T79-11201/NMK

JSC-14663
E80-10325
NASA CR-160739

"AS-BUILT" DESIGN SPECIFICATION FOR A LIST PROCESSING SYSTEM

JOB ORDER 71-475

(TIRF 78-0021)

(E80-10325) AS-BUILT DESIGN SPECIFICATION FOR A LIST PROCESSING SYSTEM (Lockheed Electronics Co.) 281 p HC A13/MF A01 CSCL 05B

N80-32816

Unclas
G3/43 00325

Prepared By

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15800
For
EARTH OBSERVATIONS DIVISION
SCIENCE AND APPLICATIONS DIRECTORATE

National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

January 1979 LEC-13076

1 Report No	2 Government Accession No	3 Recipient's Catalog No
JSC-14663		

4 Title and Subtitle	5 Report Date
"As-Built" Design Specification for a LIST Processing System	January 1979
	6 Performing Organization Code SF3
7 Author(s) C. L. Horton, C. W. Ahlers, and P. J. Aucoin Jr.	8 Performing Organization Report No LEC-13076
	10 Work Unit No
9 Performing Organization Name and Address Lockheed Electronics Company, Inc. Systems and Services Division Houston, Texas 77058	11 Contract or Grant No NAS9-15800
	13 Type of Report and Period Covered "As-Built" Design Spec.
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 (J. M. Sulester, Tech. Monitor)	14 Sponsoring Agency Code

15 Supplementary Notes

16 Abstract

This document contains an as built design specification for a LIST (LABEL IDENTIFICATION FROM STATISTICAL TABULATION) processing system. This system provides for dot labeling using quadratic discrimination based on analyst-provided answers to questions concerning an MSS pixel array (segment) and individual dots (pixels) in the segment.

17 Key Words (Suggested by Author(s))		18 Distribution Statement
LIST	Ground Truth	
EOD-LARSYS	Training	
Quadratic discrimination	Classification	
Analyst-Interpreter	Patterson-Pitt	
	209 Dots	

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

*For sale by the National Technical Information Service, Springfield, Virginia 22161

JSC-14663

"AS-BUILT" DESIGN SPECIFICATION FOR A LIST PROCESSING SYSTEM (TIRF 78-0021)

JOB ORDER 71-475

Prepared By

C. W. Ahlers
P. J. Aucoin, Jr.
C. L. Horton

APPROVED BY

James A Wilkinson for

P. L. Krumm, Acting Supervisor
Scientific Applications Section

Prepared By
Lockheed Electronics Company, Inc.
For
Earth Observations Division

January 1979

LEC-13076

CONTENTS

1. SCOPE

This document contains an "as built" design specification for a LIST (LABEL IDENTIFICATION FROM STATISTICAL TABULATION) processing system. This system provides for dot labeling using quadratic discrimination based on analyst-provided answers to questions concerning an MSS pixel array (segment) and individual dots (pixels) in the segment.

The system consists of several subsystems which communicate through disk and tape files. These subsystems are

1. The file merge processor of the EOD-LARSYS system (DAMRG)
2. The ground truth to class map and class map to labeled dots processors of the EOD-LARSYS system (GTTCN, GTDDM)
3. The LIST variable processor, (PREPPT)
4. The discriminator training program, (MPPTA)
5. The discriminator classification program, (MPTTC)
6. The EOD-LARSYS Procedure 1 system modified to accept Type dots re-labeled as boundary dots and output CCIT-type comparison reports.

The LIST processing system was written in FORTRAN IV-G and implemented on the IBM 370/148 computer at Purdue-LARS.

2. APPLICABLE DOCUMENTS

- As-Built Design Specification for a Merging Program for Formatted Image Data Files, Lockheed Electronics Company, Inc., SSD, Houston, Texas, Aug 1978, LEC-12653.
- Preliminary User Guide for the Program GTDDM (Ground Truth Dot Dumps), Lockheed Electronics Company, Inc., SSD, Houston, Texas, July 1978, LEC-12636.
- "As-Built" Design Specification for the Patterson-Pitt-Thandani Minimum Loss Classifier, Lockheed Electronics Company, Inc., SSD, Houston, Texas, May 1978, LEC-12285.
- Preliminary User Guide for the Program GTTCN (Ground Truth Tape Conversion) LEC, H,T, July 1978, LEC-12635.
- "As-Built" Design Specification for LACIE formatted Dot Cards in EOD-LARSYS, LEC, April 1978, LEC-12154.
- As-Built Specification for EOD-LARSYS Procedure 1, TIRF 77-0008, LEC-11293, JSC-13143, Oct. 1977.
- Design Specification For A LIST Processing System, Lockheed Electronics Company, Inc., SSD, Houston, Texas, August 1978, LEC-12696.
- TIRF 78-0021, LIST Processing Capabilities, May 4, 1978.

3. SYSTEM DESCRIPTION

3.1 HARDWARE

The LIST processing program is operational on the IBM 370-148 at Purdue LARS under the CMS370 operating system. The program utilizes the IBM Fortran IV-G compiler.

3.2 SOFTWARE DESCRIPTION

The LIST processing system features a semi-automatic dot labeling procedure. The quadratic discriminator is trained based on analyst-furnished responses to questions concerning the segment under investigation, and individual dots, raw MSS data and ground truth dot labels. This discriminator is then used in the test and classification phase to label dots utilizing the analyst-supplied responses.

The complete system consists of two main parts. The first part is the LIST labeling procedure. The second part utilizes the LIST labeled dots in conjunction with the EOD-LARSYS Procedure 1 system for classification of segment pixels.

3.2.1 SOFTWARE COMPONENT NO. 1 (LST EXEC)

The LIST executive file requests the tapes for each procedure, loads and starts execution of each processor, defines input and output files, saves intermediate files and prints list files. (See section 4.)

3.2.1.1 Linkages

The LIST executive uses the FORTRAN-IVG compiler, routine CMS370 software systems routines, data files and the following processor programs: DAMRG, GTTCN, GTDDM, PREPPT, MPPTA, MPPTC, TDMP, FDMP, and the EOD-LARSYS system.

3.2.1.2 Interfaces

Interface is accomplished through the following files:

16-channel Data File in EOD-LARSYS format.
Ground Truth Class Map in Universal format.
LIST Variable Input File (Table 3-5).
Loss Vector Matrix File.

3.2.1.3 Inputs

Inputs to the LIST system are described for each of the processors DAMRG, GTTCN, GTDDM, PREPPT, PPTA and PPTC. The master LIST control file is available to each processor for the types of processing (TRAIN, TEST, CLASSIFY), segment numbers and dates.

3.2.1.4 Outputs

Outputs from the LIST processor are described for each of the subprocessors DAMRG, GTTCN, GTDDM, PREPPT, PPTA and PPTC.

3.2.1.5 Storage Requirements

The LIST processor requires less than 1024k bytes of virtual memory.

3.2.1.6 Description

The LIST processor reads input control data giving the segment numbers to be processed and the type of processing to be done. If requested, the subprocessor Merging Formatted Image Data Files, DAMRG, is called to produce a 16-channel tape from portions of 4 four-channel tapes. This tape consists of one logical file for each segment.

If training or testing is to be done, the Ground Truth Tape Conversion subprocessor, GTTCN, is called to produce the Ground Truth Class Map Disc File from the one-channel Ground Truth Tape.

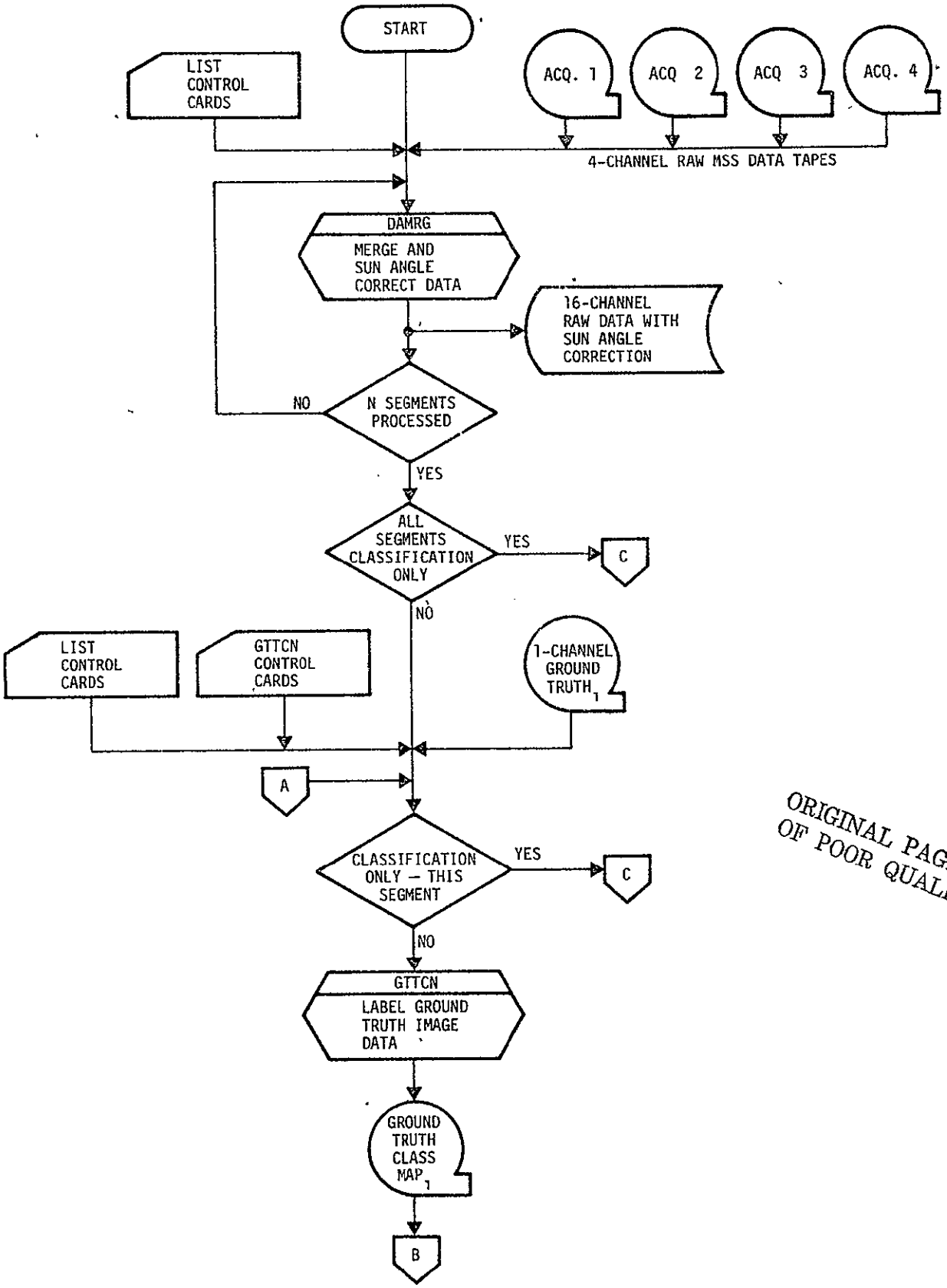
This disc file is then read by the Ground Truth Dot Dump subprocessor, GTDDM to produce the Ground Truth Dot File also consisting of one logical file for each segment.

If requested, the Preprocessor for the Patterson-Pitt-Thadani Routines, PREPRT, is called to prepare data for the quadratic discriminator. The 16-Channel Data File created by DAMRG, the AI Responses File, and, if training is to be done, the Ground Truth Dot Data File are used with the Seasonal Table and the Biostage Response Table to produce the LIST Variable Input File.

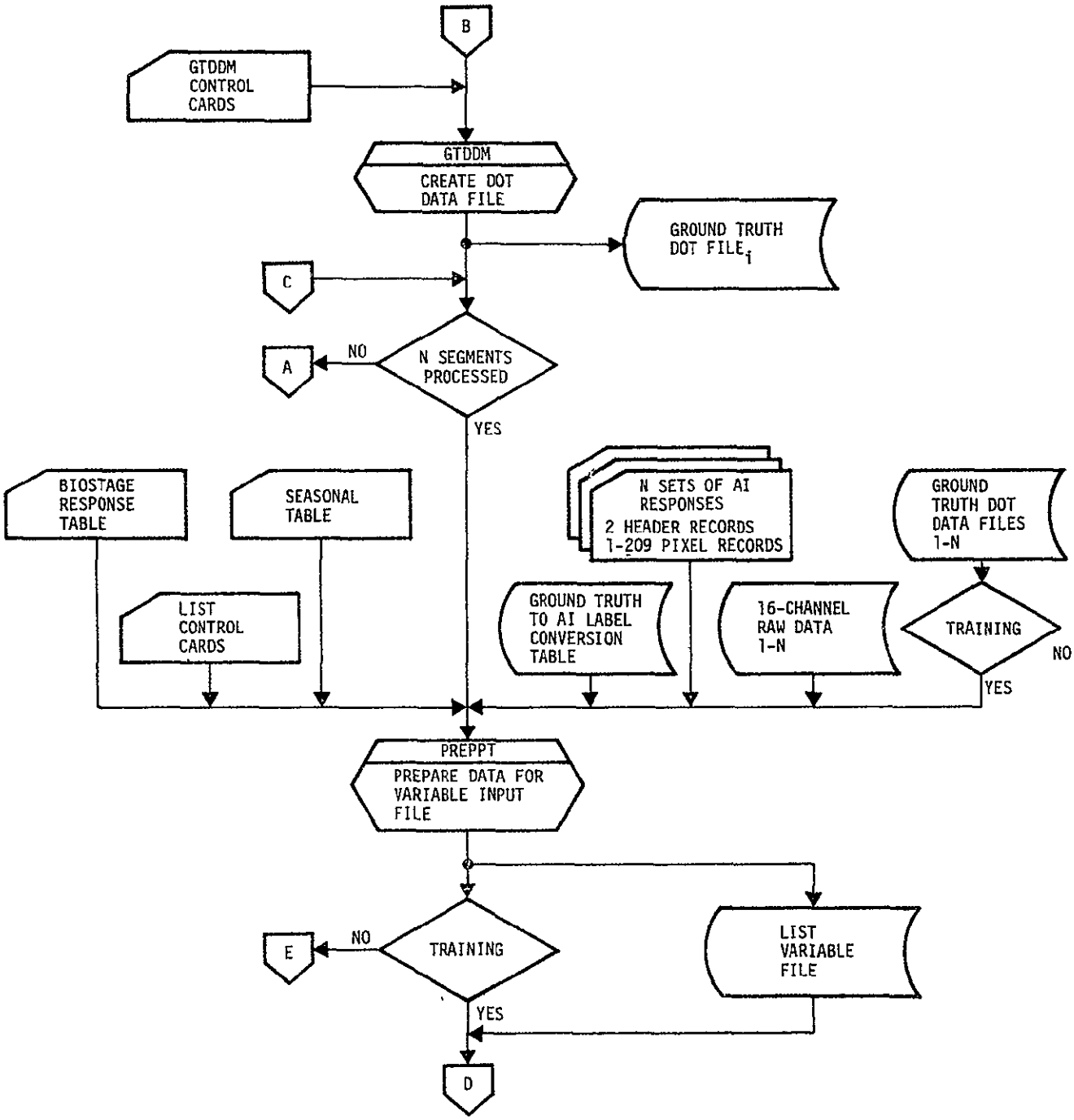
If requested, the Patterson-Pitt-Thadani Training Processor, MPPTA, is called to create the loss vector matrix, and the Patterson-Pitt-Thadani Classifying Processor, MPPTC, is called to calculate the classification losses to find the minimum loss and write the computed dot labels (classification results). MPPTC also creates LACIE format dot files for the AI lables and the classified labels.

If requested, the EOD-LARSYS System Display Processor is called the classify the entire scene using the 16-Channel Raw Data File created by DAMRG, the EOD-LARSYS control cards, the Analyst Changes to Computed Dot Labels File and if type is TEST the Ground Truth Dot Labels File. This classification produces the Class Map File, the Dot Summaries and Proportion Estimate File and a Dot Summary Listing. If type is TEST, this processor will also produce a file comparing the classification results to ground truth.

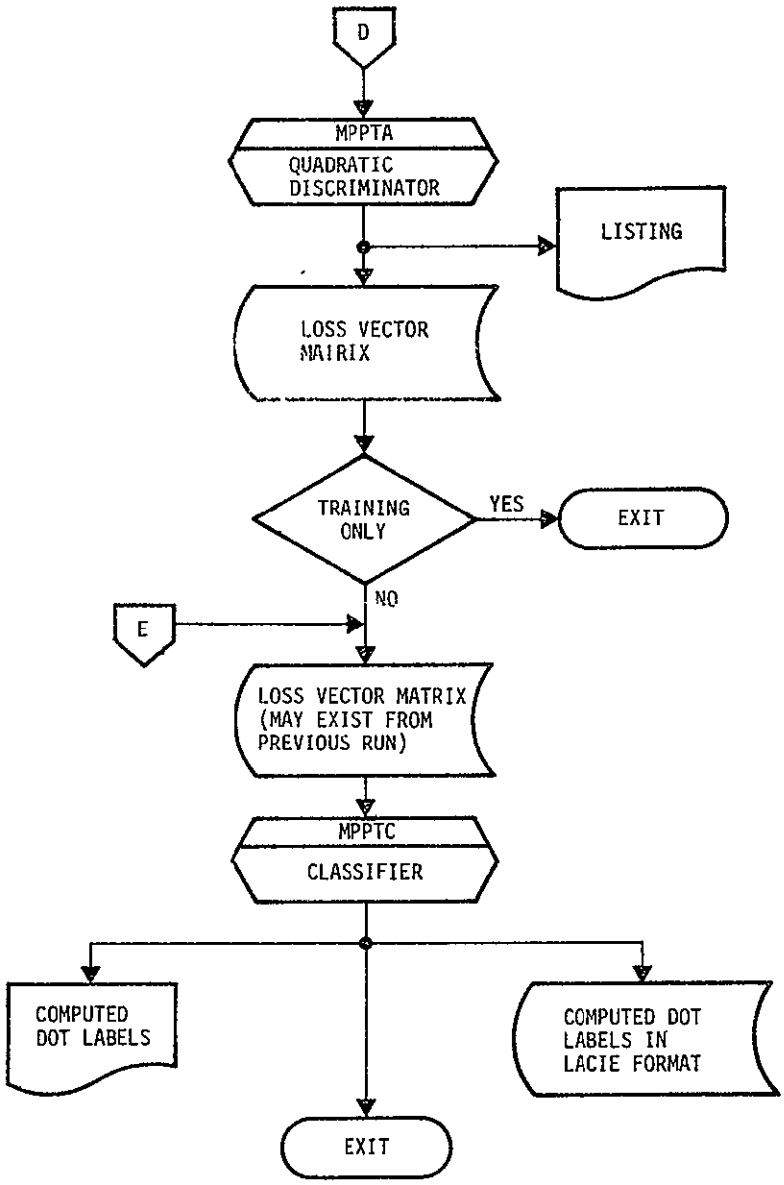
LIST PROCESSOR



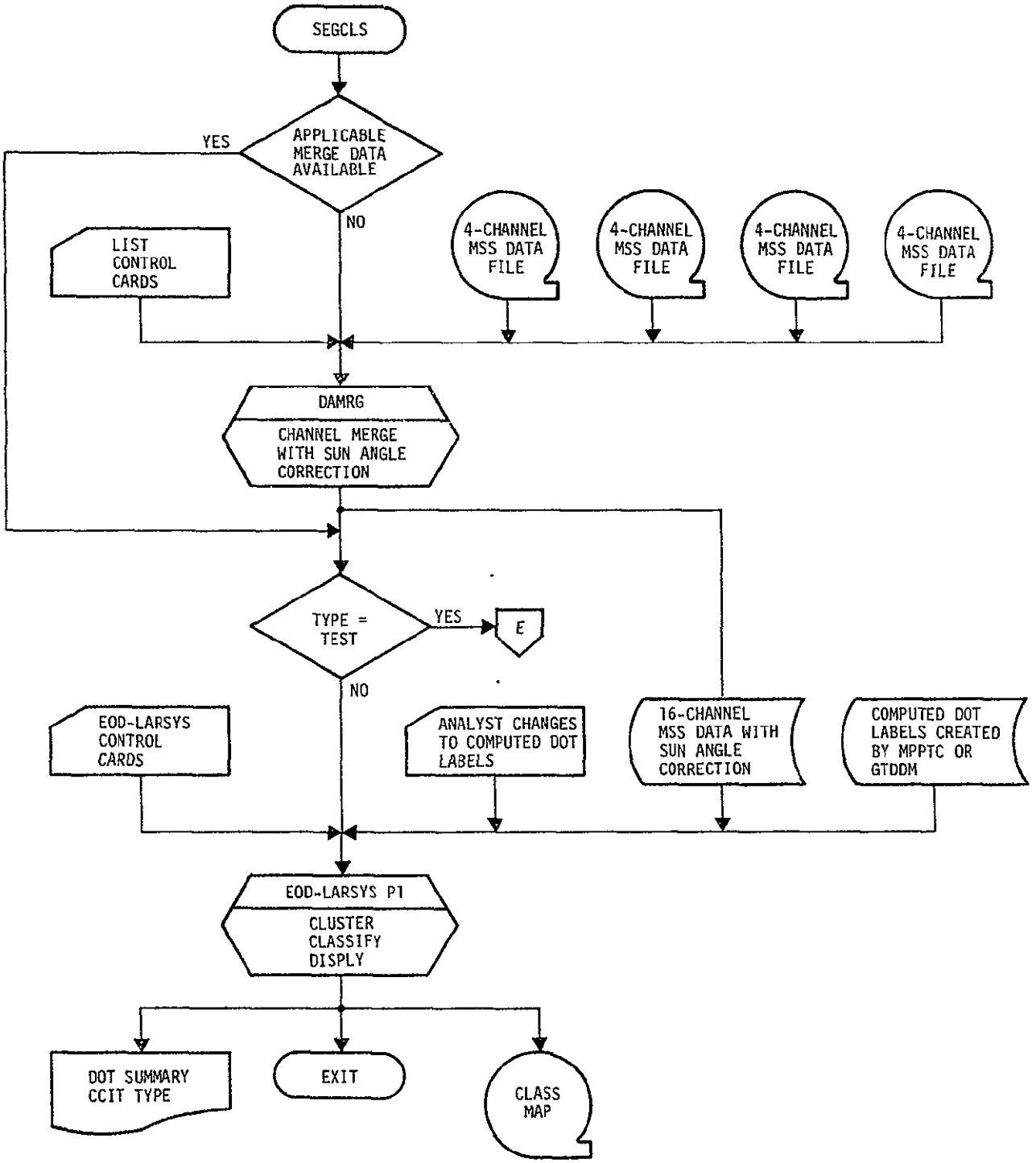
LIST PROCESSOR (Continued)



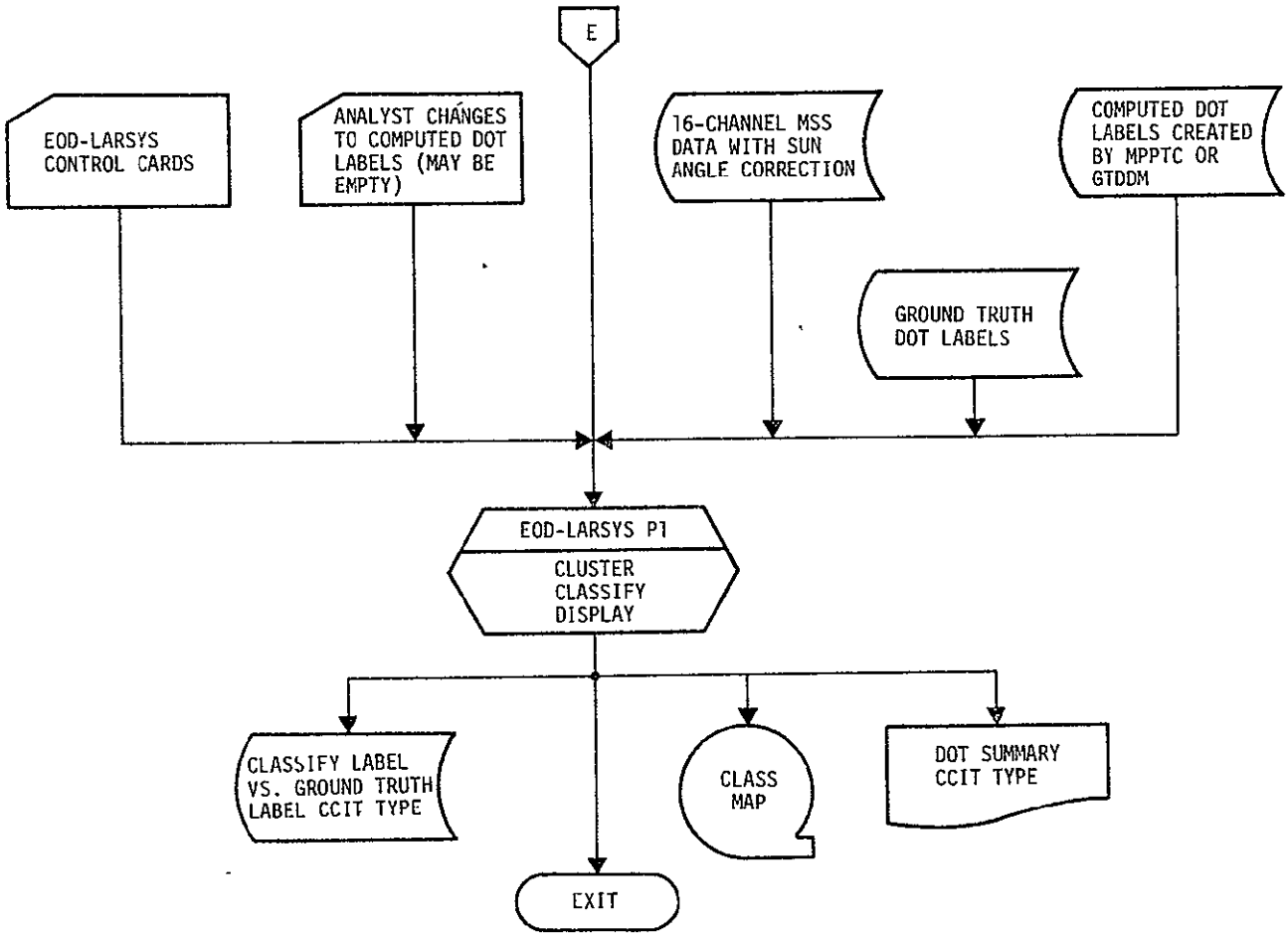
LIST PROCESSOR (Concluded)



SEGMENT CLASSIFICATION



SEGMENT CLASSIFICATION (Concluded)



3.2.2 SOFTWARE COMPONENT NO. 2 (DAMRG)

The DAMRG program furnishes a general capability to merge formatted MSS data file into a file suitable for input to pattern recognition systems such as EOD-LARSYS. Up to six files can be handled in one merge procedure; the option used in LIST processing is four channels from each of four tapes.

