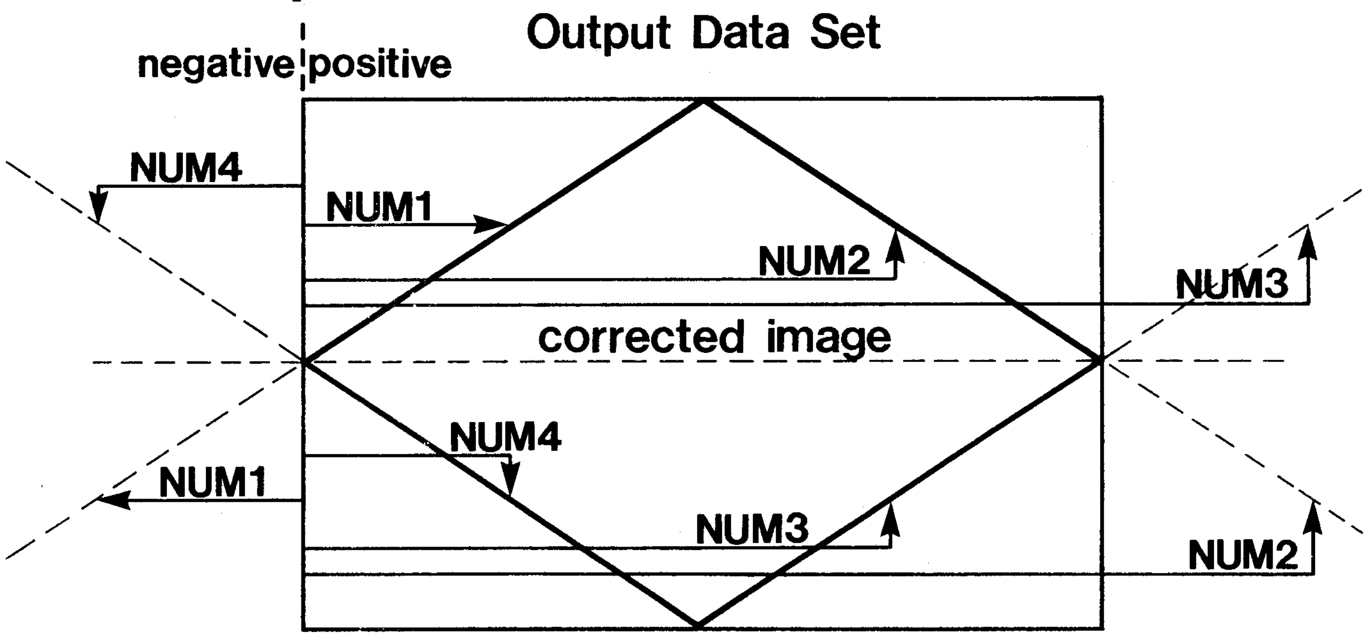
STEP 3

Calculation of the loop indices for the 3 transfer loops and then transfer of data.

In this step, the number of bytes to be filled with zeroes at the beginning and end of the output line are calculated. Also the limits of the first and last byte of output data are calculated.

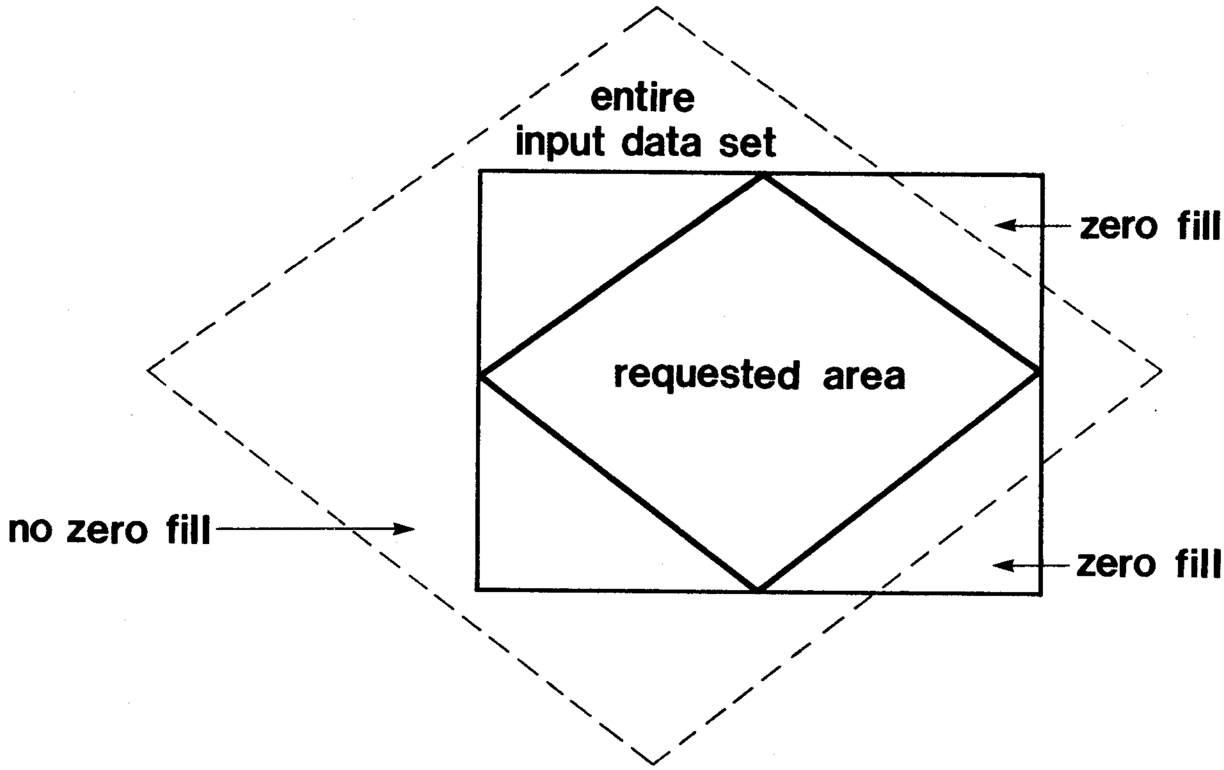
MAXO (NUM1(I), NUM4(I) = NOB(I)) = number of bytes to zero out in the front of output line I

MINU (NUM2(I), NUM3(I)) = IDEX (I) = number of bytes to zero out at the end of output line I.



If I is above midline then NOB(I) = NUM1 (num4 is negative) and IDEX(I) = NUM2. If I is below the midline then NOB(I) = NUM4 (NUM1 is negative) and IDEX(I) = NUM3.

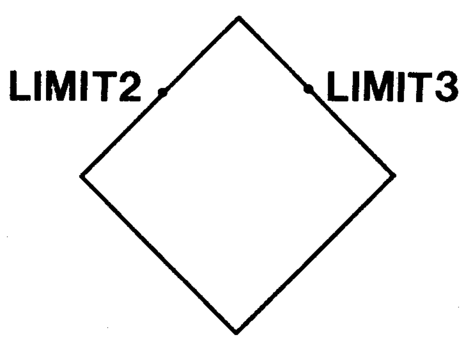
Depending on the location of the area to be corrected relative to the entire input data set, NUM1, NUM4 and NUM2, NUM3 may be out of range of the output data set limits. In the example below NUM1, NUM4 would be out of range of the output data set meaning there would be no zero fill necessary on the left side of the output. However, NUM2, NUM3 would indicate that some zero fill is necessary on the right side of the output.



Now LIMIT2 and LIMIT3 are calculated

LIMIT2 = first byte in output line where data is to be placed.

LIMIT3 = last byte in output line where data is to be placed.



Now if necessary, the XPT´s and YPT´s are re-initialized.

Now that the limits are calculated, the program then transfers data from the input buffer to the output buffer using the following 3 loops:

```
FOR K = limit2 to limit3 DO BEGIN
             (* number of lines left to do *)
     FOR i = 1 to nlltd DO BEGIN
             (* npos *)
         calculate the output buffer pointer;
             (* ibufp *)
         calculate the input buffer pointer;
             (* number of channels *)
         FOR l = 1 to nc DO BEGIN
             transfer inbuf (ibufp) to ldi (npos + ixl);
                 (* ixl is the number of bytes in a line *)
             increment ibufp and npos
         ENDDO
         update x points and y points to next reference point on each
         line;
             (* see Appendix for more information on this *)
     ENDDO
ENDDO
```

STEPS 4a and 4b

Now fill the front and end of the output lines just transfered with zeroes if needed. Also set the calibration bytes to the proper values.

STEP 5

Transfer line and roll information.

STEP 6

Write to tape the output lines transfered.

STEP 7

Check for end-of-tape on the input or output tape. If true then mount a new tape and continue.

Go To part 4 step 2

PART 5

Print out LARS form-17's for the corrected run, timing statistics, and stop.

4. SUBROUTINES Called and Entry Points

ACCNT
CPTIME
DISMAT
EOT
GCTROL
GTDATE
IDRITE
LARS17
MOUNT
MOVBYT
PAGLOC
TAPOP
- TOPBS
- TOPEF
- TOPFF
- TOPFS
- TOPRD
- TOPWR

TSTREQ
STDHDR

5. Output

1. Page and information heading.
2. Input buffer area.
3. Run, tape, and file numbers for both input and output.
4. Number of bytes locked into core.
5. Input keypts, input size, output size.
6. New input area (corrected).
7. Frame size in samples X lines, measured as a percent of the available bytes. Also how many lines the input buffer can hold at once.
8. Number of lines used in input buffer.
9. Message when input buffer lines are added at output line X.
10. Timing statistics.

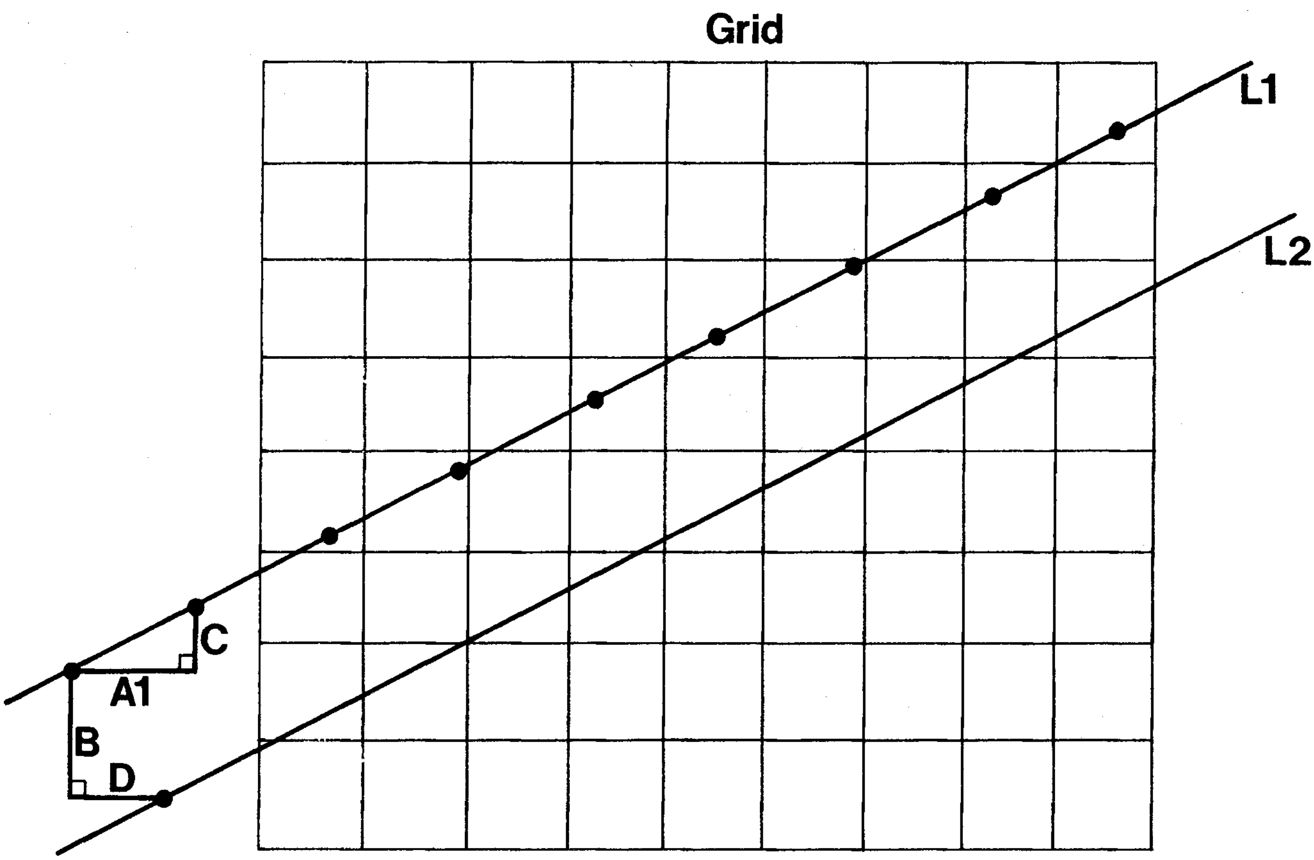
6. Errors

1. New ID write error.
2. Topfs error.
3. Buffer not large enough for this run.
4. Line read error.
5. Premature end of file on tape.
6. Inbuf - line read error.
7. Write error code X at line Y.
8. Non-standard return from GCTROL.

All errors cause GEMCOR to terminate.

Appendix

Diagram showing the function of the MAT (2,2) variables



Grid - input frame (array of pixels)

L1 and L2 - arbitrary lines of output samples

A1 - Horizontal distance to next point on same line (new X coordinate for next point on L1 or L2).

B - Vertical distance from a point on L1 such that this distance gives a new Y coordinate for the next point to process on L2.

C - Vertical distance from a point on the same line which when added to y coordinate of current point, gives the new Y coordinate for next point on the line.

D - Horizontal distance from a point on L1 to be added to the X coordinate of the current point. Gives the X coordinate of next point on L2.

Summary:

To go from point (Xo, Yo) on a line, to (X1, Y1) on same line then X1 = Xo + A1 and Y1 = Yo + C

To go from point (Xo, Yo) on a line to the next point to process on the next line (Wo, Zo) then

Wo = Xo + D   Zo = Yo + B

Note: B is negative, in the program Zo = Yo - B1, where B1 = |B|

MODULE IDENTIFICATION

Module Name: IDCAL Function Name: General Utility

Purpose: Set C0 and C1 to correct values in LARSYS ID

System/Language: Data Reformatting/IBM VM370/FORTRAN IV

Author: Walter Van Dyke Date: 8/4/81

Latest Revisor: Date:

MODULE ABSTRACT

IDCAL inserts the correct C0 and C1 values into a LARSYS ID record for a LANDSAT data set. It sets the values for all 4 bands according to the satellite number and date of data collection. If an error is detected in the satellite number or date, an error flag is set and the C0 and C1 are not inserted.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

IDCAL

Call IDCAL (SATLIT, DATE, ID, ERROR)

Input Arguments:

```
SATLIT - Interger *4 Satellite number.
      1 for LANDSAT 1
      2 for LANDSAT 2
      3 for LANDSAT 3
Date (3) - Interger *4 Date data was collected in the form
      Date (1) - month (2 digits)
           (2) - day (2 digits)
           (3) - year (2 digits)
ID (200) - Real *4 LARSYS ID record
```

Output Arguments

ID(200) Real *4 LARSYS ID record with values inserted for C0 and C1 if no errors detected. ERROR Integer *4 Error flag. Error returns are as follows:

Error	Description of Error
0	No error detected.
1	Invalid satellite number.
2	Invalid date of collection for given satellite. That is, the satellite was not collecting data on the date given.

2. Internal Description

```
SUBROUTINE idcal (*set up C0 and C1 values in ID record*)
BEGIN
     If satellite number out of range THEN
         print error message;
         error flag :=1
     ELSE
       IF date of collect out of range for given satellite THEN
         print error message;
         error flag := 2
       ELSE
         error flag :=0;
         larsys - id (53) := C0 for LANDSAT band 4;
         larsys - id (54) := C1 for LANDSAT band 4;
         larsys - id (58) := C0 for LANDSAT band 5;
         larsys - id (59) := C1 for LANDSAT band 5;
           etc.
       ENDIF
     ENDIF
END
```

3. Input Description

Not applicable.

4. Output Description

IDCAL is set up to handle only LANDSAT 1 through LANDSAT 3 with data in the order bands 4 through 7.

It will handle correctly only LANDSAT data processed through the GSFC/EROS Data Center system.

5. Supplemental Information

The C0 and C1 values are set up according to the table below. This table was prepared by Luis A. Bartolucci, LARS, in July 1981.

Table 1. Calibration Information for Different Periods of Operation of Landsat 1, 2 and 3 Systems

System and Performance Period	Wavelength Band (in view)	Minimum Radiance (in mnwatts/cm . st)	Maximum Radiance (in mwatts/cm sr)
Landsat 1	0.5 - 0.6	0	2.48
	0.6 - 0.7	0	2.00
	0.7 - 0.8	0	1.76
	0.8 - 1.1	0	4.60
Landsat 2 1/22/75 - 7/16/75	0.5 - 0.6	0.10	2.10
	0.6 - 0.7	0.07	1.56
	0.7 - 0.8	0.07	1.40
	0.8 - 1.1	0.14	4.15
Landsat 2 after 7/16/75	0.5 - 0.6	0.08	2.63
	0.6 - 0.7	0.06	1.76
	0.7 - 0.8	0.06	1.52
	0.8 - 1.1	0.11	3.91
Landsat 3 3/5/78 - 5/31/78	0.5 - 0.6	0.04	2.20
	0.6 - 0.7	0.03	1.75
	0.7 - 0.8	0.03	1.45
	0.8 - 1.1	0.03	4.41
Landsat 3 after 5/31/78	0.5 - 0.6	0.04	2.59
	0.6 - 0.7	0.03	1.79
	0.7 - 0.8	0.03	1.49
	0.8 - 1.1	0.03	3.83

MODULE IDENTIFICATION

Module Name: JTOR Function Name: General Utility

Purpose: Convert a Julian data to a Roman Calendar date

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 3/6/80

Latest Revisor: Date:

MODULE ABSTRACT

A Julian data is a five digit number where the first two digits represent the year and the last three digits represent the number of days since the beginning of the year. A Roman data may consist of three-two digit number (month, day, and year). Subroutine JTOR converts the five digit Julian date to a Roman date.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL JTOR(JULIAN, MONTH, DAY, YEAR, ERROR)

Input Arguments:

JULIAN - I*4. An integer Julian data with a range between 00001 and 99365. The last three digits must be between 1 and 365, except on leap years where the range is between 1 and 366.

Output Arguments:

MONTH - I*4. The month of the year (integer value between 1 and 12).

DAY - I*4. The day of the month (integer value between 1 and 31).

YEAR - I*4. The year (integer value between 1 and 99).

ERROR - L*4. .TRUE. if the Julian date is invalid, otherwise .FALSE.

For example, if the following code was executed,

```
         IMPLICIT INTEGER*4 (A-Z)
         LOGICAL*4 ERROR
C
         JULIAN = 66
         CALL JTOR(JULIAN, MONTH, DAY, YEAR, ERROR)
         WRITE(6,9100) MONTH, DAY, YEAR, ERROR
9100     FORMAT(1X,I2,'/',I2,'   ERROR=',L5)
         STOP
         END
```

the output would be:

```
         b3/b6/80   ERROR=FALSE
```

2. Internal Description

Not Applicable

3. Input Description

Not Applicable

4. Output Description

Not Applicable

5. Supplemental Information

Not Applicable

6. Program Algorithm

```
SUBROUTINE jtor(julian, month, day, year, error);
     (* convert julian to roman *)

BEGIN
     error  := FALSE;
     year   := julian shifted-right 3-decimal-places;
     julday := julian MOD 1000;
     IF year ≥ 0 AND year ≤ 99 THEN
         IF leap-year THEN
             February has 29 days;
             366 day-in-the-year
         ELSE
             February has 28 days;
             365 days-in-the-year
         ENDIF;
         IF julday ≥ 1 AND julday ≤ days-in-the-year THEN
             month  := 1;
             sum    := 0;
             WHILE sum + number-of-days-in-a-month julday DO
                 sum  := sum + number-of-days-in-the-month;
                 increment to next month
             ENDWHILE;
             day  := julday - sum
         ELSE
             error  := TRUE;
             PRINT(' too many julian days in the year ')
         ENDIF
     ELSE
         error  := TRUE;
         PRINT (' julian year out-of-range ')
     ENDIF
END
```

MODULE IDENTIFICATION

Module Name: LNDANC Function Name: Landsat

Purpose: Process Landsat Tape Ancillary records

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/03/80

Latest Revisor: Date:

MODULE ABSTRACT

Landsat ancillary records occur only on geometrically uncorrected CCT's. LNDANC exists as a dummy routine except for archiving.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

If archiving is desired and no archiving errors have been detected, LNDARC is called to archive the ancillary record.

2. Internal Description

Not applicable

3. Input Description

Not applicable

4. Output Description

Not applicable

5. Supplemental Information

See the "Manual on Characteristics of Landsat Computer-Compatible Tapes" produced by the EROS Data Center Digital Image Processing System, pages 24 through 38.

6. Program algorithm

Not applicable

MODULE IDENTIFICATION

Module Name: LNDANN Function Name: Landsat

Purpose: Process Landsat Annotation records

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/03/80

Latest Revisor: Date:

MODULE ABSTRACT

Values are extracted from the Landsat annotation record. These values are written to the line printer and used in creating the form 17C's

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Subroutine CASCII and MOVBYT extract the ASCII and binary data respectively. The data is then written by CONDMP to the line printer.

2. Internal Description

If archiving is desired and no archiving errors have been detected, LNDARC is invoked to archive the annotation record .

3. Input Description

The annotation record is received in the array buffer INREC.

4. Output Description

Several values from the annotation record are written to the line printer.

5. Supplemental Information

See the "Manual on Characteristics of Landsat Computer-Compatible tapes" , produced by the EROS Data Center Digital Image Processing System, pages 39 through 43.

6. Program Algorithm

```
SUBROUTINE lndann;
     (* process Landsat annotation record*)
BEGIN
     extract the data from the annotation record;
     print the data extracted;
     IF archive process desired THEN
          archive VIA LNDARC
     ENDIF
end
```

MODULE IDENTIFICATION

Module Name: LNDARC Function Name: Landsat

Purpose: Achive Landsat records

System/Language: Data Reformatting/IBM CMS 370 FORTRAN-IV

Author: Ken Dickman Date: 01/23/80

Latest Revisor: Date:

MODULE ABSTRACT

Landsat header records are saved in condensed form on a seperate output tape. The Landsat header data can then be retrieved for later use.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

If the archive process is desired, then all Landsat header records are stored as they exist on the original Landsat tape except:

1. Only the Landsat records before the first Landsat image are stored;

2. Inter-record gaps separate the records instead of end-of-file marks.

2. Internal Description

Subroutine TOPWR is called to write the Landsat data record to the archive tape.

3. Input Description

LNDARC receives one record (and its size) at a time indirectly from LNDCTL.

4. Output Description

Data is written to tape VIA the subroutine TOPWR. If TOPWR detects any errors, messages are sent to the line printer and the user's terminal. See the TAPOP module abstract for more details on the error messages.

5. Supplemental Information

Not applicable.

6. Program Algorithm

```
SUBROUTINE lndarc
     (*save Landsat data records on tape *)
BEGIN
     write to archive tape;
     IF write-error THEN
         print error message;
         turn off archive process
     ENDIF
END
```

MODULE IDENTIFICATION

Module Name: LNDBIL Function Name: LANDSAT

Purpose: Supervisor for Landsat BIL format to LARSYS format

System/Language: Data Reformatting/IBM CMS370 FORTRAN IV

Author: Ken Dickman Date: 01/04/80

Latest Revisor: Date:

MODULE ABSTRACT

Subroutine LNDBIL conducts the reformatting of EDIPS band interleaved by line (BIL) input to the LARSYS version 3.0 tape format. Multiple Landsat image data records are read and precessed by higher level subroutines LNDIMA and LNDCTL. These records are retained until written as one larger LARSYS data record.

Other processing supervised by this routine includes mounting the LARSYS output tape and writing the LARSYS header on tape, conversion of light border pixels to dark, writing the data to the output tape, and informing the user of the current job status.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LNDBIL is called from the subroutine LNDIMA after an internal decision is made by LNDIMA that the data records represent data from a band interleaved by line formatted EDIPS tape. Although LNDBIL does not directly read data from the input tape, it examines data placed in its input buffer by LNDCTL to determine the handling of further processing.

2. Internal Description

```
SUBROUTINE lndbil
     (*supervise converting Landsat records to LARSYS and outputting
       LARSYS *)
BEGIN
     IF this is the desired BIL scene to reformat THEN
         IF this is the first time lndbil has been called THEN
             initialize
         ENDIF
         IF this record is of the same image line as before THEN
             add this channel to the output buffer
         ELSE
             IF next record is the beginning of a new image line THEN
                 IF this is the first image line to reformat THEN
                     check the number of channels specified against
                     the number of channels available;
                     create the LARSYS identification header;
                     mount the LARSYS output tape;
                     write the LARSYS identification header to tape;
                     initialize pointers for inside of the output
                     buffer for LNDNEG;
                 ENDIF
                     write the output to the LARSYS tape;
                     every 100 lines, tell user how many lines reformatted
                     re-initialize the output buffer with this record
                 ELSE
                     IF bytes 7 and 8 of the image record equal zero THEN
                         skip this record - it does not contain data
                     ELSE
                         ERROR - bad scan line count
                     ENDIF
                 ENDIF
             ENDIF
         ELSE
             If this scene is after the desired scene THEN
                 assume the desired scene has been completed
             ENDIF
         ENDIF
     END
```

The image pixels (starting from byte 13 of the Landsat image record) are stored by subroutine LNDMIL in buffer OUTREC as long as SCNCNT remains the same. Once SCNCNT increases by one, then the data in OUTREC is written by subroutine LNDWRT to the output tape, and the present image pixels for the new line are stored in the beginning of OUTREC. When SCNCNT increases by one and the last value of SCNCNT was the first line to reformat, then:

1. The number of channels specified vs. the number of channels available is checked (the number of input and output channels is determined by the variables MXINCH and MXOUCH, respectively).

2. Subroutines LNDLID creates the LARSYS header;

3. Subroutine IDRITE mounts the output tape, checks the previous LARSYS header on the tape, and writes the new LARSYS header to the tape;

4. Array FIRCOL is set to point to the first byte of data per channel, in the OUTREC buffer;

5. Array LSTCOL is set to point to the last byte of data per channel, in the OUTREC buffer.

For normal operations, IDRITE is called, but for some debugging it may be easier to use XMOUNT and TOPWR. In subroutine LNDINT, the first element of array PATCH is set to 'Y' for IDRITE, and 'N' for XMOUNT and TOPWR. XMOUNT has the provisions to select the output density VIA the control card commands:

```
OUTPUT BPI(density)
       or
OUTPUT DENSITY(density)
```

These two commands are ignored when PATCH is 'Y'. The command 'OUTPUT FILE(file)' is ignored when PATCH is 'N'.