This program allows for the following three user options in merging data: channel merge, spatial merge and line merge. The option used to prepare data for LIST processing is the channel merge option whereby pixel data from the specified channels in the user-specified rectangular field are concatenated.

The output file is a Universally or LARSYS III formatted file with a rectangular field with line and sample skip factors set to the value 1 and the first pixel coordinates set to (1,1). Merge channels are renumbered starting with the value 1. Channels are merged in the order of appearance in the input control cards.

Sun angles are extracted from header records or read in from control cards. Gains and biases are unpacked for appropriate channels. Sun angle corrections are performed on option to all pixels making up to the output tape.

A channel merge of Universally formatted output has the sunangles and gains and biases written to the output header; the data and site are also written to the header.

In gains and biases and sunangle extraction, it is assumed that the first channel on any input file is channel number 1.

Reference As-Built Design Specification for a Merging Program for Image Data Files, Aug. 1978, for a detailed description of DAMRG.

3.2.3 SOFTWARE COMPONENT NO. 3 (GTTCN)

The program GTTCN converts Accuracy Assessment Ground Truth Image files in Universal format (351 by 392 in size) to Universal format image files (117 by 196 in size) by mapping six subpixels into one pixel label.

A user supplied control card-image file specifies the format and logical unit number (LUN) and file number of the first file on the input tape, the LUN and the file number of the first file on the output tape, the number of files to be converted, a list option for the crop codes listing and a vector designating which subpixels are to be used in labeling the pixel.

The conversion labels each pixel by majority rule using the user supplied vector to determine which of the subpixels are to be used in the labeling process. Ties are resolved by using the first label.

A Crop Code Listing is producted if requested in the output file.

Reference Preliminary User Guide for the Program GTTCN (Ground Truth Tape Conversion), LEC, July 1978.

3.2.4 SOFTWARE COMPONENT NO. 4 (GTDDM)

The GTDDM program furnishes a general capability to extract a 11 by 19 matrix of dots from a Converted Ground Truth Tape (size 117 by 196) prepared by the GTTCN program. This 11 by 19 matrix is then labelled using the user supplied Crop Code to Category Table in the LARSYS format which is described in the document referenced below.

The 11 by 19 matrix of labelled dots is converted to LACIE format and written to an output file.

A report file is written listing all control cards, the 11 by 19 matrix of labeled dots and the LACIE format dots.

Reference Preliminary User Guide for the Program GTDDM (Ground Truth Dot Dump) for a detailed description of GTTDM.

3.2.5 SOFTWARE COMPONENT NO. 5 (PREPPT)

PREPPT collects data from multiple files, calculates values for each pixel to describe greenness and brightness and writes the LIST variable Input File.

3.2.5.1 Linkages

PREPPT is called by the LISTRR EXEC and calls READID, READTB, READAI, READRD, CALKEY, READGT, RDLDF, and WRTQD.

3.2.5.2 Interfaces

Interface is accomplished through common blocks /AI/, /BLØSTG/, /SEASNS/, /SGMENT/.

3.2.5.3 Inputs

List Control Card File (Table 3-1), Biostage Response File (Table 3-2), Seasonal Table File (Table 3-3), AI Response File (Table 3-4), 16-channel MSS Data File, Ground Truth Dot Data File.

3.2.5.4 Outputs

LIST Variable Input File (Table 3-5).

3.2.5.5 Storage Requirement

Program size = 1502.

3.2.5.6 Description

PREPPT calls READID to read the types of processing, segment numbers and segment dates. READTB is called to read the seasonal data and biostage data and READAI is called to read the analyst response data and READRD is called to read the 16-channel MSS data file. CALKEY is called to make the calculations described in Appendix D. If testing or training is to be done, READGT is called to read the ground truth data. READGT calls RDLDF which calls PTTLAC to read the LACIE format ground truth dot file. WRTQD is called to write the LIST Variable Input File.

3.2.5.7 Flowchart

N/A

3.2.5.8 Listings

```
FILE PREPPT
C                                                                       PRE00010
C    WRITTEN BY C HORTON.                                               PRE00020
C    ADAPTED BY C AHLERS                                                PRE00030
C                                                                       PRE00040
C  THE PURPOSE OF THIS PROGRAM IS TO MERGE THE DATA FROM 6 FILES        PRE00050
C  PRODUCING THE FOLLOWING SETS OF DATA ON A 7TH FILE                   PRE00060
C                                                                       PRE00070
C    1  FILE HEADER RECORD WITH THE FOLLOWING ELEMENTS                  PRE00080
C         NUMBER OF SEGMENTS                                            PRE00090
C         SEGMENT NO. AND DATE      (N RECORDS)                         PRE00100
C                                                                       PRE00110
C    N SEGMENTS WITH THE FOLLOWING ELEMENTS                             PRE00120
C         2 HEADER RECORDS FROM FILE AI                                 PRE00130
C                                                                       PRE00140
C    FOR EACH PIXEL A RECORD CONTAINING THE FOLLOWING ELEMENTS          PRE00150
C          LINE                                                         PRE00160
C          COLUMN                                                       PRE00170
C          GROUND TRUTH CATEGORY        GTCAT                           PRE00180
C          AI CATEGORY                  AICAT                           PRE00190
C          DOT TYPE                    MASK(I,J)                        PRE00200
C          AI COMMENT                   AICMNT(I,J)                     PRE00210
C          AI CANOPY EVALUATION         IAIVEG(I)   I=1,4               PRE00220
C          BRIGHTNESS                   RDBRGT(I,J)  J=1,4              PRE00230
C          GREENESS                      GREEN(I,J)  J=1,4              PRE00240
C          SQUARE BRIGHTNESS            SQAIRB(I)                       PRE00250
C          SQUARE GREENNESS             SQAIRG(I)                       PRE00260
C          PI B                         PIFB(I)                         PRE00270
C          PI G                         PIFG(I)                         PRE00280
C          CANOPY KEY                   CANKY(I,J)      J=1,4           PRE00290
C             CANOPY TRAJECTORY         AICANK = ICANTJ(I)              PRE00300
C          ABSOLUTE BRIGHTNESS         ABRIET(I,J) J=1,4                PRE00310
C          ABSOLUTE GREENNESS         AGREEN(I,J) J=1,4                 PRE00320
C                                                                       PRE00330
0001        COMMON /LSEGS/KURTYP,KURSEG,KURDAT,LTYPE(30),LSEGNO(30),LSEGDT(30)PRE00340
           1  ,NOSEGS                                                   PRE00350
C  KURTYP,KURSEG,KURDAT  ARE THE CURRENT TYPE, SEGMENT AND DATE         PRE00360
C  LTYPE,LSEGNO,LSEGDT ARE THE REQUESTED LIST OF TYPES,SEG NOS AND DATESPRE00370
C  NOSEGS IS THE NUMBER OF SEGMENTS IN THE LIST                         PRE00380
C                                                                       PRE00390
0002        COMMON /RAWDAT/ MSTRID,RDERST,IRDERR,RDBRGT(1),19,4),       PRE00400
           1  RDGREN(1),19,4), IRDRAD, SQAIRB(1),19),SQAIRG(1),19),PIFB(1),19)PRE00410
           2  ,PIFG(1),19),ABRIET(1),19,4),AGREEN(1),19,4)              PRE00420
C                                                                       PRE00430
0003        COMMON/AI/IAIHED(73,2), IAIERR,IDAI,AICMNT(1),19),AIDATE(4) PRE00440
           1 ,NOLINE,NOPIXL,AIHDRR(4),IAIVEG(4),LABLAI(11,19),IAIEND    PRE00450
           2 ,MTCHAI,LNEROR(4),ISEASN,LNFS1(4),LNFS2(4)                 PRE00460
C                                                                       PRE00470
0004        COMMON /GT/ IGT(11,19), IGTERR, IGTFIL                      PRE00480
C                                                                       PRE00490
0005        COMMON /BIOSTG/ BIOSTG(4),BTOLMT(2,8),FIRST(6,8),SECOND(6,8) PRE00500
C                                                                       PRE00510
0006        COMMON /FLDATA/ ICNKEY(4),AICANK                            PRE00520
0007         COMMON /UN/NR1,IPPTO,NPRT,IRU,IWU                          PRE00530
C                                                                       PRE00540
0008        DATA KONSTC /1HC/                                           PRE00550
0009        DATA KONSTT /1T/                                            PRE00560
C                                                                       PRE00570
C                                                                       PRE00580
0010        WRITE(6,100)                                                PRE00590
0011    100  FORMAT(1H1,//,10X,'PROGRAM PREPPT',//)                     PRE00600
0012        WRITE(6,111)                                                PRE00610
0013    111  FORMAT (//,10X,' AT PRESENT ALL SOIL LINE NUMBERS',        PRE00620
           *  ' ARE SET TO 13.5',/,10X,' SEE SUBROUTINE CALCUL')       PRE00630
C                                                                       PRE00640
CC                                                                      PRE00650
C  INITIALIZE                                                           PRE00660
C                                                                       PRE00670
C  SET UP REREAD BUFFER                                                 PRE00680
0014        CALL REREAD (30,80)                                         PRE00690
C                                                                       PRE00700
C   READ SEGMENT NUMBERS AND DATES INTO /SEGNO/                         PRE00710
0015        REWIND 25                                                   PRE00720
0016        CALL READIO                                                 PRE00730
C                                                                       PRE00740
C   READ SEASONAL TABLE AND ROBERTSON BIOSTAGES TABLE INTO /SEASON/ AND PRE00750
C     /BIOSTG/                                                          PRE00760
```

14

FILE PREPPT

```
 0017             CALL READTR
            C
            C  INITIALIZE ERROR COUNTERS
 0018             IAIERR = 0
 0019             IRDERR = 0
 0020             IGTERR = 0
            C
            C
            C               *******  PROCESS DATA FOR EACH SEGMENT  *******
            C
 0021             DO 950 I = 1,NOSEGS
            C
 0022             KURTYP = LTYPE(I)
 0023             KURSEG = LSEGNO(I)
 0024             IDAI = KURSEG
 0025             MSTRID = KURSEG
 0026             KURDAT = LSEGDT(I)
            C
            C
 0027             IF(KURTYP.NE.KONSTC.AND.KURTYP.NE.KONSTT) GO TO 950
            C
            C  INITIALIZE VALUES FOR SEGMENT
 0028             CALL SGINIT
 0029             NOPIXL = 0
            C
            C  SET SWITCH FOR LOCATION OF LAST PIXEL TO 0 INDICATION NO PIXELS READ
            C
            C
            C  READ HEADER (ONCE PER DATA SET) AND 1 DATA CARD
 0030       10    CALL READAI
            C
 0031             IF (IAIEND .NE. 0) GO TO 1000
            C  CHECK FOR NO MATCHING SEGMENT HEADER CARDS OR PIXEL CARDS
 0032             IF (MTCHAI .NE. 1) GO TO 900
            C
 0033              IF (IAIERR .GE. 10) GO TO 990
            C
            C
            C  READ RAW DATA SET IF NEW ID
 0034             IF (IDAI .EQ. LASTID) GO TO 50
            C
 0035             CALL READRD
 0036             IF (IRDBAD .EQ. 1) GO TO 910
            C
            C  READ GROUND TRUTH UNLESS CLASSIFICATION RUN
 0037             IF (KURTYP .NE. KONSTC) CALL READGT
 0038             IF (KURTYP .NE. KONSTC  .AND. IGTFTL .NE. 0) GO TO 910
            C
            C  WRITE NEW RECORD
 0039       50    CALL WRTPPT
            C
 0040             LASTID = IDAI
            C
 0041             GO TO 10
            C
 0042       900   CONTINUE
            C
            C  END OF SEGMENT--END NEW FILE FOR SEGMENT BY WRITING NULL RECORD
            C
            C  SKIP WRITING NULL RECORD IF NO VALID RECORD FOUND FOR SEGMENT
 0043             IF (NOPIXL .EQ. 0) GO TO 950
            C
 0044              NOPIXL = 0
 0045              NOLINE = 0
 0046              CALL SGINIT
 0047              CALL WRTPPT
 0048             GO TO 950
            C
            C  BAD RAW DATA OR GROUND TRUTH, SKIP AI FILE
 0049       910   CALL SGINIT
 0050             CALL FSFMFL (5,1,ISTAT)
            C
 0051       950   CONTINUE
 0052             GO TO 1000
            C
            C  ERROR TERMINATION
```

FILE PREPPT

```
0053          990   WRITE (6,991)                                              PRE01530
0054          991   FORMAT (' PROGRAM TERMINATED DUE TO ERROR COUNT',/)        PRE01540
              C                                                                PRE01550
              C                                                                PRE01560
              C  END OF PROGRAM                                                PRE01570
0055          1000  CONTINUE                                                   PRE01580
              C                                                                PRE01590
              C  WRITE BLANK RECORD                                            PRE01600
0056                WRITE(IPPTO,707)                                           PRE01610
0057           707  FORMAT(/)                                                  PRE01620
              C                                                                PRE01630
              C  WRITE TWO ENDS OF FILE ON OUTPUT FILE                         PRE01640
0058                END FILE IPPTO                                             PRE01650
0059                END FILE IPPTO                                             PRE01660
              C                                                                PRE01670
0060                WRITE (6,1500) IAIERR,IRDERR,IGTERR                        PRE01680
0061          1500  FORMAT(' END OF PROGRAM', I4, ' AI ERRORS DETECTED',/,     PRE01690
                   1  15X,I4,' RAW DATA SEGMENTS MISSING',/,                   PRE01700
                   2  15X,I4,' GROUND TRUTH ERRORS')                          PRE01710
0062                STOP                                                       PRE01720
0063                END                                                        PRE01730
```

3.2.6 SOFTWARE COMPONENT NO. 6 (MPPTA)

The program MPPTA writes a loss vector matrix to unit no. 7 to be used by the processor MPPTC.

3.2.6.1 Linkages

The program MPPTA calls subroutines SPPTA, PPTA, CLOCK, GETIME, GETIME, GTDATE, and IDNAME. The subroutine SPPTA uses the function ALPHA and calls the subroutine SETTRM. The Subroutine PPTA calls READIT, NP, ALPHA, and PHI. Subroutines CLOCK, GETIME, GTDATE, and IDNAME are "system subroutines" and descriptive by name.

3.2.6.2 Interfaces

MPPTA interfaces with other routines through calling sequences, and common blocks UN and FV. The common blocks are initialized in PPTBLK.

3.2.6.3 Inputs

The List Variable Disk Files (Table 3-5).

3.2.6.4 Outputs

A loss vector matrix disk file is written.

3.2.6.5 Storage

Program size = 618702.

3.2.6.6 Description

The program MPPTA is the first of two processors used in sequence to classify input data using the Patterson-Pitt-Thandani algorithm for minimum loss classification. MPPTA writes a loss vector matrix to a disk data set to be used by the second processor MPPTC.

3.2.6.7 Flowchart

N/A

3.2.6.8 Listing

FILE MPPTA

```
                C   ADAPTED BY C W AHLERS
                C  THIS PROGRAM (MPPTA) USES THE FOLLOWING SUBROUTINES
                C   SPPTA
                C  SETTPM
                C  ALPHA
                C  MLO.
                C   PPTA
                C   READIT
                C   NP
                C   PHI
                C  THE PATTERSON-PITT-THADANI ALGORITHM.
                C  THIS PROGRAM USES UNITS RUNIT AND WUNIT FOR SCRATCH WORK.
                C  THE FINAL LOSS VECTOR MATRIX A IS OUTPUT TO UNIT   WUNIT.
                C  TRAINING.
                C  P(N1)....PHI FUNCTION VECTOR.
                C  Q(N1,T)....CLASS PHI SUM MATRIX.
                C  R(N1)....PN INVERSE * PHI FUNCTION VECTOR.
                C  PNI(N2)....PN INVERSE MATRIX.
   0001               INTEGER D,T,CAT,RUNIT,WUNIT
   0002               INTEGER CATREC(26)
   0003               DIMENSION ISGZ(26)
   0004               COMMON /TUN/NTRA
                      PARAMETER NN1=361,NN2=65522,TT=10,DD=30
   0005                COMMON /JN/NRDR1,NRDR2,NPRT,RUNIT,WUNIT
   0006               INTEGER FEATVC
   0007               COMMON /FV/FEATVC(30),IFMT(20),NDATA
   0008               DOUBLE PRECISION PNI(65522),P(361),Q(361,10),R(361),S(361,10)
   0009               DOUBLE PRECISION A(361,10),ALPHA,TRACE
   0010               DOUBLE PRECISION C(10,10),X(30)
   0011               DOUBLE PRECISION E
   0012                INTEGER USERID(2),NAME(4),TIME(3),DATE(3)
                C     WRITE(NTRA,999)
   0013         999   FORMAT(1H ,'MPPTA')
   0014               TIKTOK=0.
   0015               CALL CLOCK(TIKTOK)
   0016               WRITE(NPRT,100)
   0017         100   FORMAT(1H1,10X,'THE PATTERSON-PITT-THADANI ALGORITHM PROGRAM')
   0018                CALL GETIME(TIME)
   0019                CALL GTDATE(DATE)
   0020                CALL IDNAME(USERID,NAME)
   0021                WRITE(NPRT,200)USERID,NAME,DATE,TIME
   0022         200    FORMAT(//,10X,2A4,4X,4A4,4X,3A4,4X,3A4)
   0023               CALL SPPTA(D,T,ISGZ,NT,F,CATREC,C,N1,N2)
   0024               CALL PPTA(D,T,ISGZ,NT,N1,N2,PNI,P,Q,R,S,A,E,CATREC,C,X)
   0025               CALL CLOCK(TIKTOK)
   0026               WRITE(NPRT,300) TIKTOK
   0027         300    FORMAT(//,10X,'TIME FOR PPTA',F10.3)
   0028               STOP
   0029               END
```

3.2.7 SOFTWARE COMPONENT NO. 7 (MPPTC)

The program MPPTC uses the loss vector matrix file (made by the processor) MPPTA and the pixel data to compute the minimum loss classification.

3.2.7.1 Linkages

The program MPPTC calls subroutines SPPTC, PPTC, CLOSK GETIME, GTDATE, and IDNAME. The subroutine PPTC in turn calls READIT, IBETA, PRBSUM, PPTSUM, PPTDTL, PPTTAB, and PHI. The subroutines CLOCK, GETIME, and IDNAME are "system subroutines" and descriptive by name.

3.2.7.2 Interfaces

MPPTC interfaces with other routines through calling sequences and common blocks UN and FV. The common blocks are initialized in PPTBLK.

3.2.7.3 Inputs

A loss vector disk file.

3.2.7.4 Outputs

A computed dot label file in LACIE format.

3.2.7.5 Storage

Program size = 33054 bytes.

3.2.7.6 Description

The program MPPTC is the second of two processors used in sequence to classify the input data using the Patterson-Pitt-Thadani algorithm for minimum loss classification. MPPTC classifies the data using the loss vector matrix file computed by the first processor.

FILE MPPTC

```
                C   ADAPTED BY C W AHLERS                                                MPP00010
                C  THIS PROGRAM (MPPTC) USES THE FOLLOWING SUBROUTINES                   MPP00020
                C  SPPTC                                                                 MPP00030
                C  PPTC                                                                  MPP00040
                C  READIT                                                                MPP00050
                C  ETC                                                                   MPP00060
                C THE PATTERSON-PITT-THADANI CLASSIFIER.                                 MPP00070
     0001             INTEGER D,T,CAT,UNIT                                               MPP00080
                C     PARAMETER NMI=361,TT=10,DD=30                                      MPP00090
     0002             INTEGER RUNIT,WUNIT                                                MPP00100
     0003              COMMON /UM/NRDR1,NRDR2,NPRT,RUNIT,WUNIT                           MPP00110
     0004             INTEGER FEATVC                                                     MPP00120
     0005             COMMON /FV/FEATVC(30),IFMT(20),NDATA                               MPP00130
     0006             DOUBLE PRECISION A(361,10),L(10),P(361)                            MPP00140
     0007             DOUBLE PRECISION X(30)                                             MPP00150
     0008             INTEGER ISG7(10),CATREC(26)                                        MPP00160
     0009              INTEGER USERID(2),NAME(4),TIME(3),DATE(3)                         MPP00170
     0010             TIKTOK=0.                                                          MPP00180
     0011             CALL CLOCK(TIKTOK)                                                 MPP00190
     0012             WRITE(NPRT,100)                                                    MPP00200
     0013      100     FORMAT(1H1,10X,'THE PATTERSON-PITT-THADANI CLASSIFIER PROGRAM')    MPP00210
     0014              CALL GETIME(TIME)                                                 MPP00220
     0015              CALL GTDATE(DATE)                                                 MPP00230
     0016              CALL IDNAME(USERID,NAME)                                          MPP00240
     0017              WRITE(NPRT,200)USERID,NAME,DATE,TIME                              MPP00250
     0018      200     FORMAT(//,10X,2A4,4X,4A4,4X,3A4,4X,3A4)                           MPP00260
     0019             CALL SPPTC(UNIT,M,D,T,ISG7,NT,CATREC,NI,NP)                        MPP00270
     0020             CALL PPTC(M,D,T,ISG7,NT,UNIT,NI,A,L,P,CATREC,X,NP)                 MPP00280
     0021             CALL CLOCK(TIKTOK)                                                 MPP00290
     0022             WRITE(NPRT,300) TIKTOK                                             MPP00300
     0023      300     FORMAT(//,10X,'TIME FOR PPTC',F10.3)                              MPP00310
     0024             STOP                                                               MPP00320
     0025             END                                                                MPP00330
```

LIST Control Parameters

Beginning Processor	DAMRG, GTTCN, GTDDM, PREPPT, MPPTA, MPPTC or EOD-LARSYS
Ending Processor	DAMRG, GTTCN, GTDDM, PREPPT, MPPTA, MPPTC or EOD-LARSYS

For each segment the following information must be entered:

Type of processing (TRAIN, TEST, CLASSIFY)

Segment number

Segment date

Table 3-1

Robertson Biostage Table

Robertson biostage range	First class response	Second class response
1.0 – 2.0	No vegetation (0)	Green vegetation (1, 2, 3)
2.1 – 2.5	No vegetation or green vegetation (0, 1, 2, 3)	
2.6 – 3.0	Green vegetation (1, 2, 3)	No vegetation (0)
3.1 – 5.0	Green vegetation (1, 2, 3)	
5.1 – 5.5	Green vegetation or turning (1, 2, 3, 4)	Harvested (5)
5.6 – 6.0	Turning (4)	Green vegetation or harvested (1, 2, 3, 5)
6.1 – 6.9	Turning or harvested (4, 5)	
7.0	Harvested (5)	Turning (4)

Table 3-2

FOR WINTER WHEAT SITES (KANSAS)

BIO-LOWER LIMIT	BIO-UPPER LIMIT	MEAN G#	MEAN Br.	STD.DEV. G#	STD.DEV. Br.
0	1.0	1.25	74.38	2.58	10.22
1.1	2.0	(B10-1.0)*12.76/1.1 +(2.1-B10)*1.25/1.1	(B10-1.0)*75.67/1.1 +(2.1-B10)*74.38/1.1	(B10-1.0)*9.55/1.1 +(2.1-B10)*2.58/1.1	(B10-1.0)*17.04/1.1 +(2.1-B10)*10.22/1.1
2.1	2.9	12.76	75.67	9.55	17.04
3.0	3.1	(B10-2.9)*20.16/.3 +(3.2-B10)*12.76/.3	(B10-2.9)*77.06/.3 +(3.2-B10)*75.67/.3	(B10-2.9)*9.37/.3 +(3.2-B10)*9.55/.3	(B10-2.9)*13.25/.3 +(3.2-B10)*17.04/.3
3.2	3.8	20.16	77.06	9.37	13.25
3.9	4.6	(B10-3.8)*21.41/.9 +(4.7-B10)*20.16/.9	(B10-3.8)*67.7/.9 +(4.7-B10)*77.06/.9	(B10-3.8)*7.5/.9 +(4.7-B10)*9.37/.9	(B10-3.8)*8.0/.9 +(4.7-B10)13.25/.9
4.7	5.3	(B10-4.7)*13.66/.7 +(5.4-B10)*21.41/.7	(B10-4.7)*70.3/.7 +(5.4-B10)*67.7/.7	(B10-4.7)*5.7/.7 +(5.4-B10)*7.5/.7	(B10-4.7)*10.8/.7 +(5.4-B10)*8.0/.7
5.4	5.5	13.66	70.3	5.7	10.8
5.6	6.0	(B10-5.5)*5.46/.5 +(6.0-B10)*13.66/.5	(B10-5.5)*89.8/.5 +(6.0-B10)*70.3/.5	(B10-5.5)*7.4/.5 +(6.0-B10)*5.7/.5	(B10-5.5)*27.9/.5 +(6.0-B10)*10.8/.5
6.1	7.0	(B10-6.0)*(-1.36) +(7.0-B10)*5.46	(B10-6.0)*96.24 +7.0-B10)*89.8	(B10-6.0)*3.1 +(7.0-B10)*7.4	(B10-6.0)*12.5 +(7.0-B10)*27.9

Table 3-2

FOR SPRING WHEAT SITES (N.D.)

BIO-LOWER LIMIT	BIO-UPPER LIMIT	MEAN G#	MEAN Br.	ST.DEV. G#	ST.DEV. Br.
0	0.9	1.25	74.38	2.58	10.22
1.0	1.1	(B10-1.0)*12.6 +(1.2-B10)*6.25	(B10-1.0)*290.05 +(1.2-B10)*371.9	(B10-1.0)*15.5 +(1.2-B10)*12.9	(B10-1.0)*74.35 +(1.2-B10)*51.1
1.2	2.2	2.52	58.01	3.1	14.87
2.3	2.4	(B10-2.2)*38.95 +(2.4-B10)*12.6	(B10-2.2)*333.1 +(2.4-B10)*290.05	(B10-2.2)*43.1 +(2.4-B10)*15.5	(B10-2.2)*94.45 +(2.4-B10)*74.35
2.5	2.9	7.79	66.62	8.62	18.89
3.0	3.4	(B10-2.9)+44.83 +(3.5-B10)*12.98	(B10-2.9)*115.72 +(3.5-B10)*111.03	(B10-2.9)*23.27 +(3.5-B10)*14.37	(B10-2.9)*13.6 +(3.5-B10)*31.48
3.5	3.7	26.39	69.43	13.96	8.16
3.8	4.1	(B10-3.7)*65.32 +(4.2-B10)*52.78	(B10-3.7)*139.4 +(4.2-B10)*138.86	(B10-3.7)*23.48 +(4.2-B10)*27.92	(B10-3.7)*22.48 +(4.2-B10)*16.32
4.2	4.6	32.66	69.7	11.74	11.24
4.7	4.9	(B10-4.6)*60.73 +(5.0-B10)*81.65	(B10-4.6)*165.25 +(5.0-B10)*174.25	(B10-4.6)*24.53 +(5.0-B10)*29.35	(B10-4.6)*27.63 +(5.0-B10)*28.1
5.0	5.4	24.29	66.1	9.81	11.05
5.5	6.0	(B10-5.4)*15.28 +(6.0-B10)*40.48	(B10-5.4)*118.02 +(6.0-B10)*110.07	(B10-5.4)*14.98 +(6.0-B10)*16.35	(B10-5.4)*29.65 +(6.0-B10)*18.42
6.1	7.0	(B10-6.0)*4.42 +7.0-B10)*9.17	(B10-6.0)*59.04 +(7.0-B10)*70.81	(B10-6.0)*4.49 +(7.0-B10)*8.99	(B10-6.0)*6.78 +(7.0-B10)*17.79

Table 3-3

3-21
28

AI Response File

Record Type	Record Column	Type of Data
Header 1	2 - 5	Segment Number
	7	Season
	9 - 12 14 - 17 19 - 22 24 - 27	Julian Acquisition Dates
Header 2	2 - 5	Reference Acquisition Date
	11 - 12 15 - 17 20 - 22 25 - 27	Robertson Biostage Numbers for Acquisition Dates in Header 1.
Pixel	(Minimum of 1, maximum of 209 per segment)	
	2 - 3	Line Number
	6 - 7	Pixel Number
	10 - 13	AI Type of Pixel (1 value only)
	15 - 18	Vegetation Indication
	20 - 23	
	25	AI Label

Table 3-4

LIST Variable Input File

File Header Record

- Number of Segments - N
- N Segment Numbers

Segment Header Records

- Segment Header Record 1
- Segment Header Record 2

Pixel Records (Minimum of 1, maximum of 209)

- Line Number
- Column Number
- Ground Truth Category
- AI Label
- Dot Type Label
- AI Comment
- AI Vegetation Indication 1 to 4
- The following elements calculated as described in Appendix D where I = Pixel Number
  - BRIET(I,J) J=1,4
  - GREEN(I,J) J=1,4
  - SQAIRB(I)
  - SQAIRG(I)
  - PIEB(I)
  - PIEG(I)
  - CANKY(I,J) J=1,4
  - CANTJ(I)
  - ABRIET(I,J) J=1,4
  - AGREEN(I,J) J=1,4

3.2.8 SOFTWARE COMPONENT NO. 8 (EOD-LARSYS)

The modified EOD-LARSYS system constitutes the second subsystem of LIST processing. Modifications have been made to the DOTDATA and DISPLAY processors to read and process the various dot files created in the LIST system.

The DISPLAY processor has been enhanced to read any or all of the following dot files:

PPTC - output by the PPT classifier/discriminator
GT - ground truth, dot file
AI - analyst/interpreter dot file

For each, the Type 1 and Type 2 dots are displayed by category name in CCIT format vs the machine classification category name. The Type 2 dots are used to compute bias corrected proportion estimates of the "selected" category of interest.

The DOTDATA processor has been modified to accept, on option, a LACIE dot file in lieu of the in-line dots contained in the control card file. The output file created by DOTDATA is subsequently used in cluster labeling and classification under the Procedure 1 structure.

3.2.8.1 Linkages

The DISPLAY processor is called by routine MONTOR, the EOD-LARSYS system driver. It in turn calls several EOD-LARSYS system utilities, including FSBSFL, FMFSFL, WRTFLD, RDDOTS, WRTHED, WRTLN, WRTMTX, FIND12, NUMBER, and NXTCHR. It consists of routines DSPLAY, DSPLY1, DSPLY2, REDIF3, SETUP3, EMTHRS, DISTCV, CHIN, CHI, RNORM, TINORM, PCT, DESIG, FLDBOR, PRTSUM, MAPHD, FDIST, FISHIN, FISH, and PRTPCT.

Three new subprograms have been added, namely, LISTSM, LISTPR, and LISTLC. The DISPLAY processor uses blank common ARRAY (10600) and labeled common /GLOBAL/, /DISPL/, and /LISTMM/.

/LISTMM/ is a new common block included in LISTSM for the purpose of holding variable values from entry to entry.

The DOTDATA processor is called by MONTOR. It in turn calls several EOD-LARSYS utilities. It consists of subroutines DOTDAT, DOTS, SET13, FLDTYP, and WRTDOT.

It uses blank common ARRAY (10600), and labeled common /DOTVEC/, /GLOBAL/,/ISOLNK/, and /INFORM/.

No new subroutines or common blocks have been added for inclusion of LIST processing.

3.2.8.2 Interfaces

The DISPLAY processor accepts as input the following files:

a. A machine classification map (MAPTAP) written by the CLASSIFY processor.

b. On option, a dot file (DOTFIL) written by the DOTDATA processor for obtaining bias corrected proportion estimates (not used in LIST processing)

c. Under the LIST option, any or all of the following files, all in LACIE dot format, each with an 80 character header and a $END card image as a delimitor at the end of the file, can be input:

   PPTC discriminator labeled dot file
   GT ground truth dot file
   AI analyst interpreter dot file (any dot file could be used)

   Each file may have Type 1 and Type 2 or both Type 1 and 2 dots.

Output from the DISPLAY processor consists of a formatted (Universal or LARSYS II) classification display file suitable for display on the PMIS-DAS.

Printer output includes dot summaries, either under standard P1 or under the LIST structure.

The DOTDATA processor accepts a raw data file (MSS data file) as input. In LIST processing, this file may have been written by the DAMRG processor of EOD-LARSYS (processor to merge MSS data files).

Output from the DOTDATA processor consists of dot files containing named Type 1 and Type 2 dots for subsequent usage by other EOD-LARSYS processors.

3.2.8.3 Inputs

o Processor Card

$DISPLAY

o New Control Cards

Keyword	Parameter	Function
GTUN	UNIT = n, FILE = m	Unit and file number of ground truth dot file. Normally n=23, m=1 should be used.
PPUN	UNIT = n, FILE = m (no default)	Unit and file number of PPTC dot file. Normally n=27, m=1 should be used.

Keyword	Parameter	Function
AIUN	UNIT = n, FILE = m (no default)	Unit and file number of AI dot file. Normally n=28, m=1. Note: Inclusion of any one of the 3 cards above triggers the LIST processing option.
NAME	(one character name of selected category) default S	The category name of the selected category must be specified for proportion estimate computations. S in this case is taken to mean "small grains".
ALPHA	N_1, N_2 floating point values of α_1 and α_2 for bias correction. default .5,.5	Values of $\{\alpha_1\}$ used in bias correction.

o Processor Card

$DOTDATA

o Revised Control Card

Keyword	Parameter	Function
OPTION	U n (no default)	n = unit number of the LACIE formatted dot file to be used in lieu of in-line dot cards in the control card file. Normally n=29 should be used. File 1 on unit n is assumed.

3.2.8.4 Outputs

See sub-section 3.2.8.2.

3.2.8.5 Storage Requirements

The EOD-LARSYS system, overlaid by processors, requires less than 768 kilobytes of computer storage on the IBM 370/148 computer at Purdue-LARS.

3.2.8.6 Description

DISPLAY: Three new subroutines have been added to the DISPLAY processor to accomodate LIST processing. These are

1. LISTSM - called from DSPLY2, reads the dot files requested, prepares summary information, and writes out proportion estimations for the selected category. Variables computed are described as follows, for Type 2 dots.

   PMACH(1) machine classified proportion estimate of the selected category. Designated other or unidentifiable, and thresholded pixels are not included.

   PMACH(2) machine classified proportion estimate of the set of all categories other than the selected one. Designated other or unidentifiable, and thresholded pixels are not included.

   N11 number of dots labeled (by PPTC, GT, or AI) as selected category and machine classified as belonging to the selected category.

   N12 number of dots labeled as selected category but classified into one of the other categories. Designated or thresholded dots not included.

N_{21} number of dots labeled in the non-selected categories, but machine classified into the selected category.

N_{22} number of dots both labeled and classified into the non-selected categories.

N_{B1} number of dots labeled "boundary" but classified into the selected category.

N_{B2} number of dots labeled "boundary" but classified into the non-selected categories.

Regarding labeling, the symbols (names) M and R are used to denote boundary dots, and the symbol D is used to denote designated unidentifiable.

Probability estimates computed and outputted are

P11 = N11/(N11 + N21 + NB1) ,

P12 = N12/(N12 + N22 + NB2) ,

PB1 = NB1/(N11 + N21 + NB1) ,

PB2 = NB2/(N12 + N22 + NB2) ,

The proportion estimate for the selected category is computed as

```
P = PMACH(1)*(P11 + ALP(1)*PB1)
  + PMACH(2)*(P12 + ALP(2)*PB2)
```

2. LISTPR - called by LISTSM, prints the CCIT tables for Type 1 and Type 2 dots

PPC vs machine classification

GT vs machine classification

AI vs machine classification

3. LISTLC - called from LISTSM, this subprogram reads the dot files specified by control cards GTUN, PPUN, and AIUN. This subroutine is almost identical to FLDLAC in structure, however, two switches in DATA have been made calling sequence arguments.

Modification to DSPLY2 include the call to LISTSM, and transfers through the subroutine based upon setting of the LIST processing switch LISTSW. If LISTSW is on, the machine classification results for all 209 dots in the LACIE grid are retrieved and passed to LISTSM for use.

REDIF3 has been modified to accept the new control cards. Calling sequences of REDIF3, SETUP3, and DSPLY2 have been expanded to pass the extra information along. The extra variables are

```
LISTSW
GTUNIT
GTFILE
AIUNIT
AIFILE
PPUNIT
PPFILE
NAMECT
ALP(2)      REAL*4
```

A new common block /LISTMM/ has been incorporated into subroutine LISTSM to hold information previously stored during the initialization. This common block consists of

```
NPGA(3,2)
NAMPGA(209,3,2)          dot category names
                         of labeled dots in the order
                         of input
```

LINPGA(209,3,2)	dot line numbers
SAMPGA(209,3,2)	dot sample numbers
DOTLAB(209,4,2)	dot category names of labeled dots in LACIE grid position for printout.
VPGA(3)	indicates type of file PPTC=1, GT=2, AI=3
IPGA	number of files read in of PPTC, GT, AI

The "2" storage is for Type 1 and Type 2 dots.

Under LIST processing, DESIGNATED OTHER or UNIDENTIFIABLE fields can be used (as previously) to change the machine classification. Also, thresholding can be performed, as previously.

DOTDATA:

Routines DOTS, RDDOTS, and SET13 have been modified to enable handling of the new option to read in a unit number of the unit holding the LACIE formatted dots. The variable LACIE contains the unit number.