If the last line to reformat is the last line of data of the Landsat CCT sequence, then the variable FLUSH will be TRUE after all lines have been read. If FLUSH is TRUE and state 'EOF6' is reached, the OUTREC will be written VIA LNDWRT. If the last line to reformat is before the last line of data on the last Landsat tape of the CCT sequence, then the program will be done when it reads one record past the last line to reformat.

3. Input Description

This subroutine does not do any inputting itself. The subroutine LNDCTL reads the input record. Subroutine LNDIMA determines the scan line count (SCNCNT) from the image input record. LNDBIL determines the line and channel from SCNCNT and executes the proper code depending on SCNCNT. We have observed with one Landsat tape that band 8 (channel 5) contained all zeroes in the image data and in SCNCNT. Therefore, we implemented the condition that if SCNCNT was zero, then that input record would not be passed to the output. However, this could cause the LARSYS output tape to be inconsistent, in the size of the output record, if not all lines for a band were consistently zero or non-zero.

4. Output Description

The LARSYS tape format is basically an 800 byte* identification header record followed by data records. The data record starts with one half-word for the line count and a second half-word for the roll parameter. Data records produced by the Landsat to LARSYS program always have the hexadecimal value 7FFF for the roll parameter. Following the roll parameter are the first channel data samples, followed by line calibration information (hexadecimal 000080000000). Next are the second channel data samples followed by line calibration, and so on. All samples of all channels of one line are contained in one data record.

A full frame of Landsat data will not always fit on the one LARSYS 9 track 1600 bpi output tape. The frame can be reduced using the proper control cards. See the LNDRDR module abstract or the "User´s Guide for the Landsat EDIPS to LARSYS Control Card Reader" under the keyword ´INPUT´ (specifically the BAND, CHANNEL, COL, and LINE parameters).

The terms column (COL), samples, and pixel are often used interchangeably. Bytes per inch (BPI) and density are also often used interchangeably.

5. Supplemental Information

None

* One byte is equal to eight binary bits, one word is equal to four bytes.

MODULE IDENTIFICATION

Module Name: LNDCOL Function Name: Landsat

Purpose: Set defaults and check values for columns and channels

System/Language: Data Reformatting/IBM CMS370 Fotran-IV

Author: Ken Dickman Date: 01/08/80

Latest Revisor: Date:

MODULE ABSTRACT

Subroutine LNDCOL is called by LNDBIL to set default columns and channels for MSS and RBV data that is either corrected or uncorrected. several checks for columns and channels are performed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Column and channel defaults are set. Range and whether the total number of columns are evenly divisible by 4 are checked.

2. Internal Description

```
SUBROUTINE lndcol
     (* handle the number of columns and channels to reformat *)
BEGIN
     set default columns;
     set default channels;
     print the channels to reformat;
     check the range on the number of columns to reformat;
     larsys-id(6)  := the total number of columns for a LARSYS channels;
     WHILE larsys-ed(6) is not evenly divisible by 4 DO
         IF left-most pixel to reformat is NOT max THEN
             reformat one more pixel to the left
         ELSE
             IF right-most pixel to reformat is NOT max THEN
                reformat one more pixel to the right
             ELSE
                 IF column increment > 1 THEN
                    decrease column increment by 1
                 ELSE
                    set left-most, right-most and column increment
                    to pre-determined values where larsys-id(6)
                    will be evenly divisible by 4
                 ENDIF
             ENDIF
         ENDIF
         larsys-id(6) := the new total number of columns
                         for a LARSYS channel
     ENDWHILE;
     print the number of columns to reformat
END
```

Column defaults are set according to whether any were specified, whether that data is geometrically corrected or not, and whether the data is MSS or RBV. Array COLMAX gives the maximum number of columns for the particular Landsat data type. Array SODIV4 gives the maximum number of columns under the condition that when the right-most pixel is element 1 and the column increment is 1, that the total number of pixels will be evenly divisible by 4.

3. Input Description

Not applicable

4. Output Description

Information, and possibly some error, messages are sent to the line printer and user's console.

5. Supplemental Information

None

MODULE IDENTIFICATION

Module Name: LNDCOR Function Name: Landsat

Purpose: Perform line corrections to Landsat image data

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 01/19/80

Latest Revisor: Date:

MODULE ABSTRACT

If the Landsat image was not fully corrected by NASA-Goddard, LNDCOR would be invoked to perform radiometric and some limited geometric corrections. At present, LNDCOR only exists as a dummy routine because of the unavailability of uncorrected data in EDIPS format, through the EROS Data Center.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Not applicable

2. Internal Description

Not applicable

3. Input Description

The geometrically uncorrected Landsat image data is to be fed to LNDCOR in the buffer OUTREC.

4. Output Description

LNDCOR is to return geometrically corrected image data in the buffer OUTREC.

5. Supplemental Information

Not applicable

6. Program Algorithm

Not applicable.

MODULE IDENTIFICATION

Module Name: LNDCTL Function Name: Landsat

Purpose: Read and process Landsat-EDIPS formatted tapes.

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 12/27/79

Latest Revisor: Date:

MODULE ABSTRACT

LNDCTL supervises the mounting, reading, checking and processing of the Landsat-EDIPS formatted tapes. The reformatting process is directed by the contents of the Landsat input tapes.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LNDCTL is called by the main routine LNDSUP and uses the common blocks defined in subroutine LNDINT. The Landsat tape is read and checked using the PSNS table and the Landsat tape is processed by calling the appropriate subroutine.

2. Internal Description

COUNT (1) is incremented to 1 to show that state BEG has occurred. Subroutine LNDCTL is basically a CASE-statement enclosed in a WHILE-loop. The looping continues until the processing of the Landsat input is complete or a severe error occurs.

The Present-State (PS), the Next-State (NS), and the Present State-Next State table (PSNS) are initialized in subroutine LNDINT. The Present-State represents the current process. The Next-State represents the next process and is determined by reading the input tape. The intersection of the Present State and the Next State determines whether the transition from Present State to Next State is valid or not. The variable PROC is set to the value of the PSNS intersection. A detailed analysis of the PSNS table and how it is used in LNDCTL for the Landsat tapes is given in Appendix B.

The allowable ordering of Landsat records and end-of-files is shown in figure 1. The state mnemonics of figure 1 are explained in table 1. Table 1 also shows the Landsat record type codes for distinguishing between records and shows the values of EOFNUM for distinguishing between end-of-files. The transitions between states in figure 1 are identical to the transitions between states in figure 2 and figure 3. The state diagram and its relationship with the PSNS table of figure 3 is discussed in Appendix A using simple examples to demonstrate the general idea of how a PSNS table works. More details on the use of EOFNUM and figure 3 are given in Appendix B.

Figure 4 shows the actual code to initialize the PSNS table. The LNDCTL algorithm using the PSNS table is given in figure 5 and a more in-depth algorithm is given in section 6.

A frequency count is kept of the number of times a valid state is entered (COUNT) and the number of times that each specific transition has occurred (FREQ). It may be of interest to follow a trace of the states while reading a tape. This trace has been "commented-out" for general use. The IF statement cancels the trace when PS and NS are set to the Landsat image record state because this state may occur over 14,000 times per job.

PROC is checked to make sure that the transition is valid. The CASE-statement (implemented as a computed-GOTO) selects which process to perform according to PROC.

After the selected process has been executed and the computed-GOTO is exited, a check is made whether enough lines of the desired scene (BSQ or RBV subscene) have been processed.

The next input record is read. The next state is determined from the record type code, or if an end-of-file occurs, the next state is determined by the variable EOFNUM.

State Diagram for Landsat-EDIPS to LARSYS Reformatting

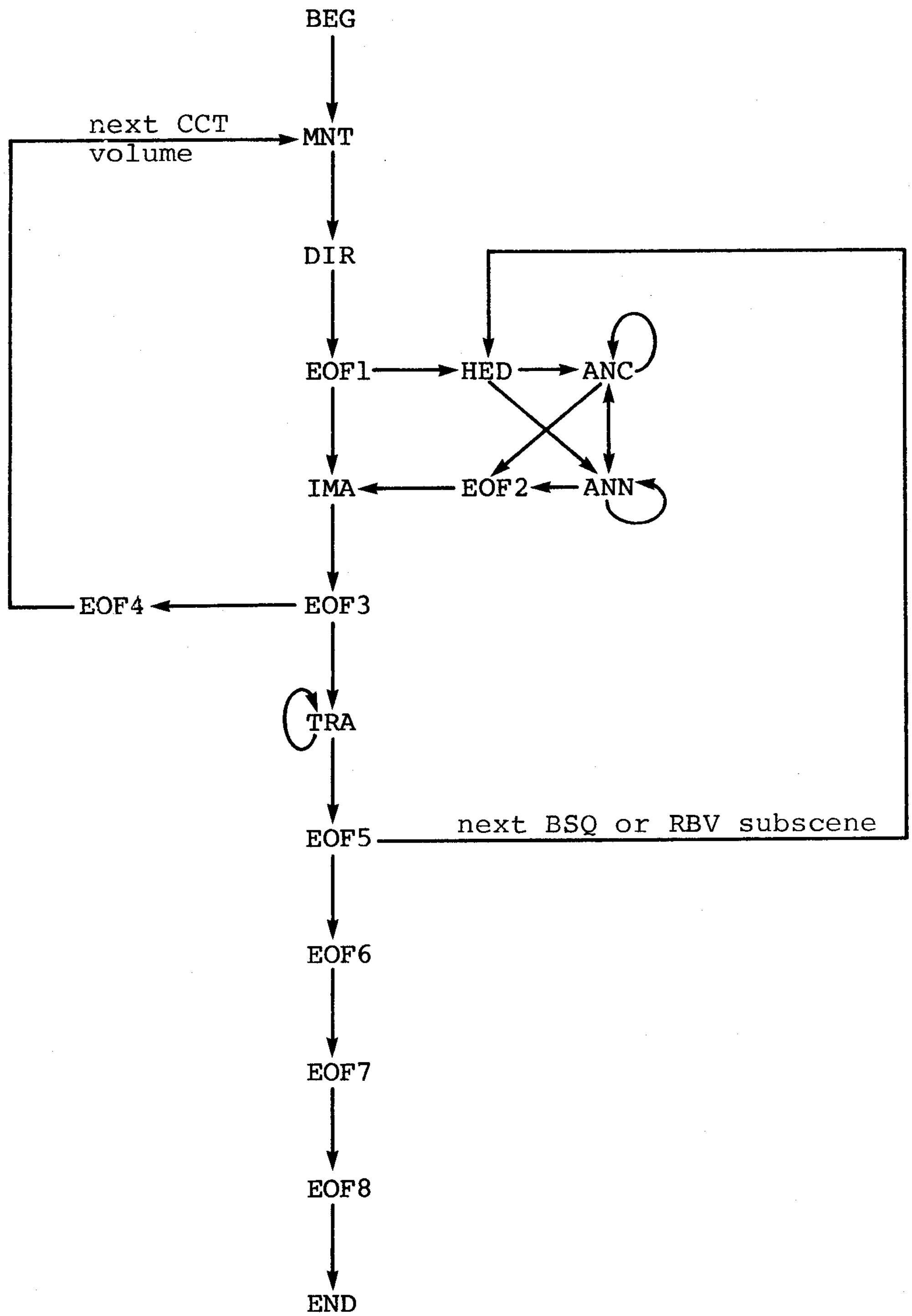


Figure 1

LANDSAT REFORMATTING STATE INFORMATION

STATE			Landsat Record Type Code (in hexadecimal)	Variable to Distinguish End-of-File States
integer	mnemonic	meaning		
1	BEG	begin processing	-	
2	MNT	mount input tape	-	EOFNUM=0
3	DIR	Landsat directory	09	
4	EOF1	first end-of-file	-	EOFNUM=EOFNUM+1
5	HED	Landsat header	12	EOFNUM=1
6	ANC	Landsat ancillary	24	
7	ANN	Landsat annotation	DB	
8	EOF2	second end-of-file	-	EOFNUM=EOFNUM+1
9	IMA	Landsat image	ED	EOFNUM=2
10	EOF3	third end-of-file	-	EOFNUM=EOFNUM+1
11	EOF4	fourth end-of-file	-	EOFNUM=EOFNUM+1
12	TRA	Landsat trailer	F6	EOFNUM=4
13	EOF5	fifth end-of-file	-	EOFNUM=EOFNUM+1
14	EOF6	sixth end-of-file	-	EOFNUM=EOFNUM+1
15	EOF7	seventh end-of-file	-	EOFNUM=EOFNUM+1
16	END	end processing	-	

Table 1.

Present State - Next State Transitions

PRESENT STATE	NEXT STATE
1. BEGIN	MOUNT
2. MOUNT	TAPE DIRECTORY
3. TAPE DIRECTORY	EOF1
4. EOF1	HEADER,IMAGE
5. HEADER	ANCILLARY, ANNOTATION, EOF2
6. ANCILLARY	ANCILLARY, ANNOTATION, EOF2
7. ANNOTATION	ANCILLARY, ANNOTATION, EOF2
8. EOF2	IMAGE
9. IMAGE	IMAGE, EOF3
10. EOF3	TRAILER, EOF4
11. EOF4	MOUNT
12. TRAILER	TRAILER, EOF5
13. EOF5	HEADER, EOF6
14. EOF6	EOF7
15. EOF7	END

end-of-volumn (EOV) = EOF3 and EOF4
end-of-set (EOS) = EOF5, EOF6, and EOF7

Except for BEGIN, END, and MOUNT, the one state represents one reading from TAPOP subroutine TOPRD.

If EOF4 is encountered, then the data is spanned over several tape volumes.

Figure 2

NEXT STATE (NS)

PRESENT STATE (PS)		BEG 1	MOUNT 2	DIR 3	EOF1 4	HED 5	ANC 6	ANN 7	EOF2 8	IMA 9	EOF3 10	EOF4 11	TRA 12	EOF5 13	EOF6 14	EOF7 15	END 16
BEG	1		2														
MOUNT	2			3													
DIR	3				4												
EOF1	4					5				9							
HED	5						6	7	8								
ANC	6						6	7	8								
ANN	7						6	7	8								
EOF2	8									9							
IMA	9									9	10						
EOF3	10											11	12				
EOF4	11		2														
TRA	12												12	13			
EOF5	13					5									14		
EOF6	14															15	
EOF7	15																16
END	16																

REPEAT DIAGONAL

Figure 3

How the Present State - Next State is actually coded in Fortran.
(Note: The array must be transposed via two DO-loops. See LNDINT.)

PRESENT STATE - NEXT STATE TABLE...

	BEG	MNT	DIR	EOF1	HED	ANC	ANN	EOF2	IMA	EOF3	EOF4	TRA	EOF5	EOF6	EOF7	END
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DATA DPSNS /																
1	0,	02,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
2	0,	0,	03,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
3	0,	0,	0,	04,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
4	0,	0,	0,	0,	05,	0,	0,	0,	09,	0,	0,	0,	0,	0,	0,	0,
5	0,	0,	0,	0,	0,	06,	07,	08,	0,	0,	0,	0,	0,	0,	0,	0,
6	0,	0,	0,	0,	0,	06,	07,	08,	0,	0,	0,	0,	0,	0,	0,	0,
7	0,	0,	0,	0,	0,	06,	07,	08,	0,	0,	0,	0,	0,	0,	0,	0,
8	0,	0,	0,	0,	0,	0,	0,	0,	09,	0,	0,	0,	0,	0,	0,	0,
9	0,	0,	0,	0,	0,	0,	0,	0,	09,	10,	0,	0,	0,	0,	0,	0,
A	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	11,	12,	0,	0,	0,	0,
B	0,	02,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,
C	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	12,	13,	0,	0,	0,
D	0,	0,	0,	0,	05,	0,	0,	0,	0,	0,	0,	0,	0,	14,	0,	0,
E	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	15,	0,
F	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	16,
G	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0 /

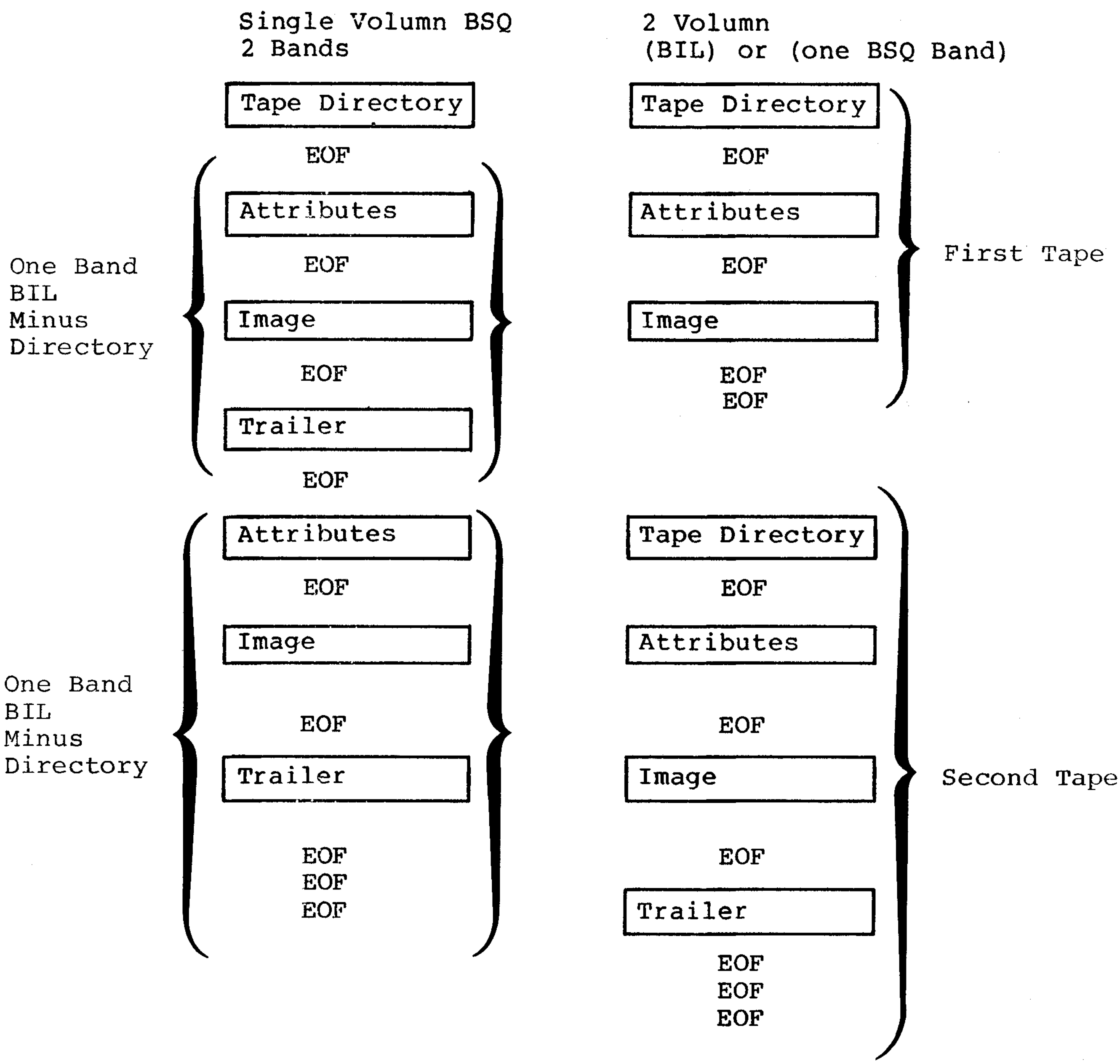
Figure 4

A Simplified Algorithm

```
SUBROUTINE lndctl

(*
    Processing control.  Reads input tape and from that determines
    processing method.
*)

    DECLARATIONS . . .

    INTEGER ARRAY psns-table(state1, state2)

    BEGIN
        done:= false
        repeat
          present -state:= begin
          next-state:= mount
          process-to-perform:= psns-table(present-state, next-state)
          IF (illegal-state in psns-table)
          THEN
            error
          ELSE
            CASE process-to-perform OF
                mount          :  call mount
                directory      :  call lnddir
                header         :  call lndhed
                ancillary      :  call lndanc
                annotation     :  call lndann
                image          :  call lndima
                trailer        :  call lndtra
                end            :  done:= ture
            ENDCASE

            present-state := next-state
            CALL toprd
            find next state from byte 6 of input record
            check record number of input record (byte 1-4)
          ENDIF
        UNTIL (done)
    RETURN
END
```

Figure 5

Data Format of EDIPS CCT's



The ATTRIBUTES file contains the HEADER, ANCILLARY, and ANNOTATION records.

If IMAGE file is followed by 2 EOF's, then the rest of the data will be found on the next input tape.

Figure 6

3. Input Description

The Landsat tape must conform to the format specifications given in the "manual on Characteristics of Landsat Computer-Compatible Tapes" produced by the EROS Data Center Digital Image Processing System, compiled by Patrick F. Holkenbrink and revised in 1978. Figure 6 is a supplement to figure 9 of the "Manual on Characteristics of Landsat Computer-Compatible Tapes." This figure shows how BIL format can be thought of as subset of BSQ format and how the input data is spanned over several tapes.

The order of the records, and the end-of-files, on the Landsat tapes is checked by the Present State - Next State table. Landsat records are recognized by the record type code located in byte 6 of each record and converted into the next state.

Each Landsat tape must have its first record contain the record type code of ´directory record´ (see table 1) and the first record must contain exactly 360 bytes. The CCT volume number is checked for proper input tape sequencing.

The Landsat formats BIL, BSQ, and RBV are treated as BIL format, so only one BSQ or RBV subscene (band) can be reformatted per job. Which subscene to reformat can be chosen with the

INPUT SCENE (subscene number)

command. (This and other commands are explained in the LNDRDR module abstract and the Landsat-EDIPS to LARSYS Control Card Description). The BAND and CHANNEL control card parameters are only implemented for BIL format. The BAND parameter is actually converted into a CHANNEL instead of reformatting bands. So when reformatting BIL format, it may be best to think in terms of CHANNELS instead of BANDS, and when reformatting BSQ or RBV, it may be best to think in terms of subscenes.

4. Output Description

Information and possibly error messages are sent to the line printer and user´s console.

5. Supplemental Information

For a description of the Landsat-EDIPS format, see the "Manual on Characteristics of Landsat Computer-Compatible Tapes" produced by the EROS Data Center Digital Image Processing System.

6. More In-Depth Algorithm

```
SUBROUTINE lndctl
        (* process Landsat-EDIPS tape input *)
     DECLARATIONS
         STATES = SET OF (beg, mnt, dir, eof1, hed, anc, ann, eof2, ima,
                          eof3, eof4, tra, eof5, eof6, eof7, end);

         input-tape    : file;
         end-of-file   : file-condition;
         ps, ns, process   : STATES;
         psns              : ARRAY(16, 16) OF STATES:   (* see lndint *)
         eofnum            : INTEGER;
         read-tape, done   : LOGICAL;
BEGIN
     ps  := beg;  ns  := mnt;  (* this initialization is done in
                              subroutine lndint *)
     done  := FALSE
     WHILE NOT done DO
         process  := psns(ps, ns);
         IF process NOT IN STATES THEN
             ERROR
         ELSE
             read-tape  := TRUE;
             CASE process OF
                  beg:  ERROR;
                  mnt:  CALL xmount; eofnum:=0
                  dir:  CALL lnddir;
                 eof1:
                  hed:  CALL lndhed;  eofnum:=1
                  and:  CALL lndanc;
                  ANN:  CALL lndann;
                 eof2:
                  ima:  CALL lndima;  eofnum:=2
                 eof3:
                 eof4:  ps:=ns;  ns:=2  read-tape:=FALSE;
                  tra:  CALL lndtra;  eofnum:=4
                 eof5:
                 eof6:
                 eof7:  done:+TRUE;
                  end:
             ENDCASE;
             IF read-tape AND NOT done THEN
                  ps := ns;
                  CALL toprd  (* read from input tape *)
                  IF end-of-file ON input-tape THEN
                      eofnum := eofnum + 1;
                      ns := eofnum
                  ELSE
                      ns := landsat-record-type-code FROM input-tape
                  ENDIF
             ENDIF
         ENDIF
     ENDWHILE
END
```

Appendix A

Use of the Present State - Next State Table

1. Introduction to the Present State - Next State Table

The purpose of this paper is to give the reader a better understanding of the Present State - Next State table, which is used for checking and processing the Landsat-EDIPS input tapes. This table is used in the subroutine LNDCTL, which is part of the Landsat-EDIPS to LARSYS reformatting program.

The approach taken here is one to use simple examples in order to demonstrate the basic concepts. State diagrams (graphs) are easy for humans to understand. Present state - next state tables are easy to program relative to other methods for implementing certain algorithms. The greatest advantage of the present state - next state table may be the ease of modification to the table.

2. Present State - Next State Examples

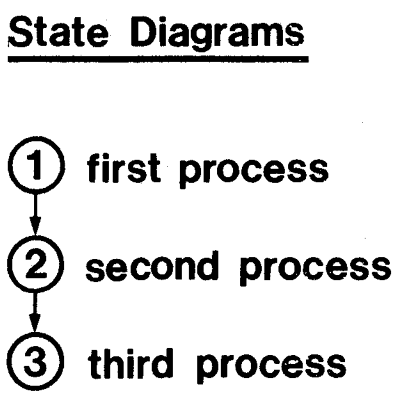


Figure 1a. Graph of forward movement.

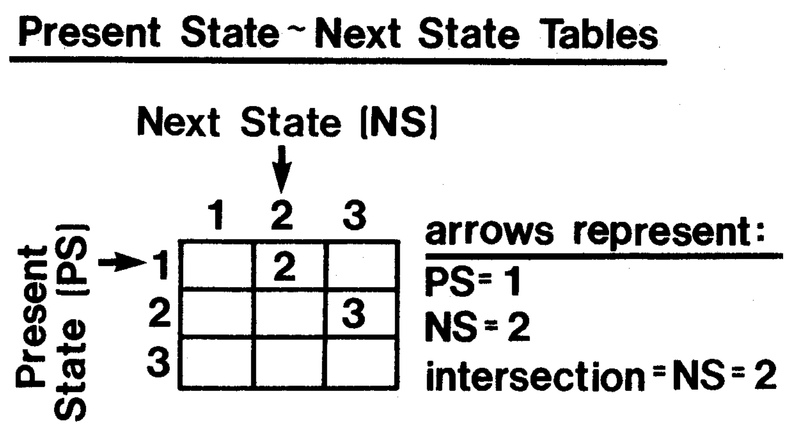


Figure 1b. Present State - Next State Table for the graph in Figure 1a.