In ordinary P1 processing, LACIE is set to the value 1 if in-line dots are LACIE formatted, and is set to the value 0 if in-line dots have the field format.

```
C      // JOB 40
C
CI-----------------------------------------------------------------
CI
CI     CALL..    SYSTEM MONITOR  (// EXEC  LARSYSAA )
CI
CI     PURPOSE.. MONITORS THE VARIOUS SYSTEM SUPERVISORS
CI
CI               ROUTINES MSCAN   CLSFY  DSPLAY  STAT
CI                        SELECT  HIST   GRAYMP  DATATR
CI                        ISOCLS  TRSTAT  NDHIST  SCTPPL
CI                .       OUTDATA   LABEL    EQUIPR  NULBAY
CI                        STICN   DAURG
CI                        ANDERR CLASY TESTSP GTDOM EXIT
CI     RETURNS.. NONE
CI
CI-----------------------------------------------------------------
C
C
      IMPLICIT INTEGER(A-H,O-Z)
      COMMON /ARRAY/
      DIMENSION ARRAY(10600)
C*
C*    'ARRAY' IS A BLOCK OF STORAGE PASSED TO EACH PROCESSOR FOR THE
C*    VARIABLE DIMENSIONING OF OTHER ARRAYS. THE ARRAY IS NEVER USED
C*    TO PASS INFORMATION  FROM ONE PROCESSOR TO ANOTHER.
C*
      DATA IOP/10600/
C     INCLUDE COMSTG,LIST
C
C
C     INCLUDE COMSTG,LIST
      COMMON/GLOBAL/HEAD(43),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
     *          HISFIL,HISKEY,TRFORM,FRIPTP,ERPKEY,MAPUNT,NOFILE,
     *    DRUMAD,DRWRDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
     *  ,INSTOP,HISTFI,SCTRUN,MAPFIL
     *  ,OUTU,T,OUTFIL,VCMPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
     *  CPUUNT,PRTUNT,HANDIO
C*
C*    GLOBAL COMMON IS USED IN EVERY PROCESSOR. IT IS ALWAYS IN CORE.
C*    ALL PARAMETERS ARE INITIALIZED IN THE MONITOR ROUTINE OR BLKCOM
C*    EXCEPT AS NOTED BELOW
C*    DEFINITIONS
C*         HEAD   - STANDARD HEADING PRINTED ON MOST OUTPUT PAGES.
C*         MAPTAP - FORTRAN UNIT NUMBER ON WHICH THE MAPTAP FILE IS
C*                  WRITTEN (=2)
C*         IMAGE - UNIT NO. FOR THE IMAGE DATA TAPE (=1)
C*         SAVTAP - UNIT NO. ON WHICH THE STATISTICS FILE IS WRITTEN (=11)
C*         BMFILE - UNIT NO. ON WHICH THE B-MATRIX FILE IS WRITTEN (=10)
C*         BMKEY - TRIGGER INDICATING THAT THE B-MATRIX FILE HAS BEEN
C*                  WRITTEN. CAN BE SET IN SELECT CLASSIFY OR DATA-TR.
C*         HISFIL - UNIT NO. ON WHICH THE HISTOGRAM FILE IS WRITTEN (=13)
C*         HISKEY - TRIGGER INDICATING THE HISTOGRAM FILE HAS BEEN
C*                  WRITTEN. SET IN HIST PROCESSOR.
C*         TRFORM - UNIT NO. ON WHICH THE TRANSFORMED IMAGE IS WRITTEN BY
C*                  THE DATA-TRANSFORMATION PROCESSOR. (=14)
C*         FRIPTP - UNIT NO. ON  WHICH THE ISOCLS PROCESSOR WRITES
C*                  CLUSTER STATISTICS FOR THE FRIPS SYSTEM. (=15)
C*         ERPKEY - TRIGGER INDICATING THAT THE FRIPS INTERFACE TAPE
C*                  HAS BEEN WRITTEN.
C*         MAPUNT - UNIT NO. ON WHICH THE ISOCLS OR DISPLAY PROCESSOR
C*                  WRITES THE CLUSTERED OR CLASSIFIED DATA
C*                  TO BE DISPLAYED ON THE PRIS DAS
C*         NOFILE - NO. OF FILES WRITTEN ON UNIT 16 (MAP OUTPUT TAPE)
C*                  BY DISPLAY AND/OR ISOCLS
C*                  SET EITHER IN ISOCLS OR DISPLAY.
C*         DRUMAD - BEGINNING ADDRESS FOR THE RANDOM ACCESS HIGH SPEED
      REAL TIME
C*                  DRUM FILE. THIS FILE IS USED AS A SCRATCH FILE IN
C*                  SEVERAL PROCESSORS. REFERENCES TO SYSTEM ROUTINES
C*                  'RREAD' AND 'RWRITE' ACCESS THIS FILE.
C*         DRWRDS - NO. OF WORDS AVAILABLE ON THE RANDOM ACCESS FILE.
C*         PAGSIZ - NO. OF LINES AVAILABLE FOR PRINTING ON A PAGE.
C*         DATFIL - NO. OF E-O-F'S TO BE READ OVER BY TAPERD ROUTINE IN
C*                  ORDER TO POSITION THE DATA TAPE TO DESIRED FILE
C*         STAFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION STAT FILE.)
C*           ASAV - UNIT NO. ON WHICH TRSTAT WRITES THE TRANSFORMED
C*                  STATS
C*         ASAVFL - NO. OF E-O-F'S TO SKIP OVER TO POSITION TRANSFORMED
```

```
C*                    STATS                                                  MON00800
C*          OUTUNT - UNIT NO. ON WHICH OUT DATA FILE (OUTFIL) IS WRITTEN     MON00810
C*          OUTFIL - NO. OF E-O-F S TO SKIP OVER TO POSITION OUTFIL FILE     MON00820
C*          NCHANS - NO. OF CHANNELS PER PASS                                MON00830
C*          TAPSFL - NO. OF E-O-F'S TO SKIP OVER FOR TAPSAM FILE             MON00840
C*          DATAFL - NO. OF E-O-F'S TO SKIP OVER FOR DMFIL FILE              MON00850
C*          HISTFL - NO. OF E-O-F'S TO SKIP OVER FOR HISFIL FILE             MON00860
C*          PUNCH - UNIT NO. FOR CARD PUNCH FILE                             MON00870
C*          CRUNIT - UNIT NO. FOR CARD READER                                MON00880
C*          SCRUNIT - SCRATCH UNIT FOR RREAD AND RWRITE ROUTINES             MON00890
C$END                                                                        MON00900
      DRUM=-1                                                                MON00910
C*                                                                           MON00920
C*    SYSTEM ROUTINE RINIT ASSIGNS THE RANDOM ACCESS DRUM FILE.              MON00930
C*                                                                           MON00940
C*          -DRUM(0)- IS THE ADDRESS TO BEGIN WRITING                        MON00950
C*          -DRMWDS- IS THE NO. OF WORDS AVAILABLE ON THE DRUM FILE.         MON00960
C*                                                                           MON00970
C*    THE FOLLOWING PROCESSORS USE THE RANDOM ACCESS DRUM FILE FOR SCRATCH   MON00980
C*          -ISOCLS-                                                         MON00990
C*          -DISPLY-                                                         MON01000
C*          -SELECT-                                                         MON01010
C*                                                                           MON01020
C*          -HRAY P-                                                         MON01030
C*          -STUFAT-                                                         MON01040
C*                                                                           MON01050
      DEFI F FILE 22(640,200,U,ID)                                           MON01060
      DRUM(1)=1                                                              MON01070
      DRMWDS=120000                                                          MON01080
      WRITE(22'1)DRUMAD                                                      MON01090
   10 CONTINUE                                                               MON01100
      TIME = 0.                                                              MON01110
      CALL CLOCK(TIME)                                                       MON01120
      CALL ASCAN(JGO,DRUG)                                                   MON01130
      GO TO (20,40,60,80,100,120,140,160,175,180,200,220,240,260,            MON01140
     * 280,290,300,310,320,330,340,350,360),JGO                              MON01150
C                                                                            MON01160
   20 CONTINUE                                                               MON01170
      CALL STAT(ARRAY,TOP)                                                   MON01180
      CALL CLOCK(TIME)                                                       MON01190
      WRITE (6,30)TIME                                                       MON01200
   30 FORMAT(' TIME FOR STAT',F10.3)                                         MON01210
      GO TO 10                                                               MON01220
C                                                                            MON01230
C                                                                            MON01240
   40 CONTINUE                                                               MON01250
      CALL CLSFY(ARRAY,TOP)                                                  MON01260
      CALL CLOCK(TIME)                                                       MON01270
      WRITE (6,50)TIME                                                       MON01280
   50 FORMAT(' TIME FOR CLASSIFY',F10.3 )                                    MON01290
      GO TO 10                                                               MON01300
C                                                                            MON01310
C                                                                            MON01320
   60 CONTINUE                                                               MON01330
      CALL DSPLAY(ARRAY,TOP)                                                 MON01340
      CALL CLOCK(TIME)                                                       MON01350
      WRITE (6,70)TIME                                                       MON01360
   70 FORMAT(' TIME FOR DISPLAY',F10.3)                                      MON01370
      GO TO 10                                                               MON01380
C                                                                            MON01390
C                                                                            MON01400
   80 CONTINUE                                                               MON01410
      CALL SELECT(ARRAY,TOP)                                                 MON01420
      CALL CLOCK(TIME)                                                       MON01430
      WRITE (6,90)TIME                                                       MON01440
   90 FORMAT(' TIME FOR SELECT',F10.3)                                       MON01450
      GO TO 10                                                               MON01460
C                                                                            MON01470
C                                                                            MON01480
  100 CONTINUE                                                               MON01490
      CALL HIST(ARRAY,TOP)                                                   MON01500
      CALL CLOCK(TIME)                                                       MON01510
      WRITE(6,110)TIME                                                       MON01520
  110 FORMAT(' TIME FOR HISTOGRAM',F10.3)                                    MON01530
      GO TO 10                                                               MON01540
C                                                                            MON01550
C     GO HERE FOR ISOCLS                                                     MON01560
C                                                                            MON01570
  120 CONTINUE                                                               MON01580
```

```
      CALL ISOCLS(ARRAY,TOP)                                            MON01590
      CALL CLOCK(TIME)                                                  MON01600
      WRITE(6,130) TIME                                                 MON01610
  130 FORMAT(' TIME FOR ISOCLS',F10.3)                                  MON01620
      GO TO 10                                                          MON01630
C                                                                       MON01640
C     GO HERE FOR GRAYMAP                                               MON01650
  140 CONTINUE                                                          MON01660
      CALL GRAYMAP(ARRAY,TOP)                                           MON01670
      CALL CLOCK(TIME)                                                  MON01680
      WRITE(6,150) TIME                                                 MON01690
  150 FORMAT(' TIME FOR GRAYMAP',F10.3)                                 MON01700
      GO TO 10                                                          MON01710
C                                                                       MON01720
C     GO HERE FOR DATA-TRANSFORMATION                                   MON01730
C                                                                       MON01740
  160 CONTINUE                                                          MON01750
      CALL DATATR(ARRAY,TOP)                                            MON01760
      CALL CLOCK(TIME)                                                  MON01770
      WRITE(6,170) TIME                                                 MON01780
  170 FORMAT(//' TIME FOR DATA-TRANSFORMATION',F10.3)                   MON01790
      GO TO 10                                                          MON01800
C*                                                                      MON01810
C*    GO HERE FOR SIGEXT MODULE                                         MON01820
C*                                                                      MON01830
  175 CONTINUE                                                          MON01840
C ***** SIGEXT                                                         MON01850
      CALL CLOCK(TIME)                                                  MON01860
      WRITE(6,176)TIME                                                  MON01870
  176 FORMAT(//' TIME FOR SIGEXT ',F10.3,' MINUTES' )                   MON01880
      GO TO 10                                                          MON01890
C                                                                       MON01900
C     GO HERE FOR TRSTAT                                                MON01910
C                                                                       MON01920
  180 CONTINUE                                                          MON01930
      CALL TRSTAT(ARRAY,TOP)                                            MON01940
      CALL CLOCK(TIME)                                                  MON01950
      WRITE(6,190)TIME                                                  MON01960
  190 FORMAT(//' TIME FOR TRSTAT',F10.3,' MINUTES')                     MON01970
      GO TO 10                                                          MON01980
C                                                                       MON01990
C     GO HERE FOR NOHIST                                                MON02000
C                                                                       MON02010
  200 CONTINUE                                                          MON02020
      CALL NOHIST(ARRAY,TOP)                                            MON02030
      CALL CLOCK(TIME)                                                  MON02040
      WRITE(6,210)TIME                                                  MON02050
  210 FORMAT(//' TIME FOR NOHIST',F10.3,' MINUTES')                     MON02060
      GO TO 10                                                          MON02070
C                                                                       MON02080
C     GO HERE FOR SCTRPL                                                MON02090
C                                                                       MON02100
  220 CONTINUE                                                          MON02110
      CALL SCTRPL(ARRAY,TOP)                                            MON02120
      CALL CLOCK(TIME)                                                  MON02130
      WRITE(6,230)TIME                                                  MON02140
  230 FORMAT(//' TIME FOR SCTRPT',F10.3,' MINUTES')                     MON02150
      GO TO 10                                                          MON02160
C                                                                       MON02170
C                                                                       MON02180
C     GO HERE FOR DOTDATA                                               MON02190
C                                                                       MON02200
  240 CONTINUE                                                          MON02210
      CALL DOTDAT(ARRAY,TOP)                                            MON02220
      CALL CLOCK(TIME)                                                  MON02230
      WRITE(6,250)TIME                                                  MON02240
  250 FORMAT(//' TIME FOR DOTDATA',F10.3,' MINUTES')                    MON02250
      GO TO 10                                                          MON02260
C                                                                       MON02270
C     GO HERE FOR LABEL                                                 MON02280
C                                                                       MON02290
  260 CONTINUE                                                          MON02300
      CALL LABEL(ARRAY,TOP)                                             MON02310
      CALL CLOCK(TIME)                                                  MON02320
      WRITE(6,270)TIME                                                  MON02330
  270 FORMAT(//' TIME FOR LABEL',F10.3,' MINUTES')                      MON02340
      GO TO 10                                                          MON02350
C                                                                       MON02360
C      GO HERE FOR EQUI-PROBABLE BLOCKS CLASSIFIER                      MON02370
```

```
C
  280  CONTINUE                                                          MON02380
       CALL EQUPRH(ARRAY,TOP)                                            MON02390
       CALL CLOCK(TIME)                                                  MON02400
       WRITE(6,285) TIME                                                 MON02410
  285  FORMAT(//' TIME FOR EQUPRH',F10.3,' MINUTES')                     MON02420
       GO TO 10                                                          MON02430
C                                                                        MON02440
CC     GO HERE FOR MULTI-TEMPORAL CLASSIFIER                             MON02450
C                                                                        MON02460
  290  CONTINUE                                                          MON02470
       CALL MULMAY(ARRAY,TOP)                                            MON02480
       CALL CLOCK(TIME)                                                  MON02490
       WRITE(6,295) TIME                                                 MON02500
  295  FORMAT(//' TIME FOR MULMAY',F10.3,' MINUTES')                     MON02510
       GO TO 10                                                          MON02520
C                                                                        MON02530
CC     GO HERE FOR GROUND TRUTH TO MAPFIL                                MON02540
C                                                                        MON02550
  300  CONTINUE                                                          MON02560
       CALL GTTOM(ARRAY,TOP)                                             MON02570
       CALL CLOCK(TIME)                                                  MON02580
       WRITE(6,305) TIME                                                 MON02590
  305  FORMAT(//' TIME FOR GTTOM',F10.3,' MINUTES')                      MON02600
       GO TO 10                                                          MON02610
C                                                                        MON02620
CC     GO HERE FOR IMAGE DATA MERGE                                      MON02630
  310  CONTINUE                                                          MON02640
       CALL DAMRG(ARRAY,TOP)                                             MON02650
       CALL CLOCK(TIME)                                                  MON02660
       WRITE(6,315) TIME                                                 MON02670
  315  FORMAT(//' TIME FOR DAMRG',F10.3,' MINUTES')                      MON02680
       GO TO 10                                                          MON02690
C                                                                        MON02700
CC     GO HERE FOR AMOEBA                                                MON02710
C                                                                        MON02720
  320  CONTINUE                                                          MON02730
       CALL AMOEBA(ARRAY,TOP)                                            MON02740
       CALL CLOCK(TIME)                                                  MON02750
       WRITE(6,325) TIME                                                 MON02760
  325  FORMAT(//' TIME FOR AMOEBA',F10.3,' MINUTES')                     MON02770
       GO TO 10                                                          MON02780
C                                                                        MON02790
CC     GO HERE FOR CLASY                                                 MON02800
C                                                                        MON02810
  330  CONTINUE                                                          MON02820
       CALL CLASY(ARRAY,TOP)                                             MON02830
       CALL CLOCK(TIME)                                                  MON02840
       WRITE(6,335) TIME                                                 MON02850
  335  FORMAT(//' TIME FOR CLASY',F10.3,' MINUTES')                      MON02860
       GO TO 10                                                          MON02870
C                                                                        MON02880
CC     GO HERE FOR TESTSP                                                MON02890
C                                                                        MON02900
  340  CONTINUE                                                          MON02910
       CALL TESTSP(ARRAY,TOP)                                            MON02920
       CALL CLOCK(TIME)                                                  MON02930
       WRITE(6,345) TIME                                                 MON02940
  345  FORMAT(//' TIME FOR TESTSP',F10.3,' MINUTES')                     MON02950
       GO TO 10                                                          MON02960
C                                                                        MON02970
CC     GO HERE FOR GROUND TRUTH DOT UNLOAD                               MON02980
C                                                                        MON02990
  350  CONTINUE                                                          MON03000
       CALL GTDOM(ARRAY,TOP)                                             MON03010
       CALL CLOCK(TIME)                                                  MON03020
       WRITE(6,355) TIME                                                 MON03030
  355  FORMAT(//' TIME FOR GTDOM',F10.3,' MINUTES')                      MON03040
       GO TO 10                                                          MON03050
C                                                                        MON03060
CC     GO HERE TO EXIT                                                   MON03070
C                                                                        MON03080
  360 IF (NOFILE .GT. 0) REWIND MAPUNT                                   MON03090
      END                                                                MON03100
                                                                         MON03110
```

4. OPERATION

The LIST processor, implemented as a system of processors called by an executive file LST, uses the Purdue-LARS 370/148 computer system.

The following input files are required:

Card image file giving type of processing, segment numbers and dates

4-channel MSS data files (4)

1-channel ground truth file

Robertson Biostage file

Seasonal constants file

AI responses file

MPPTA control cards

MPPTC control cards

PREPPT output file

Classification interface file

The card image file containing the type of processing, segment numbers and dates is the control card file for the software component PREPPT (unit no. 25).

The 4-channel MSS data files (4) are made into one 16-channel merged file by the software component DAMRG. This file is then used by software component PREPPT. The 16-channel file is on unit number 12.

The one-channel ground truth file (unit no. 11) (Accuracy Assessment form) is used by the software component GTTCN to make another image file (unit no. 12) which is LANDSAT size. This file is used by software component GTDDM to make a LACIE format dot file (unit no. 23) which is used by PREPPT. This file might also be input by hand.

The Robertson Biostage file (unit no. 24) is a permanent file used by software component PREPPT.

The seasonal file (unit no. 26) is a permanent file used by software component PREPPT. This file contains tables for computing the mean and standard deviations for greenness and brightness from biostage number.

The AI response files (unit no. 5) contain label, registration, vegetation information, and biostage numbers for each segment. These files (one for each segment) are used by software component PREPPT.

The PREPPT output file (unit no. 15) is the file made by PREPPT from the above files to be used by the training (PPTA) and the classification software components.

The MPPTA control card file is the control card file for the training program (unit no. 8).

The MPPTC control card file is the control card file for the classification program (unit no. 8).

The classification interface file (unit no. 7) is made by the training program and is to be used by the classification program. It contains the loss vector matrix and other data needed by the classification program.

The executive file LST is a prompting executive that queries the user and thus leads him through the whole LIST process by logical steps.

LST expects the input files to be in place at executio- time, but allows for editing control card files. LST gives file specifications, checks tape header records and file contents on request. By using secondary responses the user may branch to certain sections of LST, thus, by-passing certain processors.

APPENDIX A

A.1 SOFTWARE SUBPROGRAM NO. 1 (READID)

READID reads the LIST control cards.

A.1.1 Linkages

READID is called by PREPPT.

A.1.2 Interfaces

Interface is accomplished through common block /SGMENT/.

A.1.3 Inputs

List control file giving types of processing, segment numbers and dates from unit number 25.

A.1.4 Outputs

None

A.1.5 Storage Requirement

Program size = 414.

A.1.6 Description

READID stores the data from the LIST Control File in common block /SGMENT/.

A.1.7 Flowchart

N/A

A.1.8 Listings

FILE PREPRT

```
0001               SUBROUTINE READIO                                                PRE11420
            C                                                                      PRE11430
            C  PURPOSE:  READ SEGMENT NUMBER TYPES, NUMBERS AND DATES               PRE11440
            C            REQUESTED BY USER                                          PRE11450
            C                                                                      PRE11460
0002               COMMON /LSEGS/KURTYP,KURSEG,KURDAT,LTYPE(30),LSEGNO(30),LSEGDT(30)PRE11470
                  1   ,NOSEGS                                                      PRE11480
            C                                                                      PRE11490
0003               NOSEGS = 0                                                       PRE11500
            C                                                                      PRE11510
0004        10     NOSEGS = NOSEGS + 1                                              PRE11520
0005               READ (25,20,END=900) LTYPE(NOSEGS), LSEGNO(NOSEGS),LSEGDT(NOSEGS) PRE11530
0006        20     FORMAT (1X,A4,1X,A4,1X,A4)                                       PRE11540
            C                                                                      PRE11550
0007               GO TO 10                                                         PRE11560
            C                                                                      PRE11570
0008        900    NOSEGS = NOSEGS - 1                                              PRE11580
0009               RETURN                                                           PRE11590
0010               END                                                              PRE11600
```

A.2 SOFTWARE SUBPROGRAM 2 (READTB)

READTB reads the Seasonal Table File and the Biostage Response File.

A.2.1 Linkages

READTB is called by PREPPT.

A.2.2 Interfaces

Interface is accomplished through common blocks /BIOSTG/, /SEASNS/.

A.2.3 Inputs

Seasonal Table File
Biostage Response File
From unit number 26

A.2.4 Outputs

None

A.2.5 Storage Requirement

Program size = 900.

A.2.6 Description

The data from the Biostage Response File is stored in /BIOSTG/.
The data from the Seasonal Table File is stored in /SEASNS/.

A.2.7 Flowchart

N/A

A.2.8 Listings

FILE PREPPT

```
0001              SUBROUTINE READTB
           C
           C
           C   THE PURPOSE OF THIS SUBROUTINE IS TO READ THE SEASONS AND BIOSTAGES
           C
0002              COMMON /BIOSTG/ BIOSTG(4), BIOLMT(2,8), FIRST(6,8), SECOND(6,8)
0003              COMMON /SEASNS/ XBIO(4,13,2), XMEANG(13,2), XMENG2(13,2),
                1  XMEANB(13,2), XMENB2(13,2), SDG(13,2), SDG2(13,2), SDB(13,2),
                2  SDB2(13,2),LNESEA(4)
           C
           C   READ ROBERTSON BIOSTAGE TABLE
0004              DO 110 J = 1,9
0005              READ (24,100,END=900) BIOLMT(1,J), BIOLMT(2,J), (FIRST(I,J),I=1,6)
                1 ,(SECOND(I,J),I=1,6)
0006       100    FORMAT (F2.1,3X,F2.1,2X,6A1,4X,6A1)
0007       110    CONTINUE
           C
           C   READ SEASONS TABLE
0008       120    DO 300 ISEASN = 1,2
           C
0009              DO 300 LINE = 1,13
0010              READ (26,200,END=950) (XBIO(I,LINE,ISEASN),I=1,4), XMEANG(LINE,
                1  ISEASN),XMENG2(LINE,ISEASN),XMEANB(LINE,ISEASN),
                2  XMENB2(LINE,ISEASN),SDG(LINE,ISEASN),SDG2(LINE,ISEASN),
                3  SDB(LINE,ISEASN),SDB2(LINE,ISEASN)
0011       200    FORMAT (F3.1,F4.1,F3.1,F4.1,8(F7.2))
0012       300    CONTINUE
           C
0013              RETURN
           C
           C     ************  ERROR MESSAGES   **********
           C
           C   BIOSTAGE TABLE EMPTY
0014       900    WRITE (5,910)
0015       910    FORMAT (' BIOSTAGE TABLE EMPTY')
0016              GO TO 120
           C
           C   SEASONS TABLE EMPTY
0017       950    WRITE (6,960)
0018       960    FORMAT (' SEASONS TABLE EMPTY')
0019              RETURN
0020              END
```

A.3 SOFTWARE SUBPROGRAM NO. 3 (READAI)

READAI reads one set of responses from the AI Response File.

A.3.1 Linkages

READAI is called by PREPPT.

A.3.2 Interfaces

Interface is accomplished through common block /AI/.

A.3.3 Inputs

AI Response File from unit no. 5.

A.3.4 Outputs

None

A.3.5 Storage Requirement

Program size = 4286.

A.3.6 Description

One set of responses from the AI Response File is stored in common block /AI/.

A.3.7 Flowchart

N/A

A.3.8 Listings

FILE PREPPT

```
0001                SUBROUTINE READAI                                                   PRE01740
                C                                                                       PRE01750
                CCC                                                                     PRE01760
                C     PURPOSE--READ FILE AI, VALIDATE DATA, FIND INDICES TO TABLES      PRE01770
                C        FIRST CALL TO READAI FOR SEGMENT                               PRE01780
                C           FIND FIRST MATCHING HEADER CARD                             PRE01790
                C           VALIDATE AND SAVE SEASON                                    PRE01800
                C           READ NEXT CARD, MUST BE SECOND HEADER CARD                  PRE01810
                C           VALIDATE AND SAVE ROBERTSON BIOSTAGES                       PRE01820
                C           VALIDATE AND SAVE SEASONAL BIOSTAGES                        PRE01830
                C           READ FIRST PIXEL AND CALL READPX TO PROCESS IT              PRE01840
                C        SUCCESSIVE CALLS TO READAI FOR SAME SEGMENT                    PRE01850
                C           READ PIXEL AND CALL READPX TO PROCESS IT                    PRE01860
                C                                                                       PRE01870
                C     MTCHAI IS SET TO 1 IF A GOOD PIXEL CARD IS READ FOR THE SEGMENT   PRE01880
                C     MTCHAI IS SET TO 0 IF NO HEADER CARDS ARE FOUND OR IF NO VALID    PRE01890
                C                        PIXEL CARD IS READ.                            PRE01900
                C                                                                       PRE01910
                C                                                                       PRE01920
                C     ERROR MESSAGES ARE WRITTEN FOR ALL ERRORS DETECTED.               PRE01930
                C     READAI RETURNS WITH MTCHAI SET TO 0 IF AN ERROR IS FOUND ON A HEADERPRE01940
                C     IF AN ERROR IS FOUND ON A PIXEL CARD, AN ERROR MESSAGE IS WRITTEN ANDPRE01950
                C     THE NEXT PIXEL CARD IS READ AND PROCESSED.                        PRE01960
                C     IF THERE ARE NO MORE VALID PIXEL CARDS, MTCHAI IS SET TO 0.       PRE01970
                CCC                                                                     PRE01980
                C                                                                       PRE01990
0002                  COMMON /LSEGS/KURTYP,KURSEG,KURDAT,LTYPE(30),LSEGNO(30),LSEGDT(30)PRE02000
                     1  ,NOSEGS                                                        PRE02010
0003                  COMMON/AI/IAIHED(73,2), IAIERR,IDAI,AICMNT(11,19),AIDATE(4)     PRE02020
                     1 ,NOLINE,NOPIXL,AIHDBR(4),IAIVEG(4),LABLAI(11,19),IAIEND       PRE02030
                     2 ,MTCHAI,LNEPOR(4),ISEASN,LNES1(4),LNES2(4)                     PRE02040
                C                                                                       PRE02050
0004                  COMMON /BIOSTG/ BIOSTG(4),BIOLMT(2,8),FIRST(6,8),SECOND(6,8)    PRE02060
                C                                                                       PRE02070
0005                  COMMON /SEASNS/ XBIO(4,13,2), XMEANG(13,2), XMENG2(13,2),        PRE02080
                     1  XMEANB(13,2), XMENB2(13,2), SDG(13,2), SDG2(13,2), SDB(13,2),  PRE02090
                     2  SDB2(13,2),INFSEA(4)                                         PRE02100
                C                                                                       PRE02110
0006                  DIMENSION KEEP(73)                                               PRE02120
                C                                                                       PRE02130
0007                  DATA  KONSTA/1HA/, KONSTB/1HB/,KONSTS/1HS/,KONSTW/1HW/           PRE02140
0008                  DATA IBLANK/1H /, KONST1/1H1/, KONST2/1H2/                       PRE02150
                CCC                                                                     PRE02160
                CCC                                                                     PRE02170
                C                                                                       PRE02180
                C     SET MATCHING AI RECORD SWITCH TO 0 INDICATING  NO VALID, MATCHING PRE02190
                C     HEADER AND PIXEL CARD FOUND                                       PRE02200
0009                  MTCHAI = 0                                                       PRE02210
0010                  IAIEND = 0                                                       PRE02220
                C                                                                       PRE02230
                C     SET SWITCH THAT NO ERROR HAS BEEN FOUND IN SEGMENT NUMBER         PRE02240
0011                  ISEGSW = 0                                                       PRE02250
                C                                                                       PRE02260
                CCC                                                                     PRE02270
                C       FIRST RECORD ON FILE MUST BE A HEADER CARD                      PRE02280
                C       HEADER CARDS MUST OCCUR IN PAIRS                                PRE02290
                C       THE NUMBER OF DATA CARDS IS NOT FIXED                           PRE02300
                CCC                                                                     PRE02310
                C     HEADER MAY HAVE BEEN READ ON PREVIOUS CALL TO READAI, IF SO NOPIXL PRE02320
                C     WILL BE SET NONZERO                                               PRE02330
0012                  IF (NOPIXL .NE. 0) GO TO 380                                     PRE02340
                C                                                                       PRE02350
                C     RESET REREAD BUFFER AND READ A CARD                               PRE02360
0013                  10 READ (5,15,END = 220) (KEEP(I),I = 1,73)                      PRE02370
0014                     REWIND 30                                                     PRE02380
0015                     WRITE (30,15) (KEEP(I),I=1,73)                                PRE02390
0016                  15 FORMAT (1X,A4,71A1,A4)                                        PRE02400
                C                                                                       PRE02410
                C     CHECK FOR HEADER 1 CARD                                           PRE02420
0017                  IF (KEEP(24) .EQ. KONSTA) GO TO 200                              PRE02430
                CCC                                                                     PRE02440
                C     ERROR--1ST CARD NOT HEADER, IGNORE IF LOOKING FOR FILE OUT OF ORDERPRE02450
0018                  IF (ISEGSW .NE. 0) GO TO 270                                     PRE02460
0019                  WRITE (6,30)                                                     PRE02470
0020               30 FORMAT(' AI FILE ERROR--FIRST CARD NOT HEADER')                  PRE02480
                                                                                        PRE02490
```

52

FILE PREPPT

```
0021                WRITE (6,31) (KEEP(I),I=1,73)
0022          31    FORMAT (1X,A4,71A1,A4)
              C
0023          60    IAIERR = IAIERR + 1
0024                CALL FSFMFL (5,1,ISTAT)
0025                RETURN
              C
              C
              C  ONE HEADER CARD READ. MOVE IT TO IHEAD
0026          200   DO 210 I=1,73
0027                IAIHED(I,1) = KEEP(I)
0028          210   CONTINUE
              C
              C  REREAD CARD AS FIRST HEADER CARD
0029                REWIND 30
0030                READ (30,211) ISEGNO,ISEASN, (AIDATE(I),I=1,4)
0031                REWIND 30
0032          211   FORMAT (1X,A4,1X,A1,4(1X,A4))
              C
              C  CHECK SEGMENT NUMBER FOR CURRENT SEGMENT
0033                IF(ISEGNO.NE.IDAI) WRITE (6,9989) ISEGNO,IDAI
0034          9989  FORMAT (' READAI-COMPARISON OF SEG.NO. 1ST CARD, SEG.NO. REQUESTED
                   1 ', A4,1X,A4)
0035                IF (ISEGNO .EQ. IDAI) GO TO 290
              C
              C  INCORRECT SEGMENT
              C    IF FIRST INCORRECT SEGMENT, REWIND FILE AND START LOOKING
              C    IF NOT 1ST INCORRECT SEGMENT, CHECK FOR LAST FILE AND KEEP LOOKING
0036                IF (ISEGSW .NE. 0) GO TO 270
0037          220   ISEGSW = 1
0038                REWIND 5
0039                GO TO 10
              C
              C  NOT FIRST INCORRECT SEGMENT
0040          270   IF (ISEGSW .EQ. NOSEGS) REWIND 5
0041                IF (ISEGSW .EQ. NOSEGS) GO TO 7000
0042                ISEGSW = ISEGSW + 1
0043                CALL FSFMFL(5,1,ISTAT)
0044                GO TO 10
              C
              C
              C  CORRECT SEGMENT FOUND, WAS IT FOUND ON FIRST TRY
0045          290   IF (ISEGSW .EQ. 0) GO TO 294
0046                WRITE (6,291)
0047          291   FORMAT(' EXCESSIVE PROCESSING TIME DUE TO ERROR IN ORDER OF AI FIL
                   1ES', /, ' IN PROCEDURE FILE', /)
              C
              C
              C  CHECK SEASON MUST BE 1 FOR WINTER OR 2 FOR SPRING
0048          294   IF (ISEASN .EQ. KONSTW  .OR.  ISEASN .EQ. KONST1) ISEASN = 1
0049                IF (ISEASN .EQ. KONSTS  .OR.  ISEASN .EQ. KONST2) ISEASN = 2
0050                IF (ISEASN .LT. 1 .OR. ISEASN .GT. 2) GO TO 9030
              C
              C     .
              C  READ SECOND CARD AND CHECK ID NUMBERS
0051                READ(5,15,END=9000) (IAIHED(I,2),I=1,73)
0052                IF (IAIHED(24,2) .NE. KONSTB) GO TO 9010
              C
0053                REWIND 30
0054                WRITE (30,15) (IAIHED(I,2),I=1,73)
0055                REWIND 30
0056                READ (30,295)  (AIHDRB(I),I=1,4), ISEGH2
0057                REWIND 30
0058          295   FORMAT (9X, (4(F3.1,2X),47X,A4))
0059                IF (ISEGNO .NE. ISEGH2) GO TO 9010
              C
              C  CHECK ROBERTSON BIOSTAGES FOR VALID VALUES
0060                DO 320 I = 1,4
0061                DO 310 J = 1,8
0062                JJ = J
0063                IF (AIHDRB(I) .GE. BIOLMT(1,J) .AND. AIHDRB(I) .LE. BIOLMT(2,J))
                   1  GO TO 320
0064          310   CONTINUE
              C
              C  INVALID ROBERTSON BIOSTAGE
0065                WRITE (6,315) ISEGNO,AIHDRB(I)
```

FILE PREPPT

```
0066          315   FORMAT (' HEADER ERROR--ROBERTSON BIOSTAGE FOR ',A4,' IS ', F4.1)PRE03260
0067                GO TO 9035                                                      PRE03270
                C                                                                   PRE03280
0068          320   LMERGR(I) = JJ                                                  PRE03290
                C                                                                   PRE03300
                C  CHECK SEASONAL BIOSTAGES FOR VALID VALUES                        PRE03310
0069                DO 340 I = 1,4                                                  PRE03320
0070                DO 330 J = 1,13                                                 PRE03330
0071                JJ = J                                                          PRE03340
0072                IF(AIHORB(I).GE.XBIO(1,J,ISEASN).AND.AIHORB(I).LE.XBIO(2,J,ISEASN)PRE03350
                   1  ) GO TO 340                                                  PRE03360
0073          330   CONTINUE                                                        PRE03370
                C                                                                   PRE03380
                C  INVALID SEASONAL BIOSTAGE                                        PRE03390
0074                WRITE (6,335) ISEGNO,AIHORB(I)                                  PRE03400
0075          335   FORMAT (' HEADER ERROR--SEASONAL BIOSTAGE FOR ',A4,' IS ',F4.1) PRE03410
0076                GO TO 9035                                                      PRE03420
                C                                                                   PRE03430
0077          340   LMSEA(I) = JJ                                                   PRE03440
                C                                                                   PRE03450
                C                                                                   PRE03460
                C  READ FIRST PIXEL CARD AS POSSIBLE HEADER                         PRE03470
0078          350   READ(5,15,END=390) (KEEP(I),I=1,73)                             PRE03480
0079                READ(5,30)                                                      PRE03490
0080                WRITE (30,15) (KEEP(I),I=1,73)                                  PRE03500
0081                IF (KEEP(24) .EQ. IBLANK  .AND.  IAIEND .EQ. 0) GO TO 400       PRE03510
                C                                                                   PRE03520
0082                  IF (IAIEND .NE. 1) WRITE (6,381) ISEGNO                       PRE03530
0083          381   FORMAT (' ERROR IN PIXEL CARD FOR ', A4)                        PRE03540
0084                IF (IAIEND .NE. 1) WRITE (6,31) (KEEP(I),I=1,72)                PRE03550
0085                IAIEND = 1                                                      PRE03560
0086                GO TO 380                                                       PRE03570
                C                                                                   PRE03580
                C                                                                   PRE03590
                C  END OF PIXELS, HAVE ANY PIXEL BEEN READ?                         PRE03600
0087          390   IF (NOPIXL .EQ. 0) GO TO 9020                                   PRE03610
0088                RETURN                                                          PRE03620
                C                                                                   PRE03630
                C  PROCESS PIXEL CARD                                               PRE03640
0089          400   IF (KEEP(73) .NE. ISEGN2) GO TO 9050                            PRE03650
0090                CALL READPX(IPXERR)                                             PRE03660
0091                IF (IPXERR .NE. 0) GO TO 380                                    PRE03670
                C                                                                   PRE03680
                C  SET SWITCH FOR VALID MATCHING HEADER AND PIXEL TO 1              PRE03690
0092                MTCHAI = 1                                                      PRE03700
                C                                                                   PRE03710
0093                 RETURN                                                         PRE03720
                C                                                                   PRE03730
                C  END OF FILE                                                      PRE03740
                C                                                                   PRE03750
                C                                                                   PRE03760
                C    ********  ERROR OR END OF FILE CONDITIONS  *********           PRE03770
                C                                                                   PRE03780
0094          7000  WRITE (6,7100) ISEGNO                                           PRE03790
0095          7100  FORMAT (' NO DATA ON FILE AI FOR SEGMENT ',A4)                  PRE03800
0096                GO TO 9035                                                      PRE03810
                C                                                                   PRE03820
                C                                                                   PRE03830
                C  EOF WHILE LOOKING FOR NEXT HEADER                                PRE03840
0097          8000  WRITE (6,8100) IDAI                                             PRE03850
0098          8100  FORMAT (' END OF FILE WHILE LOOKING FOR HEADER ON FILE AI ', A4)PRE03860
0099                 GO TO 9035                                                     PRE03870
                C                                                                   PRE03880
                C                                                                   PRE03890
0100          9000  WRITE (6,9005)                                                  PRE03900
0101          9005  FORMAT (' END OF FILE WHILE READING 2ND HEADER CARD')           PRE03910
0102                WRITE (6,31) (KEEP(I),I=1,73)                                   PRE03920
0103                GO TO 9035                                                      PRE03930
                C                                                                   PRE03940
                C  NO DATA CARD AFTER HEADER                                        PRE03950
0104          9020  WRITE (6,9021)                                                  PRE03960
0105          9021  FORMAT (' NO DATA AFTER HEADER CARD')                           PRE03970
0106                WRITE (6,31) (IAIHED(I,1),I=1,73)                               PRE03980
                C                                                                   PRE03990
0107                GO TO 9035                                                      PRE04000
                C                                                                   PRE04010
```

FILE PREPRT

```
                C                                                                       PRE04020
                C  HEADER CARDS DO NOT MATCH                                            PRE04030
0108            9010  WRITE (6,9011) ISEGNO, ISEGH2                                     PRE04040
0109            9011  FORMAT (' HEADER CARDS DO NOT MATCH', 2X,A4,2X,A4)                PRE04050
0110                  WRITE (6,31) (IAIHED(I,1),I = 1,73)                               PRE04060
0111                  WRITE (6,31) (IAIHED(I,2),I = 1,73)                               PRE04070
0112                  GO TO 9035                                                        PRE04080
                C                                                                       PRE04090
                C  SEASON INCORRECT                                                     PRE04100
0113            9030  WRITE (6,9031) ISEASN,ISEGNO                                      PRE04110
0114            9031  FORMAT (' SEASON IS ', A1, ' FOR SEG NO. ',A4)                    PRE04120
                C                                                                       PRE04130
                C  FATAL ERROR, SET SWITCH AND RETURN                                   PRE04140
0115            9035  IAIERR = IAIERR + 1                                               PRE04150
0116                  RETURN                                                            PRE04160
                C                                                                       PRE04170
                C  READAI CALLED AFTER EOF, PROGRAMMING ERROR                           PRE04180
                C                                                                       PRE04190
                C  SEGMENT NUMBER ON PIXEL CARD NOT EQUAL TO SEGMENT NUMBER ON HEADER    PRE04200
0117            9050  WRITE (6,9051) ISEGNO,KEEP(73)                                    PRE04210
0118            9051  FORMAT (' HEADER SEGMENT NUMBER=',A4,' PIXEL SEGMENT NUMBER=',A4) PRE04220
0119                  GO TO 340                                                         PRE04230
0120                  END                                                               PRE04240
```

A.4 SOFTWARE SUBPROGRAM NO. 4 (READRD)

READRD reads one set of data from the 16-channel MSS Data File.

A.4.1 Linkages

READRD is called by PREPRT.

A.4.2 Interfaces

Interface is accomplished through common block /RAWDTA/.

A.4.3 Inputs

16-channel MSS Data File from unit no. 12.

A.4.4 Outputs

None

A.4.5 Storage Requirement

Program size = 6530.

A.4.6 Description

One set of data is read from the 16-channel Data File.

A.4.7 Flowchart

N/A

A.4.8 Listings

```
FILE PREPRT
0001            SUBROUTINE READRD                                                PRE06220
                C                                                                PRE06230
                C                                                                PRE06240
                C  PURPOSE:READ SET OF RAW DATA RECORDS (SEG.NO.=ID) FROM 16-CHAN FILE  PRE06250
                C  AND CALCULATE GREENESS, BRIGHTNESS, PIF AND SQUARE            PRE06260
                C                                                                PRE06270
                C   ********WARNING.  THIS SUBROUTINE WILL STOP THE PROGRAM IF THE  PRE06280
                C                     SEGMENTS ON THE RAW DATA TAPE ARE NOT IN THE PRE06290
                C                     SAME ORDER AS THE SEGMENTS REQUESTED       PRE06300
                C                                                                PRE06310
0002             COMMON /RAWDAT/ MSTRID,RDFRST,IRDERR,PDARGT(11,19,4),           PRE06320
                1  PDGREN(11,19,4), IRDRAD, SQATRR(11,19),SQAIRG(11,19),PIER(11,19)PRE06330
                2  ,PIFG(11,19),ARWIFT(11,19,4),AGREEN(11,19,4)                  PRE06340
                C                                                                PRE06350
0003             COMMON /MEANSD/ YMEANG(4), YMEANR(4), YSDG(4), YSDR(4)          PRE06360
0004             COMMON /TAPERD/ IUNIT,IFRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,    PRE06370
                *  LINC,ID(200),DSL,LBUF(30),JREC(30),IBYTE(30),NBUFS,FILENO,LINENDPRE06380
                *  ,LININC,NSAMP,NOCHAN,IFORMT                                   PRE06390
                C                                                                PRE06400
0005             COMMON /LSEGS/KURTYP,KURSEG,KURDAT,LTYPE(30),LSEGNO(30),LSEGDT(30)PRE06410
                1  ,NOSEGS                                                       PRE06420
                C                                                                PRE06430
0006             DIMENSION FETVEC(30)                                            PRE06440
0007             DIMENSION BLOCK(6)                                              PRE06450
                C                                                                PRE06460
0008             INTEGER FETVEC,BLOCK                                            PRE06470
                C                                                                PRE06480
0009             DATA LASTFL /0/                                                 PRE06490
0010             DATA BLOCK/10,110,10,10,190,10/                                 PRE06500
0011             DATA FETVEC/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16/             PRE06510
0012             DATA NOFEAT /16/                                                PRE06520
                C                                                                PRE06530
0013             DIMENSION  LDATA(4),IDATA(19,4,4),FL(12), KDATA(400)            PRE06540
0014             COMMON /PPPTU/ITU                                               PRE06550
0015             INTEGER DATA,FL,SAMSTR,SAMEND,SAMINC,IE,JR                      PRE06560
                C                                                                PRE06570
                C                                                                PRE06580
0016             DATA IFIRST /0/                                                 PRE06590
                C                                                                PRE06600
0017              DATA LS/0/                                                     PRE06610
0018             DIMENSION ISOIL(4)                                              PRE06620
0019             COMMON /CK/ICK,JCK                                              PRE06630
                C                                                                PRE06640
                C                                                                PRE06650
                C  SET RAW DATA RAD SWITCH TO 1                                  PRE06660
0020             IRDRAD = 1                                                      PRE06670
                C                                                                PRE06680
                C  READ HEADER                                                   PRE06690
0021             IFILE = LASTFL                                                  PRE06700
                C  IF THERE ARE NO MORE FILES TO BE READ, REWIND TAPE TO TRY 1ST FILE  PRE06710
0022             IF (LASTFL .GE. NOSEGS) IFILE = 0                               PRE06720
                C                                                                PRE06730
                C                                                                PRE06740
0023             ITAPE=ITU                                                       PRE06750
0024             IUNIT =ITU                                                      PRE06760
                C                                                                PRE06770
                C                                                                PRE06780
0025            1     CALL TAPHDR(ITAPE,IFILE)                                   PRE06790
0026             LASTFL = LASTFL + 1                                             PRE06800
                C                                                                PRE06810
                C  REFORMAT AND CHECK FOR CORRECT SEGMENT                        PRE06820
0027             ISITE = ID(28)                                                  PRE06830
0028             WRITE(6,999) ISITE                                              PRE06840
0029            999   FORMAT (' RAW DATA TAPE SEG.NO.=',I5)                      PRE06850
                C                                                                PRE06860
                C  MOVE MSTRID IN A4 FORMAT TO IDREC IN I4 FORMAT                PRE06870
0030             REWIND 30                                                       PRE06880
0031             WRITE (30,30) MSTRID                                            PRE06890
0032            30    FORMAT (A4)                                                PRE06900
0033             REWIND 30                                                       PRE06910
0034             READ(30,31) IDREC                                               PRE06920
0035            31    FORMAT (I4)                                                PRE06930
                C      WRITE(6,9944) MSTRID,IDREC                                PRE06940
0036            9944   FORMAT(1H ,'MSTRID=', A4,' IDREC=',A4)                    PRE06950
                C                                                                PRE06960
                C  CHECK SEGMENT NUMBERS (I4 FORMAT)                             PRE06970
```

57

FILE PREPPT

```
0037              IF (ISITE .NE. IDREC) GO TO 9000
          C
          C  CORRECT SEGMENT FOUND. CALCULATE SOIL LINE NUMBERS
0038      699      DO 700 I=1,4
0039       700     ISOIL(I)=ID(I+160-1)
0040               WRITE(6,701) (ISOIL(I),I=1,4)
0041       701     FORMAT(1H ,10X,' THE SOIL LINE NUMBERS ARE ',4I5)
          C
          C  SET FILE TO FIRST LINE OF DATA
0042              CALL FLOINT(BLOCK,FETVEC,NOFEAT)
          C       WRITE (6,9945)
0043      9945     FORMAT('  EXIT FLOINT')
          C
          C  READ AND PROCESS LINES 1 THROUGH 11
          C
          C  SET LINE INFORMATION
          C
0044              DO 290 ILINE =1,11
0045              CALL LINERD (KDATA, ENDTAP)
0046              IF (ENDTAP .EQ. -1) GO TO 800
          C
          C  STORE DATA ON THIS LINE INTO 1 X 19 X 4 X 4 BUFFER
0047              DO 60 I = 1,19
0048              DO 59 K = 1,4
0049              DO 59 J = 1,4
0050              INDEX=I+(J-1)*19+(K-1)*76
0051              IDATA(I,J,K) = KDATA(INDEX)
0052              IF(ILINE.EQ.ICK.AND.I.EQ.JCK) WRITE(6,9950) I,J,K,IDATA(I,J,K),
                 *   INDEX
0053      9950     FORMAT(1H ,10X,3I5,' IDATA ',I5,' INDEX ',I5)
0054       59      CONTINUE
          C
0055      60       CONTINUE
          C
          C  CALCULATE BRIGHT AND GREEN FOR EACH SET OF 4 NUMBERS
0056              DO 210 ISET = 1,4
0057              DO 200 IPIXEL = 1,19
          C
          C    MOVE FROM TRIPLE DIMENSION ARRAY TO SINGLE DIMENSION ARRAY
0058              DO 150 KK = 1,4
0059      150     LDATA(KK) = IDATA(IPIXEL,KK,ISET)
          C
          C    ACTUALLY CALCULATE BRIGHT AND GREEN
0060              LS=ISOIL(ISET)
0061              CALL CALCUL (LDATA, LS, BRIT, GRN)
0062              IF(ILINE.EQ.ICK.AND.IPIXEL.EQ.JCK) WRITE(6,9951)
                 *   (LDATA(I),I=1,4),LS
0063      9951     FORMAT(1H ,10X,'LDATA ',4I5,' SOILL ',I5)
0064              RDBRGT(ILINE, IPIXEL, ISET) = (BRIT - YMEANB(ISET)) / YSDB(ISET)
0065              RDGREN (ILINE, IPIXEL, ISET) = (GRN - YMEANG(ISET)) / YSDG(ISET)
0066              IF(ILINE.EQ.ICK.AND.IPIXEL.EQ.JCK) WRITE(6,9953)
                 * ILINE,IPIXEL,ISET,BRIT,YMEANB(ISET),YSDB(ISET)
0067              IF(ILINE.EQ.ICK.AND.IPIXEL.EQ.JCK) WRITE(6,9952)
                 * ILINE,IPIXEL,ISET,GRN,YMEANG(ISET),YSDG(ISET)
0068      9952     FORMAT(1H ,3I5,' GRN ',F10.4,' MEANG ',F10.4,' SDG ',F10.4)
0069      9953     FORMAT(1H ,3I5,' BRIT ',F10.4,' MEANB ',F10.4,' SDB ',F10.4)
          C
0070              ABRIFT(ILINE,IPIXEL,ISET) = ABS(RDBRGT(ILINE,IPIXEL,ISET))
0071              AGREEN(ILINE,IPIXEL,ISET) = ABS(RDGREN(ILINE,IPIXEL,ISET))
          C
0072      200     CONTINUE
          C
0073      210     CONTINUE
          C
          C  CALCULATE SQUARE AND PIE
0074              DO 250 IPIXEL = 1,19
          C
          C  SUM SQUARE AND PIE FOR 4 SETS
0075              SQAIRB(ILINE,IPIXEL) = 0.
0076              SQAIRG(ILINE,IPIXEL) = 0.
0077              PIEB(ILINE,IPIXEL) = 1.
0078              PIEG(ILINE,IPIXEL) = 1.
          C
0079              DO 240 K = 1,4
0080              SQAIRB(ILINE,IPIXEL)=SQAIRB(ILINE,IPIXEL)+(RDBRGT(ILINE,IPIXEL,K))
                 1  ** 2
```

58

```
FILE PREPPT

0041                SQAIRG(ILINE,IPIXEL)=SQAIRG(ILINE,IPIXEL)+(RDGREN(ILINE,IPIXEL,K))PRE07740
                   1  ** 2                                                          PRE07750
0042                PIFR(ILINE,IPIXEL)=PIFR(ILINE,IPIXEL)*(ARRIFT(ILINE,IPIXEL,K)+1.) PRE07760
0043                PIFG(ILINE,IPIXEL)=PIFG(ILINE,IPIXEL)*(AGREEN(ILINE,IPIXEL,K)+1.) PRE07770
0044          240   CONTINUE                                                        PRE07780
              C                                                                     PRE07790
              C                                                                     PRE07800
0045          250   CONTINUE                                                        PRE07810
              C                                                                     PRE07820
0046          290   CONTINUE                                                        PRE07830
              C                                                                     PRE07840
0047                IRDRAD = 0                                                      PRE07850
              C     WRITE(6,9946)                                                   PRE07860
0048          9946  FORMAT(' EXIT READRD')                                         PRE07870
0049                RETURN                                                          PRE07880
              C                                                                     PRE07890
              C       ***************  ERROR MESSAGES  **************               PRE07900
              C                                                                     PRE07910
              C  ID NOT FOUND, WAS THERE ANY DATA ON FILE?                          PRE07920
004A          300   IF (IFIRST .EQ. 0) GO TO 340                                    PRE07930
004B                WRITE (6,310) ISEGNO                                            PRE07940
004C           310  FORMAT(' NO DATA ON RAW DATA FILE WHILE LOOKING FOR SEGMENT  ',A4)PRE07950
004D                IRDERR = IRDERR + 1                                             PRE07960
004E                RETURN                                                          PRE07970
              C                                                                     PRE07980
              C  ID NOT ON FILE                                                     PRE07990
004F          340   WRITE (6,345) MSTRID                                            PRE08000
0050          345   FORMAT (1X, A4, ' NOT ON RAW DATA FILE')                        PRE08010
0051                IRDERR = IRDERR + 1                                             PRE08020
0052                RETURN                                                          PRE08030
              C                                                                     PRE08040
              C  INSUFFICIENT LINE INFORMATION, MUST BE PROGRAMMING ERROR           PRE08050
0053          800   WRITE (6,810)                                                   PRE08060
0054          810   FORMAT (' INSUFFICIENT LINE INFORMATION ON RAW DATA FILE')      PRE08070
0055                STOP                                                            PRE08080
              C                                                                     PRE08090
              C  FILES OUT OF ORDER OR FILE MISSING FOR THIS SEGMENT                PRE08100
0056          9000  REWIND ITAPE                                                    PRE08110
              C                                                                     PRE08120
0057                DO 9100 I = 1,NOSEGS                                            PRE08130
0058                IFILE = I -1                                                    PRE08140
              C                                                                     PRE08150
              C  SKIP OVER IFILE FILES AND READ HEADER RECORD                       PRE08160
0059                CALL TAPHDR (ITAPE,IFILE)                                       PRE08170
              C                                                                     PRE08180
              C  SAVE NUMBER OF LAST FILE                                           PRE08190
005A                LASTFL = I                                                      PRE08200
              C                                                                     PRE08210
              C  CHECK FOR CORRECT SEGMENT                                          PRE08220
005B                ISITE = IO(28)                                                  PRE08230
005C                WRITE(6,999) ISITE                                              PRE08240
005D                IF (ISITE .EQ. IDREC) GO TO 9200                                PRE08250
              C                                                                     PRE08260
005E          9100  CONTINUE                                                        PRE08270
              C                                                                     PRE08280
005F                WRITE (6,9105) MSTRID                                           PRE08290
0060          9105  FORMAT (' NO RAW DATA FOR ',A4)                                PRE08300
0061                RETURN                                                          PRE08310
              C                                                                     PRE08320
              C  CORRECT FILE FOUND, WRITE WARNING MESSAGE AND GO TO CONTINUE PROCESS PRE08330
0062          9200  WRITE (6,9205)                                                  PRE08340
0063          9205  FORMAT (' EXCESSIVE PROCESSING TIME DUE TO ERROR IN ORDER OF    PRE08350
                   1 RAW DATA FILES', /, ' IN PROCEDURE FILE',/)                    PRE08360
0064                GO TO 699                                                       PRE08370
              C                                                                     PRE08380
0065                END                                                             PRE08390
```

A-1

69

A.5 SOFTWARE SUBPROGRAM NO. 5 (CALKEY)

CALKEY calculates greenness and brightness factors.

A.5.1 Linkages

CALKEY is called by PREPPT.

A.5.2 Interfaces

Interface is accomplished through common blocks /BIOSTG/, /AI/, /SEASNS/ .

A.5.3 Inputs

See Appendix D.

A.5.4 Outputs

See Appendix D.

A.5.5 Storage Requirement

Program size = 1338.

A.5.6 Description

See Appendix D.