Figure 1a shows a state diagram for three serial processors. The present state-next state table that corresponds to figure 1a is shown in figure 1b. To follow the operation of the present state - next state table, start process 1 on the left vertical margin in figure 1b. The next process desired is process 2. Starting with the present state of 1, move right until the column under the top horizontal margin (representing process 2) is found. The intersection of PS=1 and NS=2 is 2. Thus, this transition from process 1 to process 2 is valid. Process 2 now becomes the new present state and process 3 becomes the new next state. Repeat the process for PS=2 and NS=3. If the intersection of PS=1 and NS-2 would have been blank, then the transition would have been invalid.

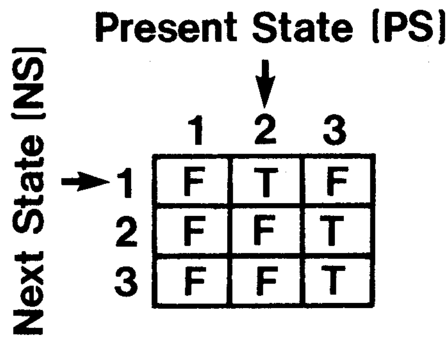


Figure 1c. PS-NS table similar to Figure 1b except that valid transitions are represented by truth (T).

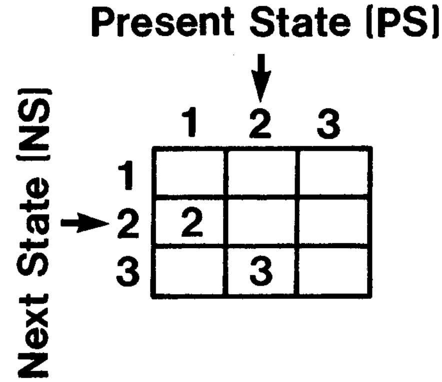


Figure 1d. PS-NS table similar to Figure 1b except the axes are reversed. (Note that Figure 1d is symmetrical to Figure 1b).

Figure 1c is similar to figure 1b except that the intersection of PS and NS returns true (T) or false (F) for the transitions. The NS is assumed to be the next process to take place as long as the intersection is true.

Figure 1d is similar to figure 1b except that the margins have been reversed (the top horizontal margin represents the present state and the left vertical margin represents the next state). This reversing produces a table that is symmetrical.

Figure 2a. Graph of backward movement.

PS \ NS	1	2	3
1			
2	1		
3		2	

Figure 2b. PS-NS table for Figure 2a.

Figure 3a. Graph of backward and forward movement.

PS \ NS	1	2	3
1		2	
2	1		3
3		2	

Figure 3b. PS-NS table for Figure 3a.

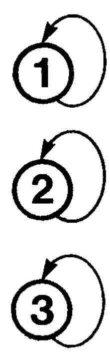


Figure 4a. Graph of repeated processes.

PS \ NS	1	2	3
1	1		
2		2	
3			3

Figure 4b. PS-NS table for Figure 4a.

3. More PS - NS Examples

Figure 2 illustrates the backward (upward) movement of a graph. Combining the abilities of the first two graphs results in the ability to have forward and backward transitions (see figure 3). In order to complete the process in a valid way, the next state must be found in the intersection from the top-left to the bottom-right diagonal (figure 4).

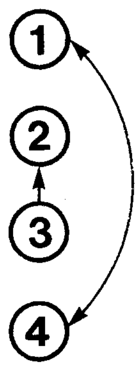


Figure 5a. Symmetrical graph about the center.

NS

PS	1	2	3	4
1				4
2			3	
3		2		
4	1			

Figure 5b. PS-NS table for Figure 5a.

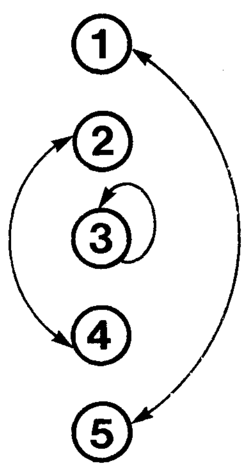


Figure 6a. Symmetrical Graph about the center

NS

PS	1	2	3	4	5
1					5
2				4	
3			3		
4		2			
5	1				

Figure 6b. PS-NS table for Figure 6b.

Figures 5 and 6 illustrate symmetrical transitions about the center process. Figure 7 depicts a graph that allows for transitions to any process. Figure 8 shows a graph of forward movement only. Another interesting graph is figure 9. Figures 10 and 11 are examples drawn from the Landsat-EDIPS reformatting program.

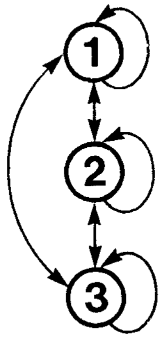


Figure 7a. Graph that allows any transition.

		NS 1	NS 2	NS 3
PS	1	1	2	3
PS	2	1	2	3
PS	3	1	2	3

Figure 7b. PS-NS table for Figure 7a.

		PS 1	PS 2	PS 3
NS	1	1	1	1
NS	2	2	2	2
NS	3	3	3	3

Figure 7c. PS-NS table for Figure 7a with axes reversed. (Note symmetry)

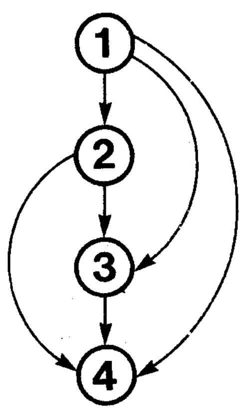


Figure 8a. Graph of any forward Movement.

		NS 1	NS 2	NS 3	NS 4
PS	1		2	3	4
PS	2			3	4
PS	3				4
PS	4				

Figure 8b. PS-NS table for Figure 81.

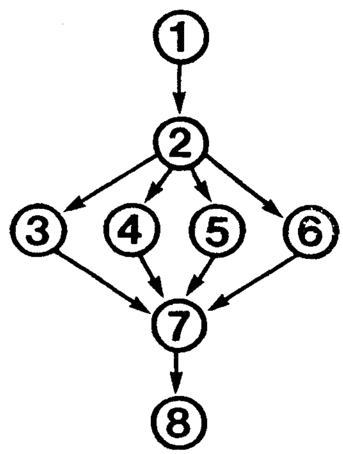


Figure 9a. Graph of general interest.

PS \ NS	1	2	3	4	5	6	7	8
1		2						
2			3	4	5	6		
3							7	
4							7	
5							7	
6							7	
7								8
8								

Figure 9b. PS-NS table for Figure 9a.

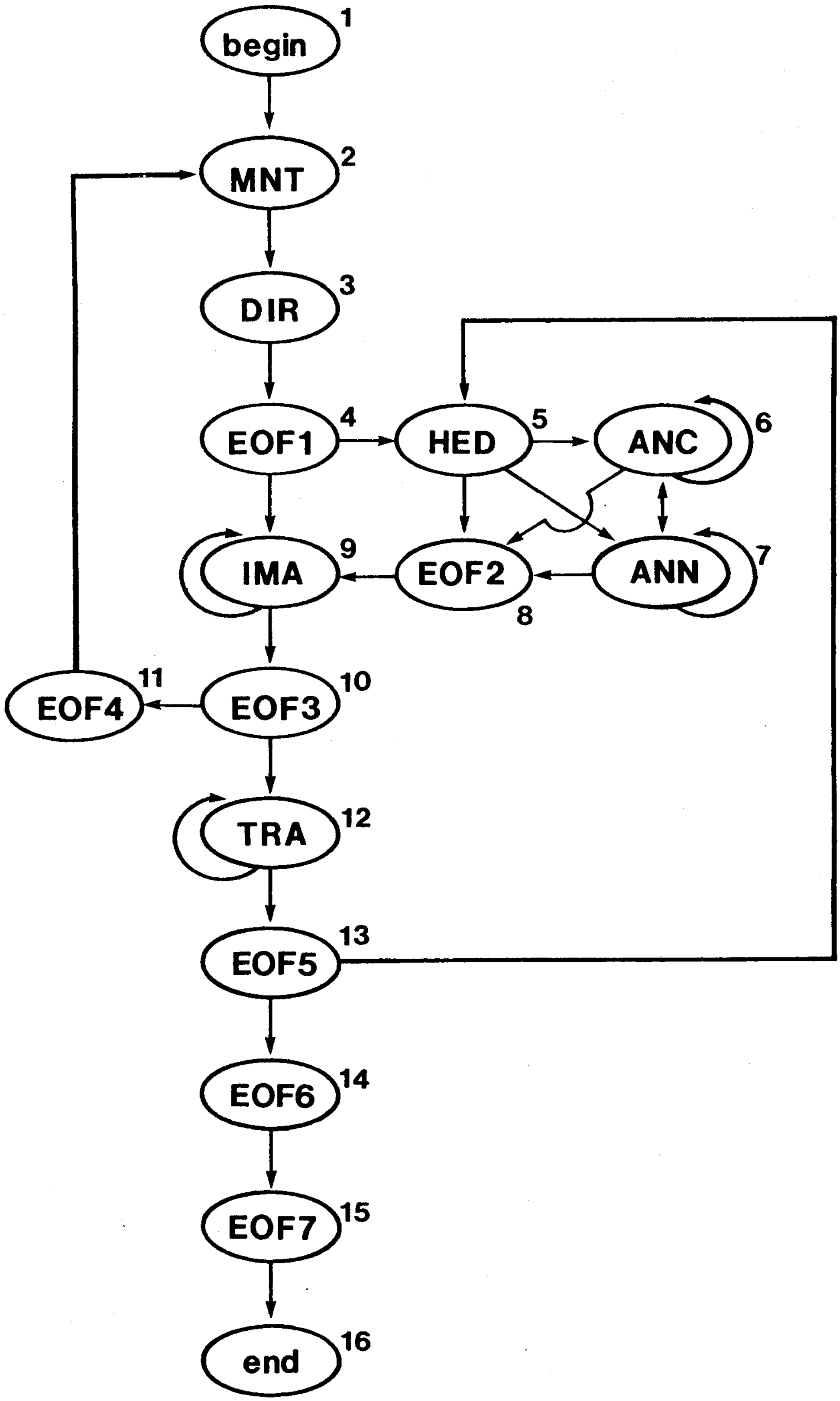


Figure 10a. Graph for the Landsat-EDIPS tapes.

PS	NS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
beg	1		2														
MNT	2			3													
DIR	3				4												
EOF1	4					5				9							
HED	5						6	7	8								
ANC	6						6	7	8								
ANN	7						6	7	8								
EOF2	8									9							
IMA	9									9	10						
EOF3	10											11	12				
EOF4	11		2														
TRA	12												12	13			
EOF5	13					5									14		
EOF6	14															15	
EOF7	15																16
end	16																

Figure 10b. PS-NS table for the Landsat-EDIPS tapes.

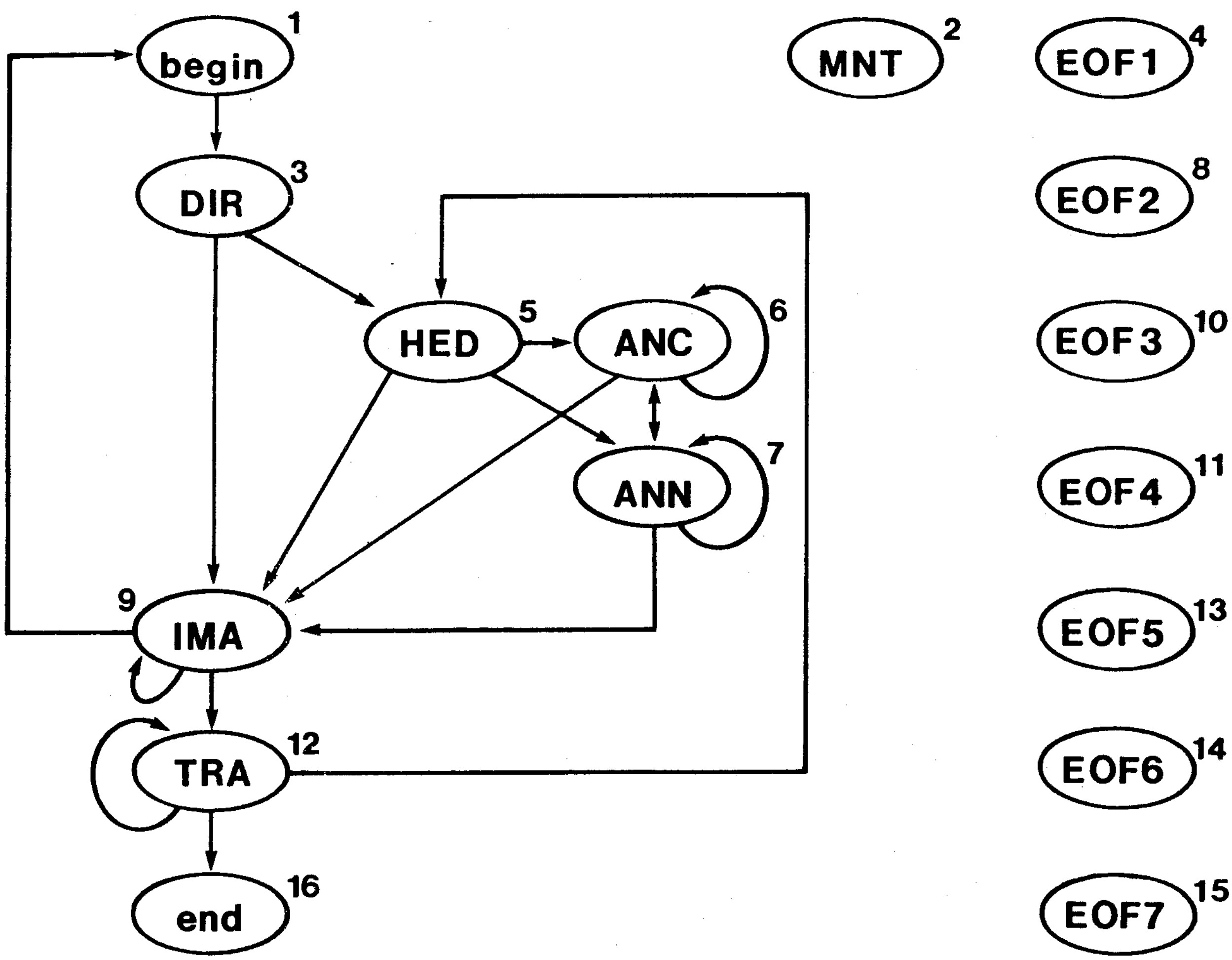


Figure 11a. The Landsat-EDIPS graph without end-of-files or a tape mount.

PS		NS: 1 beg	2	3 DIR	4	5 HED	6 ANC	7 ANN	8	9 IMA	10	11	12 TRA	13	14	15	16 end
beg	1			3													
	2																
DIR	3					5				9							
	4																
HED	5						6	7		9							
ANC	6						6	7		9							
ANN	7						6	7		9							
	8																
IMA	9			3						9			12				
	10																
	11																
TRA	12					5							12				16
	13																
	14																
	15																
end	16																

Figure 11b. The Landsat-EDIPS PS-NS table without EOF´s or a tape mount

PS \ NS	beg	DIR	HED	ANC	ANN	IMA	TRA	end
beg		DIR						
DIR			HED			IMA		
HED				ANC	ANN	IMA		
ANC				ANC	ANN	IMA		
ANN				ANC	ANN	IMA		
IMA		DIR				IMA	TRA	
TRA			HED				TRA	end
end								

Figure 11c. The Landsat-EDIPS PS-NS table without EOF's, modified to show no trace of EOF's or tape mounts.

Present State	Next State
1. begin	tape directory
3. tape directory	header, image
5. header	ancillary, annotation, image
6. ancillary	ancillary, annotation, image
7. annotation	ancillary, annotation, image
9. image	image, trailer, tape directory
12. trailer	trailer, end, header

Figure 11d. The Landsat-EDIPS transitions.

4. ADVANTAGES FOR USING A PRESENT STATE - NEXT STATE TABLE

1. The technique is very fast, because there is no searching required.

2. The technique also helps to factor application problems into several smaller, simpler, logical groups. This factoring helps to modularize the program code.

3. Evolution and maintenance of the table is quick and easy.

4. Error handling can be easily incorporated in a straight forward way.

5. Ability to provide a potential n by n unique operations from n states.

6. Generality for allowing mixing of types (eg. mixing of record type codes with end-of-files).

7. The amount of storage needed for the table is sometimes a disadvantage, but when the amount of storage needed for the program code without the table is considered, the total amount of storage needed for certain applications may actually be less with the table.

8. There are no ambiguities in the table as there would be in additional code. How these ambiguities with code are resolved depends on the version of compiler, the language, and the machine.

9. Programming language and machine independence.

10. It is easier to prove program correctness with a table because it greatly reduces the ambiguities of the algorithm.

5. ADDITION REFERENCES

1. Aho, Hopcroft, and Ullman, "The Design and Analysis of Computer Algorithms," Addison-Wesley, Reading, Massachusetts, 1974.

2. Gries, David, "The Use of Transition Matrices in Compiling", communications of the Association for Computing Machinery (CACM) 11, January 1968, pp 26-34.

3. Gries, David, "Compiler Construction for Digital Computers", John Wiley and Sons, 1971.

4. Halstead, Maurice H., "A Laboratory Manual for Compiler and Operating System Implementation", Elsevier, 1974, chapter 11.

5. Halstead, Maurice H., "Machine-Independent Computer Programming", Spartan Books, Washington, D.C., 1962.

6. Horowitz, Ellis and Sahni, Sartaj, "Fundamentals of Data Structures" , Computer Science Press, 1977, pp 287-288.

7. Kolman, Bernard, "Introductory Linear Algebra with Applications," Macmillan Publishing, 1976, section 7.3.

Appendix B

Interpretation of the Present State - Next State Table

Used for the

Landsat-EDIPS to LARSYS Reformatting Program

INTERPRETATION OF PRESENT STATE - NEXT STATE TABLE

The purpose of this discussion is to describe the use of the PRESENT STATE - NEXT STATE (PSNS) TABLE used to control the input processing within the LANDSAT reformatting process. The PSNS table is also known as a Current Operator-Next Operator (CONO) table, a transition matrix, and an adjacency matrix. Further discussions of the table can be found in the topics of finite state machines and compiler design. Figure 1 shows a representation of this PSNS table.

I. AXES OF THE TABLE

An axis is set-up much like the format of the input tape.

1. 'PRESENT STATE' can be thought of as what was most recently completed.

2. 'NEXT STATE' can be thought of as what is now desired to do.

3. The 'INTERSECTION' of the two states on the table is the process that is to be done now.

II. TRAVERSING THE TABLE

A. When the intersection of the axes is a blank (or zero), then an illegal state has occurred.

B. When the intersection of the axes is an integer number (greater than zero), then the section of code with the label corresponding to this number will be executed. The present state will be set to this integer number and the new next state will be determined by reading the next record on the input tape and checking the record type code (byte 6).

1. ADVANCING THROUGH THE TABLE

If the intersection number is greater than the present state value, then the process is advancing forward. To increment by one state, the intersection number is greater than the present state by one. Most cases where the states are incrementing by one can be seen on the diagonal on the table. Example . . .

PRESENT STATE = 1
NEXT STATE = 2
INTERSECTION = 2

This condition means that the input tape was just mounted, an input record was read, and it is now time to process the tape directory record.

NEXT STATE (NS)

PRESENT STATE (PS)		BEG 1	MOUNT 2	DIR 3	EOF1 4	HED 5	ANC 6	ANN 7	EOF2 8	IMA 9	EOF3 10	EOF4 11	TRA 12	EOF5 13	EOF6 14	EOF7 15	END 16
BEG	1		2														
MOUNT	2			3													
DIR	3				4												
EOF1	4					5				9							
HED	5						6	7	8								
ANC	6						6	7	8								
ANN	7						6	7	8								
EOF2	8									9							
IMA	9									9	10						
EOF3	10											11	12				
EOF4	11		2														
TRA	12												12	13			
EOF5	13														14		
EOF6	14															15	
EOF7	15																16
END	16																

REPEAT DIAGONAL

Figure 1

The cases where the states are incremented by more than one are:

1. PS=4, NS=9, INTER=9
   (from first EOF to IMAGE)

2. PS=5, NS=7, INTER=7
   (from HEADER to ANNOTATION)

3. PS=5, NS=8, INTER=8
   (from HEADER to second EOF)

4. PS=6, NS=8, INTER=8
   (from ANCILLARY to second EOF)

5. PS=10, NS=12, INTER=12
   (from third EOF to TRAILER)

2. REPEATING A STATE (Tight Loop)

There may be times when there will be many occurances of the same record type grouped together. For this case the same process will be repeated several times. These cases are. . .

1. PS=6, NS=6, INTER=6
   (ANCILLARY records)

2. PS=7, NS=7, INTER=7
   (ANNOTATION records)

3. PS=9, NS=9, INTER=9
   (IMAGE records)

4. PS=12, NS=12, INTER=12
   (TRAILER records)

Cases where states are repeated are shown by a diagonal on the table assuming that PS=NS=INTER.

3. BACK-UP ONE OR MORE STATES

To repeat several states, the intersection number will be less than the present state. These cases are:

1. PS=11, NS=2, INTER=2
   (end-of-tape on input, mount next input tape)

2. PS=7, NS=6, INTER=6
   (process ANCILLARY after ANNOTATION)

3. PS=13, NS=5, INTER=5
   (process next BSQ band)

C. PSNS DIAGONAL

If the intersection number equals the next state, then some easy generalizations can be made. Any intersection lying on the diagonal going from top left to bottom implies that the state is being repeated (see II B 2. above). Any intersection to the top or right of this diagonal is advancing (see II B 1. above) and any intersection to the bottom or left of this diagonal is back-tracking (see II B 3. above).

III. HOW TO ADJUST THE LANDSAT REFORMATTING PROGRAM TO EASILY ADAPT

To adapt to the evolving input format, most of the information can be found from reading the above material.

A. CHANGES FOR RECORDS

Simply change the PSNS table as described above.

B. CHANGES FOR END-OF-FILE

The variable 'EOFNUM' is used to count the number of EOD's read. The following statements are currently used . . .

1. EOFNUM = 0
   (IO initialize the eof count,
   found within the input tape mount)

2. EOFNUM = 1
   (to initialize for the next BSQ band,
   found within the HEADER processing)

3. IF (IOFNUM) .LE. 1) EOFNUM = 3
   (to skip the attribute file,
   found under the IMAGE processing)

4. IF (EOFNUM .LE.3) EOFNUM = 4
   (to skip the eof that indicates the need to mount next input tape, found under the TRAILER processing)

MODULE IDENTIFICATION

Module Name: LNDDIR Function Name: Landsat

Purpose: Process Landsat Tape Directory Records

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 01/21/80

Latest Revisor: Date:

MODULE ABSTRACT

Values are extracted from the Landsat tape directory record. These values are written to the line printer and checked for correctness. Mounting and positioning of the archive tape is done when the archive process is desired.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Some values extracted from the directory record are to be printed for the user's benefit only and some values are critical to the operation of the reformatting task. The key header record variables HSENTY, GCAPPL, AND HLEVTY (scene type, whether geometric corrections were applied and interleaving type indicator) are initialized from the respective directory record values. Since BSQ format is a subset of BIL and BSQ can be partitioned into BIL, the variables DLEVTY and HLEVTY are set to represent the simpler BIL format.

2. Internal Description

Checking is performed to insure that the input tape is of proper Landsat-EDIPS format and is inputted in the proper sequence. For the first record of each tape, LNDCTL examines the record type code for the value representing the directory record. LNDDIR verifies that the directory record contains exactly 360 bytes and verifies that byte 19 of the directory record contains the correct CCT volumn number.

3. Input Description

The directory record is received from subroutine LNDCTL in the buffer INREC. The numeric data is extracted with subroutine MOVBYT. Character data, in ASCII, is extracted and converted to EBCDIC with subroutine CASCII.

4. Output Description

Values from the directory record are written to the line printer using the subroutine CONDMP. Additional information and error message may be written to the line printer and user's terminal. The mounting, positioning, and initial writing for archiving is done within LNDDIR with the help of XMOUNT, FILSRH, LNDARC.

5. Supplemental Information

See the "Manual on Characteristics of Landsat Computer-Compatible Tapes" produced by the EROS Data Center Digital Image Processing System, pages 11 and 12.

6. Program Algorithm

```
SUBROUTINE lnddir;
     (* process the Landsat tape directory record *)
BEGIN
     IF this is the first tape directory encountered THEN
         set FORM-17C data to blanks
     ENDIF:

     IF this tape directory is the correct size THEN
         extract the data from the directory record;
         print the data extracted;
         IF the Landsat tapes have not been mounted in proper order THEN
             print error message;
             begin abort procedures
         ENDIF:

         IF archiving is desired THEN
             mount the archive tape;
             position the archive tape;
             perform the archive
         ENDIF
     ELSE
         print error message
     ENDIF

     set flag to assume all data is band interleaved by line (BIL)
END
```

MODULE IDENTIFICATION

Module Name: LNDERR Function Name: Landsat

Purpose: Error handling support routine

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/09/80

Latest Revisor: Date:

MODULE ABSTRACT

The number of error occurrences and the error severity levels detected by the Landsat program is counted. If an error of greater severity than previously detected is encountered, a message is printed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LNDERR has one parameter set, before entry, to the value corresponding to the level of error severity detected. To invoke LNDERR, use the following FORTRAN statement:

```
CALL LNDERR (HOWBAD)
```

where HOWBAD is an integer *4 variable. For more information about error severity levels, refer to the LNDSUM module abstract.

2. Internal Description

The number of errors and the error severity level is set to zero in LNDINT. If the error severity level reaches the level to abort, present state and the next state (see the LNDCTL module abstract) are printed and the tape drives are detached from the user's virtual machine.

3. Input Description

Not applicable.

4. Output Description

Messages are printed to the line printer and the user's terminal when the error severity level increases and when the error severity level reaches the level to abort.

5. Supplemental Information

Not applicable.

6. Program Algorithm

```
SUBROUTINE lnderr(howbad);
     (*
BEGIN
     increment count of error occurrances;
     IF howbad > error-severity-level THEN
         accumulate error-severity-level + howbad;
         PRINT the new error-severity-level;
         IF error-severity-level > = abort-level THEN
             PRINT a very noticeable error message to abort;
             begin abort procedures
         ENDIF
     ENDIF
END
```

MODULE IDENTIFICATION

Module Name: LNDF17 Function Name: Landsat

Purpose: Print LARSYS Form-17c

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/09/80

Latest Revisor: Date:

MODULE ABSTRACT

Write a specified number of FORM-17c´s to the line printer.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Much of the FORM-17c is taken from the LARSYS identification header (see LNDLID module abstract). Other data are collected by LNDDIR, LNDHED, LNDANN, and GETACT. The default number of FORM-17c's to be printed is 7, but 1 to 30 forms may be specified by using an 'OUTPUT FORM17C(n)' card where n is the number of forms desired.

2. Internal Description

Not applicable

3. Input Description

None

4. Output Description

Several copies of a one page FORM-17C is printed. The letter "I" is printed in the four corners of the form for paper cutting to an 8 1/2 x 11 sheet.

5. Supplemental Information

Not applicable.

6. Program Algorithm

```
SUBROUTINE lndf17;
     (* print the LARSYS FORM-17C *)
BEGIN
     compute GMT and local time data was taken;
     find reformatting dept. work project number and master account;
     i :=1;
     WHILE i <= number of forms desired DO
         PRINT a FORM-17C
         i := i + 1
     ENDWHILE
END
```

MODULE IDENTIFICATION

Module Name: LNDHED Function Name: Landsat

Purpose: Process Landsat header reocords

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/01/80

Latest Revisor: Date:

MODULE ABSTRACT

Values are extracted from the Landsat header record. These values are written to the line printer and later used for controlling the reformatting process.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Subroutine CASCII and MOVBYT extract the ASCII and binary data, respectively. This data is printed by CONDMP.

2. Internal Description

If archiving is desired and no archiving errors have been detected, LNDARC is called to archive the header record. Variable HLEVTY is set to type BIL to treat all input formats as BIL format.

3. Input Description

The header record is received from LNDCTL in the array INREC.

4. Output Description

Several values from the header record are written to the line printer VIA the subroutine CONDMP.

5. Supplemental Information

See the "Manual on Characteristics of Landsat Computer-Compatible Tapes" produced by the EROS Data Center Digital Image Processing System, pages 13 through 23.

6. Program Algorithm

```
SUBROUTINE lndhed;
     (* process the Landsat header record *)
BEGIN
     extract the data from the header record;
     print the data extracted;
     IF archive process desired THEN
         archive VIA LNDARC
     ENDIF;
     set flag to assume all data is band interleaved by line (BIL)
END
```

MODULE IDENTIFICATION

Module Name: LNDIMA Function Name: Landsat

Purpose: Process Landsat Image records

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/03/80

Latest Revisor: Date:

MODULE ABSTRACT

The image line number is extracted from the image record. Subroutine LNDBIL is called to process the image record further.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LNDIMA is only set up to handle geometrically corrected data.

2. Internal Description

Code to extract the left and right fill counts is still present but has been "commented out".

3. Input Description

The image record is received from LNDCTL in buffer INREC. The scan line count (SCNCNT) is set to the current line number within the Landsat frame.

4. Output Description

If the variable GCAPPL from the header record is set to represent uncorrected data, an error message is printed and execution halts.

5. Supplemental Information

See the "Manual on Characteristics of Landsat Computer-Compatible Tapes" produced by the EROS Data Center Digital Image Processing System, page 46 through 49.

6. Program Algorithm

```
SUBROUTINE lndima;
     (* process the Landsat image record *)
BEGIN
     IF the data is geometrically corrected THEN
          extract scan line count
     ELSE
          ABORT
     ENDIF:
     CALL lndbil
END
```

MODULE IDENTIFICATION

Module Name: LNDINT Function Name: Landsat

Purpose: Initialize global variables for the Landsat reformatting process

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/08/80

Latest Revisor: Date:

MODULE ABSTRACT

For each job, LNDSUP calls LNDINT to initialize all global constants and nearly all global variables. A description of these globals is also given.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

The larger array initialization is done in the block data routine appended to LNDINT. These arrays are copied to new arrays, in transposed form, by LNDINT.

2. Internal Description

The four byte character to integer function CHAR is called for initializing small literal strings.

3. Input Description

Not applicable.

4. Output Description

Not applicable.

5. Supplemental Information

Not applicable

6. Program Algorithm

Not applicable

MODULE IDENTIFICATION

Module Name: LNDLID Function Name: Landsat

Purpose: Create a LARSYS identification header record

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/09/80

Latest Revisor: Date:

MODULE ABSTRACT

Data collected in LNDDIR, LNDHED, LNDANN, LNDPRM, and LNDVAL are used to build an 800 byte LARSYS header. The LARSYS header is the last record to be archived.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

In the section of code that creates the LARSYS header, a comment begins each part of the code for each grouping that identifies the words in the header and a description for the grouping. Writing the header to the output tape is handled in LNDBIL. The information in the header is used again to generate FORM-17c's (see the LNDF17 module abstract).

2. Internal Description

The code for the satellite ground heading is taken from the reformatting program REFERTS. LNDLID completes the archiving by writing two end-of-file marks and detaching the tape drive from the user's virtual machine.

3. Input Description

Not applicable.

4. Output Description

Messages are written to the line printer and user's terminal when: 1) errors are detected; 2) lines to reformat are determined; 3) subroutine IDRITE is not used; & 4) when the archive process is completed.

5. Supplemental Information

Refer to LARSYS Version 3.0 tape format.

6. Program Algorithm

```
SUBROUTINE lndlid;
     (* create a LARSYS ID for the LARSYS output tape *)
BEGIN
     CREATE LARSYS header;
     IF idrite is not to be called by lndbil THEN
         PRINT LARSYS header;
         (* otherwise idrite prints the header *)
     ENDIF;
     COPY the LARSYS header to lower part of the FORM-17c array;
     IF archive desired THEN
         archive the LARSYS header;
         wrap-up archive process for this job
     ENDIF;
END
```

MODULE IDENTIFICATION

Module Name: LNDMIL Function Name: Landsat

Purpose: Save the desired Landsat channels for a line for outputting as one LARSYS record

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 01/08/80

Latest Revisor: Date:

MODULE ABSTRACT

LNDMIL is called by subroutine LNDBIL to merge Landsat image data records for one line. Landsat band interleaved by line (BIL) format is assumed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

One image data record in Landsat-EDIPS format represents one channel for a particular line. In LARSYS format, all channels for a particular line are represented by one data record. LNDMIL stores the multiple Landsat data records for one line into a single output buffer.

2. Internal Description

```
SUBROUTINE lndmil
     (* store Landsat channels of one line for later outputtng in LARSYS
     format *)
BEGIN
     IF this is the first channel THEN
         begin saving pixels at the beginning of the output buffer;
     ENDIF;

     move the pixels from the input buffer to the output buffer;
     move the six calibration bytes to the output buffer;
     keep track of how full the output buffer is
END
```

Subroutine MOVBYT (not a part of the Landsat function) does the actual moving of image pixels and calibration bytes.

3. Input Description

Not applicable

4. Output Description

Not applicable

5. Supplemental Information

See the MOVBYT module abstract.

MODULE IDENTIFICATION

Module Name: LNDPAG Function Name: Landsat

Purpose: Write a LARSYS like header at the top of the line printer page.

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/08/80

Latest Revisor: Date:

MODULE ABSTRACT

Prints user, job, date and time information at the top of each output page.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

By taking the difference between the time of day, terminal connect time, and CPU time, one can determine how much time is spent processing the data corresponding to a page of output. If the -COMMENT command is used in the control cards, the first two lines of the -COMMENT will be printed with the page header.

2. Internal Description

Not applicable.

3. Input Description

Not applicable.

4. Output Description

Information is printed at the top of each output page.

5. Supplemental Information

See the LNDRDR module abstract and the Landsat-EDIPS to LARSYS control Card Description for more information about the -COMMENT command.

6. Program Algorithm

Not applicable.

MODULE IDENTIFICATION

Module Name: LNDPRM Function Name: Landsat

Purpose: Process the control parameters on a control card.

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 2/05/80

Latest Revisor: Date:

MODULE ABSTRACT

After the keywork has been read by LNDRDR, the commands following the keyword are to be processed by LNDPRM. If any integer or string constants are expected, LNDVAL or LNDSTR are called, respectively.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Initially, all control parameters are valid for any keyword. After CTLPRM is called, the array LEGAL is used to determine whether the control parameter is valid for the given keyword.

2. Internal Description

The user is not required to separate control parameters with commas, so he/she can make control cards more pleasant to the eye. CTLCBC was written to handle the burden of placing commas in awkward places for compatability with CTLPRM (entry in the LARSYS support routine CTLWRD).

Much of the code in LNDRDR is in preparation for LNDPRM and much of the code in LNDPRM is in preparation for LNDVAL.

3. Input Description

One line is received from LNDRDR in the buffer LN20A4 with COL pointing just before the first control parameters.

4. Output Description

If an error is detected, messages are written to the line printer and the user´s terminal.

5. Supplemental Information

None.

6. Program Algorithm

```
SUBROUTINE lndprm;
     (* process control parameters for a given keyword *)
BEGIN
     make sure at least one comma separates each control-parameter;
     WHILE NOT end-of-line DO
         extract the next control-parameter;
         IF this control-parameter is valid for current keyword THEN
             CASE control-parameter OF
                 'BANDS'     :  process channels;
                 'BPI'       :  process density;
                 'CHANNELS'  :  process channels;
                 'COLUMNS'   :  process columns;
                     .
                     .
                     .
                 'TAPE'      :  process tape
             ENDCASE
             IF integer value expected next THEN
                  CALL lndval
         ELSE
             ERROR-'control-parameter not valid for this keyword'
         ENDIF
     ENDWHILE
END
```

MODULE IDENTIFICATION

Module Name: LNDRDR Function Name: Landsat

Purpose: Supervise the reading of the user's control cards

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 2/3/80

Latest Revisor: Date:

MODULE ABSTRACT

The card reader gives the user control of the Landsat to LARSYS process by reading commands and setting the appropriate variables in common blocks.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

LNDRDR reads control cards, processes the first token (keyword) of the card, and calls LNDPRM to process the rest of the card. Implementation of the control card reader is centered around the module CTLWRD.

2. Internal Description

The Landsat to LARSYS program was designed with possible future extensions in mind. To process more than one BSQ band would require at most two scratch tapes. A SCRATCH keyword has already been implemented for this purpose.

The control card is read in four bytes per word (buffer LN20A4) to be compatible with existing system software (i.e., CTLWRD and BCDVAL). Since the enhancement of allowing indentation is implemented into Landsat to LARSYS card reader, entry point CTLPRM of module CTLWRD is used instead of the main entry point of CTLWRD. A blank is concatenated to the beginning of the card copy (LN21A4) to be compatible with CTLPRM. To remove the restriction of requiring the user to have commas between words, LNDRDR must insert the commas before calling CTLPRM. Another copy of the control card (LN72A1) is made with one byte per word to be easier to process in FORTRAN.

The control card input is echoed to the line printer. It may be desirable to also echo the control cards to the user's terminal. This has been implemented but commented out because if an error is detected by CTLWRD or BCDVAL, they already echo the control card. So to avoid the confusion of double echoing, the control card input is not echoed to the user's terminal.

To allow indentation in the control cards, and still be able to use CTLWRD and BCDVAL, the column pointer (COL) must be set to point to the blank immediately preceeding any non-blank commands.

Blank lines and lines with '**' or '* ' as the first non-blank characters are ignored. Lines with '-COMMENT' keywords have a special purpose and are not completely ignored.

After the code for the keyword is executed, LNDPRM is called to complete processing of the line.

3. Input Description

The control cards consist of one keyword followed by control parameters (for that keyword) on each line. This arrangement is very similar to the control card decks used in the LARSYS system except for some enhancements. For more details, see the "Landsat-EDIPS to LARSYS Control Card Description" in Appendix A.

4. Output Description

The control cards are echoed to the line printer with columns 73 through 80 spaced out to the right so the user can notice any control card lines exceeding column 72 (input after column 72 is not scanned for commands). If any errors are detected, messages are sent to the line printer and user's terminal.

5. Supplemental Information

The "Landsat-EDIPS to LARSYS Control Card Description" tells how to set up a control card deck for Landsat reformatting.

6. Program Algorithm

```
SUBROUTINE lndrdr;
     (* supervise the reading of the user´s control cards *)
BEGIN
     initialize;
     WHILE NOT end-of-file ON control-cards AND
         NOT end-card-encountered AND
         error-level   abort
     DO
         READ one control card;
         IF NOT end-of-file ON control-cards THEN
             WRITE the control card to the line-printer;
             SKIP preceeding blanks in the control-card;
             IF NOT empty control-card THEN
                 IF first two non-blank characters = (´**´ or ´* ´) THEN
                     extract the first token of the control-card;
                     IF first token is not a legal command THEN
                         ERROR-´invalid keyword´
                     ELSE
                         CASE token OF
                             *LANDSAT : prepare for Landsat job;
                             -COMMENT : process comment;
                             INPUT    : prepare for input commands;
                             OUTPUT   : prepare for output commands;
                             ARCHIVE  : prepare for archive commands;
                             CP       : process CP command;
                             END      : process end card;
                         ENDCASE;
                         CALL LNDPRM to process rest of card
                     ENDIF
                 ENDIF
             ENDIF
         ENDIF
     ENDWHILE
END
```

APPENDIX A

Landsat-EDIPS to LARSYS

Control Card Description

Written November 24, 1979

Revised February 14, 1980

by Ken Dickman

Purdue LARS

1. INTRODUCTION

This guide describes the input required from the user to control the reformatting process to convert from Landsat-EDIPS tape format to LARSYS tape format. The user input is similar to that used in the LARSYS analysis system. The following is an example of the Landsat to LARSYS control card format.

THIS IS THE GENERAL SYNTAX
FOR THE
LANDSAT TO LARSYS
CONTROL CARD READER

```
*LANDSAT
-COMMENT     text
INPUT        parameters  and/or  parameter(values)
OUTPUT       parameters  and/or  parameter(values)
ARCHIVE      parameters  and/or  parameter(values)
CP           text
END
```

Keywords are to be typed on the control cards as shown above. Parameters and text are to be replaced with valid text or control parameters. The text following the -COMMENT keyword can be of any form, whereas the text following the CP keyword must be valid for an IBM VM/370 CP command. The INPUT, OUTPUT, and ARCHIVE keywords require at least one control parameter after the keyword. Several control parameters require one or more integer (or character) values enclosed in parentheses immediately following the control parameters. The meanings of the keywords, parameters, and text will be given in subsequent sections as well as in the Summary of Landsat-EDIPS to LARSYS Control Commands.

2. FREEDOMS/RESTRICTIONS (environment)

This section illustrates some of the similarities with the LARSYS control card reader along with some added features. The first thing to appear on a line must be a keyword. A keyword and its associated data can begin and end anywhere on a card between columns 1 and 72, inclusive. Blank lines are ignored, as are lines beginning with a ´**´ or ´* ´. Only the first four characters of a keyword and its parameter words are "significant" (any characters after the first four and before a blank or comma are ignored). However, all characters of text found after the -COMMENT and CP keywords are "significant".

Below is an example of the maximum number of values for the TAPE control parameter following each tape I/O keyword.

```
INPUT TAPE(1001, 1002, 1003)
OUTPUT TAPE(4480)
ARCHIVE TAPE(5237)
```

Any number of control parameters are permitted for any keyword. For one job, any number of keywords are allowed with one exception: only one end card is allowed to be used per job. Any number of jobs separated by END cards are allowed per card deck, although for correct FORM-17C´s, only one job should be run per card deck. The minimum required card deck is:

```
*LANDSAT
INPUT  TAPE(all tapes of the CCT set)
OUTPUT RUN(eight digit run number)
END

     or

*LANDSAT
INPUT  TAPE(all tapes of the CCT set)
OUTPUT OFF
END
```

For the first minimum control card deck, since no tape was specified, a tape would be selected automatically and the converted data would be written to the selected tape using all default parameters and values. The second minimum control card deck would do the same as the first deck except that no output tape would be selected, mounted, or written to.

Below is an example of a control card deck depicting more than is required to complete the job.

```
*LANDSAT-TO-LARSYS
  **THIS LINE IS IGNORED
  * THIS LINE IS ALSO IGNORED
  *THIS LINE IS IN ERROR
  -COMMENT ECHOED TO THE TERMINAL AND LINE PRINTER
  CP MESSAGE * THIS LINE IS ECHOED TO THE TERMINAL
  CP MESSAGE CP THIS LINE IS ECHOED TO THE OPERATOR
  *
  INPUT LANDSAT-EDIPS FROM TAPE(100, 101)
  INPUT BPI(1600), DENSITY(1600), MODE(16)
  INPUT BANDS(4,5,6,7), CHANNELS(1,2,3,4)
  INPUT LINES(1, 1000, 1), COLUMNS(1, 2000, 1)
  *
  OUTPUT TO LARSYS RUN(1234567), TAPE(5971)
```

```
    OUTPUT FILE(1)
    OUTPUT BPI(800), DENSITY(800), MODE(17)
    OUTPUT FLIGHTLINE(0123456789ABCDEF)
    OUTPUT NODETACH, OFF, FORM17C(1)
    *
    ARCHIVE TO TAPE(113), FILE(1)
    ARCHIVE NODETACH, OFF
  END
    CP MESSAGE * ALL DONE
```

There are several reasons why the above control card deck is more than necessary. The BPI, DENSITY, and MODE control parameters have the same meaning. MODE(16) sets the tape density to 1600 bpi and MODE(17) sets the tape density to 800 bpi. BANDS and CHANNELS control parameters also have similar meanings. Bands represent the NASA assigned spectral bands, and are mapped into channels by subtracting 3 from each integer value. The archiving process is not activated unless the ARCHIVE keyword appears. A card containing ´ARCHIVE OFF´ de-activates the archiving process.

A more typical control card deck might be:

```
  *LANDSAT-EDIPS
    -COMMENT  KEN DICKMAN OF LARS  09/16/79
    -COMMENT  CARD FILE = ´30001 CC´
       INPUT  FROM LANDSAT TAPE(30001), DENSITY (1600)
       OUTPUT LARSYS TO RUN(79000300), TAPE(2670)
       OUTPUT FLIGHTLINE(335815512    MS), FILE(1)
    CP MESSAGE CP TAPE 30001 IS IN MY MAILBOX.
    CP MESSAGE * 30001 CC MESSAGE SENT TO OPERATOR
  END
```

3. TAPE I/O KEYWORDS

INPUT, OUTPUT, and ARCHIVE comprise the tape I/O keywords. These three keywords directly affect tape operation. The INPUT and OUTPUT tape I/O keywords are required for a valid control card deck. Not all parameters that are allowed for one keyword, are allowed for another (see Table 1). Table 1 gives a list of the most important parameters and the number of values required for each parameters.

3.1 The 'INPUT' Keyword

The first input tape through to the tape containing the last line (or last requested line + 1) to reformat must be specified, in order (i.e., CCT volume 1, CCT volume 2, ...), in the control cards.

3.2 The 'OUTPUT' Keyword

If the output tape is not specified, the LARS-Reformatting tape management system will select an output tape and an output file automatically. The run number must be specified. If the OFF control parameter is specified, the LARSYS output is suspended and the run number is not required.

3.3 The 'ARCHIVE' Keyword

When the ARCHIVE keyword is used, all Landsat records before the first image record plus the LARSYS identification record are written to the archive tape at the file specified. If the file is not specified, the records are written to the first empty file on the tape. Two end-of-the-file marks are written at the end of the last archive file, whether or not the file has been specified.

Table 1. Valid Parameters for Each I/O Keyword and the Number of Values for Each Parameter.

TAPE I/O KEYWORDS

INPUT	OUTPUT	ARCHIVE	parameters
4-7	-	-	BANDS
-	1	-	BPI
1-30	-	-	CHANNEL
1-3	-	-	COLUMN
1	1	-	DENSITY
-	1	1	FILE
-	1	-	FLIGHTLINE
-	1	-	FORM17/C
0	-	-	FROM
0	-	-	LANDSAT
-	0	-	LARSYS
1-3	-	-	LINE
1	1	-	MODE
0	0	0	NODETACH
-	0	0	OFF
-	1	-	RUN
1	-	-	SCENE
1-3	1	1	TAPE
-	0	0	TO
-	-	0	USING

Legend:

The inside of the table shows the range of the number of values allowed for each parameter per keyword.

A zero means that the parameter is allowed for the keyword but the parameter must not have any values associated with it.

An '-' means that the parameter is invalid for the keyword.

4. NON-TAPE I/O KEYWORDS

-COMMENT, *LANDSAT, CP, and END comprise the non-tape I/O keywords. These four keywords are not directly or exclusively for tape operations. The *LANDSAT and END non-tape I/O keywords are required for each job.

4.1 The '-COMMENT' Keyword

Lines starting with -COMMENT will have the text string printed at the user's terminal as well as at the top of each subsequent page for that job. Any -COMMENT keywords after the first two are ignored.

4.2 The '*LANDSAT' Keyword

This keyword invokes the Landsat to LARSYS processor and should be used to identify the job.

4.3 The 'CP' Keyword

The text following the CP keyword will be sent immediately, after that CP keyword is read, to the operating system's control program, VIA the subroutine CPFUNC. The following is an example of one of its uses:

```
CP SPOOL CONSOLE START *
```

Messages sent from the system's control program to the user's console will not be recorded unless the user previously "spools console on". The command 'CP SPOOL CONSOLE START *' will begin the recording session. The command 'CP SPOOL CONSOLE STOP CLOSE' will halt the recording session and store the session in the user's virtual reader.

4.4 The 'END' Keyword

When the END card is read, control card reading is suspended and the Landsat to LARSYS conversion begins.

5. ERROR DETECTED BY THE CONTROL CARD READER

When errors are found, a dollar sign will be printed beneath the approximate column where the error occurred in the card. At least one message will be printed to help the user determine the problem.

SUMMARY OF

LANDSAT-EDIPS TO LARSYS CONTROL COMMANDS

Written November 27, 1979

Revised February 14, 1980

by Ken Dickman

Purdue LARS

LANDSAT-EDIPS TO LARSYS CONTROL COMMANDS

REQ	KEY WORD	REQ	CONTROL PARAMETER	FUNCTION	DEFAULT
+	*LANDSAT		(none)	Convert Landsat-EDIPS tape format to LARSYS tape format.	(none)
	-COMMENT		anything	Echoes whatever follows the keyword at the terminal and on the line printer output. -COMMENT after the first two (per control card set) are ignored.	BLANK
	**		anything	Anything after two asterisks (or one asterisk and a blank) is ignored.	N/A
+	INPUT			Read from the Landsat tape.	(none)
			BAND(b)	Select which spectral bands to reformat. If the scan line count in the image record is zero then that record is skipped (this has been noticed with band 8).	BIL: 4,5, 6,7; BSQ or RBV (first subscene);
			BPI(d)	Select density of either 800 bpi or 1600 bpi. Any one of the BPI, DENSITY, or MODE control parameters is sufficient.	Density of 800 BPI (dual density drive)
			CHANNEL(c)	Select which LARSYS channels to reformat. Actually bands (see band control parameter above) are mapped into channel notation in the Landsat-EDIPS to LARSYS card reader. The default channels are the first four of any tape format or fewer if fewer exist (eg. RBV).	BIL MSS: 1,2,3,4; BSQ MSS:1; RBV: 1;

R E Q	KEY WORD	R E Q	CONTROL PARAMETER	FUNCTION	DEFAULT
			COL(x,y,z)	Data from column x to y with interval z. Maximum number of columns for MSS, RBV uncorrected, and RBV corrected are 3548, 5322, and 5375 respectively. The number of columns + 6 reformatted should be divisible by 4 so all of the LARSYS processors can handle the data properly. If the columns are not evenly divisible by r, the Landsat-EDIPS to LARSYS program will attempt to increase the number of columns specified.	MSS: (53, 3494,1); RBV uncorrected: (1,5374,1) RBV corrected: (1,5322,1)
			DENSITY(d)	Select the number of bytes per inch to read the input tape. This is equivalent to the BPI control parameter above.	Density of 800 bpi (dual density drive)
			FROM	Used to make control cards more readable. (eg INPUT FROM LANDSAT TAPE(x) or INPUT LANDSAT FROM TAPE(x))	(none)
			LANDSAT	Specify the input tape format.	Landsat-EDIPS
			LINE(x,y,z)	Data from line x to y with interval z. The default values taken are the maximum number of lines specified in the "Manual on Characteristics of Landsat CCT´s." At present, it is not known at LARS the number of lines in a RBV uncorrected tape so, for now, the default is 9999 lines.	MSS corrected: (1,2983,1) MSS uncorrected: (1,2400,1) RBV corrected: (1,5322,1) RBV uncorrected: (1,9999,1)

LANDSAT-EDIPS TO LARSYS CONTROL COMMANDS

R E Q	KEY WORD	R E Q	CONTROL PARAMETER	FUNCTION	DEFAULT
			MODE(m)	Select the tape mode. MODE(16) selects a density of 1600 bpi and MODE(17) selects a density of 800 bpi. Any one of the BPI, DENSITY, or MODE control parameters is sufficient.	MODE(17)
			NODETACH	The tape drive is not detached from the user's virtual machine after completion of every job.	Input tape drive is detached.
			SCENE(s)	Selects the scene (or in case of MSS BSQ and RBV - select the subscene) desired to reformat.	First
		+	TAPE(x,y,z)	Specify the one to three Landsat-EDIPS tapes in order.	(none)

LANDSAT-EDIPS TO LARSYS CONTROL COMMANDS

R E KEY Q WORD	R E CONTROL Q PARAMETER	FUNCTION	DEFAULT
+OUTPUT		Read from the Landsat tape.	(none)
	BPI(d)	Same as for INPUT BPI(d) above except the default density is 1600 bpi. Any one of the BPI, DENSITY, or MODE control parameters is sufficient.	Density of 1600 bpi
	DENSITY(d)	Same as for INPUT DENSITY(d) and BPI(d) above except the default density is 1600 bpi.	Density of 1600 bpi
	FILE(f)	Specify the output tape position.	If tape specified then file 1 is default.
	FLIGHT(fl)	Specify the flight line segment to be inserted into the LARSYS header words 7 through 10. The string must be surrounded by parentheses, and blanks are significant within the string	The first 16 or less characters of the tape id, (bytes 7 - 22) of the Landsat directory record.
	FORM17C(n)	Specify the number of FORM - 17C pages to print	7 pages
	LARSYS	Specify the output tape format.	LARSYS format
	MODE(m)	Same as for INPUT MODE(m) above except the default mode is 16.	MODE(16)
	NODETACH	The tape drive is not detached from the user's virtual machine after completion of every job.	Output tape drive is detached.