A.5.7 Flowchart

N/A

A.5.8 Listings

FILE PREPPT

```
0001            SUBROUTINE CALKEY(ITYPE)                                                PRE10290
                C                                                                       PRE10300
                C  PURPOSE: TO CALCULATE VARIOUS KEYS                                   PRE10310
                C                                                                       PRE10320
0002                COMMON /FLDATA/ ICNKEY(4),AICANK                                    PRE10330
                C                                                                       PRE10340
0003                COMMON /MEANSD/ YMEANG(4), YMEANR(4), YSDG(4), YSDR(4)              PRE10350
                C                                                                       PRE10360
0004                COMMON /BIOSTG/BIOSTG(4),BIOLMT(2,8),FIRST(6,8),SECOND(6,8)         PRE10370
                C                                                                       PRE10380
0005                COMMON /AI/IAIHED(73,2), IAIERR,IDAI,AICMNT(11,19),AIDATE(4)        PRE10390
                   1 ,NOLINE,NOPIXL,AIHDBR(4),IAIVEG(4),LABLAI(11,19),IAIEND            PRE10400
                   2 ,MTCHAI,LNEPOB(4),ISEASN,LNES1(4),LNES2(4)                         PRE10410
                C                                                                       PRE10420
0006                COMMON /SEASNS/ XMIO(4,13,2), XMEANG(13,2), XMEANG2(13,2),          PRE10430
                   1  XMEANR(13,2), XMEANR2(13,2), SDG(13,2), SDG2(13,2), SDR(13,2),  PRE10440
                   2  SDR2(13,2),LNESEA(4)                                             PRE10450
                C                                                                       PRE10460
0007                DATA HYPHEN/1H-/                                                    PRE10470
                C                                                                       PRE10480
                C                                                                       PRE10490
                C     INPUT                                                             PRE10500
                C      ITYPE = AITYPE (THIS IS NOT CURRENTLY USED)                      PRE10510
                C                                                                       PRE10520
                C  OUTPUT                                                               PRE10530
                C       RIGHT                                                           PRE10540
                C       GREEN                                                           PRE10550
                C       CANKY                                                           PRE10560
                C       AICANK                                                          PRE10570
                C  MATCH CLASS RESPONSE ON DATA CARD TO TABLE                           PRE10580
0008                ISUM = 0                                                            PRE10590
0009                AICANK = 0.                                                         PRE10600
                C                                                                       PRE10610
                C********** BRANCHING ON AI TYPE AS OF SEPTEMBER 26, 1978. MIKE PORF*******PRE10620
                C  IF AI TYPE IS NOT P, ZERO VALUES                                     PRE10630
                C     IF (ITYPE .NE. 1) GO TO 1000                                      PRE10640
                C                                                                       PRE10650
                C  4 RESPONSE INTERVALS IN DATA CARD                                    PRE10660
0010                DO 400 I = 1,4                                                      PRE10670
                C                                                                       PRE10680
                C  SET LINE NUMBER ON SEASONAL TABLE FROM BIOSTAGE ON HEADER CARD       PRE10690
0011                LINETB = LNEPOB(I)                                                  PRE10700
                C                                                                       PRE10710
                C  SET ROW POSITION IN BIOSTAGE TABLE FROM PIXEL CARD                   PRE10720
0012                IVRESP = IAIVEG(I) + 1                                              PRE10730
                C                                                                       PRE10740
                C  CK FOR FIRST CLASS RESPONSE                                          PRE10750
0013                IF (FIRST(IVRESP,LINETB) .NE. HYPHEN) GO TO 350                     PRE10760
                C                                                                       PRE10770
                C  CK FOR SECOND CLASS RESPONSE                                         PRE10780
0014                IF (SECOND(IVRESP,LINETB).NE. HYPHEN) GO TO 360                     PRE10790
                C                                                                       PRE10800
                C  NEITHER FIRST NOR SECOND CLASS RESPONSE                              PRE10810
0015                ISUM = ISUM + 10                                                    PRE10820
0016                ICNKEY(I) = 10                                                      PRE10830
0017                AICANK = 1                                                          PRE10840
0018                GO TO 400                                                           PRE10850
                C                                                                       PRE10860
                C  FIRST CLASS RESPONSE                                                 PRE10870
0019            350   ICNKEY(I) = 0                                                     PRE10880
0020                GO TO 400                                                           PRE10890
                C                                                                       PRE10900
                C  SECOND CLASS RESPONSE                                                PRE10910
                C  SET AICANK                                                           PRE10920
0021            360   AICANK = 0                                                        PRE10930
0022                  IF (ISUM .GT. 1) AICANK = 1                                       PRE10940
0023                  ICNKEY(I) = 5                                                     PRE10950
0024                  ISUM = ISUM + 5                                                   PRE10960
                C                                                                       PRE10970
                C                                                                       PRE10980
                C  CALCULATE SEASONAL CONSTANTS                                         PRE10990
0025            400   CONTINUE                                                          PRE11000
                C                                                                       PRE11010
                C    CALCULATE MEANS                                                    PRE11020
                C     IF BIO CONSTANT IS ZERO, MEAN AND STANDARD DEV. ARE CONSTANTS     PRE11030
                C                                                                       PRE11040
```

FILE PREPPT

```
                C                                                                       PRE11050
0024                  DO 510 I = 1,4                                                    PRE11060
0027                  LINETR = LNFSEA(I)                                                PRE11070
                C                                                                       PRE11080
0028                  IF (XRIO(3,LINETR,ISEASN) .NE. 0) GO TO 500                       PRE11090
                C                                                                       PRE11100
0029                  YMEANG(I) = XMEANG(LINETR,ISEASN)                                 PRE11110
002A                  YMEANR(I) = XMEANR(LINETR,ISEASN)                                 PRE11120
002B                  YSDG(I) = SDG(LINETR,ISEASN)                                      PRE11130
002C                  YSDR(I) = SDR(LINETR,ISEASN)                                      PRE11140
002D                  GO TO 510                                                         PRE11150
                C                                                                       PRE11160
                C   RIO CONSTANT IS NONZERO. MEAN AND STANDARD DEV. MUST BE CALCULATED PRE11170
002E            500   CONTINUE                                                          PRE11180
002F                  DIFF1 =  -XRIO(3,LINETR,ISEASN) + AIHORR(I)                       PRE11190
0036                  DIFF2 = -AIHORR(I) + XRIO(4,LINETR,ISEASN)                        PRE11200
                C                                                                       PRE11210
0037                  YMEANG(I)=DIFF1*XMEANG(LINETR,ISEASN)+DIFF2*XMEANG2(LINETR,ISEASN) PRE11220
003D                  YMEANR(I)=DIFF1*XMEANR(LINETR,ISEASN)+DIFF2*XMEANR2(LINETR,ISEASN) PRE11230
0043                  YSDG(I) = DIFF1 * SDG(LINETR,ISEASN) + DIFF2 * SDG2(LINETR,ISEASN)PRE11240
0049                  YSDR(I) = DIFF1 * SDR(LINETR,ISEASN) + DIFF2 * SDR2(LINETR,ISEASN)PRE11250
                C                                                                       PRE11260
004F            510   CONTINUE                                                          PRE11270
0052                  RETURN                                                            PRE11280
                C                                                                       PRE11290
                C  AI TYPE NOT P. ZERO VALUES                                           PRE11300
0053            1000  CONTINUE                                                          PRE11310
                C                                                                       PRE11320
0054                  AICARK = 0                                                        PRE11330
0055                  DO 1100 I = 1,4                                                   PRE11340
0058                  ICLKEY(I) = 0                                                     PRE11350
005D                  YMEANG(I) = 0.                                                    PRE11360
0062                  YMEANR(I) = 0.                                                    PRE11370
0067                  YSDG(I) = 0.                                                      PRE11380
006C                  YSDR(I) = 0.                                                      PRE11390
0071            1100  CONTINUE                                                          PRE11400
0072                  END                                                               PRE11410
```

62

A.6 SOFTWARE COMPONENT NO. 6 (READGT)

READGT reads one set of ground truth data.

A.6.1 Linkages

READGT is called by PREPPT.

A.6.2 Interfaces

Interface is accomplished through common block /GT/.

A.6.3 Inputs

Ground Truth Dot Data File from unit number 23.

A.6.4 Outputs

None

A.6.5 Storage Requirement

Program size = 1158.

A.6.6 Description

The ground truth dot data is read into a common block /GT/.

A.6.7 Flowchart

N/A

A.6.8 Listings

FILE PREPPT

```
0001            SUBROUTINE READGT                                                PRE04590
                C                                                                PRE04600
                C                                                                PRE04610
                C  PURPOSE--READ A SET OF GROUND TRUTH DATA                      PRE04620
                C                                                                PRE04630
                C                                                                PRE04640
0002            COMMON/LSEGS/ KURTYP,KURSEG,KURDAT,LTYPE(30),LSEGNO(30),LSEGDT(30)PRE04650
               1  ,NOSEGS                                                       PRE04660
0003            COMMON /GT/ IGT(11,19), IGTERR, IGTFTL                           PRE04670
                C                                                                PRE04680
0004            COMMON /MSK/MASK(11,19)                                          PRE04690
                C                                                                PRE04700
                C                                                                PRE04710
0005            COMMON /LU/LUNIT                                                 PRE04720
0006            DATA IDOT /'DOT '/,  KSEG /'SITE'/, LASTSG /'FRST'/              PRE04730
                C                                                                PRE04740
                C     GROUND TRUTH DOTS READ FROM    LUNIT                       PRE04750
                C                                                                PRE04760
                C      WRITE (6,9948)                                            PRE04770
0007        9948   FORMAT(' ENTER READGT')                                       PRE04780
                C                                                                PRE04790
                C  SET GROUND ERROR SWITCH TO 0                                  PRE04800
0008            IGTFTL = 0                                                       PRE04810
                C  READ SEGMENT NUMBER AND CERTIFY THAT THIS IS A HEADER RECORD  PRE04820
0009        10  READ (LUNIT ,20,END = 9100)  ISEG, IDGT                         PRE04830
0010        20  FORMAT  (A4, 4X, A4)                                            PRE04840
0011            IF (ISEG .NE. KSEG) GO TO 9500                                   PRE04850
                C                                                                PRE04860
                C  SEGMENT CARD READ.  IS IT THE RIGHT ONE                       PRE04870
0012        25    LASTSG = IDGT                                                  PRE04880
0013              IF (IDGT .NE. KURSEG) GO TO 9100                               PRE04890
                C                                                                PRE04900
                C  MATCHING RECORD, READ GROUND TRUTH                            PRE04910
                C                                                                PRE04920
0014        100   CONTINUE                                                       PRE04930
0015              WRITE(6,150)                                                   PRE04940
0016         150   FORMAT(//,10X,'GROUND TRUTH HEADER')                         PRE04950
                C     WRITE(6,9949)                                              PRE04960
0017        9949   FORMAT(' ENTER RDLDF')                                        PRE04970
0018              BACKSPACE LUNIT                                                PRE04980
0019              CALL RDLDF                                                     PRE04990
                C                                                                PRE05000
0020              CALL FSFMFL (LUNIT,1,ISTAT)                                    PRE05010
                C     WRITE(6,9947)                                              PRE05020
0021        9947   FORMAT(' EXIT READGT')                                        PRE05030
0022              RETURN                                                         PRE05040
                C                                                                PRE05050
                C                                                                PRE05060
                C      **********  ERROR CONDITIONS  **********                  PRE05070
                C                                                                PRE05080
                C                                                                PRE05090
                C  SEQUENCE NUMBERS DO NOT MATCH, REWIND TAPE AND LOOK AT OTHER  PRE05100
                C  GROUND TRUTH FILES                                            PRE05110
0023        9100  CONTINUE                                                       PRE05120
                C                                                                PRE05130
0024              REWIND LUNIT                                                   PRE05140
0025              KOUNT = 1                                                      PRE05150
                C  READ FIRST RECORD AND CHECK SEGMENT NUMBER                    PRE05160
0026        9110  READ (LUNIT,20,END=9300) ISEG,IDGT                             PRE05170
0027              IF (ISEG .NE. KSEG) GO TO 9500                                 PRE05180
                C                                                                PRE05190
                C  SEGMENT CARD READ.  IS IT THE RIGHT ONE?                      PRE05200
0028              LASTSG = IDGT                                                  PRE05210
0029              IF (IDGT .EQ. KURSEG) GO TO 9225                               PRE05220
                C                                                                PRE05230
                C  IF THERE ARE MORE FILES, SKIP TO END OF FILE AND GO TO READ NEXT FILEPRE05240
0030              IF (KOUNT .EQ. NOSEGS) GO TO 9300                              PRE05250
0031              KOUNT = KOUNT + 1                                              PRE05260
                C  MORE GROUND TRUTH FILES, SKIP TO END OF THIS FILE             PRE05270
0032              CALL FSFMFL (LUNIT,1,ISTAT)                                    PRE05280
0033              GO TO 9110                                                     PRE05290
                C                                                                PRE05300
                C  CORRECT FILE FOUND, WRITE WARNING MESSAGE                     PRE05310
0034        9225  WRITE (6,9226)                                                 PRE05320
0035        9226  FORMAT (' EXCESSIVE PROCESSING TIME DUE TO ERROR IN SPECIFYING ORDPRE05330
               1ER OF FILES IN PROCEDURE FILE')                                  PRE05340
```

FILE PREPPT

```
0036                GO TO 100                                                    PRE05350
                    C                                                            PRE05360
                    C  FILE NOT FOUND                                            PRE05370
0037                9300  WRITE (6,9301) KURSEG                                  PRE05380
0038                9301  FORMAT (' NO GROUND TRUTH FILE FOR SEGMENT ',A4)       PRE05390
0039                      IGTERR = IGTERR + 1                                    PRE05400
0040                      IGTFTL = 1                                             PRE05410
0041                      RETURN                                                 PRE05420
                    C                                                            PRE05430
                    C  CARDS OUT OF ORDER                                        PRE05440
0042                9500  WRITE (6,9501) LASTSG                                  PRE05450
0043                9501  FORMAT (' GROUND TRUTH CARDS OUT OF ORDER FOLLOWING SEGMENT ',A4) PRE05460
0044                      CALL ERREX (LUNIT+1,ISTAT)                             PRE05470
                    C  INCREMENT TOTAL COUNT OF ERRORS, SET FATAL ERROR FLAG     PRE05480
0045                      IGTERR = IGTERR + 1                                    PRE05490
0046                      IGTFTL = 1                                             PRE05500
0047                      RETURN                                                 PRE05510
0048                      END                                                    PRE05520
```

A.7 SOFTWARE COMPONENT NO. 7 (WRTPPT)

WRTPPT writes one record to the LIST Variable Input File (FILE PPTIN).

A.7.1 Linkages

WRTPPT is called by PREPPT.

A.7.2 Interfaces

Interface is accomplished through common blocks.

A.7.3 Inputs

A.7.4 Outputs

LIST Variable Input File to unit number 15.

A.7.5 Storage Requirement

Program size = 1830.

A.7.6 Description

The header record and data files are written to the LIST Variable Input File as described in Table 3-5.

A.7.7 Flowchart

See Appendix A.

A.7.8 Listings

FILE PREPPT

```
0001          SUBROUTINE WRTPPT
        C
        C
        C  PURPOSE--TO WRITE THE NEW FILE
        C          FORMAT DESCRIBED AT THE BEGINNING OF THE MAIN PROGRAM
        C
0002          COMMON/AI/IAIHED(73,2), IAIERR,IDAI,AICMNT(11,19),AIDATE(4)
             1 ,NOLINE,NOPIXL,AIHOPR(4),IAIVEG(4),LABLAI(11,19),IAIEND
             2 ,MTCHAI,LNEROR(4),ISEASN,LNES1(4),LNES2(4)
        C
0003          COMMON /GT/ IGT(11,19), IGTERR, IGTFIL
        C
0004          COMMON /RAWDAT/ MSTRID,RDFRST,IRDERR,RDBRGT(11,19,4),
             1  RDGREN(11,19,4), IRDBAD, SDAIRB(11,19),SDAIRG(11,19),PIFB(11,19)
             2  ,PIFG(11,19),ABRIFT(11,19,4),AGREEN(11,19,4)
        C
0005          COMMON /FLDATA/ ICNKEY(4),AICANK
0006          DIMENSION CANKY(4)
0007             COMMON /UN/NROR1,IOUT,NPPT,IU1,IU2
        C
0008          COMMON /MEANSD/ YMEANG(4), YMEANB(4), YSDG(4), YSDB(4)
        C
        C        COMMON /PREVID/  PREVID
        C
0009          COMMON /MSK/MASK(11,19)
0010          DATA IFIRST/0/
0011          COMMON/FV/IFVEC(30),IFMT(30),NDATA
        C
        C
0012             IFIRST = 1
        C
        C  DO NOT WRITE HEADER IF NULL RECORD
0013          IF (NOPIXL .EQ. 0) GO TO 200
        C
        C
        C  WRITE HEADER AND RECORD FROM AI
0014          IF (IDAI .NE. PREVID) WRITE (IOUT,10)((IAIHED(I,J),I=1,73), J=1,2)
0015          IF(IDAI.NE.PREVID) WRITE(6,696)
0016     696   FORMAT(//,'  A. I. HEADER RECORDS')
0017          IF (IDAI .NE. PREVID) WRITE (6,10)((IAIHED(I,J),I=1,73), J=1,2)
0018     10    FORMAT (1X,A4,71A1,A4)
0019          IF(IDAI.NE.PREVID) WRITE(6,697)
0020     697   FORMAT(//)
        C
0021          PREVID = IDAI
        C
0022          NLINE=NOLINE
0023          NPIXL=NOPIXL
0024          DO 50 I=1,4
0025     50    CANKY(I)=ICNKEY(I)
0026     100  WRITE (IOUT,IFMT) NLINE,NPIXL,IGT(NOLINE,NOPIXL),
             1  LABLAI(NOLINE,NOPIXL), MASK(NOLINE,NOPIXL),
             2  (IAIVEG(I),I=1,4),AICMNT(NOLINE,NOPIXL),
             1 (RDBRGT(NOLINE,NOPIXL,J),J=1,4), (RDGREN(NOLINE,NOPIXL,J),J=1,4)
             2  ,SDAIRB(NOLINE,NOPIXL),SDAIRG(NOLINE,NOPIXL),PIFB(NOLINE,NOPIXL)
             3  ,PIFG(NOLINE,NOPIXL),(CANKY(I),I=1,4), AICANK,
             4(ABRIFT(NOLINE,NOPIXL,I),I=1,4),  (AGREEN(NOLINE,NOPIXL,I),I=1,4)
        C
        C
        C
0027          RETURN
0028     200   CONTINUE
        C        SET UP FOR WRITING NULL RECORD
0029           NLINE = 0
0030           NPIXL = 0
0031           NOLINE=1
0032           NOPIXL=1
0033           GO TO 100
0034          END
```


67

A.8 SOFTWARE COMPONENT NO. 8 (RDLDF)

RDLDF in conjunction with PPTLAC reads LACIE formatted dot files.

A.8.1 Linkages

RDLDF is called by READGT and calls PPTLAC.

A.8.2 Interfaces

RDLDF interfaces with other routines through the common blocks DOTVEC, INFORM, GT, MSK, and LV.

A.8.3 Inputs

Inputs to RDLDF are from the program PPTLAC.

A.8.4 Outputs

N/A

A.8.5 Storage Requirement

Program size = 14460

A.8.6 Description

RDLDF in conjunction with PPTLAC read a LACIE format dot file from unit LUNIT.

A.8.7 Flowchart

N/A

A.8.8 Listing

```
 FORTRAN IV G LEVEL  21                RDLDF            DATE = 79003          16/55/04            PAGE 0001
FILE RDLDF

                      C       THIS SUBROUTINE READS LACIE FORMAT DOT FILES                                  RDL00010
 0001                        SUBROUTINE RDLDF                                                              RDL00020
 0002                        IMPLICIT INTEGER (A-Z)                                                        RDL00030
 0003                        COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6)                         RDL00040
                     *   ,PPTKEY,SIZE,LACIE                                                                RDL00050
 0004                        COMMON /INFORM/NOCLS2,NOSUB2,NOFET2,VARS22,TOTVT2,NOFLD2,                     RDL00060
                     *AVAR2,COVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTX2,                                     RDL00070
                     *FETVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60),                                         RDL00080
                     *KEPPTS(50),NOGRP,GRPNAM(60),GRPDFX(61),                                              RDL00090
                     *GRPCHK(61),GROUPS(124)                                                               RDL00100
 0005                        DIMENSION FIELDS(4,418),VERTEX(1672)                                          RDL00110
 0006                        COMMON /GT/IGT(11,19),IGTERR                                                  RDL00120
 0007                        COMMON /MSK/MASK(11,19)                                                       RDL00130
 0008                        COMMON /LUN/LUNIT                                                             RDL00140
 0009                        DIMENSION IHEAD(20)                                                           RDL00150
 0010                        DATA IDOT/'DOT '/,IEND/'$END'/                                                RDL00160
                      C                                                                                    RDL00170
                      C                                                                                    RDL00180
 0011                        NPRT=6                                                                        RDL00190
 0012                        IPT=0                                                                         RDL00200
 0013                        STAMNT=1                                                                      RDL00210
 0014                        TYPE=1                                                                        RDL00220
 0015                        NOCAT=0                                                                       RDL00230
 0016                        NOFLD2=0                                                                      RDL00240
 0017                        TOTVT2=0                                                                      RDL00250
 0018                        TOTVEC=0                                                                      RDL00260
                      C                                                                                    RDL00270
                      C**     RPTLAC SETS IPT=NOCAT=0   UPON FINDING TYPE CHANGE                           RDL00280
                      C                                                                                    RDL00290
                      C       WRITE(6,9950)                                                                RDL00300
 0019                 9950    FORMAT(*  ENTER RPTLAC*)                                                     RDL00310
                      C                                                                                    RDL00320
                      C                                                                                    RDL00330
 0020                           READ(LUNIT,200) IHEAD                                                     RDL00340
 0021                  200      FORMAT(20A4)                                                              RDL00350
 0022                         IF(IHEAD(1).EQ.IEND) GO TO 150                                              RDL00360
 0023                         IF(IHEAD(1).EQ.IDOT) BACKSPACE LUNIT                                        RDL00370
 0024                         IF(IHEAD(1).NE.IDOT)  WRITE(NPRT,201) IHEAD                                 RDL00380
 0025                   201    FORMAT(1H ,10X,20A4)                                                      RDL00390
 0026                   100    CALL RPTLAC(FIELDS,STAMNT,&130,&140,&150,                                 RDL00400
                     *   IPT,VERTEX)                                                                       RDL00410
                      C                                                                                    RDL00420
                      C**     A DOT HAS BEEN PROCESSED                                                     RDL00430
                      C                                                                                    RDL00440
 0027                   130  ILINE=FLDINF(1)/10                                                           RDL00450
 0028                        IPIXEL=FLDINF(4)/10                                                           RDL00460
 0029                       IF(ILINE.LT.1.OR.ILINE.GT.11) GO TO 160                                       RDL00470
 0030                       IF(IPIXEL.LT.1.OR.IPIXEL.GT.19) GO TO 160                                     RDL00480
 0031                        IGT(ILINE,IPIXEL)=CATNAM(NOCAT)                                              RDL00490
 0032                        MASK(ILINE,IPIXEL)=TYPE                                                       RDL00500
                      C                                                                                    RDL00510
                      C**     NOFLD2  HAS TOTAL NUMBER OF DOTS PROCESSED                                   RDL00520
                      C                                                                                    RDL00530
                      C**     DOT TYPE CHANGED                                                             RDL00540
                      C       WRITE(6,9995)ILINE,IPIXEL,CATNAM(NOCAT),TYPE                                 RDL00550
 0033                 9995    FORMAT(1H ,2I5,5X,A1,I5)                                                     RDL00560
                      C                                                                                    RDL00570
 0034                   140  GO TO 100                                                                    RDL00580
                      C                                                                                    RDL00590
                      C**     $END CARD REACHED                                                            RDL00600
                      C                                                                                    RDL00610
 0035                   150  RETURN                                                                       RDL00620
 0036                   160  CONTINUE                                                                     RDL00630
 0037                        WRITE(NPRT,9966)                                                              RDL00640
 0038                 9966    FORMAT(*   LACIE NUMBER OUT OF RANGE*)                                       RDL00650
 0039                        RETURN                                                                        RDL00660
 0040                        END                                                                           RDL00670
```

A.9 SOFTWARE COMPONENT NO. 9 (PPTLAC)

The new subprogram PPTLAC will read and decode the LACIE formatted field (dot) cards. This is the same program as FLDLAC except it read from unit LUNIT instead of unit 21.