LANDSAT-EDIPS TO LARSYS CONTROL COMMANDS

R E Q	KEY WORD	R E Q	CONTROL PARAMETER	FUNCTION	DEFAULT
			OFF	Suppresses mounting of the output tape and writing to the output tape. This could be used to inspect the input tape, or to archive without reformatting.	Not OFF
		+	RUN(r)	Specify the LARSYS run number. Must be a value between 1 and 99999999. Run is not required if the OFF control parameter is specified.	(none)
			TAPE(t)	Specify the output tape.	Output tape and file selected automa- tically.
			TO	Used to make control cards more readable. (eg. OUTPUT TO LARSYS RUN(r) or OUTPUT LARSYS TO RUN(r)).	(none)
	ARCHIVE			Write the Landsat records up to (but not including) the image record to a tape.	(archiving is not done)
			FILE(f)	Specify the ARCHIVE tape position.	The first empty file.
			NODETACH	The tape drive is not detached from the user's virtual machine afteris completion of every job.	Archive tape drive detached
			OFF	Suppress the ARCHIVE process (the archive tape is not mounted). Archiving is normally suppressed unless an ARCHIVE tape is specified.	(archiving is not done)

LANDSAT-EDIPS TO LARSYS CONTROL COMMANDS

R E KEY Q WORD	R E CONTROL Q PARAMETER	FUNCTION	DEFAULT
	TAPE(t)	Specify the ARCHIVE tape.	(archiving is not done).
	TO	Used to make control cards more readable. (eg. ARCHIVE TO TAPE(t)).	(none)
	USING	Used to make control cards more readable. (eg. ARCHIVE USING TAPE(t))	(none)
CP	CP command	To execute any CP command. (eg. CP QUERY READER CP QUERY TAPES CP MESSAGE TAPES CP MESSAGE CP HELLO CP MESSAGE * WAKE UP).	(none)
+ END	(none)	End of function.	(none)

MODULE IDENTIFICATION

Module Name: LNDSTR Function Name: Landsat

Purpose: Extract a string of characters, surrounded by parenthesis, from a control card.

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Data: 02/06/80

Latest Revisor: Date:

MODULE ABSTRACT

LNDSTR reads a character string from a control card. The string must be surrounded by parentheses.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Compatability with the method used in CTLWRD and BCDVAL has been kept except that CARD is 72 fullwords with one character per word rather than one character per byte. The arguments passed to LNDSTR are as follows:

CALL LNDSTR (CARD, STRING, STRSZ, ERROR)

ENTRY ARGUMENTS

CARD - 72 fullwords with one character per word, of a control card line.

COL - index pointing within CARD to the left of the string as for use of BDCVAL.

STRZ - the length of STRING in words.

EXIT ARGUMENTS

STRING -the buffer containing the string extracted from CARD.

ERROR -true if an error was detected, false otherwise.

2. Internal Description

Besides using parentheses to delimit the string on the control card, single quotes, double quotes, and slashes can be used as delimiters but CTLWRD and BCDVAL only recognize parentheses. If the control card string is greater than STRSZ, only the leftmost characters will be stored in string. If the control card string is less than STRSZ, the right most elements of STRING will be set to blank. An empty string will be returned as well as blanks.

3. Input Description

The control card is read in 72A1 format by the calling routine.

4. Output Description

If an error is detected, messages are written to the line printer and user´s terminal.

5. Supplemental Information

None

6. Program Algorithm

```
SUBROUTINE lndstr (card, col, string, strsz, error);
     (*
         starting at col, copy strsz or fewer characters (that are
         surrounded by parentheses) in card to string
     (*
DECLARATIONS
     card       : one byte/word ARRAY(1 to 72 OF CHARACTER: (* one control
                  card line
     col        : INTEGER;                     (* index into card
     string     : one byte/word ARRAY (1 to strsz) OF CHARACTER;   (*
                  desired characters from card

     strsz     : INTEGER;                                        (* max
                 size of string
     error     : LOGICAL;                                        (* TRUE=
                 characters not copied


BEGIN
     col    :=col +1
     error :=FALSE;
     WHILE card(col) = ´ ´ DO
         col   := col + 1                  (* skip blanks *)
     ENDWHILE:
     IF col is at end-of-line THEN
         error := TRUE                     (* string not found *)
     ELSE
         IF card(col) = ´(´ THEN
             error := TRUE                     (* ´(´ not found *)
         ELSE
             col := col +1;
             i   := 1;
             WHILE NOT end-of-line AND card(col) = ´)´ DO
                 IF i < strsz THEN
                     string(i) := card(col); (* copy *)
                     i  := i +1
                 ENDIF;
                 col := col + 1
             ENDWHILE:
             col := col + 1; (* increment past ´)´ *)
             WHILE i - strsz DO
                 string (i) := ´ ´;                  (* pad blanks *)
                 i := i +1
             ENDWHILE;
             WHILE NOT end-of-line AND card(col) = ´ ´ DO
                 col :=col + 1                       (* skip blanks *)
             ENDWHILE;
             col := col + 1
         ENDIF
     ENDIF
END
```

MODULE IDENTIFICATION

Module Name: LNDSUM Function Name: Landsat

Purpose: Print job summary information

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 02/08/80

Latest Revisor: Date:

MODULE ABSTRACT

Information is written to the line printer giving the user an idea of how the job ran, and the contents of the input and output tapes.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

If errors are detected, the number of errors is printed. Each level of error severity reached is printed.

The number of times each state in the present-state next-state table was encountered is printed. (For more details about the states and counting their frequencies, see the LNDCTL module abstract. For details about input and output lines and channels, see LNDBIL and LNDWRT module abstracts.)

This information is also printed as a two-dimensional table. This is merely an expansion of the one-dimensional printing of states described above. Summing down the columns of the table should produce the same number given as state occurrances. Following this table (as explained in the LNDCTL module abstract) gives a good indication of the structure of each Landsat CCT set. The following page shows the table with valid transitions marked with an X. If a transition occurs where an X does not appear, then reformatting aborts and summary information is printed.

2. Internal Description

The error severity levels are stored as powers of two, thereby requiring only one word in memory to retain the levels. For example, if error levels 2, 4, 4, and 16 were encountered, the variable SEVERTY would contain binary 0...010110. Binary ones occur in the 2(1), 2(2), and 2(4) positions.

3. Input Description

All values printed are gathered from within the Landsat to LARSYS program.

4. Output Description

Summary information is sent to the line printer.

5. Supplemental Information

The LNDCTL module abstract.

6. Program Algorithm

```
SUBROUTINE lndsum;
     (* print job summary information *)
BEGIN
     IF no errors detected THEN            (* summarize errors *)
         PRINT 'NO ERRORS DETECTED'
     ELSE
         PRINT number of errors detected;
         PRINT severity levels of errors detected
     ENDIF:
     PRINT count;                  (*occurrances of each input state *)
     PRINT number of records, EOO's, lines, channels;
     PRINT freq;                      (* occurrances of  transitions between
states *)
     PRINT  psns;                     (*  legal transitions  defined by  the
table *)
END
```

MODULE IDENTIFICATION

Module Name: LNDSUP Function Name: LANDSAT

Purpose: Supervisor routine for Landsat-EDIPS to LARSYS Reformatting

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman/Charles R. Smith Date: 12/09/79

Latest Revisor: Date:

LANDSAT FUNCTION ABSTRACT

The function LANDSAT reformats tapes received from the EROS Data Center that conform to the "Manual on Characteristics of Landsat Computer Compatible Tapes"1 into LARSYS Version 3.0 tape format. The EDIPS band interleaved by line (BIL) format conversion has been fully implemented. Single channel band sequential (BSQ) reformatting is also implemented. The reformatting process is controlled directly from information obtained from the input data tape.

A more complete description of the reformatting process and of the EDIPS-format input tape can be found in the LNDCTL module abstract. A complete description of the control card input is documented with the LNDRDR module abstract.

For proper setup and execution under CMS370, the following EXEC (control) routine is recommended:

```
&CONTOL OFF
GLOBAL TXTLIB LNDLIBH REFRMLBH FORTMOD2 CMSLIB FORTRAN REFRMLIB
FILEDEF 5 READER
FILEDEF 6 PRINTER (RECFM FA
FILEDEF 7 PUNCH
FILEDEF 11 TAP1
FILEDEF 12 TAP2
FILEDEF 13 TAP3
FILEDEF 16 TERMINAL
LOAD BLKDAT (CLEAR
INCLUDE LNDSUP TAPOP CTLWRD BCDVAL
START
&EXIT
```

To use the above EXEC, control cards must be placed in the user's virtual reader (unit 5). A sample set of control cards to run the function follows:

---

1. Produced by the EROS Data Center Digital Image System, compiled by Patrick F. Holkenbrink, and revised December 1978.

```
*LANDSAT
INPUT FROM TAPES (2341,2342), DENSITY (1600)
OUTPUT TO LARSYSRUN (79001200)
END
```

As noted previously, a complete description of the control card input is documented with the LNDRDR module abstract.

MODULE ABSTRACT

LNDSUP is the main routine that oversees the operation of converting Landsat-EDIPS formatted tapes to LARSYS version 3.0 formatted tapes.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Control cards that describe the job to be performed are read from logical file unit 5. Landsat-EDIPS tape(s) are reformatted into a properly initialized tape in LARSYS format. Terminal and line printer output describes the process results.

2. Internal Description

```
PROGRAM lndsup
          (* Landsat to LARSYS program *)


       DECLARATIONS
         control-cards : file;
         end-of-file   : file-condition;
         error-level, abort  : error-level-indicator;
         start, stop   : time;
         dummy, clock, totcpu, vircpu  : computer-usage;


BEGIN
      CALL cptime(dummy, clock, totcpu, vircpu);
      CALL getime(start);
      REPEAT
          CALL lndint;  (* initialize *)
          CALL lndrdr;  (* read control card file *) IF NOT(end-of-file ON
          control-cards) THEN
            IF error-level < abort THEN
               CALL lndctl;  (* reformat Landsat *)
               CALL lndwup  (* wrap-up reformatting *)
            ENDIF;
            CALL lndsum  (* summarize the job *)
          ENDIF
      UNTIL end-of-file ON control-cards OR
                error-level >= abort;
      CALL usage(clock, totcpu,vircpu);  (* print computer time used *)
      CALL getime(stop);
      CALL lndf17  (* print FORM-17 *)
END
```

5. SUPPLEMENTAL INFORMATION

Figure 1 shows how the routines of the Landsat-EDIPS to LARSYS PROGRAM ARE GROUPED. Figures 2 through 7 take the groups of routines and names the most important individual routines. The routines and what they do are shown in Figure 8.

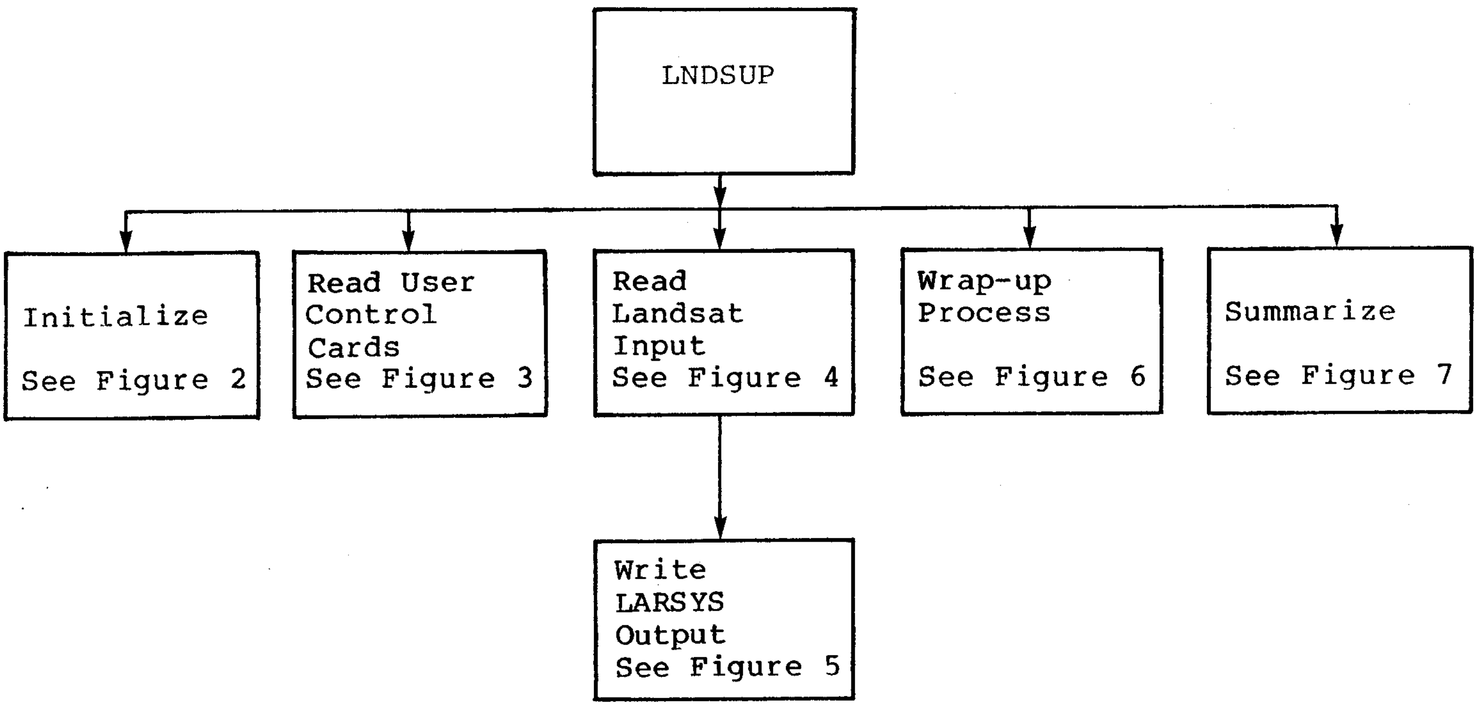


Figure 1. The Landsat-EDIPS to LARSYS Program Structure

Subroutine CPTIME is called to get the present time, the present total CPU usage, and the present virtual CPU usage. These values will be used again with the subroutine USAGE. The first argument of CPTIME (DUMMY) is ignored.

The start time of the process is gotten from GETIME. All global constants are defined and initialized in subroutine LNDINT.

For every control card job, the REPEAT-UNTIL loop will be executed once. LNDINT initializes global constants and variables for every control card job. If the error level ever gets equal to or greater than ABORT, then the proper error measures will be taken and jobs following in the card deck will not be processed (execution of the REPEAT-UNTIL loop will cease).

To read one control card set of the deck, LNDRDR is called. When LNDRDR reads an END card (EOFRDR=false) or an end-or file (EOFRDR=true), reading is suspended and LNDRDR exits.

LNDCTL is the heart of the Landsat-EDIPS to LARSYS process. It reads the Landsat-EDIPS tape, and from the contents of the tape, determines how to process the data.

After LNDCTL exits, LNDWUP cleans up any loose ends. An execution summary is printed by LNDSUM, and if no severe errors were detected, the loop is executed again.

USAGE prints the connect time for the job(s), the total CPU time for the job(s), and the virtual CPU time for the job(s). The stop time is printed and FORM-17´s are printed for the final job.

3. Input Description

None

4. Output Description

A few messages are sent to the line printer and user´s terminal.

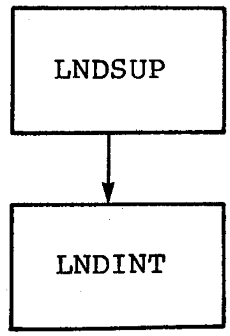


Figure 2. The Landsat-EDIPS to LARSYS Initialization

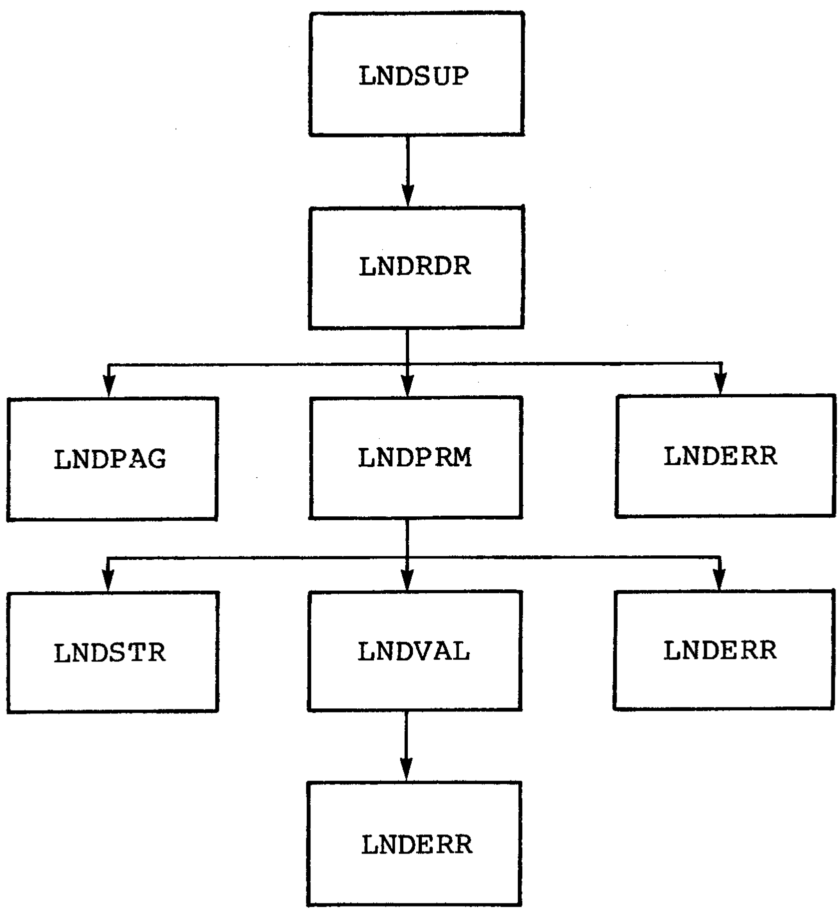


Figure 3. The Landsat-EDIPS to LARSYS Processor Control

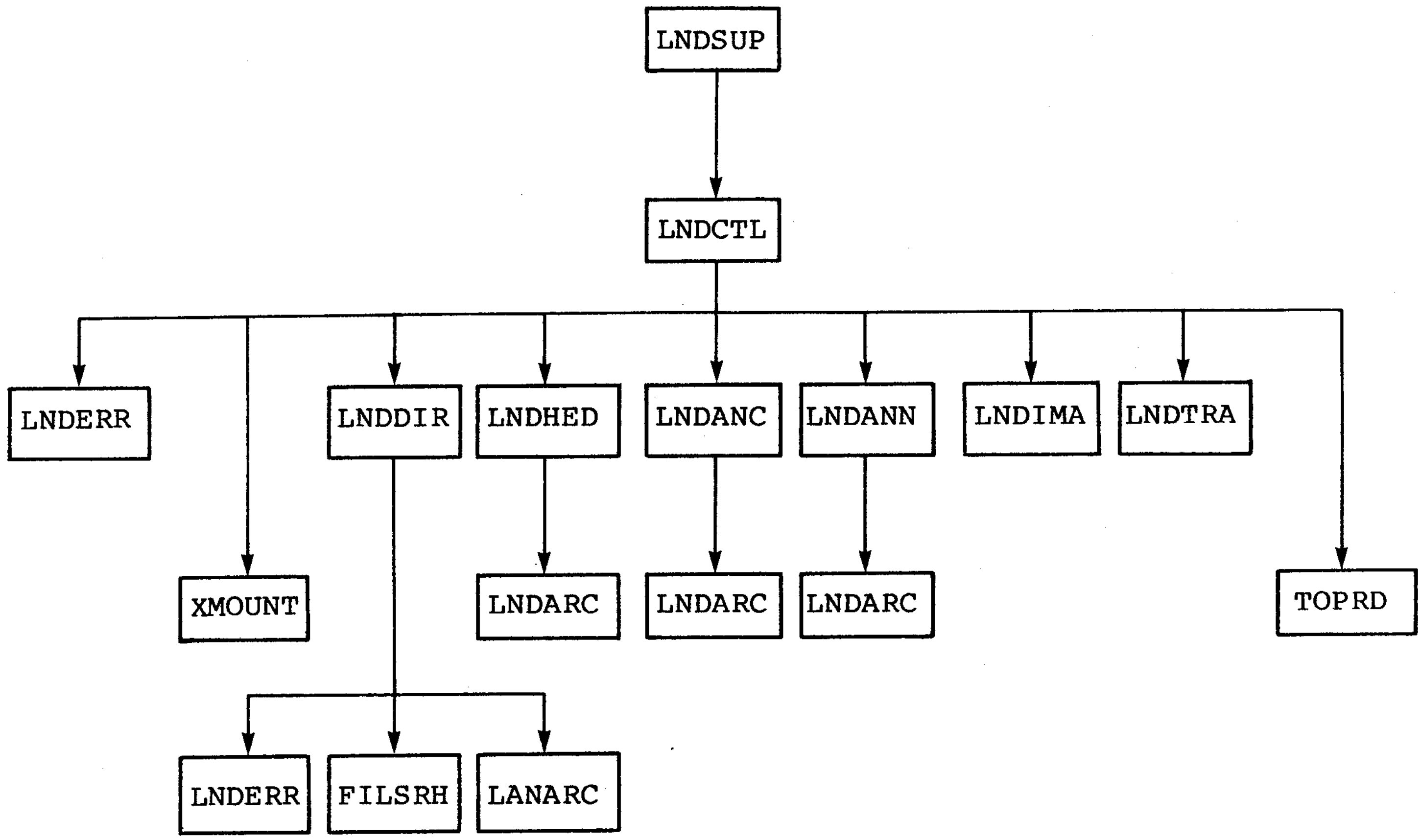


Figure 4. The Landsat-EDIPS to LARSYS Input Processing

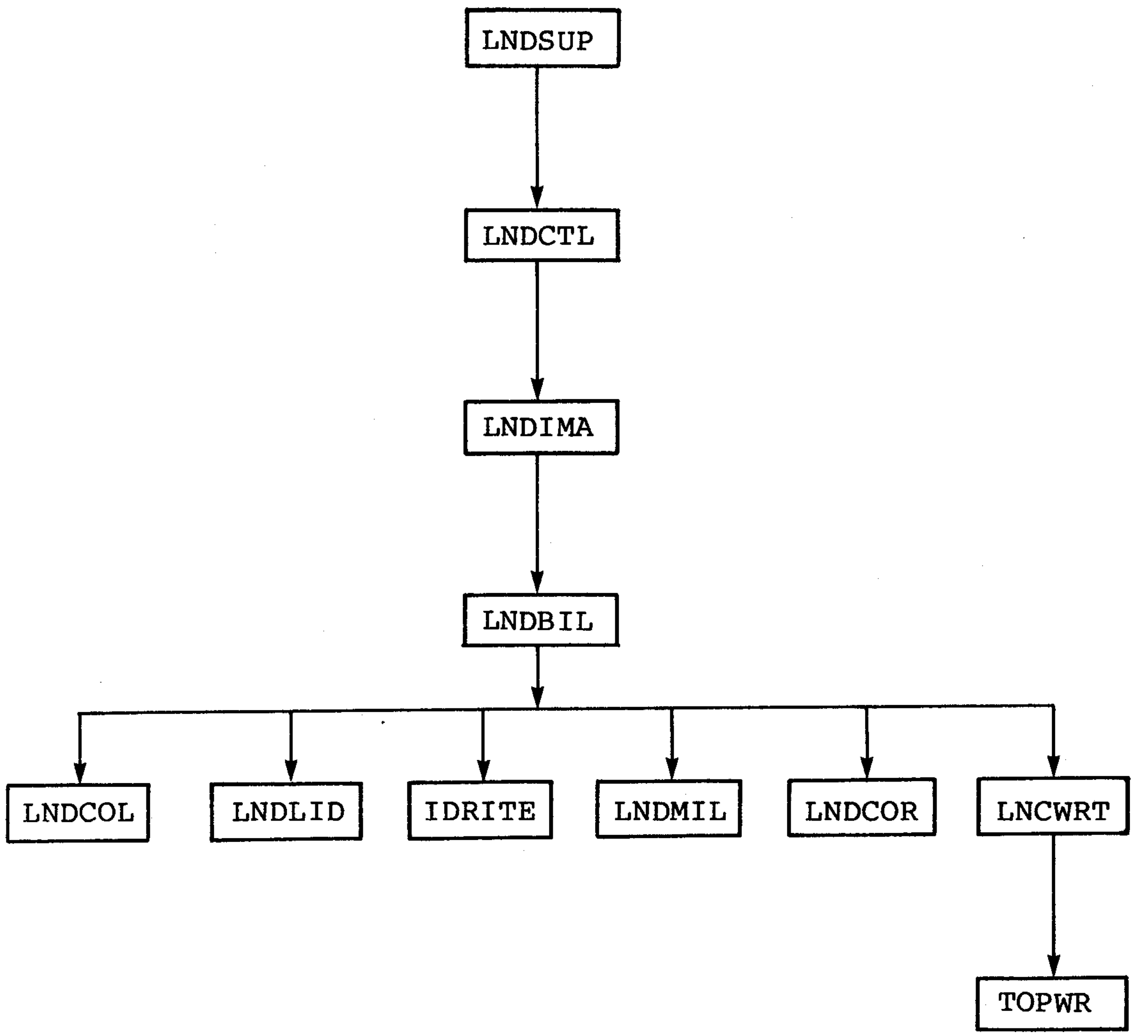


Figure 5. The LANDSAT-EDIPS to LARSYS Output Processing

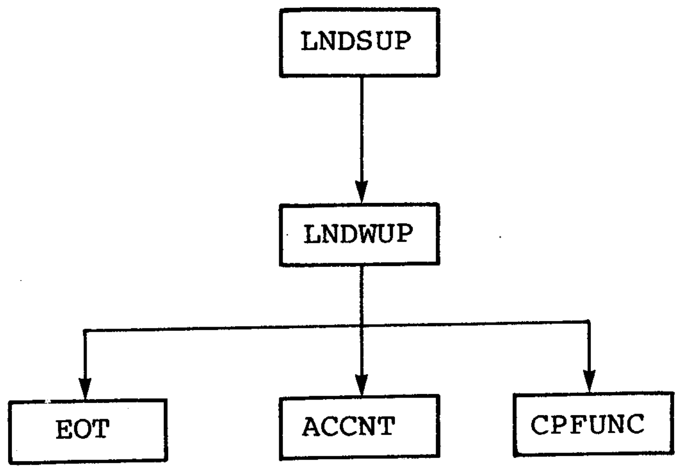


Figure 6. The Landsat-EDIPS to LARSYS Process Wrap-up.

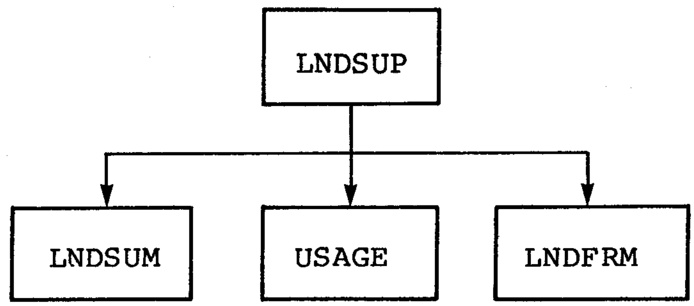


Figure 7. The Landsat-EDIPS to LARSYS Process Summary

Routines Used for Reformatting Landsat into LARSYS

ROUTINE	PURPOSE
BINSRH	Perform a binary Search.
CASCII	Convert ASCII characters (bytes) to EBCDIC characters
CHAR	Assign a literal character string
CONDMP	Dump contents of an array with additional info
*CPTIME	Get connect time, virtual and total CPU time usage
CTLCBC	Insert commas in the control card so CTLWRD will work
*CTLPRM	(Entry point of CTLWRD) read parameters from control cards
CTLSPN	Span over columns (also so CTLWRD will work)
CTOIA1	Convert character (in A1 format) to integer
*DARSIN	FORTRAN double precision arc sine
*DATAN	FORTRAN double precision arc tangent
*DCOS	FORTRAN double precision cosine
*EOT	Write end-of-tape and more
ERRPTR	Print a dollar sign beneath the error in the control card
FILSRH	Find the first occurance of an empty file on a tape
*GETIME	Get time of day
*GTDATE	Get date
*GTUNIT	(Entry point of TAPOP) Find the device address of a tape
*IDRITE	Check LARSYS header record and more
*IVAL	(Entry point of BCDVAL) read values on control cards
JTOR	Convert a Julian data to Roman (i.e. month, day, year)
LNDANC	Process the Landsat ancillary record
LNDANN	Process the Landsat annotation record
LNDARC	Write to archive tape
LNDBIL	Output process for BIL format
LNDCOL	Check the columns specified
LNDCOR	Perform line corrections
LNDCTL	Control according to the Landsat input tape structure
LNDDIR	Process the Landsat directory record
LNDERR	Records errors encountered
LNDF17	Generate run completion forms (eg. FORM17S)
LNDHED	Process the Landsat header record
LNDIMA	Process the Landsat image record
LNDINT	Initialize values
LNDLID	Create LARSYS ID record from Landsat-EDIPS dir & hed records
LNDMIL	Merge and retain BIL bands for a line
LNDPAG	Print a header at the top of the output page
LNDPRM	Read control parameters on control cards
LNDRDR	Control card reader
LNDSTR	Get a delimited string from an array in A4 format
LNDSUM	Prints info for the job most recently run
LNDSUP	Landsat reformatting supervisor
LNDTRA	Process the Landsat trailer record
LNDVAL	Read control parameter values on control cards
LNDWRT	Write line image in LARSYS format
LNDWUP	Wrap-up Landsat reformatting
*MOUNT	Mount a tape on a tape drive
*MOVBYT	Copy bytes
*RINGIN	(Entry point of TAPOP) check the write protect ring

Routines Used for Reformatting Landsat into LARSYS (continued)

```
*TOPEF============= (Entry point of TAPOP) write an end-of-file to tape
*TOPFF============= (Entry point of TAPOP) forward file a tape
*TOPRD============= (Entry point of TAPOP) read a record from tape
*TOPRU============= (Entry point of TAPOP) rewind and unload a tape
*TOPWR============= (Entry point of TAPOP) write a record to tape
 USAGE============= Print job info such as connect, virtual, and total CP time
 XMOUNT============ Call MOUNT with additional information
```

* No abstract available in this document

Figure 8. The Landsat-EDIPS to LARSYS Routines, Support Routines and their purpose.

MODULE IDENTIFICATION

Module Name: LNDTRA Function Name: Landsat

Purpose: Process Landsat Trailer records

System/Language: Data Reformatting/IBM CNM370 FORTRAN-IV

Author: Ken Dickman Date: 02/03/80

Latest Revisor: Date:

MODULE ABSTRACT

A dummy routine was created so if any processing of the trailer record was needed, it would be in LNDTRA.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Not applicable

2. Internal Description

Not applicable

3. Input Description

Not applicable

4. Output Description

Not applicable

5. Supplemental Information

See the "Manual on Characteristics of Landsat Computer-Compatible Tapes" produced by the EROS Data Center Digital Image Processing System, page 52.

MODULE IDENTIFICATION

Module Name: LNDVAL Function Name: Landsat

Purpose: Process integer values associated with control parameters

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 2/05/80

Latest Revisor: Date:

MODULE ABSTRACT

A vector of integer values, enclosed by parenthesis, separated by commas, is expected after certain control parameters. LNDVAL extracts these integers from the control card line and assigns the values to the appropriate variables in common blocks.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

Entry point IVAL of module BCDVAL is called to extract the vector of integers.

2. Internal Description

A computed-goto statement transfers execution to the proper code for the control parameter.

3. Input Description

The control card is received from LNDPRM in the array LN20A4 with COL pointing just before the integer vector.

4. Output Description

If an error is detected, messages are written to the line printer and the user's terminal.

5. Supplemental Information

None

6. Program Algorithm

```
SUBROUTINE lndval;
     (* process integer values in the Landsat to LARSYS control cards *)
BEGIN
     extract a vector of integer values;
     CASE control-parameter OF
          'BANDS'     :  set channel flags;
          'BPI'       :  set density;
          'CHANNELS'  :  set channel flags;
          'COLUMNS'   :  set columns;
              .
              .
              .
          'TAPE'      :  set tape numbers
     ENDCASE
END
```

MODULE IDENTIFICATION

Module Name: LNDWRT Function Name: Landsat

Purpose: Output the LARSYS data records to tape

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 01/19/80

Latest Revisor: Date:

MODULE ABSTRACT

Add the final details to the LARSYS data record and write the LARSYS data record to the output tape. If a write error is encountered, corresponding error messages are printed.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

The data record is passed VIA the buffer OUTREC which represents one line of data. The first four bytes of the data record are set to include the line number and a dummy roll parameter.

2. Internal Description

The LARSYS record count (bytes 1-2) is initialized in subroutine LNDINT and is incremented to 1 for the first data record, 2 for the second data record, 3 for the third data record, and so on. The LARSYS roll parameter (bytes 3-4) is set to hexadecimal 7FFF.

3. Output Description

The LARSYS data record (in final form) is passed VIA array OUTREC to the subroutine TOPWR to be written to the output tape. This final form conforms to the LARSYS version 3.0 tape format. If an error occurs while writing to tape, error messages are written to the line printer and the user's terminal.

4. Supplemental Information

For a description of tape routines and tape I/O error messages, see the TAPOP module abstract.

5. Program Algorithm

```
SUBROUTINES lndwrt;
     (* write the LARSYS data record to the output tape *)
BEGIN
     LARSYS-record-count := LARSYS-record-count +1
     pack the LARSYS-record-count into the first half-word;
     pack the LARSYS-roll-parameter into the second half-word;
     output the LARSYS data record;
     IF an error occurred writing to output tape THEN
             print error message
     ENDIF
END
```

MODULE IDENTIFICATION

Module Name: LNDWUP Function Name: Landsat

Purpose: Wrap-up the Landsat to LARSYS reformatting process.

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken DickmanData: 02/03/80

Latest Revisor: Date:

MODULE ABSTRACT

Add the finishing touches to the Landsat-EDIPS to LARSYS reformatting process. After the execution of LNDWUP, the output tape will be in the proper LARSYS format.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

The final LARSYS record count (from the first half-word of each LARSYS data record-see LNDWRT module abstract) is compared with word 20 of the LARSYS header record for consistency. Two end-of-file marks (end-of-tape) are written to the output tape. Both Landsat and LARSYS tape drives are detached from the user's virtual machine. Internal accounting procedures and tape management are accomplished with calls to ACCNT and EOT.

2. Internal Description

To compare RECCNT (Integer *2) and LARSID (20) (Integer *4), the intermediate variable I4WORD (Integer *4) is used. Subroutine EOT keeps track of the output tape usage.

3. Input Description

Not applicable

4. Output Description

If word 20 of the LARSYS header doesn't agree with the number of lines written, an error message is printed. Two end-of-file marks are written to the LARSYS tape with TOPEF and EOT.

5. Supplemental Information

None

6. Program Algorithm

```
SUBROUTINE lndwup;
     (* wrap-up Landsat to LARSYS processing *)
BEGIN
     IF the number of LARSYS data records written =
         the number of image lines requested THEN
             print error message
     ENDIF:
     write end-of-tape mark;
     perform accounting;
     detach the tape drives
END
```

MODULE IDENTIFICATION

Module Name: STDHDR Function Name: Reformatting Support

Purpose: Print a standard header at top of page

System/Language: VM370 FORTRAN-IV

Author: Ken Dickman Date: 7/22/81

Latest Revisor: Date:

MODULE ABSTRACT

STDHDR prints a standard header banner at the top of a printout page folowed by one line of user supplied header information.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

CALL STDHDR(NCHAR, STRING)

Input Arguments

NCHAR - INTEGER*4. The number of characters in the user supplied header information.

STRING - LOGICAL*1. Array dimensioned to NCHAR elements. Contains the user supplied header information in character format.

Output arguments

None

2. Internal Description

STDHDR centers the character string (STRING) supplied by the calling routine. It then collects the time of day, date, user ID, and user name from various subroutines. The information is printed out on Fortran unit 6 as follows:

```
            (top of page)
USER ID     LABORATORY FOR APPLICATION OF REMOTE SENSING    Date
USER NAME                PURDUE UNIVERSITY                  Time
            (3 blank lines)
            User supplied character string
            (2 blank lines)
```

3. Subroutines Used

CPTIME
IDNAME
GTDATE
GETIME (entry in GTDATE)

MODULE IDENTIFICATION

Module Name: USAGE Function Name: General Utility

Purpose: Write the connect, total CPU, and virtual CPU time

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 3/05/80

Latest Revisor: Date:

MODULE ABSTRACT

Three lines are written to the line printer informing the user of the":

1. Connect time - time read from any ordinary wall clock;
2. Total CPU time - computer central processing unit usage including virtual cpu time and system overhead; and
3. Virtual CPU time - Computer central processing unit usage for only working on the user's job (no system overhead included);

since the subroutine CPTIME was called.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. CALL USAGE(CLOCK, TOTCPU, VIRCPU)

Input Arguments:

CLOCK - I*2. A nine element array for storing the connect time.

TOTCPU - I*4. Two element array for storing the total CPU time.

VIRCPU - I*4. Two element array for storing the virtual CPU.

Output Arguments:

None

The following presents additional information about the arguments passed to USAGE and the local variables REALCP and VIRCP.

		START TIME	STOP TIME	DIFFERENCE (STOP-START)
connect	hours	CLOCK(1)	CLOCK(4)	CLOCK(7)
	minutes	CLOCK(2)	CLOCK(5)	CLOCK(8)
	seconds	CLOCK(3)	CLOCK(6)	CLOCK(9)
total	seconds	TOPCPU(1)	TOTCPU(2)	REALCP
virtual	seconds	VIRCPU(1)	VIRCPU(2)	VIRTCP

Below is code that measures the connect, total, and virtual time between the call of CPTIME (start time) and the call of USAGE (stop time). USAGE then calculates the difference between the start and stop time, and prints the information

```
      IMPLICIT INTEGER*4 (A-Z)
      INTEGER*4 DUMMY(2), TOTCPU(2), VIRCPU(2)
      INTEGER*2 CLOCK(9)
C
      CALL CPTIME(DUMMY, CLOCK, TOTCPU, VIRCPU)
      CALL USAGE(CLOCK, TOTCPU, VIRCPU)
      STOP
      END
```

2. Internal Description

The variable DUMMY is not used in USAGE. Several calculations are done to adjust for the number of seconds in a minute, minutes in an hour, hours in a day.

3. Input Description

Not Applicable

4. Output Description

Information is written to the line printer about connect, total CPU, and virtual CPU time.

5. Supplemental Information

The code for USAGE was taken from the LARS UNIFORM reformatting program.

6. Program Algorithm

Not Applicable

MODULE IDENTIFICATION

Module Name: XMOUNT Function Name: General Utility

Purpose: Mount a tape with additional information for the user.

System/Language: Data Reformatting/IBM CMS370 FORTRAN-IV

Author: Ken Dickman Date: 2/29/80

Latest Revisor: Date:

MODULE ABSTRACT

XMOUNT prints the current time of day and the parameters passed to it, then MOUNT is called to ask for a tape to be put on a tape drive. After MOUNT returns control to XMOUNT, the current time of day is again printed along with verification data about the mount.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

The arguments passed to XMOUNT are the same arguments for MOUNT.

CALL XMOUNT(TAPE, UNIT, RING, DENSITY)

Input Arguments:

TAPE - I*4. The tape number of the tape to be mounted.

UNIT - I*4. The tape drive unit number where units 11 through 15 are for virtual addresses 181 through 185.

RING - I*4. Either the characters 'RI' (protection ring in) for read/write access or the characters 'RO' (protection ring out) for read only access.

Density - I*4. The bytes per inch to be read from the tape by the tape drive. '800' requests a tape drive that reads 800 bytes per inch and 1600 bytes per inch.

MOUNT does not write any messages to the line printer, but XMOUNT does. When the user spools his/her console, the message sent to the user's terminal is not recorded, but XMOUNT's message is. There is no way to record the mounting of a tape using MOUNT unless something like XMOUNT is used or the user's terminal is a "hardcopy" console. XMOUNT's additional information helps determine tape drive trouble and helps determine time spent mounting the tape.

2. Internal Descrption

Subroutine MOUNT handles the tape mounting operation. Entry points RINGIN and GTUNIT of module TAPOP verify that the tape was mounted properly.

3. Input Description

Not Applicable.

4. Output Description

An informative message is written to the line printer and user's terminal before and after MOUNT is called.

5. Supplemental Information

Refer to the CHAR, GETIME, MOUNT, and TAPOP module abstracts for subroutine information.

6. Program Algorithm

Not applicable.

SECTION 3

Example Outputs

Section 3 contains example output from the Landsat and Geometric Correction processors. The printer pages are an example of the output sent to Fortran Unit 6. The grayscale plots are not part of the processors; they are included here only as verification of the images written to tape.

Landsat Processor

```
BATONITE                                  LABORATORY FOR APPLICATIONS OF REMOTE SENSING                     NOV  25,1980
GOOD 305 702                                              PURDUE UNIVERSITY                                  03 41 04  AM

LANDSAT REFORMATTING VERSION    1.03    12/08/79

CONNECT TIME   =      0 HOURS   0 MINUTES   1 SECONDS
TOTAL    CPU   =        0.080   SECONDS
VIRTUAL  CPU   =        0.060   SECONDS

                                          *** REFORMATTING LANDSAT-EDIPS TO LARSYS ***

*LANDSAT
INPUT TAPES(5023,5024),BPI(1600)
OUTPUT RUN(79000200)
OUTPUT FLIGHT(648215101        FL)
END


I0002 ALL CONTROL CARDS HAVE BEEN READ     (LNDRDR)
I0001 JOB BEGINS     (LNDCTL)

I0001 AT 03 41 04 AM   TAPE    5023 REQUESTED.
      RING OUT ON UNIT 11 AT 1600BPI      (XMOUNT)
I0002 AT 03 43 15 AM   TAPE    5023 READY.
      WRITE PROTECT = YES, AT DEVICE ADDRESS 181      (XMOUNT)


I0001 LANDSAT DIRECTORY RECORD WILL NOW BE DUMPED
      LINES BEGINNING WITH AN ASTERISK ('*') ARE CRITICAL TO THE OPERATION OF THE REFORMATTING PROGRAM.
      LINES BEGINNING WITH A PLUS ('+') ARE USED BY THE REFORMATTING PROGRAM.     (LNDDIR)

          7-18       TAPE ID . . . . . . . . . . . . . . . . . . . . . . . L2MC?7921701
    *      9         SENSOR TYPE    (MSS OR RBV)  . . . . . . . . . . . . . MSS
    *     10-11      DATA WITH/WITHOUT GEOMETRIC CORRECTIONS . . . . . . . WITH

    *     19         CCT VOLUME NUMBER . . . . . . . . . . . . . . . . . . 1
          20         NUMBER OF VOLUMES IN THE CCT SET  . . . . . . . . . . 2

     DATE OF CCT TAPE GENERATION

          27         DAY . . . . . . . . . . . . . . . . . . . . . . . . . 5
          28         MONTH . . . . . . . . . . . . . . . . . . . . . . . . 8
          29         YEAR  . . . . . . . . . . . . . . . . . . . . . . . . 79

    *     31         INTERLEAVING TYPE INDICATOR    (BSQ OR BIL)  . . . . BIL
    +     35-44      SCENE ID  . . . . . . . . . . . . . . . . . . . . . . 2148?15101
          45-52      WRS DESIGNATOR (ASCEND/DESCEND,NOM PATH,NOM ROW)       0018039

         359         EDIPS SOFTWARE PROCESSING VERSION . . . . . . . . . . 0
         360         EDIPS DOCUMENT VERSION  . . . . . . . . . . . . . . . 0
```

```
BATONITE                              LABORATORY FOR APPLICATIONS OF REMOTE SENSING                    NOV 25,1980
0000 305 702                                        PURDUE UNIVERSITY                                  03 41 04 AM

LANDSAT REFORMATTING VERSION   1.03   12/08/79

CONNECT TIME  =    0 HOURS  2 MINUTES 14 SECONDS
TOTAL   CPU   =     0.450  SECONDS
VIRTUAL CPU   =     0.220  SECONDS

                               *** REFORMATTING LANDSAT-EDIPS TO LARSYS ***

I0001 DUMP OF LANDSAT HEADER RECORD VALUES PRESENTLY USED    (LNDHED)

        7-18    IMAGE ID . . . . . . . . . . . . . . . . . . . . . .  214
  +    19-26    WRS DESIGNATOR (ASCEND/DESCEND,NOM PATH,NOM ROW)   D013039
  *    37-39    SENSOR TYPE . . . . . . . . . . . . . . . . . . . . MSS
       49-52    ACTIVE CAMERA STATUS   (IN HEXADECIMAL) . . . . . . FFFFFF00  00000000

       58-59    PIXELS/SCAN LINE (ORIGINAL GEO UNCORRECTED IMAGE) 3247
  +    77-92    TIME OF EXPOSURE (5-JULIAN,2-HR,2MIN,2-SEC,3-MS)  79043151013704

  *   107       GEOMETRIC CORRECTIONS APPLIED . . . . . . . . . . . YES
      108       GEOMETRIC CORRECTION DATA PRESENT . . . . . . . . . NO
      109       RADIOMETRIC CORRECTIONS APPLIED . . . . . . . . . . YES
      110       RADIOMETRIC CORRECTION DATA PRESENT . . . . . . . . NO

      117       IMAGE DATA FORMAT . . . . . . . . . . . . . . . . . RECTANGULAR
  *   120       INTERLEAVING TYPE INDICATOR . . . . . . . . . . . . BIL
  +   123       RESAMPLING APPLIED  . . . . . . . . . . . . . . . . CUBIC
      124       MAP PROJECTION APPLIED  . . . . . . . . . . . . . . ????

      131-132   NUMBER OF PIXELS PER SCAN LINE  . . . . . . . . . . 3548
      162       CAL WEDGE MODE  . . . . . . . . . . . . . . . . . . LOW   COMPRES
      163-164   SCENE ID  . . . . . . . . . . . . . . . . . . . . .

               FOR BYTES 183 THROUGH 214...
               (PRINTED AS HALFWORDS)
                                      CURRENT IMAGE              REFERENCE IMAGE
                                      -------------              ---------------
TEMPORAL REGISTRATION          SCAN LINE      PIXEL        SCAN LINE        PIXEL
        POINT                  NUMBER         NUMBER       NUMBER           NUMBER

         #1                       0              0            0                0
         #2                       0              0            0                0
         #3                       0              0            0                0
         #4                       0              0            0                0

      232       QUALITY ASSESSMENT    (0=LOWEST,9=HIGHEST) . . . . 3

     3582       EDIPS PERFORMED CONTRAST ENHANCEMENT  . . . . . . . FALSE
     3583       EDIPS PERFORMED ATMOSPHERE SCATTER COMPENSATION . FALSE
     3584       EDIPS PERFORMED EDGE ENHANCEMENT  . . . . . . . . . FALSE


I0005 BIL SCENE   1     RELATIVE TO THE FIRST TAPE REFORMATTED     (LNDCTL)
I0001 DUMP OF LANDSAT ANNOTATION RECORD VALUES PRESENTLY USED      (LNDANN)

  +    15-30    LATITUDE AND LONGITUDE AT CENTER OF IMAGE . . . . C N30-11/W082-19
  +    41-56    NOMINAL LATITUDE AND NOMINAL LONGITUDE  . . . . . . N N30-12/W082-21

  +    68-81    SUN ANGLE AND SUN AZIMUTH . . . . . . . . . . . . . SUN EL32 A134
  +    82       TYPE OF CORRECTION APPLIED  . . . . . . . . . . . . GEOMETRIC GCP'S
  +    83       SCALE OF IMAGE  . . . . . . . . . . . . . . . . . . 185*170 KM
  +    84       PROJECTION  . . . . . . . . . . . . . . . . . . . . HOTINE OBLIQUE MERCATOR
  +    86       RESAMPLING ALGORITHM  . . . . . . . . . . . . . . . CUBIC
  +    87       TYPE EPHEMERIS DATA USED TO COMPUTE IMAGE CENTER  PREDICTIVE


I0001   4 CHANNEL(S) TO BE REFORMATTED     (LNDCOL)
                      (CHANNEL  1)
                      (CHANNEL  2)
                      (CHANNEL  3)
                      (CHANNEL  4)


I0002 REFORMAT FROM COLUMN   53 TO 3494 BY    1     (LNDCOL)
I0001 LINES     1 TO 2983 BY    1 WILL BE REFORMATTED     (LNDLID)
I0001 CALL TO SUBROUTINE IDRITE     (LNDBIL)


********** AUTOMATIC TAPE SELECTION REQUESTED. **********


   *****RUN WILL TAKE  96 PER CENT OF THE TAPE.          2305.0 FEET

A RUN 79000200 ALREADY EXISTS -- EXECUTION CONTINUES.

I0001 AT 03 43 21 AM   TAPE   3202 REQUESTED,
      RING IN  ON UNIT 12 AT 1600BPI     (XMOUNT)
I0002 AT 03 45 55 AM   TAPE   3202 READY,
       WRITE PROTECT = NO , AT DEVICE ADDRESS 182     (XMOUNT)
```

```
                       LARSYS DATA TAPE ID INFORMATION
                       -------------------------------

  TAPE NUMBER............... 3202          FILE NUMBER...............   1
  RUN NUMBER........... 79000200           CONTINUATION CODE.......... 0
  NUMBER OF DATA CHANNELS....  4           NUMBER OF DATA SAMPLES... 3448
  FLIGHT LINE.. 648215101      FL          DATE DATA TAKEN......  2/12/79
  TIME DATA TAKEN.... 1510 HOURS           SENSOR ALTITUDE.. 3062000 FEET
  GROUND HEADING.... 189 DEGREES           REFORMATING DATE. NOV  25,1980
  LINES OF DATA............. 2983          RUN CENTER... 30.18 N/ 82.32 W


                    SPECTRAL                                  CALIBRATION
                      BAND                                    PULSE VALUES

  CHANNEL         LOWER        UPPER                  C0          C1          C2

     1            0.50         0.60                   0.0         2.48        0.0
     2            0.60         0.70                   0.0         2.00        0.0
     3            0.70         0.80                   0.0         1.76        0.0
     4            0.80         1.10                   0.0         2.30        0.0
I0002  100 OF 2983 INPUT LINES READ.    100 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  200 OF 2983 INPUT LINES READ.    200 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  300 OF 2983 INPUT LINES READ.    300 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  400 OF 2983 INPUT LINES READ.    400 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  500 OF 2983 INPUT LINES READ.    500 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  600 OF 2983 INPUT LINES READ.    600 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  700 OF 2983 INPUT LINES READ.    700 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  800 OF 2983 INPUT LINES READ.    800 OUTPUT LINES WRITTEN     (LNDBIL)
I0002  900 OF 2983 INPUT LINES READ.    900 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1000 OF 2983 INPUT LINES READ.   1000 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1100 OF 2983 INPUT LINES READ.   1100 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1200 OF 2983 INPUT LINES READ.   1200 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1300 OF 2983 INPUT LINES READ.   1300 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1400 OF 2983 INPUT LINES READ.   1400 OUTPUT LINES WRITTEN     (LNDBIL)
I0005 ALL   1491 INPUT LINES. UP TO VOLUME 1. TAPE   5023. HAVE BEEN READ.
      READING OF INPUT CONTINUES ON VOLUME 2 OF      2     (LNDCTL)

I0001 AT 03 56 04 AM   TAPE    5024 REQUESTED.
      RING OUT ON UNIT 11 AT 1600BPI      (XMOUNT)
I0002 AT 04 04 52 AM   TAPE    5024 READY.
       WRITE PROTECT = YES. AT DEVICE ADDRESS 181      (XMOUNT)


I0001 LANDSAT DIRECTORY RECORD WILL NOW BE DUMPED
      LINES BEGINNING WITH AN ASTERISK ('*') ARE CRITICAL TO THE OPERATION OF THE REFORMATTING PROGRAM.
      LINES BEGINNING WITH A PLUS ('+') ARE USED BY THE REFORMATTING PROGRAM.     (LNDDIR)

         7-18      TAPE ID . . . . . . . . . . . . . . . . . . . . . . . . L2MC27921701
   *      9        SENSOR TYPE    (MSS OR PBV)  . . . . . . . . . . . . . MSS
   *     10-11     DATA WITH/WITHOUT GEOMETRIC CORRECTIONS . . . . . . WITH

   *     19        CCT VOLUME NUMBER . . . . . . . . . . . . . . . . . . 2
         20        NUMBER OF VOLUMES IN THE CCT SET  . . . . . . . . . . 2

    DATE OF CCT TAPE GENERATION

         27        DAY . . . . . . . . . . . . . . . . . . . . . . . . . 5
         28        MONTH . . . . . . . . . . . . . . . . . . . . . . . . 8
         29        YEAR  . . . . . . . . . . . . . . . . . . . . . . . . 79

   *     31        INTERLEAVING TYPE INDICATOR    (BSQ OR BIL)   . . . BIL
   +     35-44     SCENE ID  . . . . . . . . . . . . . . . . . . . . . . 2148215101
         45-52     WRS DESIGNATOR (ASCEND/DESCEND,NOM PATH,NOM ROW)   D018039

        359        EDIPS SOFTWARE PROCESSING VERSION . . . . . . . . . . 0
        360        EDIPS DOCUMENT VERSION  . . . . . . . . . . . . . . . 0


I0002 1500 OF 2983 INPUT LINES READ.   1500 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1600 OF 2983 INPUT LINES READ.   1600 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1700 OF 2983 INPUT LINES READ.   1700 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1800 OF 2983 INPUT LINES READ.   1800 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 1900 OF 2983 INPUT LINES READ.   1900 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2000 OF 2983 INPUT LINES READ.   2000 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2100 OF 2983 INPUT LINES READ.   2100 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2200 OF 2983 INPUT LINES READ.   2200 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2300 OF 2983 INPUT LINES READ.   2300 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2400 OF 2983 INPUT LINES READ.   2400 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2500 OF 2983 INPUT LINES READ.   2500 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2600 OF 2983 INPUT LINES READ.   2600 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2700 OF 2983 INPUT LINES READ.   2700 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2800 OF 2983 INPUT LINES READ.   2800 OUTPUT LINES WRITTEN     (LNDBIL)
I0002 2900 OF 2983 INPUT LINES READ.   2900 OUTPUT LINES WRITTEN     (LNDBIL)
I0007 ALL INPUT HAS BEEN READ. JOB EXECUTION WRAP-UP BEGINS     (LNDCTL)

  EOT CALLED TO END THE TAPE.

TAPE FILE UPDATED AT EOT.


I0002 LANDSAT TO LARSYS COMPLETE     (LNDSUP)
```

```
BATONITE                         LABORATORY FOR APPLICATIONS OF REMOTE SENSING                  NOV  25,1980
5000 305 702                                   PURDUE UNIVERSITY                                  03 41 04 AM

LANDSAT REFORMATTING VERSION    1.03    12/08/79

CONNECT TIME   =      0 HOURS 34 MINUTES 49 SECONDS
TOTAL    CPU   =    201.540   SECONDS
VIRTUAL  CPU   =    172.670   SECONDS

                             *** REFORMATTING LANDSAT-EDIPS TO LARSYS ***


T0001 BEGIN SUMMARY     (LNDSUM)


      NO ERRORS DETECTED



      THE FOLLOWING IS A COUNT OF THE NUMBER OF TIMES EACH STATE WAS ENCOUNTERED

             STATE             COUNT
             -----             -----

         1    BEG                  1
         2    MMT                  2
         3    DIR                  2
         4    EOF1                 2
         5    HED                  1
         6    ANC                  0
         7    ANN                  1
         8    EOF2                 1
         9    IMA              11932
        10    EOF3                 2
        11    EOF4                 1
        12    TRA                  4
        13    EOF5                 1
        14    EOF6                 1
        15    EOF7                 1
        16    END                  1

      11940 TOTAL  INPUT  RECORDS READ
          9 TOTAL  INPUT  END-OF-FILES READ
       2983 TOTAL  INPUT  IMAGE LINES READ
       2983 TOTAL  WRITES (NOT  INCLUDING  ID HEADERS)
       2983 ESTIMATED  TOTAL  OUTPUT LINES (NOT INCLUDING THE ID HEADERS)
       2983 ACTUAL  TOTAL  OUTPUT LINES (NOT INCLUDING THE ID HEADER)

          4 TOTAL  INPUT CHANNELS
          4 ESTIMATED  TOTAL  OUTPUT  CHANNELS
          4 ACTUAL  TOTAL  OUTPUT CHANNELS

            EXECUTION HALTED AT...
               INPUT LINE       2983
               INPUT CHANNEL       4
               OUTPUT LINE      2982

      LIGHT BOARDER PIXELS (7F HEX) CONVERTED TO DARK (00 HEX)...
       3449 TOTAL  PIXELS PER CHANNEL PER LINE
               (INCLUDING 6 CALIBRATION BYTES AT THE END OF EACH CHANNEL)
        195 MAXIMUM LEFT   BOARDER PIXELS
         12 MINIMUM LEFT   BOARDER PIXELS
        201 MAXIMUM RIGHT  BOARDER PIXELS
         12 MINIMUM RIGHT  BOARDER PIXELS
```