A.9.1 Linkages

PPTLAC is called by subprogram RDLDF. Each call to PPTLAC will provide, upon return

1. a dot (field) description (first return)
2. transfer to dot file writing (second turn)
3. transfer to dot file writing (third return)

A.9.2 Interfaces

PPTLAC will interface with other routines through a calling sequence and common blocks DOTVEC, LU, and INFORM.

A.9.3 Inputs

Calling sequence:

SUBROUTINE PPTLAC (FIELDS, STAMNT, $, $, $, IPT, VERTEX)

Parameter	Dimension	In/Out	Description
FIELDS	(4,250)	Out	Category name and dot type for dot I stored in FIELDS (1,I) and FIELDS (4,I).
STAMNT	1	In/Out	Initially set equal to 1, switch to indicate dots being taken from currently read card.
$			Returns to DOTS.
$			Returns to DOTS.
$			Returns to DOTS.
PIT	1	In/Out	Initially set equal to 1, index number for field vertex information.

Parameter	Dimension	In/Out	Description
VERTEX	1000	Out	Vertex information for each dot.

In addition, PPTLAC stores the FLDINF vector in common block DOTVEC with rectangular coordinates of field enclosing each dot field.

A.9.4 Outputs

N/A

A.9.5 Storage Requirement

Program Size

A.9.6 Description

The subroutine reads of dot cards of the form

```
DOT      (TYPE)      (CATEGORY NAME)      ({LACIE NUMBERS}), i.e.,
DOT  1     W     2   5    10       29      32      54      110
```

"TYPE" cards are used to prefix a set of dots. This will remain as the default option. The association between LACIE numbers and training field coordinates is as follows.

SAMPLE NUMBER

LINE NUMBER	10	20	30	.	.	.	190
10	1	2	3				19
20	20						38
30	39						57
:			LACIE				
110	191		NUMBERS				209

It is intended that two expansions of the LACIE card format be incorporated. These are

1. free-field locations of all information cards, cols 1-72, data items separated by at least one blank, with the restriction that DOT identifiers start in col 1, and the dot type appear in column 5.

2. In order to cover pixels not included in the LACIE numbering scheme, input dot numbers will be represented as the numerical equivalents of

   $N = L*10^8 + S*10^4 + LACIE$

   where

   LI = #lines to be incremented (up or down) from the line number mapped from the LACIE number. The convention will will be

LI	negative	to	increment up
LI	positive	to	increment down
LI	zero	to	avoid incrementation

   SI = #samples to be incremented (right or left)

SI	negative	to	increment left
SI	positive	to	increment right
SI	zero	to	avoid incrementation.

For example, LI=2, SI= -3, LACIE = 38 yields
$N = 2*10^8 - 3*10^4 + 38 = 199970038$
would correspond to the pixel at (187,22), ie, the pixel at sample number 187 and line number 22.
Letting LI= SI=0, LACIE = 38, obtain
N = 38, correspond to the pixel at (190,20).

Reduction of the value of N to sample and line coordinates will proceed as follows.

$N_1 = |N|/10^8$ (truncated to integer)

if $|N|-N, *10^8 \geq 10^7$ set $N_1 = N_1+1$

$LI = N_1 * \text{sign } (N)$

$|N_2| = N-L*10^8$

$N_3 = |N_2|/10^4$ (truncated to integer)

if $|N_2|-N_3*10^4 \geq 10^3$ set $N_3 = N_3+1$

$SI = N_3 * \text{sign } (N_2)$

$LACIE = N_2-S*10^4$

LR = LACIE ROW# = (LACIE/19+1) *10

LS = LACIE COL# = (LACIE-((LR-1)*19)/10) *10

where truncated divides are specified.

Finally,

L = LR+LI line number corresponding to N

S = LS+SI sample (column) number corresponding to N

In the scheme to follow, each dot will be considered to be a field. All type 1 dots will occur prior to type 2 dots; i.e., the input cards cannot be scrambled with respect to dot type. Otherwise, arbitrary order to cards and LACIE numbers on each card are permitted.

The function of the various parameters is as follows.

IPT index number for dot (field) vertex information

NOFLD2 number of fields (dots) for dots of current type (common block INFORM)

SWCHG — number of times dot type has changed. This must be no greater than 1 or an input error will have occurred.

SWITCH — flags a dot type change. The second return will be taken for subsequent writing of a dot field. (internal)

STAMNT — if = 1, a new dot card has been ready if = 2, dots are being processed from a previously read card.

TYPE — dot type being processed (common block DOTVEC)

The calling sequence of PPTLAC is the same as that for FLDTYP, and the meaning of FIELDS and VERTEX remains the same.

```
      IF (STAMNT.EQ.2)  GΦ  TΦ  30
      IF (.NOT.SWITCH)  GΦ  TΦ  20

10    READ A CARD, extract TYPES from column 5
      If (TYPE.EQ.TYPES) GΦ  TΦ  20
      If (SWCHG.NE.O)  error exit
      TYPE=TYPES

20    RE-READ CARD, extract
            CATNM      category name
            NDCARD     # dots on this card
            NDOTS(I), I=1,NDCARD   dots on this card
      If (NDCARD.EQ.O)  GΦ  TΦ  10
      ICNT = 0
      STAMNT = 2
      SWITCH = .TRUE.
      GΦ  TΦ  100
```

```
30   If (ICNT.LT.NDCARD)  GO  TO  100
     STAMNT = 1
     ICNT   = 0
     READ A CARD, extract first 4 characters and store in IDUM,
     extract TYPES
     If (IDUM.EQ.ENDBCD) RETURN 5
     If (TYPE.EQ.TYPES)  GΦ  TΦ  20
     SWITCH. = .FALSE.
     SWCHG = SWCHG+1
     If (SWCHG.GT.1)  error exit
     NΦFLD2 = 0; TYPE = TYPES
     IPT = 1
     RETURN 4

100  INCNT = ICNT+1
     NOFLD2 = NOFLD2+1
     Find sample and line numbers S and L from NDOTS (ICNT) as
     described previously.
     Store
     FIELDS (1,NOFLD2) = CATNM
     FIELDS (4,NOFLD2) = 2
     FLDINF (1) = L
     FLDINF (2) = L
     FLDINF (3) = 1
     FLDINF (4) = S
     FLDINF (5) = S
     FLDINF (6) = 1
```

rectangular bodering field (dot)

```
VERTEX (IPT)   = S
VERTEX (IPT+1) = L
VERTEX (IPT+2) = S
VERTEX (IPT+3) = L
IPT = IPT+4
RETURN
END
```

Regarding the extraction of dot numbers NDOTS(I), I=1, NDCARD, a new routine, NUMBR, similar to existing function NUMBER will be provided. The differences will be

1. NUMBR will recognize blanks instead of commas as delimiters
2. NUMBR will process only one card of information

A.9.7 Flowchart

N/A

A.9.8 Listing

```
FILE PPTLAC
                C     FIELDS - CATEGORY NAME AND DOT TYPE FOR DOT I STORED IN           PPT00010
                C              FIELD(1,I) AND FIELD(4,I)                                PPT00020
                C     STAMNT - INITIALLY SET TO 1, SWITCHED TO INDICATE DOTS BEING     PPT00030
                C              TAKEN FROM CURRENTLY READ CARD.                          PPT00040
                C     IPT - INITIALLY SET TO 1,INDEX NUMBER FOR FIELD VERTEX INFORMATIONPPT00050
                C     VERTEX - VERTEX INFORMATION FOR EACH DOT.                        PPT00060
                C                                                                      PPT00070
0001                  SUBROUTINE PPTLAC(FIELDS,STAMNT,*,*,*,IPT,VERTEX)                PPT00080
0002                  IMPLICIT INTEGER (A-Z)                                           PPT00090
0003                  REAL DUM                                                         PPT00100
0004                  DIMENSION FIELDS(4,1),VERTEX(1),CARD(75),NDOTS(30)               PPT00110
0005                  DIMENSION ACARD(80)                                              PPT00120
0006                  LOGICAL SWITCH                                                   PPT00130
0007                  DATA SWITCH/.TRUE./,SWCHG/0/,ENDBCD/'$EN'/,                      PPT00140
                     *CATNM1/' '/                                                      PPT00150
0008                  DATA ISBLANK/' '/                                                PPT00160
                C     INCLUDE COMK14                                                   PPT00170
                C     INCLUDE COM=<1                                                   PPT00180
0009                  COMMON/INFO3W/NDCLS2,NDSUB2,NDFET2,VAPS72,TOTVT2,NOFLD2,         PPT00190
                     *              AVAR2,COVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTX2,  PPT00200
                     *              FETVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60),      PPT00210
                     *              KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),           PPT00220
                     *              GRPCHR(61),GROUPS(124)                            PPT00230
0010                  COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY     PPT00240
                     *                ,SIZE ,LACIE                                    PPT00250
                C(REAL)                                                                PPT00260
                C                                                                      PPT00270
0011                  COMMON /LU/LUNIT                                                 PPT00280
                C                                                                      PPT00290
0012                  IF(STAMNT.EQ.2)GO TO 30                                          PPT00300
0013                  IF(.NOT.SWITCH)GO TO 20                                          PPT00310
0014                  CALL PREAD(30,80)                                                PPT00320
0015                10 READ(LUNIT,103)(ACARD(I),I=1,80)                               PPT00330
0016                103 FORMAT(80A1)                                                   PPT00340
0017                  **WRITE(30,103)(ACARD(I),I=1,80)                                 PPT00350
0018                  REWIND 30                                                        PPT00360
0019                  READ(30,1000)ID,TYPES,CARD                                       PPT00370
                C     **WRITE(6,9998) ID,ENDBCD                                        PPT00380
0020                9998 FORMAT(1H ,2A4)                                               PPT00390
0021                  IF(ID.EQ.ENDBCD) GO TO 333                                       PPT00400
0022                  REWIND 30                                                        PPT00410
0023                1000 FORMAT(A3,1X,I1,75A1)                                         PPT00420
0024                  IF(TYPE.EQ.TYPES )GO TO 20                                       PPT00430
0025                  IF(SWCHG.NE.0)GO TO 40                                           PPT00440
0026                  TYPE = TYPES                                                     PPT00450
                C                                                                      PPT00460
                C       READ CARD                                                      PPT00470
                C                                                                      PPT00480
0027                 20 COL = 0                                                        PPT00490
0028                  CATNM = NXTCHR(CARD,COL)                                         PPT00500
                C*    IF NEXT CHAR IS NOT A CAT. NAME, CORRECT COL COUNT TO READ NUM   PPT00510
0029                  IF(CATNM.GT.0)GO TO 21                                           PPT00520
0030                  IF(CATNM.EQ.CATNM1)GO TO 23                                      PPT00530
0031                  NOCAT= NOCAT + 1                                                 PPT00540
0032                  CATNAM(NOCAT)=CATNM                                              PPT00550
0033                  CATNM1 = CATNM                                                   PPT00560
0034                  GO TO 23                                                         PPT00570
0035                 21 COL=COL - 1                                                    PPT00580
0036                 23 NDCARD=0                                                       PPT00590
0037                  CALL NUMBR(NDOTS,NDCARD,CARD,COL)                                PPT00600
0038                  IF(NDCARD.EQ.0)GO TO 10                                          PPT00610
0039                  ICNT = 0                                                         PPT00620
0040                  STAMNT = 2                                                       PPT00630
0041                  SWITCH = .TRUE.                                                  PPT00640
0042                  GO TO 100                                                        PPT00650
                C                                                                      PPT00660
                C     TEST FOR END OF DOTS TO BE PROCESSED ON CARD                     PPT00670
                C                                                                      PPT00680
0043                 30 IF(ICNT.LT.NDCARD)GO TO 100                                    PPT00690
                C                                                                      PPT00700
                C     READ NEXT CARD                                                   PPT00710
                C                                                                      PPT00720
0044                  STAMNT = 1                                                       PPT00730
0045                  ICNT = 0                                                         PPT00740
0046                  READ(LUNIT,103)(ACARD(I),I=1,80)                                 PPT00750
0047                  WRITE(30,103)(ACARD(I),I=1,80)                                   PPT00760
```

```
FILE PPTLAC
0048                REWIND 30                                                   PPT00770
0049                READ(30,1000 )ID,TYPES,CARD                                 PPT00780
                C   WRITE(6,9998) ID,ENDRCD                                     PPT00790
0050                REWIND 30                                                   PPT00800
0051                IF(ID.EQ.ENDRCD)GO TO 333                                   PPT00810
0052                IF(TYPE.EQ.TYPES)GO TO 20                                   PPT00820
0053                SWITCH = .FALSE.                                            PPT00830
0054                S.CHG = S.CHG + 1                                           PPT00840
0055                IF(S.CHG.GT.1)GO TO 40                                      PPT00850
0056                TYPE = TYPES                                                PPT00860
0057                 IPT = 0                                                    PPT00870
                C***** CHANGED JUNE 28 1978                                     PPT00880
0058                RETURN 2                                                    PPT00890
                C                                                               PPT00900
                C                                                               PPT00910
0059                 100  ICNT = ICNT + 1                                      PPT00920
0060                NOFLD2 = NOFLD2 + 1                                         PPT00930
                C                                                               PPT00940
                CC      COMPUTE LINE INCREMENT                                  PPT00950
                C                                                               PPT00960
0061                NN = NDOTS(ICNT)                                            PPT00970
0062                NI =IABS( NN) / 100000000                                   PPT00980
0063                LI = IABS(NN) - NI * 100000000                              PPT00990
0064                IF(LI.GE.100000000)NI = NI +1                               PPT01000
                C                                                               PPT01010
                CC      COMPUTE SAMPLE INCREMENT                                PPT01020
                C                                                               PPT01030
0065                KK=1                                                        PPT01040
0066                IF(NN.LT.0)KK=-1                                            PPT01050
0067                LI = NI * KK                                                PPT01060
0068                N2 = NN - LI * 100000000                                    PPT01070
0069                N3 = IABS( N2)/10000                                        PPT01080
0070                SI = IABS(N2)-N3 * 10000                                    PPT01090
0071                IF(SI.GE.1000)N3 = N3 + 1                                   PPT01100
0072                KK=1                                                        PPT01110
0073                IF(N2.LT.0)KK=-1                                            PPT01120
0074                SI = N3 *KK                                                 PPT01130
0075                LACI = N2 - SI * 10000                                      PPT01140
0076                LR = (LACI-1)/10                                            PPT01150
0077                LP = (LR+1) * 10                                            PPT01160
0078                LS = LP - 1                                                 PPT01170
0079                LS = LS /10                                                 PPT01180
0080                LS = 10 * (LACI - (LS*10))                                  PPT01190
0081                L = LP -LI                                                  PPT01200
0082                S = LS + SI                                                 PPT01210
                CC      STORE DOT INFO                                          PPT01220
                C                                                               PPT01230
                                                                                PPT01240
0083                FIELDS(1,NOFLD2) = CATNM                                    PPT01250
0084                FIELDS(4,NOFLD2) = 2                                        PPT01260
0085                FLOINF(1) = L                                               PPT01270
0086                FLOINF(2) = L                                               PPT01280
0087                FLOINF(3) = L                                               PPT01290
0088                FLOINF(4) = S                                               PPT01300
0089                FLOINF(5) = S                                               PPT01310
0090                FLOINF(6) = 1                                               PPT01320
0091                IF(IPT.NE.0)GO TO 35                                        PPT01330
0092                IPT = -3                                                    PPT01340
0093                 35   IPT = IPT + 4                                        PPT01350
0094                VERTEX(IPT) = S                                             PPT01360
0095                VERTEX(IPT+1)=L                                             PPT01370
0096                VERTEX(IPT+2)=S                                             PPT01380
0097                VERTEX(IPT+3)=L                                             PPT01390
0098                RETURN 1                                                    PPT01400
0099                 40   WRITE(6,2000)                                        PPT01410
0100                 2000 FORMAT(//5X,'ERROR HAS OCCURRED IN READING LACIE FORMATTED DOT CAPPT01420
                    *DS - SUBROUTINE PPTLAC - EXIT TAKEN')                      PPT01430
0101                 333   CONTINUE                                           PPT01440
0102                SWITCH=.TRUE.                                               PPT01450
0103                S.CHG =0                                                    PPT01460
0104                CATNM1=IBLANK                                               PPT01470
0105                RETURN 3                                                    PPT01480
0106                END                                                         PPT01490
```

APPENDIX B

B.1 SOFTWARE SUBPROGRAM NO. 1 (SPPTA)

Subroutine SPPTA reads the input cards and sets option switches for the processor MPTTA.

B.1.1 Linkages

SPPTA is called by the program MPPTA and uses data initialized in PPTBLK. SPPTA uses the function ALPHA and calls the subroutine SETTRM.

B.1.2 Interfaces

SPPTA interfaces with MPPTA through a calling sequence and interfaces with MPPTA and PPTBLK through common blocks UN, PF, TR and FV.

B.1.3 Inputs

Calling sequence: Subr. SPPTA (D, T, ISGZ, NT, E, CATREC, C, N1, N2)

Parameter	Dimension	In/Out	Description
D	1	Out	No. of channel
T	1	Out	No. of classes
ISGZ	26	Out	No. of pixels of acertain
NT	1	Out	Total no. of samples category
E	1	Out	Error Tolerance
C	(10,10)	Out	Cost Matrix
CATREC	26	Out	An array use to indicate which categories are being used.
N1	1	Out	A number that determines certain array sizes

Parameter	Dimension	In/Out	Description
N2	1	Out	A number that determines certain array sizes.

Common Blocks:

See PPTBLK for information about the common blocks.

Input cards (unit NRDR1):

	Variables	Format	Function
1.	PFLAG	I5	0- for short printout 1- for printout
2.	D,T,	4I5 2I5	D- no. channels T- no. of classes (at present T=2)
3.	ISYM,ICREC	(AI,I3)	Category symbol and associated index (ICREC=0 indicates an end of file for this information).
4.	((C(I,J),J=1,T),	10F5.2	The cost matrix (one row per card)
5.	IDEF	A1	D- use default data vector input format N- input an input format
6.	(use if IDEF=N) NDATA	I5	Number of data points per pixel
7.	(use if IDEF=n) (IFMT(I), I=1,20)	20A4	Input format
8.	IDEF	A1	D- use default feature index vector N- input a feature index vector

	Variables	Format	Function
9.	(Use if IDEF=N) (FEATVC(I), I-1,D)	30I2	The feature index vector.

B.1.4 Outputs

Input information is printed out.

B.1.5 Storage

Program size = 2900.

B.1.6 Description

SPPTA is the input subroutine for all except the pixel data. If default options are not used this subroutine inputs the format for the pixel data and the feature index vector.

B.1.7 Flowchart

N/A

B.1.8 Listing

FILE SPPTA

```
C     ADAPTED BY C W AHLERS
      SUBROUTINE SPPTA(D,T,ISGZ,NT,E,CATREC,C,N1,N2)
      INTEGER D,T,CAT,RUNIT,WUNIT
      INTEGER TERMS
      INTEGER ALPHA
      INTEGER CATREC(26)
      INTEGER ISGZ(26)
       COMMON /UN/NRDR1,NRDR2,NPRT,RUNIT,WUNIT
      COMMON /TUN/NTRM
      COMMON /TR/TERMS(361)
      INTEGER PFLAG
      COMMON /PF/PFLAG
      INTEGER FEATVC,TT,DD
      COMMON /FV/FEATVC(30),IFMT(30),NDATA
C     PARAMETER NN1=361,NN2=65522,TT=10,DD=30
C       NN1 & NN2 BASED ON 25 CHANNELS
C DIMENSIONS CHECKED IN THIS SUBROUTINE
      DATA NN1/361/,NN2/65522/,TT/10/,DD/30/
      DATA IY/'Y'/
      DOUBLE PRECISION C(10,10)
      DOUBLE PRECISION E
C     WRITE(NTRM,1101)
 1101  FORMAT(1H ,'SPPTA')
      REWIND NRDR1
      REWIND NRDR2
      DO 111 I=1,26
  111 CATREC(I)=0
      DO 112 I=1,NN1
  112  TERMS(I)=0
C READ INPUT DATA.
      READ(NRDR1,10) PFLAG
  10   FORMAT(3I5)
      WRITE(NPRT,837) PFLAG
 837   FORMAT(//,10X,'PFLAG=',I5)
        WRITE(NPRT,20) RUNIT,WUNIT
  20  FORMAT(//,10X,'RUNIT=',I5,5X,'WUNIT=',I5)
      READ(NRDR1,21) D,T
  21  FORMAT(4I5)
      WRITE(NPRT,22) D,T
  22   FORMAT(//,10X,'NO. OF CHANNELS=',I5,//,10X,'NO. OF CLASSES=',
     *  I5)
       IF(D.GT.DD) WRITE(NPRT,101) DD
       IF(T.GT.TT) WRITE(NPRT,102) TT
 101    FORMAT(//,10X,' ERROR -- THE NO. OF CHANNELS EXCEEDS ',I5)
 102    FORMAT(//,10X,' ERROR -- THE NO. OF CLASSES EXCEEDS ',I5)
      IF(D.GT.DD) STOP
      IF(T.GT.TT) STOP
      WRITE(NPRT,168)
 168   FORMAT(///,12X,'CATEGORY',8X,'INDEX',/)
 169   CONTINUE
      READ(NRDR1,166) ISYM,ICREC
      WRITE(NPRT,167) ISYM,ICREC
 167   FORMAT(1H ,15X,A1,10X,I5)
      IF(ICREC.EQ.0) GO TO 165
 166  FORMAT(A1,I3)
      ICAT=ALPHA(ISYM)
      CATREC(ICAT)=ICREC
      GO TO 169
 165   CONTINUE
C     WRITE(NPRT,269) CATREC
 269   FORMAT(1H ,26I2)
C     READ(NRDR1,1) E
    1 FORMAT(F10.7)
C     WRITE(NPRT,15) E
 15   FORMAT(//,10X,'E = ',F16.8)
    2 FORMAT(10F5.2)
      WRITE(NPRT,35)
 35    FORMAT(//,10X,'THE COST MATRIX')
      DO 150 I=1,T
      READ(NRDR1,2) (C(I,J),J=1,T)
      WRITE(NPRT,25) (C(I,J),J=1,T)
 150   CONTINUE
 25   FORMAT(//,10X,10F5.2)
      READ(NRDR1,506) IDEF
 506  FORMAT(A1)
      IF(IDEF.EQ.IY) GO TO 507
```

83

```
FILE SPPTA

C  READ IN NO. OF DATA POINTS PER PIXEL                                  SPP00770
      READ (NRDR),21) NDATA                                              SPP00780
  444  FORMAT(//,10X,'NDATA=',I5)                                        SPP00790
C    READ IN THE FORMAT FOR THE FEATURE VECTOR                           SPP00800
      READ(NRDR),450) (IFMT(I),I=1,20)                                   SPP00810
  450 FORMAT(20A4)                                                       SPP00820
  507  CONTINUE                                                          SPP00830
      WRITE(NPRT,444) NDATA                                              SPP00840
      WRITE(NPRT,459)                                                    SPP00850
      WRITE(NPRT,451) IFMT                                               SPP00860
  459 FORMAT(//,10X,'FEATURE VECTOR FORMAT  ')                           SPP00870
  451  FORMAT(//,10X,20A4)                                               SPP00880
      READ(NRDR),504) IDEF                                               SPP00890
      IF(IDEF.EQ.IY) GO TO 509                                           SPP00900
C  READ IN THE FEATURE  INDEX VECTOR                                     SPP00910
      READ(NRDR),452) (FEATVC(I),I=1,D)                                  SPP00920
  452 FORMAT(3012)                                                       SPP00930
  509  CONTINUE                                                          SPP00940
      WRITE(NPRT,453) (FEATVC(I),I=1,D)                                  SPP00950
  453 FORMAT(//,10X,'THE FEATURE INDEX VECTOR ',//,10X,3013)             SPP00960
      DO 300 I=1,D                                                       SPP00970
      IF(FEATVC(I).LE.0) WRITE(NPRT,301)                                 SPP00980
      IF(FEATVC(I).LE.0) STOP                                            SPP00990
  300 CONTINUE                                                           SPP01000
  301 FORMAT(//,10X,'ERROR -- FEATURE VECTOR FORMAT - OR -',             SPP01010
     *' FEATURE INDEX VECTOR IS WRONG')                                  SPP01020
      N1=D+1+(D*(D+1))/2                                                 SPP01030
      N2=N1+(N1*(N1+1))/2                                                SPP01040
      WRITE(NPRT,33) N1,N2                                               SPP01050
  33  FORMAT(//,10X,'N1=',I5,5X,'N2=',I5)                               SPP01060
      IF(N1.GT.NN1) WRITE(NPRT,103) NN1                                  SPP01070
      IF(N2.GT.NN2) WRITE(NPRT,104) NN2                                  SPP01080
      IF(N2.GT.NN2) N2= NN2                                              SPP01090
      IF(N1.GT.NN1) STOP                                                 SPP01100
      IF(N2.GT.NN2) STOP                                                 SPP01110
      CALL SETTRM(0)                                                     SPP01120
  103  FORMAT(//,10X,' ERROR -- N1 EXCEEDS ',I10)                        SPP01130
C104   FORMAT(//,10X,' ERROR -- N2 EXCEEDS ',I10)                        SPP01140
  104  FORMAT(//,10X,' N2 REPLACED BY ',I10)                             SPP01150
      RETURN                                                             SPP01160
      END                                                                SPP01170
```

43

B.2 SOFTWARE COMPONENT NO. 2 (PPTA)

Subroutine PPTA is the main computational subroutine of the processor MPTTA. Input from SPPTA is passed to PPTA. PPTA with the aid of other subroutines calculates the loss vector matrix and writes it out to unit WUNIT.