BATONITE
GOOD 305 702

LABORATORY FOR APPLICATIONS OF REMOTE SENSING
PURDUE UNIVERSITY

NOV 25,1980
03 41 04 AM

LANDSAT REFORMATTING VERSION 1.03 12/08/79

CONNECT TIME = 0 HOURS 34 MINUTES 49 SECONDS
TOTAL CPU = 201.730 SECONDS
VIRTUAL CPU = 172.750 SECONDS

*** REFORMATTING LANDSAT-EDIPS TO LARSYS ***

FREQUENCY OF PRESENT STATE - NEXT STATE TRANSITIONS

		REG 1	MNT 2	DIR 3	N EOF1 4	E HED 5	X ANC 6	T ANN 7	EOF2 8	S IMA 9	T EOF3 10	A EOF4 11	T TRA 12	E EOF5 13	EOF6 14	EOF7 15	END 16
	REG	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	MNT 2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
R	DIR 3	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
E	EOF1 4	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
S	HED 5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
E	ANC 6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	ANN 7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
T	EOF2 8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	IMA 9	0	0	0	0	0	0	0	0	11930	2	0	0	0	0	0	0
S	EOF3 10	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
T	EOF4 11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	TRA 12	0	0	0	0	0	0	0	0	0	0	0	3	1	0	0	0
T	EOF5 13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
E	EOF6 14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	EOF7 15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	END 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RATONITE
0000 305 702

LABORATORY FOR APPLICATIONS OF REMOTE SENSING
PURDUE UNIVERSITY

NOV 25,1980
03 41 04 AM

LANDSAT REFORMATTING VERSION 1.03 12/08/79

CONNECT TIME = 0 HOURS 34 MINUTES 49 SECONDS
TOTAL CPU = 201.910 SECONDS
VIRTUAL CPU = 172.840 SECONDS

*** REFORMATTING LANDSAT-EDIPS TO LARSYS ***

LEGAL PRESENT STATE - NEXT STATE TRANSITIONS

NEXT STATE

PRESENT STATE		REG 1	MNT 2	DIR 3	EOF1 4	HED 5	ANC 6	ANN 7	EOF2 8	IMA 9	EOF3 10	EOF4 11	TRA 12	EOF5 13	EOF6 14	EOF7 15	END 16
REG	1		X														
MNT	2			X													
DIR	3				X												
EOF1	4					X				X							
HED	5						X	X	X								
ANC	6						X	X	X								
ANN	7						X	X	X								
EOF2	8									X							
IMA	9									X	X						
EOF3	10											X	X				
EOF4	11		X														
TRA	12												X	X			
EOF5	13					X									X		
EOF6	14															X	
EOF7	15																X
END	16																

T0003 END SUMMARY (LNDSUM)

```
BATONITE                          LABORATORY FOR APPLICATIONS OF REMOTE SENSING                    NOV  25,1980
GODD 305 702                                    PURDUE UNIVERSITY                                  04 15 52 AM

LANDSAT REFORMATTING VERSION    1.03    12/08/79

CONNECT TIME   =     0 HOURS 34 MINUTES 49 SECONDS
TOTAL   CPU    =   202.150   SECONDS
VIRTUAL CPU    =   172.990   SECONDS

                          *** REFORMATTING LANDSAT-EDIPS TO LARSYS ***


I0003 END-OF-FILE ON CONTROL CARD INPUT     (LNDRDR)

CONNECT TIME   =     0 HOURS 34 MINUTES 50 SECONDS
TOTAL   CPU    =   202.200   SECONDS
VIRTUAL CPU    =   173.010   SECONDS




I0003 EXECUTION STOPS 04 15 53 AM      (LNDSUP)
```

FORM - 17C

DATA STORAGE TAPE FILE

TAPE NUMBER................ 3202 FILE NUMBER.................
RUN NUMBER...............79000200 DATA SAMPLES/CHANNEL/LINE.....3448
FLIGHT LINE.......648215101 FL DATE DATA TAKEN........... 2/12/79
TIME DATA TAKEN......... B= (GMT) TIME DATA TAKEN.......1010 (LOCAL)
PLATFORM ALTITUDE...3062000 (FEET) GROUND HEADING.......189 (DEGREES)
DATE TAPE GENERATED...NOV 25,1980 LINES OF DATA.................2983
LAT. AT FRAME CENTER........ 30.18 LONG. AT FRAME CENTER....... 82.32
LAT. NOMINAL ....N30-12 LONG. NOMINAL..............W082-21
SUN ELEVATION...................32 SUN AZIMUTH....................134
SCENE FRAME ID..........2148215101 WRS DESIGNATOR............ 0018039
SENSOR TYPE....................MSS TIME OF EXPOSURE........7904315101
RESAMPLING APPLIED.........CUBIC CAL WEDGE MODE...........LOW COMPR
LINE LENGTH ADJUST.............YES COMPRESSED DATA.................YES
CALIBRATION....................YES EPHEMERIS DATA TYPE.....PREDICTIVE
SCALE OF IMAGE..........185*170 KM
PROJECTION..........HOTINE OBLIQUE MERCATOR

	SPECTRAL BAND		CALIBRATION PULSE VALUES		
CHANNEL	LOWER	UPPER	C0	C1	C2
1	0.500	0.600	0.0	2.480	0.0
2	0.600	0.700	0.0	2.000	0.0
3	0.700	0.800	0.0	1.750	0.0
4	0.800	1.100	0.0	2.300	0.0

RUN CONDITIONS AND COMMENTS--

WP 305

FORM - 17C

DATA STORAGE TAPE FILE

TAPE NUMBER.................. 3202	FILE NUMBER.................... 1
RUN NUMBER................79000200	DATA SAMPLES/CHANNEL/LINE.....3448
FLIGHT LINE.......648215101 FL	DATE DATA TAKEN........... 2/12/79
TIME DATA TAKEN......... B= (GMT)	TIME DATA TAKEN.......1010 (LOCAL)
PLATFORM ALTITUDE...3062000 (FEET)	GROUND HEADING.......189 (DEGREES)
DATE TAPE GENERATED...NOV 25,1980	LINES OF DATA..................2983
LAT. AT FRAME CENTER........ 30.18	LONG. AT FRAME CENTER....... 82.32
LAT. NOMINAL...............N30-12	LONG. NOMINAL..............W082-21
SUN ELEVATION.....................32	SUN AZIMUTH......................134
SCENE FRAME ID..........2148215101	WRS DESIGNATOR............ D018039
SENSOR TYPE....................MSS	TIME OF EXPOSURE........7904315101
RESAMPLING APPLIED.........CUBIC	CAL WEDGE MODE..........LOW COMPR
LINE LENGTH ADJUST.............YES	COMPRESSED DATA.................YES
CALIBRATION......................YES	EPHEMERIS DATA TYPE.....PREDICTIVE
SCALE OF IMAGE..........185*170 KM	
PROJECTION..........HOTINE OBLIQUE MERCATOR	

CHANNEL	SPECTRAL BAND		CALIBRATION PULSE VALUES		
	LOWER	UPPER	C0	C1	C2
1	0.500	0.600	0.0	2.480	0.0
2	0.600	0.700	0.0	2.000	0.0
3	0.700	0.800	0.0	1.760	0.0
4	0.800	1.100	0.0	2.300	0.0

RUN CONDITIONS AND COMMENTS--

W2 305

FORM - 17C

DATA STORAGE TAPE FILE

TAPE NUMBER.................. 7202	FILE NUMBER.................. 1
RUN NUMBER..................79000200	DATA SAMPLES/CHANNEL/LINE.....3448
FLIGHT LINE.......648215101 FL	DATE DATA TAKEN........... 2/12/79
TIME DATA TAKEN......... 0= (GMT)	TIME DATA TAKEN.......1210 (LOCAL)
PLATFORM ALTITUDE...3062000 (FEET)	GROUND HEADING.......180 (DEGREES)
DATE TAPE GENERATED...NOV 25,1980	LINES OF DATA.................2983
LAT. AT FRAME CENTER........ 30.18	LONG. AT FRAME CENTER....... 82.32
LAT. NOMINAL................N30-12	LONG. NOMINAL...............W092-21
SUN ELEVATION....................32	SUN AZIMUTH.....................134
SCENE FRAME ID..........2148215101	WRS DESIGNATOR........... D018030
SENSOR TYPE....................MSS	TIME OF EXPOSURE.......7904315101
RESAMPLING APPLIED.........CUBIC	CAL WEDGE MODE..........LOW COMOD
LINE LENGTH ADJUST..............YES	COMPRESSED DATA.................YES
CALIBRATION......................YES	EPHEMERIS DATA TYPE.....PREDICTIVE
SCALE OF IMAGE..........185*170 KM	
PROJECTION..........HOTINE OBLIQUE MERCATOR	

CHANNEL	SPECTRAL BAND LOWER	UPPER	CALIBRATION PULSE VALUES C0	C1	C2
1	0.500	0.600	0.0	2.420	0.0
2	0.600	0.700	0.0	2.000	0.0
3	0.700	0.800	0.0	1.760	0.0
4	0.800	1.100	0.0	2.300	0.0

RUN CONDITIONS AND COMMENTS---------------------------------------

WP 305

FORM - 17C

DATA STORAGE TAPE FILE

TAPE NUMBER.................. 3202 FILE NUMBER.................... 1
RUN NUMBER...............79000200 DATA SAMPLES/CHANNEL/LINE.....3448
FLIGHT LINE.......648215101 FL DATE DATA TAKEN........... 2/12/79
TIME DATA TAKEN......... 8= (GMT) TIME DATA TAKEN.......1010 (LOCAL)
PLATFORM ALTITUDE...3062000 (FEET) GROUND HEADING.......189 (DEGREES)
DATE TAPE GENERATED...NOV 25,1980 LINES OF DATA.................2983
LAT. AT FRAME CENTER......... 30.18 LONG. AT FRAME CENTER....... 82.32
LAT. NOMINAL................N30-12 LONG. NOMINAL..............W082-21
SUN ELEVATION...................32 SUN AZIMUTH....................134
SCENE FRAME ID..........2148215101 WRS DESIGNATOR............. D018039
SENSOR TYPE....................MSS TIME OF EXPOSURE........7904315101
RESAMPLING APPLIED.........CUBIC CAL WEDGE MODE..........LOW COMPR
LINE LENGTH ADJUST.............YES COMPRESSED DATA..................YES
CALIBRATION....................YES EPHEMERIS DATA TYPE.....PREDICTIVE
SCALE OF IMAGE..........185*170 KM
PROJECTION..........HOTINE OBLIQUE MERCATOR

CHANNEL	SPECTRAL BAND LOWER	UPPER	CALIBRATION PULSE VALUES C0	C1	C2
1	0.500	0.600	0.0	2.480	0.0
2	0.600	0.700	0.0	2.000	0.0
3	0.700	0.800	0.0	1.760	0.0
4	0.800	1.100	0.0	2.300	0.0

RUN CONDITIONS AND COMMENTS---

WP 305

FORM - 17C

DATA STORAGE TAPE FILE

TAPE NUMBER................ 3202 FILE NUMBER................. 1
RUN NUMBER..............79000200 DATA SAMPLES/CHANNEL/LINE.....3448
FLIGHT LINE.......648215101 FL DATE DATA TAKEN.......... 2/12/79
TIME DATA TAKEN........ 8= (GMT) TIME DATA TAKEN.......1010 (LOCAL)
PLATFORM ALTITUDE...3062000 (FEET) GROUND HEADING.......189 (DEGREES)
DATE TAPE GENERATED...NOV 25,1980 LINES OF DATA..............2983
LAT. AT FRAME CENTER....... 30.18 LONG. AT FRAME CENTER...... 82.32
LAT. NOMINAL..............N30-12 LONG. NOMINAL.............W082-21
SUN ELEVATION..................32 SUN AZIMUTH.................134
SCENE FRAME ID.........2148215101 WRS DESIGNATOR.......... D018039
SENSOR TYPE..................MSS TIME OF EXPOSURE.......7904315101
RESAMPLING APPLIED........CUBIC CAL WEDGE MODE.........LOW COMPR
LINE LENGTH ADJUST.............YES COMPRESSED DATA..............YES
CALIBRATION...................YES EPHEMERIS DATA TYPE.....PREDICTIVE
SCALE OF IMAGE..........185*170 KM
PROJECTION.........HOTINE OBLIQUE MERCATOR

CHANNEL	SPECTRAL BAND LOWER	UPPER	CALIBRATION PULSE VALUES C0	C1	C2
1	0.500	0.600	0.0	2.480	0.0
2	0.600	0.700	0.0	2.000	0.0
3	0.700	0.800	0.0	1.760	0.0
4	0.800	1.100	0.0	2.300	0.0

RUN CONDITIONS AND COMMENTS--

WP 305

FORM - 17C

DATA STORAGE TAPE FILE

TAPE NUMBER.................. 3202
FILE NUMBER.................. 1
RUN NUMBER..................79000200
DATA SAMPLES/CHANNEL/LINE.....3448
FLIGHT LINE.......648215101 FL
DATE DATA TAKEN........... 2/12/79
TIME DATA TAKEN........ B= (GMT)
TIME DATA TAKEN.......1010 (LOCAL)
PLATFORM ALTITUDE...3062000 (FEET)
GROUND HEADING.......189 (DEGREES)
DATE TAPE GENERATED...NOV 25,1980
LINES OF DATA..................2983
LAT. AT FRAME CENTER........ 30.18
LONG. AT FRAME CENTER....... 82.32
LAT. NOMINAL...............N30-12
LONG. NOMINAL.............W082-21
SUN ELEVATION..................32
SUN AZIMUTH...................134
SCENE FRAME ID..........2148215101
WRS DESIGNATOR........... D018039
SENSOR TYPE....................MSS
TIME OF EXPOSURE........7904315101
RESAMPLING APPLIED.........CUBIC
CAL WEDGE MODE..........LOW COMPR
LINE LENGTH ADJUST............YES
COMPRESSED DATA...............YES
CALIBRATION....................YES
EPHEMERIS DATA TYPE.....PREDICTIVE
SCALE OF IMAGE..........185*170 KM
PROJECTION.........HOTINE OBLIQUE MERCATOR

CHANNEL	SPECTRAL BAND LOWER	SPECTRAL BAND UPPER	CALIBRATION PULSE VALUES C0	C1	C2
1	0.500	0.600	0.0	2.480	0.0
2	0.600	0.700	0.0	2.000	0.0
3	0.700	0.800	0.0	1.760	0.0
4	0.800	1.100	0.0	2.300	0.0

RUN CONDITIONS AND COMMENTS--

W? 305

FORM - 17C

DATA STORAGE TAPE FILE

TAPE NUMBER................. 3202

RUN NUMBER.................79000200

FLIGHT LINE.......648215101 FL

TIME DATA TAKEN......... 8= (GMT)

PLATFORM ALTITUDE...3062000 (FEET)

DATE TAPE GENERATED...NOV 25,1980

LAT. AT FRAME CENTER........ 30.18

LAT. NOMINAL................N30-12

SUN ELEVATION.....................32

SCENE FRAME ID..........2148215101

SENSOR TYPE.......... ....MSS

RESAMPLING APPLIED..........CUBIC

LINE LENGTH ADJUST.............YES

CALIBRATION.....................YES

SCALE OF IMAGE..........185*170 KM

PROJECTION..........HOTINE OBLIQUE MERCATOR

FILE NUMBER................. 1

DATA SAMPLES/CHANNEL/LINE.....3448

DATE DATA TAKEN.......... 2/12/79

TIME DATA TAKEN.......1010 (LOCAL)

GROUND HEADING.......199 (DEGREES)

LINES OF DATA...............2983

LONG. AT FRAME CENTER....... 82.32

LONG. NOMINAL..............W082-21

SUN AZIMUTH.....................134

WRS DESIGNATOR........... D018039

TIME OF EXPOSURE........7904315101

CAL WEDGE MODE..........LOW COMPR

COMPRESSED DATA...............YES

EPHEMERIS DATA TYPE.....PREDICTIVE

CHANNEL	SPECTRAL BAND LOWER	SPECTRAL BAND UPPER	CALIBRATION PULSE VALUES C0	C1	C2
1	0.500	0.600	0.0	2.480	0.0
2	0.600	0.700	0.0	2.000	0.0
3	0.700	0.800	0.0	1.760	0.0
4	0.800	1.100	0.0	2.300	0.0

RUN CONDITIONS AND COMMENTS---

WP 305

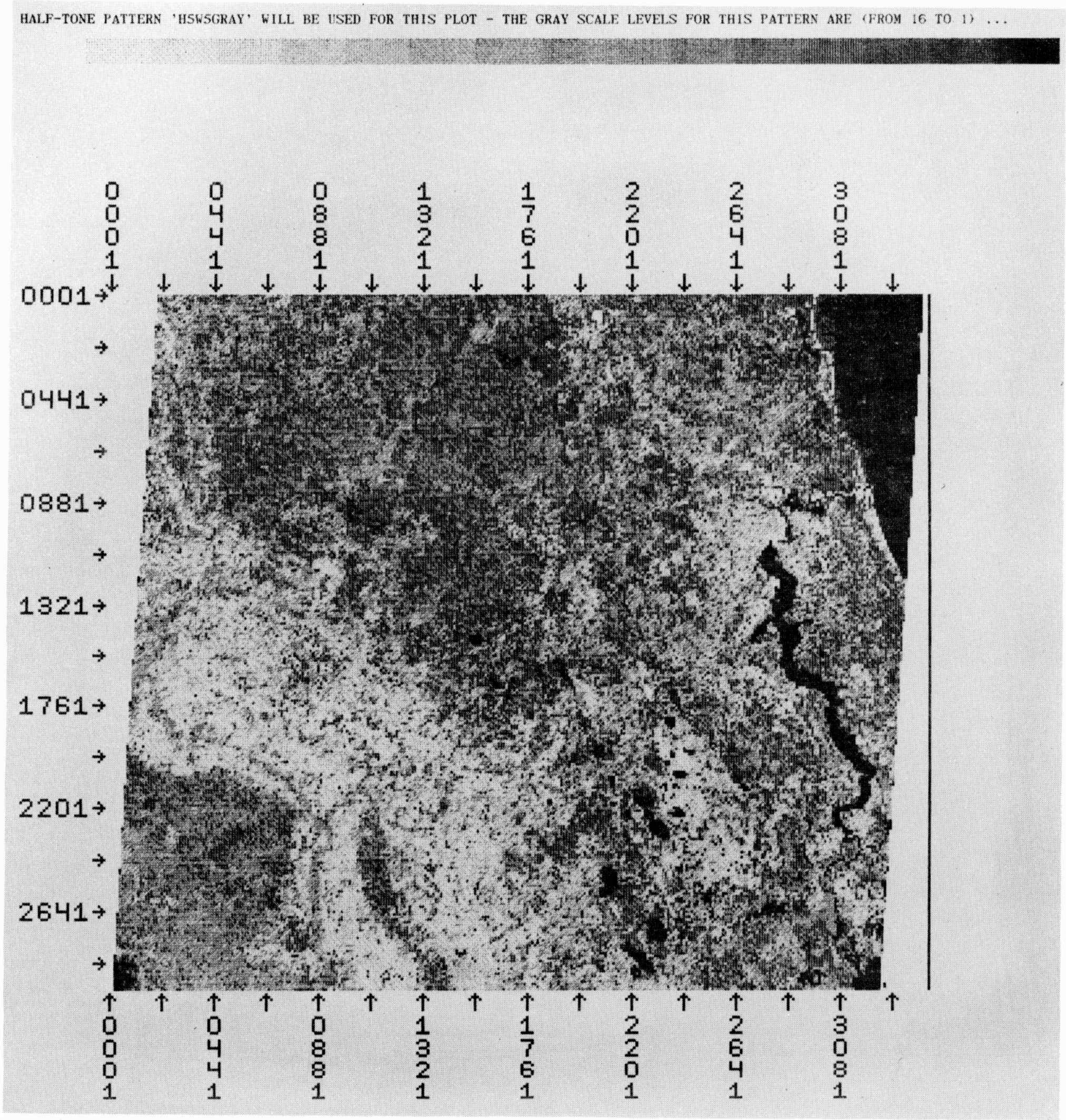


Figure 1. Grayscale plot of output from Landsat processor. The control card deck is shown on opposite page.

```
    *GDATA
    DISPLAY RUN(79000200),LINES(1,2983,11),COLS(1,3448,11)
    CHAN 2
    BLOCK RUN(79000200),LINES(1,2983,29),COLS(1,3448,34)
    GRAYPARM PATTERN(H5W5GRAY)
    END

YOU HAVE SELECTED THE FOLLOWING GRAYSCALE PLOT OPTIONS
    CALCULATE A NEW HISTOGRAM FOR SETTING GRAY LEVELS

GDATA INFORMATION
    NUMBER OF CHANNELS...   1
    THE HALFTONE PATTERN SELECTED...H5W5GRAY
I0033  DATA IS IN LARSYS FORMAT    (GADRUN)
```

Geometric Correction Processor

```
REFORM13                LABORATORY FOR APPLICATIONS OF REMOTE SENSING          MAR  10,1982
KOZL 113                            PURDUE UNIVERSITY                     TIME  12 03 21 AM

                                                  GEOMETRIC CORRECTION

*******************************************************************************
*                                                                             *
*                         CONTROL CARDS                                       *
*                                   (GCTROL)                                  *
*                                                                             *
*******************************************************************************


    *GEOMETRICCORRECTION
    GCORRECTION RUN(79000200)
    OUTPUT        RUN(79000201),TAPE(3841),FILE(1)
    AREA          LINES(600,1700),COLS(1000,2150)
    END


I0033  DATA IS IN LARSYS FORMAT   (GADRUN)
```

GEOMETRIC CORRECTION

```
*****************************************************************************
*                                                                           *
*                      JOB CONTROL SUMMARY                                  *
*                           (GCTROL)                                        *
*                                                                           *
*****************************************************************************
```

LINE 600 THROUGH LINE 1700 WILL BE USED.

COLUMN 1000 THROUGH COLUMN 2150 WILL BE USED.

LATITUDE OF 30.20 DERIVED FROM THE ID RECORD

DATA WILL NOT BE DESKEWED.

DATA WILL BE ROTATED.

CALIBRATION BYTES WILL BE ADDED.

FLIGHTLINE DERIVED FROM ID RECORD IS 648215101 FL

LANDSAT VEHICLE NUMBER IS 2

OUTPUT SCALE IS 1 TO 24000.0000

DATA WAS TAKEN IN EDIPS FORMAT.

GEOMETRIC CORRECTION

GEOMETRIC DISTORTION CORRECTION SUMMARY
(DISMAT)

LATITUDE ANGLE IN DEGREES = 30.20

N-S CORRECTION ANGLE (THETA) = 10.67211

SIN(THETA)	COS(THETA)
0.18519	0.98270

SKEW FACTOR = 0.07298

CORRECTION MATRICES

MATRIX NO. 1

SCALE CORRECTION MATRIX

57 X 57 TO 79 X 79

1.38680	0.0
0.0	1.38680

MATRIX NO. 2

ROTATION CORRECTION MATRIX

0.98270	0.18519
-0.18519	0.98270

MATRIX NO. 3

OUTPUT CORRECTION MATRIX

0.80000	0.0
0.0	1.00000

MATRIX NO. 4

RESCALING CORRECTION MATRIX

0.95270	0.0
0.0	0.95270

DISTORTION MATRIX

1.03868	0.24467
-0.19574	1.29835

GEOMETRIC CORRECTION

```
*****RUN WILL TAKE  13 PER CENT OF THE TAPE.          312.2 FEET

A RUN 79000201 ALREADY EXISTS -- EXECUTION CONTINUES.

I0001 AT 12 25 30 AM  TAPE   3841 REQUESTED.
      RING IN  ON UNIT 11 AT 1600BPI     (XMOUNT)
I0002 AT 12 28 25 AM  TAPE   3841 READY.
      WRITE PROTECT = NO . AT DEVICE ADDRESS 181     (XMOUNT)
```

TAPE NUMBER.............. 3841
RUN NUMBER........... 79000201
NUMBER OF DATA CHANNELS.... 4
FLIGHT LINE.. 648215101 FL
TIME DATA TAKEN.... 1510 HOURS
GROUND HEADING.... 180 DEGREES
LINES OF DATA............ 980

FILE NUMBER............... 1
CONTINUATION CODE........... 0
NUMBER OF DATA SAMPLES... 1268
DATE DATA TAKEN...... 2/12/79
SENSOR ALTITUDE.. 3062000 FEET
REFORMATING DATE. MAR 10,1982
RUN CENTER... 30.20 N/ 82.35 W

	SPECTRAL BAND		CALIBRATION PULSE VALUES		
CHANNEL	LOWER	UPPER	C0	C1	C2
1	0.50	0.60	0.08	2.63	0.0
2	0.60	0.70	0.06	1.76	0.0
3	0.70	0.80	0.06	1.52	0.0
4	0.80	1.10	0.11	3.91	0.0

GEOMETRIC CORRECTION

DATA FOR RUN 79000201 OF TAPE 3841 FILE 1

PRODUCED FROM RUN 79000200 OF TAPE 2672 FILE 1.

NUMBER OF BYTES REQUIRED = 76140
PAGES 000 TO 000 WILL BE LOCKED INTO CORE

GEOMETRIC CORRECTION

```
*********************************************************************
*                                                                   *
*               PROCESSING SUMMARY                                  *
*                    (GEMCOR)                                       *
*                                                                   *
*********************************************************************

          INPUT KEYPOINTS    (LINE,COL)      ( 1150, 1575)

          SIZE               (LINE,COL)      ( 1521, 1551)

          OUTPUT SIZE        (LINE,COL)      (  980, 1268)


        THE REQUESTED RUN WILL CONTAIN  THE AREA BOUNDED  BY
        LINES  600 - 1700 AND  COLUMNS 1000 - 2150

        THE INPUT BUFFER WILL BE (ILIN,LLIN,ISAM,LSAM)--   389  1909   800  2350

        FROM CENTER ( 1575, 1150) A FRAME OF  1268 SAMPLES AND  980 LINES WOULD TAKE
         9436284  OF THE   9600000 AVAILABLE BYTES.

        THE INPUT BUFFER COULD HOLD  1547 LINES AT ONCE.


  293 LINES WILL BE USED IN THE INPUT BUFFER.
        20 BUFFER LINES ADDED AT OUTPUT LINE   31
        20 BUFFER LINES ADDED AT OUTPUT LINE   46
        20 BUFFER LINES ADDED AT OUTPUT LINE   61
        20 BUFFER LINES ADDED AT OUTPUT LINE   76
        20 BUFFER LINES ADDED AT OUTPUT LINE   91
        20 BUFFER LINES ADDED AT OUTPUT LINE  106
        20 BUFFER LINES ADDED AT OUTPUT LINE  121
        20 BUFFER LINES ADDED AT OUTPUT LINE  136
        20 BUFFER LINES ADDED AT OUTPUT LINE  151
        20 BUFFER LINES ADDED AT OUTPUT LINE  166
        20 BUFFER LINES ADDED AT OUTPUT LINE  181
        20 BUFFER LINES ADDED AT OUTPUT LINE  196
        20 BUFFER LINES ADDED AT OUTPUT LINE  211
        20 BUFFER LINES ADDED AT OUTPUT LINE  226
        20 BUFFER LINES ADDED AT OUTPUT LINE  241
        20 BUFFER LINES ADDED AT OUTPUT LINE  256
        20 BUFFER LINES ADDED AT OUTPUT LINE  271
        20 BUFFER LINES ADDED AT OUTPUT LINE  286
        20 BUFFER LINES ADDED AT OUTPUT LINE  301
        20 BUFFER LINES ADDED AT OUTPUT LINE  316
        20 BUFFER LINES ADDED AT OUTPUT LINE  331
        20 BUFFER LINES ADDED AT OUTPUT LINE  346
        20 BUFFER LINES ADDED AT OUTPUT LINE  361
        20 BUFFER LINES ADDED AT OUTPUT LINE  376
        20 BUFFER LINES ADDED AT OUTPUT LINE  391
        20 BUFFER LINES ADDED AT OUTPUT LINE  406
        20 BUFFER LINES ADDED AT OUTPUT LINE  421
        20 BUFFER LINES ADDED AT OUTPUT LINE  436
        20 BUFFER LINES ADDED AT OUTPUT LINE  451
        20 BUFFER LINES ADDED AT OUTPUT LINE  466
        20 BUFFER LINES ADDED AT OUTPUT LINE  481
        20 BUFFER LINES ADDED AT OUTPUT LINE  496
        20 BUFFER LINES ADDED AT OUTPUT LINE  526
        20 BUFFER LINES ADDED AT OUTPUT LINE  541
        20 BUFFER LINES ADDED AT OUTPUT LINE  556
        20 BUFFER LINES ADDED AT OUTPUT LINE  571
        20 BUFFER LINES ADDED AT OUTPUT LINE  586
        20 BUFFER LINES ADDED AT OUTPUT LINE  601
        20 BUFFER LINES ADDED AT OUTPUT LINE  616
        20 BUFFER LINES ADDED AT OUTPUT LINE  631
        20 BUFFER LINES ADDED AT OUTPUT LINE  646
        20 BUFFER LINES ADDED AT OUTPUT LINE  661
        20 BUFFER LINES ADDED AT OUTPUT LINE  676
```

```
20 BUFFER LINES ADDED AT OUTPUT LINE   691
20 BUFFER LINES ADDED AT OUTPUT LINE   706
20 BUFFER LINES ADDED AT OUTPUT LINE   721
20 BUFFER LINES ADDED AT OUTPUT LINE   736
20 BUFFER LINES ADDED AT OUTPUT LINE   751
20 BUFFER LINES ADDED AT OUTPUT LINE   766
20 BUFFER LINES ADDED AT OUTPUT LINE   781
20 BUFFER LINES ADDED AT OUTPUT LINE   796
20 BUFFER LINES ADDED AT OUTPUT LINE   811
20 BUFFER LINES ADDED AT OUTPUT LINE   826
20 BUFFER LINES ADDED AT OUTPUT LINE   841
20 BUFFER LINES ADDED AT OUTPUT LINE   856
20 BUFFER LINES ADDED AT OUTPUT LINE   871
20 BUFFER LINES ADDED AT OUTPUT LINE   886
20 BUFFER LINES ADDED AT OUTPUT LINE   901
20 BUFFER LINES ADDED AT OUTPUT LINE   916
20 BUFFER LINES ADDED AT OUTPUT LINE   931
20 BUFFER LINES ADDED AT OUTPUT LINE   946
20 BUFFER LINES ADDED AT OUTPUT LINE   961
20 BUFFER LINES ADDED AT OUTPUT LINE   976

      30490.49 POINTS / TOTAL CPU SEC
       2662.32 POINTS / CONNECT SEC


CONNECT   0 HRS   31 MINS    7 SECS
VIRTUAL CPU TIME      153.98 SECS
TOTAL CPU TIME        163.02 SECS
```

LARS FORM - 17D

DATA STORAGE TAPE FILE

RUN NUMBER..................	79000201	FLIGHTLINE ID......... 648215101	FL
DATE TAPE GENERATED.....	MAR 10,1982	DATE DATA TAKEN..............	2/12/79
TAPE NUMBER....................	3841	TIME DATA TAKEN............	1510 HOURS
FILE NUMBER......................	1	PLATFORM ALTITUDE.........	3062000 FEET
LINES OF DATA..................	980	GROUND HEADING.............	180 DEGREES
SECONDS OF DATA...............	18.62	FIELD OF VIEW...........	0.096 RADIANS
MILES OF DATA.................	65.50	DATA SAMPLES PER CHANNEL PER LINE	1268
LINE RATE...........	52.62 LINES/SEC	SAMPLE RATE........	0.09 MILLIRADIANS
FRAME CENTER LATITUDE.........	30.20	FRAME CENTER LONGITUDE..........	82.35

SPECTRAL BANDWIDTH IN MICROMETERS..

CHAN	LOWER	UPPER	CHAN	LOWER	UPPER	CHAN	LOWER	UPPER
(1)	0.50	0.60	(2)	0.60	0.70	(3)	0.70	0.80
(4)	0.80	1.10	(5)	-----	-----	(6)	-----	-----
(10)	-----	-----	(11)	-----	-----	(12)	-----	-----
(13)	-----	-----	(14)	-----	-----	(15)	-----	-----
(16)	-----	-----	(17)	-----	-----	(18)	-----	-----
(19)	-----	-----	(20)	-----	-----	(21)	-----	-----
(22)	-----	-----	(23)	-----	-----	(24)	-----	-----
(25)	-----	-----	(26)	-----	-----	(27)	-----	-----
(28)	-----	-----	(29)	-----	-----	(30)	-----	-----

DATA TAPE COMMENTS...

THIS RUN IS A GEOMETRIC CORRECTION PRODUCED BY GEMCOR WITH A PRINTER ASPECT RATIO FROM LARSYS RUN 79000200, LINES 389-1909 AND COLUMNS 800-2350.

WP-113

LARS FORM - 170

DATA STORAGE TAPE FILE

RUN NUMBER.................. 79000201
DATE TAPE GENERATED..... MAR 10,1982
TAPE NUMBER..................... 3841
FILE NUMBER........................ 1
LINES OF DATA................... 980
SECONDS OF DATA................ 18.62
MILES OF DATA.................. 65.50
LINE RATE............ 52.62 LINES/SEC
FRAME CENTER LATITUDE......... 30.20

FLIGHTLINE ID......... 648215101 FL
DATE DATA TAKEN............... 2/12/79
TIME DATA TAKEN............. 1510 HOURS
PLATFORM ALTITUDE......... 3062000 FEET
GROUND HEADING............ 180 DEGREES
FIELD OF VIEW........... 0.096 RADIANS
DATA SAMPLES PER CHANNEL PER LINE 1268
SAMPLE RATE........ 0.09 MILLIRADIANS
FRAME CENTER LONGITUDE.......... 82.35

SPECTRAL BANDWIDTH IN MICROMETERS..

CHAN	LOWER	UPPER	CHAN	LOWER	UPPER	CHAN	LOWER	UPPER
(1)	0.50	0.60	(2)	0.60	0.70	(3)	0.70	0.80
(4)	0.80	1.10	(5)	-----	-----	(6)	-----	-----
(10)	-----	-----	(11)	-----	-----	(12)	-----	-----
(13)	-----	-----	(14)	-----	-----	(15)	-----	-----
(16)	-----	-----	(17)	-----	-----	(18)	-----	-----
(19)	-----	-----	(20)	-----	-----	(21)	-----	-----
(22)	-----	-----	(23)	-----	-----	(24)	-----	-----
(25)	-----	-----	(26)	-----	-----	(27)	-----	-----
(28)	-----	-----	(29)	-----	-----	(30)	-----	-----

DATA TAPE COMMENTS...

THIS RUN IS A GEOMETRIC CORRECTION PRODUCED BY GEMCOR WITH A PRINTER ASPECT RATIO FROM LARSYS RUN 79000200, LINES 389-1909 AND COLUMNS 800-2350.

WP-113

LARS FORM - 170

DATA STORAGE TAPE FILE

RUN NUMBER.................. 79000201	FLIGHTLINE ID......... 648215101 FL
DATE TAPE GENERATED..... MAR 10,1982	DATE DATA TAKEN............... 2/12/79
TAPE NUMBER..................... 3841	TIME DATA TAKEN............. 1510 HOURS
FILE NUMBER...................... 1	PLATFORM ALTITUDE......... 3062000 FEET
LINES OF DATA.................. 980	GROUND HEADING............. 180 DEGREES
SECONDS OF DATA............... 18.62	FIELD OF VIEW........... 0.096 RADIANS
MILES OF DATA................. 65.50	DATA SAMPLES PER CHANNEL PER LINE 1268
LINE RATE............ 52.62 LINES/SEC	SAMPLE RATE........ 0.09 MILLIRADIANS
FRAME CENTER LATITUDE......... 30.20	FRAME CENTER LONGITUDE.......... 82.35

SPECTRAL BANDWIDTH IN MICROMETERS..

CHAN	LOWER	UPPER	CHAN	LOWER	UPPER	CHAN	LOWER	UPPER
(1)	0.50	0.60	(2)	0.60	0.70	(3)	0.70	0.80
(4)	0.80	1.10	(5)	-----	-----	(6)	-----	-----
(10)	-----	-----	(11)	-----	-----	(12)	-----	-----
(13)	-----	-----	(14)	-----	-----	(15)	-----	-----
(16)	-----	-----	(17)	-----	-----	(18)	-----	-----
(19)	-----	-----	(20)	-----	-----	(21)	-----	-----
(22)	-----	-----	(23)	-----	-----	(24)	-----	-----
(25)	-----	-----	(26)	-----	-----	(27)	-----	-----
(28)	-----	-----	(29)	-----	-----	(30)	-----	-----

DATA TAPE COMMENTS...

THIS RUN IS A GEOMETRIC CORRECTION PRODUCED BY GEMCOR WITH A PRINTER ASPECT RATIO FROM LARSYS RUN 79000200, LINES 389-1909 AND COLUMNS 800-2350.

WP-113

LARS FORM - 17D

DATA STORAGE TAPE FILE

RUN NUMBER..................	79000201	FLIGHTLINE ID.........	648215101 FL
DATE TAPE GENERATED.....	MAR 10,1982	DATE DATA TAKEN..............	2/12/79
TAPE NUMBER.....................	3841	TIME DATA TAKEN............	1510 HOURS
FILE NUMBER......................	1	PLATFORM ALTITUDE.........	3062000 FEET
LINES OF DATA..................	980	GROUND HEADING............	180 DEGREES
SECONDS OF DATA................	18.62	FIELD OF VIEW..........	0.096 RADIANS
MILES OF DATA.................	65.50	DATA SAMPLES PER CHANNEL PER LINE	1268
LINE RATE...........	52.62 LINES/SEC	SAMPLE RATE........	0.09 MILLIRADIANS
FRAME CENTER LATITUDE........	30.20	FRAME CENTER LONGITUDE..........	82.35

SPECTRAL BANDWIDTH IN MICROMETERS..

CHAN	LOWER	UPPER	CHAN	LOWER	UPPER	CHAN	LOWER	UPPER
(1)	0.50	0.60	(2)	0.60	0.70	(3)	0.70	0.80
(4)	0.80	1.10	(5)	-----	-----	(6)	-----	-----
(10)	-----	-----	(11)	-----	-----	(12)	-----	-----
(13)	-----	-----	(14)	-----	-----	(15)	-----	-----
(16)	-----	-----	(17)	-----	-----	(18)	-----	-----
(19)	-----	-----	(20)	-----	-----	(21)	-----	-----
(22)	-----	-----	(23)	-----	-----	(24)	-----	-----
(25)	-----	-----	(26)	-----	-----	(27)	-----	-----
(28)	-----	-----	(29)	-----	-----	(30)	-----	-----

DATA TAPE COMMENTS...

THIS RUN IS A GEOMETRIC CORRECTION PRODUCED BY GEMCOR WITH A PRINTER ASPECT RATIO FROM LARSYS RUN 79000200, LINES 389-1909 AND COLUMNS 800-2350.

WP-113

LARS FORM - 17D

DATA STORAGE TAPE FILE

RUN NUMBER.................. 79000201
DATE TAPE GENERATED..... MAR 10,1982
TAPE NUMBER..................... 3841
FILE NUMBER......................... 1
LINES OF DATA.................... 980
SECONDS OF DATA................ 18.62
MILES OF DATA.................. 65.50
LINE RATE........... 52.62 LINES/SEC
FRAME CENTER LATITUDE......... 30.20

FLIGHTLINE ID......... 648215101 FL
DATE DATA TAKEN............... 2/12/79
TIME DATA TAKEN............. 1510 HOURS
PLATFORM ALTITUDE......... 3062000 FEET
GROUND HEADING............. 180 DEGREES
FIELD OF VIEW........... 0.096 RADIANS
DATA SAMPLES PER CHANNEL PER LINE 1268
SAMPLE RATE........ 0.09 MILLIRADIANS
FRAME CENTER LONGITUDE.......... 82.35

SPECTRAL BANDWIDTH IN MICROMETERS..

CHAN	LOWER	UPPER	CHAN	LOWER	UPPER	CHAN	LOWER	UPPER
(1)	0.50	0.60	(2)	0.60	0.70	(3)	0.70	0.80
(4)	0.80	1.10	(5)	-----	-----	(6)	-----	-----
(10)	-----	-----	(11)	-----	-----	(12)	-----	-----
(13)	-----	-----	(14)	-----	-----	(15)	-----	-----
(16)	-----	-----	(17)	-----	-----	(18)	-----	-----
(19)	-----	-----	(20)	-----	-----	(21)	-----	-----
(22)	-----	-----	(23)	-----	-----	(24)	-----	-----
(25)	-----	-----	(26)	-----	-----	(27)	-----	-----
(28)	-----	-----	(29)	-----	-----	(30)	-----	-----

DATA TAPE COMMENTS...

THIS RUN IS A GEOMETRIC CORRECTION PRODUCED BY GEMCOR WITH A PRINTER ASPECT RATIO FROM LARSYS RUN 79000200, LINES 389-1909 AND COLUMNS 800-2350.

WP-113

LARS FORM - 17D

DATA STORAGE TAPE FILE

RUN NUMBER.................. 79000201	FLIGHTLINE ID.......... 648215101 FL
DATE TAPE GENERATED..... MAR 10,1982	DATE DATA TAKEN............... 2/12/79
TAPE NUMBER..................... 3841	TIME DATA TAKEN............. 1510 HOURS
FILE NUMBER...................... 1	PLATFORM ALTITUDE......... 3062000 FEET
LINES OF DATA.................. 980	GROUND HEADING............. 180 DEGREES
SECONDS OF DATA................ 18.62	FIELD OF VIEW........... 0.096 RADIANS
MILES OF DATA................. 65.50	DATA SAMPLES PER CHANNEL PER LINE 1268
LINE RATE........... 52.62 LINES/SEC	SAMPLE RATE....... 0.09 MILLIRADIANS
FRAME CENTER LATITUDE........ 30.20	FRAME CENTER LONGITUDE.......... 82.35

SPECTRAL BANDWIDTH IN MICROMETERS..

CHAN	LOWER	UPPER	CHAN	LOWER	UPPER	CHAN	LOWER	UPPER
(1)	0.50	0.60	(2)	0.60	0.70	(3)	0.70	0.80
(4)	0.80	1.10	(5)	-----	-----	(6)	-----	-----
(10)	-----	-----	(11)	-----	-----	(12)	-----	-----
(13)	-----	-----	(14)	-----	-----	(15)	-----	-----
(16)	-----	-----	(17)	-----	-----	(18)	-----	-----
(19)	-----	-----	(20)	-----	-----	(21)	-----	-----
(22)	-----	-----	(23)	-----	-----	(24)	-----	-----
(25)	-----	-----	(26)	-----	-----	(27)	-----	-----
(28)	-----	-----	(29)	-----	-----	(30)	-----	-----

DATA TAPE COMMENTS...

THIS RUN IS A GEOMETRIC CORRECTION PRODUCED BY GEMCOP WITH A PRINTER ASPECT RATIO FROM LARSYS RUN 79000200, LINES 389-1909 AND COLUMNS 800-2350.

WP-113

LARS FORM - 17D

DATA STORAGE TAPE FILE

RUN NUMBER.................. 79000201

DATE TAPE GENERATED..... MAR 10,1982

TAPE NUMBER..................... 3841

FILE NUMBER......................... 1

LINES OF DATA.................. 980

SECONDS OF DATA............... 18.62

MILES OF DATA................. 65.50

LINE RATE........... 52.62 LINES/SEC

FRAME CENTER LATITUDE......... 30.20

FLIGHTLINE ID......... 648215101 FL

DATE DATA TAKEN............... 2/12/79

TIME DATA TAKEN............. 1510 HOURS

PLATFORM ALTITUDE......... 3062000 FEET

GROUND HEADING............. 180 DEGREES

FIELD OF VIEW........... 0.096 RADIANS

DATA SAMPLES PER CHANNEL PER LINE 1268

SAMPLE RATE........ 0.09 MILLIRADIANS

FRAME CENTER LONGITUDE.......... 82.35

SPECTRAL BANDWIDTH IN MICROMETERS..

CHAN	LOWER	UPPER	CHAN	LOWER	UPPER	CHAN	LOWER	UPPER
(1)	0.50	0.60	(2)	0.60	0.70	(3)	0.70	0.80
(4)	0.80	1.10	(5)	-----	-----	(6)	-----	-----
(10)	-----	-----	(11)	-----	-----	(12)	-----	-----
(13)	-----	-----	(14)	-----	-----	(15)	-----	-----
(16)	-----	-----	(17)	-----	-----	(18)	-----	-----
(19)	-----	-----	(20)	-----	-----	(21)	-----	-----
(22)	-----	-----	(23)	-----	-----	(24)	-----	-----
(25)	-----	-----	(26)	-----	-----	(27)	-----	-----
(28)	-----	-----	(29)	-----	-----	(30)	-----	-----

DATA TAPE COMMENTS...

THIS RUN IS A GEOMETRIC CORRECTION PRODUCED BY GEMCOR WITH A PRINTER ASPECT RATIO FROM LARSYS RUN 79000200, LINES 389-1909 AND COLUMNS 800-2350.

WP-113

```
  EOT CALLED TO END THE TAPE.

TAPE FILE UPDATED AT EOT.
```

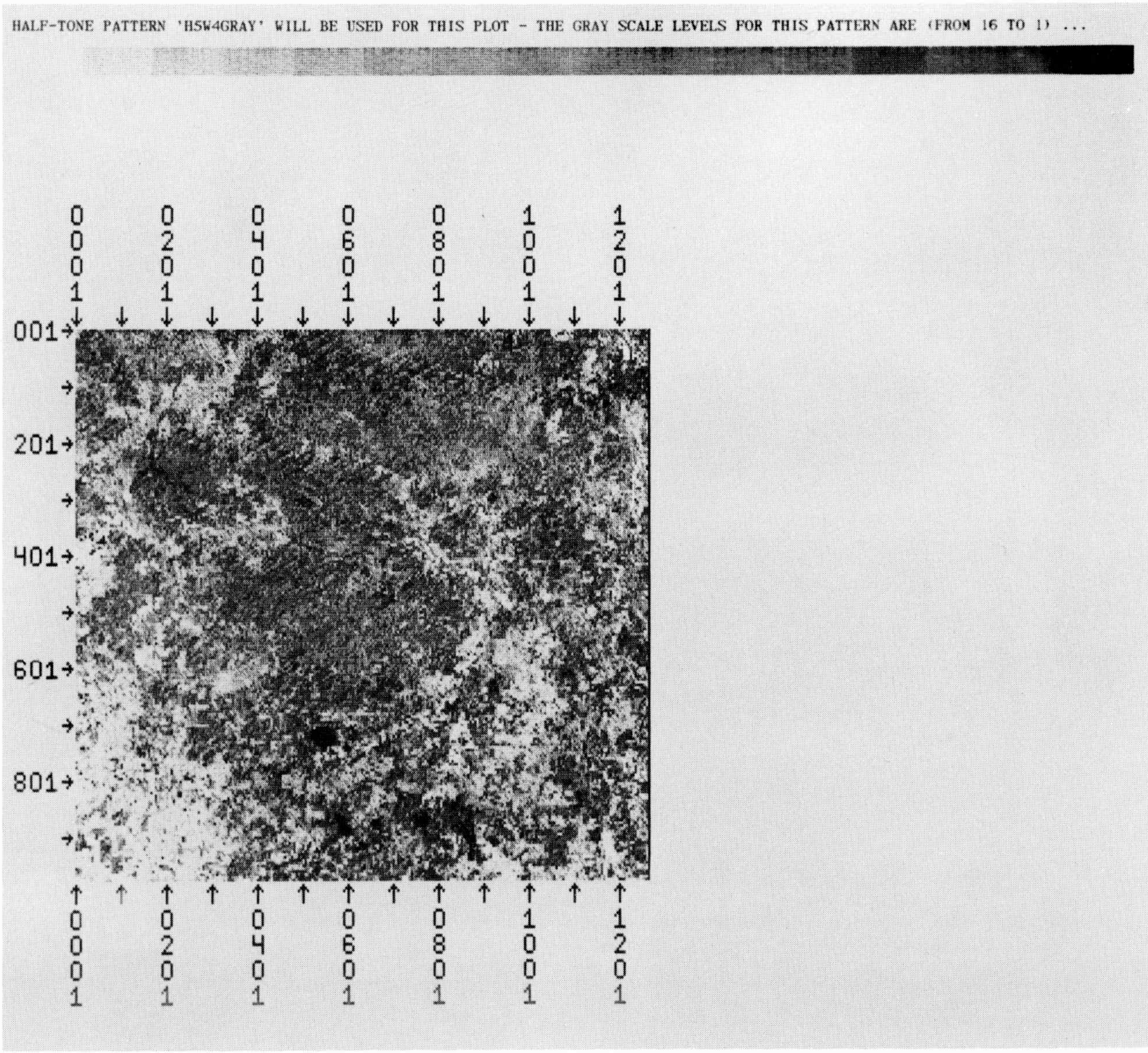


Figure 2. Grayscale plot of output from Geometric Correction processor. The control card deck is shown on the opposite page.

```
*GDATA
DISPLAY RUN(79000201),LINE(1,980,5),COL(1,1262,5)
CHAN 2
GRAYPARM  PATTERN(H5W4GRAY)
END
```

YOU HAVE SELECTED THE FOLLOWING GRAYSCALE PLOT OPTIONS

CALCULATE A NEW HISTOGRAM FOR SETTING GRAY LEVELS

GDATA INFORMATION

NUMBER OF CHANNELS... 1

THE HALFTONE PATTERN SELECTED...H5W4GRAY

10033 DATA IS IN LARSYS FORMAT (GADRUN)

REFORM13	LABORATORY FOR APPLICATIONS OF REMOTE SENSING	MAR 12,1982
KOZL 113	PURDUE UNIVERSITY	12 36 10 AM
		LARSYS VERSION 3

RUN NUMBER........... 79000201	DATE DATA TAKEN... FEB 12,1979
FLIGHT LINE... 648215101 FL	TIME DATA TAKEN..... 1510 HOURS
DATA TAPE/FILE NUMBER.. 3841/ 1	PLATFORM ALTITUDE..3062000 FEET
REFORMATTING DATE. MAR 10,1982	GROUND HEADING..... 180 DEGREES

CHANNEL 2 SPECTRAL BAND 0.60 TO 0.70 MICROMETERS CALIBRATION CODE = 1 C0 = .6000E-01