B.2.1 Linkages

Subroutine PPTA is called by MPPTA and is passed information from SPPTA. PPTA calls subroutines READIT, PHI, COLINV, and NP.

B.2.2 Interfaces

PPTA interfaces with other routines through a calling sequence and common blocks UN and PF.

B.2.3 Inputs

Calling sequence:

Subr. PPTA (D, T, ISGZ, NT, N1, N2, PNI, P, Q, R, S, A, E, CATREC, C, X)

Parameter	Dimension	In/Out	Description
D	1	In	No. of channels
T	1	In	No. of classes
ISGZ	26	In	No. of pixels of a certain category
NT	1	In	Total no, of samples
N1	1	In	Dimension for some arrays
N2	1	In	Dimension for some arrays
PNI	N2	In	PN inverse
P	N1	In	Phi function vector
Q	(N1,T)	In	Class phi sum matrix
R	N1	In	PNI*P
S	(N1,T)	In	Working storage

Parameter	Dimension	In/Out	Description
A	(N1,T)	Out	The loss vector matrix
E	1	In	Error Tolerance
CATREC	26	In	Records categories used.
C	(10,10)	In	Cost matrix
X	D	In	The feature vector

Common Blocks:

See PPTBLK for information about the common blocks.

B.2.4 Outputs

The loss vector matrix is printed out and written to unit WUNIT. Optional information is printed out if PFLAG = 1.

B.2.5 Storage

Program size = 5808 bytes

B.2.6 Description

PPTA uses the input of SPPTA and READIT as principle input to compute the loss vector matrix and writes it to unit WUNIT. The method for computing the inverse of matrices in this routine will be changed.

B.2.7 Flowchart

N/A

B.2.8 Listing

```
FILE PPTA

                C   ADAPTED BY C W AHLERS                                                PPT00010
                C THE PATTERSON-PITT-THADANI ALGORITHM.                                  PPT00020
                C THE FINAL LOSS VECTOR MATRIX A IS OUTPUT TO UNIT   WUNIT.              PPT00030
                C TRAINING.                                                              PPT00040
                C P(N1)....PHI FUNCTION VECTOR.                                          PPT00050
                C Q(N1,T)....CLASS PHI SUM MATRIX.                                       PPT00060
                C R(N1)....NOT USED                                                      PPT00070
                C PHI(N2)....PHI INVERSE MATRIX.                                         PPT00080
                C CATREC(26)....A RECORD OF CATEGORY SYMBOLS                             PPT00090
                C ISG7(26)....A COUNT VECTOR FOR CATEGORIES                              PPT00100
0001                  SUBROUTINE PPTA(D,T,ISG7,NT,N1,N2,PHI,P,Q,R,S,A,E,                 PPT00110
                     *  CATREC,C,X)                                                      PPT00120
0002                  INTEGER D,T,CAT,RUNIT,WUNIT                                        PPT00130
0003                  INTEGER CATREC(26)                                                 PPT00140
0004                  INTEGER ALPHA                                                      PPT00150
0005                  INTEGER SAMPLE,GTCAT,AICAT                                         PPT00160
0006                  COMMON /UN/NRDR2,NPRT,RUNIT,WUNIT                                  PPT00170
0007                  COMMON /LT/LETTER(26)                                              PPT00180
0008                  INTEGER PFLAG                                                      PPT00190
0009                  INTEGER TERMS                                                      PPT00200
0010                  INTEGER TYPE                                                       PPT00210
0011                  COMMON /TUN/NTHM                                                   PPT00220
0012                  COMMON /TRM/TERMS(361)                                             PPT00230
0013                  DIMENSION IHEAD(20)                                                PPT00240
0014                  DIMENSION ISG7(26)                                                 PPT00250
0015                  COMMON /PF/PFLAG                                                   PPT00260
0016                  DOUBLE PRECISION PHI(N2),P(N1),Q(N1,T),R(N1),S(N1,T)               PPT00270
0017                  DOUBLE PRECISION A(N1,T),ALPHA1,TRACE                              PPT00280
0018                  DIMENSION IALPHA(4)                                                PPT00290
0019                  DOUBLE PRECISION C(10,10),X(D)                                     PPT00300
0020                  DOUBLE PRECISION E,DET                                             PPT00310
0021                  DATA IRLNK/'    '/                                                 PPT00320
0022                  DATA IHK/' '/                                                      PPT00330
                C     WRITE(NTHM,888)                                                    PPT00340
0023            888   FORMAT(1H ,'PPTA')                                                 PPT00350
                C     WRITE(NPRT,999) D,T,NT,RUNIT,WUNIT,N1,N2,E,C                       PPT00360
0024            999   FORMAT(//,2X,7I5,F10.7,10(/,10F5.2))                             PPT00370
                C INITIALIZE PHI,P,Q,R,A                                                 PPT00380
0025                  DO 3 I=1,N2                                                        PPT00390
0026                  PHI(N2)=0.000                                                      PPT00400
0027                3 CONTINUE                                                           PPT00410
0028                  DO 4 I=1,N1                                                        PPT00420
0029                  DO 4 J=1,T                                                         PPT00430
0030                  Q(I,J)=0.000                                                       PPT00440
0031                  A(I,J)=0.000                                                       PPT00450
0032                  S(I,J)=0.000                                                       PPT00460
0033                4 CONTINUE                                                           PPT00470
0034                  DO 369 I=1,26                                                      PPT00480
0035                  ISG7(I)=0                                                          PPT00490
0036            369   CONTINUE                                                           PPT00500
                C     WRITE(NPRT,669) CATREC                                             PPT00510
0037             669  FORMAT(1H ,'CATREC',26I3)                                          PPT00520
0038                  ITXX=T                                                             PPT00530
                C     T=0                                                                PPT00540
0039                  M=0                                                                PPT00550
                C COMPUTE NO. OF PHI FUNCTIONS.                                          PPT00560
0040                  DO 369 I=1,N1                                                      PPT00570
0041             368    IF(TERMS(I).EQ.1) M=M+1                                         PPT00580
                C                                                                        PPT00590
0042                  DO 5 I=1,N1                                                        PPT00600
0043                  P(I)=0.000                                                         PPT00610
0044                  P(I)=0.000                                                         PPT00620
0045                5 CONTINUE                                                           PPT00630
0046                  TUNU=0                                                             PPT00640
0047                  TUNGT=0                                                            PPT00650
0048                  IF(PFLAG.GE.1) WRITE(NPRT,937)                                     PPT00660
0049             937   FORMAT(//,10X,'THE INPUT DATA',//)                              PPT00670
0050                  KOUNT=0                                                            PPT00680
0051             707    CONTINUE                                                        PPT00690
0052                  DO 464  I = 1,2                                                    PPT00700
0053                   READ(NRDR2,462) IHEAD                                           PPT00710
0054                  IF(IHEAD(1).EQ.IRLNK) GO TO 10                                     PPT00720
0055             462  FORMAT(20A4)                                                      PPT00730
0056                   WRITE(NPRT,463) IHEAD                                           PPT00740
0057            463   FORMAT(//,10X,20A4)                                              PPT00750
0058            464     CONTINUE                                                       PPT00760
```

87

FILE PPTA

```
0059          11    CONTINUE
             C      WRITE(NTRM,10001) KOUNT
0060         10001  FORMAT(1X,'KOUNT',6X,I10)
             C      WRITE(NTRM,988) D
0061          988   FORMAT(1H ,'D = ', I5)
0062                CALL READIT(LINE,SAMPLE,GTCAT,AICAT,TYPE,X,D,ICMNT,IAIVEG)
0063                 IF(LINE.EQ.0) GO TO 707
0064                 KOUNT=KOUNT+1
             C      WRITE(NTRM,789)
0065          789   FORMAT(1H ,'OUT OF READIT')
             C      WRITE(NTRM,988) D
0066                IF(PFLAG.GE.1) WRITE(NPRT,822) (X(I),I=1,D)
0067          822    FORMAT(1H ,2X,'X',2X,15F6.1)
0068                IF(PFLAG.GE.1) WRITE(NPRT,801) LINE,SAMPLE,GTCAT,AICAT,TYPE,ICMNT,
                  *  (IAIVEG(I),I=1,4)
0069          801   FORMAT(1H ,5X,2I4,1X,A1,1X,A1,I2,1X,A4,1X,4I1)
0070                ICAT=ALPHA(GTCAT)
0071                IF(GTCAT.EQ.IPK) IUNGT=IUNGT+1
0072                IF(GTCAT.EQ.IPK) GO TO 11
0073                IF(ISGZ(ICAT).EQ.0) IUNU= IUNU+ 1
             C      IF(T.GT.ITXX)  WRITE(NPRT,511) GTCAT
             C      IF(T.GT.ITXX) WRITE(NPRT,10001) KOUNT
             C      IF(T.GT.ITXX) STOP
0074                ISGZ(ICAT)=ISGZ(ICAT)+1
0075                CAT=CATREC(ICAT)
             C      IF(CAT.EQ.0) WRITE(NPRT,511) GTCAT
             C      IF(CAT.EQ.0) WRITE(NPRT,533) ICAT,T,ISGZ,CATREC
0076          533    FORMAT(1H ,'ICAT=',I5,'  T=',I5,//,26I2,//,26I2)
0077          511    FORMAT(//,10X,'CATEGORY ',A1,' WAS NOT ALLOWED FOR')
0078                 IF(CAT.EQ.0) GO TO 11
             C COMPUTE PHI FUNCTION VECTOR.
0079            12 CALL PHI(X,P,D,NI)
0080               IF(PFLAG.GE.2) WRITE(NPRT,802) (P(I),I=1,M)
0081           802 FORMAT(/1X,'PHI VECTOR',/,(1X,3D20.10))
             C UPDATE PHI SUM MATRIX Q.
0082               DO 13 I=1,M
0083               A(I,CAT)=A(I,CAT)+P(I)
0084            13 CONTINUE
0085               K=0
0086               DO 14 I=1,M
0087               DO 14 J=1,I
0088               K=K+1
0089               PNI(K)=P(I)*P(J) + PNI(K)
0090          14    CONTINUE
0091               IK=(KOUNT/5)*5
0092               IF(KOUNT .EQ. IK.AND.PFLAG.GE.1) WRITE(NPRT,10001) KOUNT
0093                GO TO 11
0094            10 CONTINUE
0095                WRITE(NPRT,111) KOUNT
0096          111   FORMAT(//,10X,'THE NO. OF SAMPLES = ',I8)
             C        COMPUTE PN INVERSE
0097                CALL COLINV(PNI,M,IERR,3,DET)
0098               IF(IERR.NE.0) WRITE(NPRT,807)
0099          807   FORMAT(///,10X,'PN IS NOT POSITIVE DEFINITE')
0100               IF(IERR.NE.0) STOP
0101               DO 50 I=1,T
0102               DO 50 J=1,T
0103               DO 70 K=1,M
0104               S(K,I)=S(K,I) + C(I,J)*A(K,J)
0105            70 CONTINUE
0106            60 CONTINUE
0107            50 CONTINUE
0108           230 DO 23 I=1,T
0109               DO 24  J=1,M
0110               DO 25 K=1,M
0111               IE=NLOW(J,K,M)
0112               IF(K.GT.J) IE=NLOW(K,J,M)
0113               Q(J,I)=Q(J,I)+PNI(IE)*A(K,I)
0114            25 CONTINUE
0115            24 CONTINUE
0116            23 CONTINUE
0117           103 CONTINUE
             C     WRITE(NPRT,241)
0118          241   FORMAT(///,10X,' THE Q MATRIX',//)
             C     DO 240 I=1,M
             C     WRITE(NPRT,238) (Q(I,J),J=1,T)
```

88

```
FILE PPTA

                C240   CONTINUE                                                  PPT01530
                C COMPUTE A.                                                     PPT01540
   0119            26 DO 28 I=1,T                                                PPT01550
   0120               DO 29 K=1,M                                                PPT01560
   0121               A(K,I)=0.0D0                                               PPT01570
   0122               DO 30 J=1,T                                                PPT01580
   0123               A(K,I)=A(K,I)+C(I,J)*Q(K,J)                                PPT01590
   0124            30 CONTINUE                                                   PPT01600
   0125            29 CONTINUE                                                   PPT01610
   0126            28 CONTINUE                                                   PPT01620
   0127               TRACE=0.0D0                                                PPT01630
   0128               DO 80 K=1,T                                                PPT01640
   0129               DO 80 I=1,M                                                PPT01650
   0130               TRACE=TRACE + A(I,K)*S(I,K)                                PPT01660
   0131            80 CONTINUE                                                   PPT01670
   0132            80 CONTINUE                                                   PPT01680
   0133               TRACE=(TRACE/KOUNT)                                        PPT01690
                C WRITE M,Q,T, LOSS VECTOR MATRIX A TO WUNIT.                    PPT01700
                C     ENDFILE WUNIT                                              PPT01710
   0134               REWIND WUNIT                                               PPT01720
   0135                WRITE(NPRT,410)                                           PPT01730
   0136            410  FORMAT(//,15X,'CATEGORY  OCCURANCES')                    PPT01740
   0137               DO 411 I=1,26                                              PPT01750
   0138               IF(CATREC(I).EQ.0) GO TO 411                               PPT01760
   0139               WRITE(NPRT,412) LETTER(I),ISG7(I)                          PPT01770
   0140            412  FORMAT(1H ,15X,A1,10X,I5)                               PPT01780
   0141            411  CONTINUE                                                PPT01790
   0142               WRITE( PRT,413) IUNGT                                      PPT01800
   0143            413  FORMAT(//,10X,I5,'  PIXELS WITHOUT GROUND TRUTH')       PPT01810
   0144               WRITE( PRT,414)  IUNU                                      PPT01820
   0145            414  FORMAT(//,10X,I5,'  UNUSED GROUND TRUTH SYMBOLS')       PPT01830
   0146               WRITE(WUNIT,31) M,Q,T                                      PPT01840
   0147            31 FORMAT(3I3)                                                PPT01850
   0148               WRITE(WUNIT,311) (CATREC(I),I=1,26)                        PPT01860
   0149               WRITE(WUNIT,311) (TERMS(I),I=1,N1)                         PPT01870
   0150               WRITE(WUNIT,32) (((C(I,J),J=1,T),I=1,T)                    PPT01880
   0151               IF(PFLAG.GE.1)  WRITE(NPRT,313)                            PPT01890
   0152            313  FORMAT(//,10X,'THE TERM KEY FOR THE PHI FUNCTION')      PPT01900
   0153               IF(PFLAG.GE.1) WRITE(NPRT,312) (TERMS(I),I=1,N1)           PPT01910
   0154            312  FORMAT(1H ,10X,50I1)                                    PPT01920
   0155            311  FORMAT(50I1)                                            PPT01930
   0156               WRITE(WUNIT,32) (((A(I,J),J=1,T),I=1,M)                    PPT01940
   0157               IF(PFLAG.GE.1) WRITE(NPRT,237)                             PPT01950
   0158            237  FORMAT(//,10X,'NEW PN INVERSE',//)                      PPT01960
                C     NINV=M*(M+1)                                               PPT01970
   0159               NINV=M*(M+1)/2                                             PPT01980
   0160               IF(PFLAG.GE.1) WRITE(NPRT,238) (PNI(I),I=1,NINV)           PPT01990
   0161            238  FORMAT(1H ,5X,3D20.10)                                  PPT02000
   0162               WRITE(NPRT,555)                                            PPT02010
   0163            555  FORMAT(//,10X,'THE LOSS VECTOR MATRIX',/)               PPT02020
   0164               WRITE(NPRT,332)(((A(I,J),J=1,T),I=1,M)                     PPT02030
   0165               32 FORMAT(2020.10)                                         PPT02040
   0166            332 FORMAT(1H ,2020.10)                                       PPT02050
   0167               ENDFILE WUNIT                                              PPT02060
                C TRAINING OVER.                                                 PPT02070
                C CLASSIFICATION PROGRAM WILL READ LOSS VECTOR MATRIX A          PPT02080
                C FROM UNIT WUNIT.                                               PPT02090
   0168               WRITE(NPRT,33) WUNIT                                       PPT02100
   0169            33 FORMAT(/1X,'TRAINING OVER',//,                            PPT02110
                     *1X,'LOSS VECTOR MATRIX RESIDES ON UNIT ',I8/)              PPT02120
   0170               WRITE(NPRT,34) M,Q,T                                       PPT02130
   0171            34 FORMAT(/1X,'M,Q,T',3I8/)                                   PPT02140
   0172               WRITE(NPRT,110) TRACE                                      PPT02150
   0173           110 FORMAT(/1X,'UPPER BOUND ON BAYES RISK',6X,D20.10/)         PPT02160
   0174               RETURN                                                     PPT02170
   0175               END                                                        PPT02180
```

B.3 SOFTWARE PROGRAM NO. 3 (READIT)

Subroutine READIT reads in a vector of data about a pixel, using the input format IFMT, and stores it in the feature vector using the feature index vector.

B.3.1 Linkages

READIT is called by PPTA and PPTC.

B.3.2 Interfaces

READIT interfaces with PPTA and PPTC through a calling sequence and PPTBLK through the common blocks UN, PF, and FV. READIT reads data from unit NRDR2.

B.3.3 Inputs

Calling sequence:

Subr. READIT (LINE, SAMPLE, GTCAT, AICAT, X, ND, ICMT, IAIVEG)

Parameter	Dimension	In/Out	Description
LINE	1	Out	Line number
SAMPLE	1	Out	Sample number
GTCAT	1	Out	Ground Truth
AICAT	1	Out	Category Label A.I. Category Label
Type	1	Out	Type
X	ND	Out	The feature vector.
ND	1	In	The number of channels.
ICMT	1	Out	AI Comment
IAIVEG	4	Out	Vegetation indicators

Common Blocks:

COMMON/FV/FEATVC(30,IFMT(20);NDATA

Parameter	Dimension	In/Out	Description
FEATVC	30	In	The feature index vector
IFMT	2	In	The data input format
NDATA	1	In	Number of data points per pixel

See PPTBLK for information on the other common blocks.

Input cards (unit NRDR2):

Variables	Format	Function
(XX(I),I=1,ND)	IFMT	Input data for a pixel.

B.3.4 Outputs

If PFLAG = 1 then the vector XX is printed out.

B.3.5 Storage

Program size = 1264.

B.3.6 Description

READIT reads in a vector data (length NDATA) about a pixel using the input format IFMT and stores it in the feature vector using the feature index vector as a set of pointers.

B.3.7 Flowchart

N/A

B.3.8 Listing

FILE READIT

```
                C   ADAPTED BY C W AHLERS                                                REA00010
0001                SUBROUTINE READIT(LINE,SAMPLE,GTCAT,AICAT,TYPE,X,ND,ICMNT,IAIVEG)    REA00020
                C THIS SUBROUTINE READS DATA FOR SUPER PAT-PIT-THAD.                     REA00030
0002                DOUBLE PRECISION X(ND),XX(30)                                        REA00040
0003                COMMON /TUN/NTAM                                                     REA00050
0004                DIMENSION IAIVEG(4)                                                  REA00060
0005                INTEGER PUNIT,WUNIT                                                  REA00070
0006                INTEGER SAMPLE,GTCAT,AICAT                                           REA00080
0007                COMMON /UN/NRDR1,NRDR2,NPRT,RUNIT,WUNIT                              REA00090
0008                INTEGER PFLAG                                                        REA00100
0009                COMMON /PF/PFLAG                                                     REA00110
0010                INTEGER TYPE                                                         REA00120
0011                INTEGER FEATVC(30)                                                   REA00130
0012                COMMON /FV/FEATVC,IFMT(30),NDATA                                     REA00140
                C   WRITE(NTAM,999)                                                     REA00150
0013                FORMAT(1H ,'READIT')                                                 REA00160
0014                READ(NRDR2,IFMT) LINE,SAMPLE,GTCAT,AICAT,TYPE,(IAIVEG(I),I=1,4),    REA00170
                   *   ICMNT,(XX(I),I=1,NDATA)                                          REA00180
0015                IF(PFLAG.EQ.1) WRITE(NPRT,3)                                         REA00190
0016              3   FORMAT(1H )                                                       REA00200
0017                IF(PFLAG.EQ.1) WRITE(NPRT,2) (XX(I),I=1,NDATA)                       REA00210
0018              2   FORMAT(1H ,2X,'XX ',15F6.1)                                      REA00220
                C   WRITE(NTAM,777) ND                                                 REA00230
0019              777   FORMAT(1H ,'ND = ',I5)                                         REA00240
0020                IF(ND.LT.1.OR.ND.GT.30) STOP                                         REA00250
0021                DO 1 I=1,ND                                                          REA00260
0022              1   X(I)=XX(FEATVC(I))                                               REA00270
                C   WRITE(NTAM,999)                                                     REA00280
0023                RETURN                                                           REA00290
0024                END                                                              REA00300
```

96

B.4 SOFTWARE COMPONENT NO. 4 (PHI)

Subroutine PHI computes the quadratic function vector.

B.4.1 Linkages

PHI is called by subroutines PPTA and PPTC.

B.4.2 Interfaces

PHI interfaces with other routines through a calling sequence.

B.4.3 Inputs

Calling sequence

Subr. PHI(X,P,D,NP)

Parameter	Dimension	In/Out	Description
X	D	In	The feature vector.
P	NP	Out	The phi function vector.
D	1	In	Number of channels.
NP	1	In	Number of terms in the phi vector (N1).

Common blocks:

COMMON/TRM/TERMS(300)

Parameter	Dimension	In/Out	Description
TERMS	300	In	Indicates terms of the Phi function to be used.

B.4.4 Outputs

N/A

B.4.5 Storage

Program size = 904 bytes

B.4.6 Description

PHI computes the quadratic function vector. This vector consists of squared terms, cross product terms, first order terms, and one.

FILE PHI

```
0001            SUBROUTINE PHI(X,P,D,NP)                                        PHI00010
                C   THIS ROUTINE COMPUTES THE TERMS OF THE QUADRIC XT*A*X + BT*X + C.PHI00020
                C   OR (XI**2, XI*XJ, XI, 1).                                   PHI00030
0002            DOUBLE PRECISION P(NP)                                          PHI00040
0003            COMMON /TUN/NBTM                                                PHI00050
0004            INTEGER D,TERMS                                                 PHI00060
0005            DOUBLE PRECISION X(D)                                           PHI00070
0006            COMMON /TRM/TERMS(361)                                          PHI00080
                C   *PTIF(NBTM,999)                                            PHI00090
0007          999 FORMAT(1H ,'PHI')                                             PHI00100
0008            INDEX=0                                                         PHI00110
0009            L=D*(D+1)/2                                                     PHI00120
0010            DO 10 IJ=1,L                                                    PHI00130
0011            IF(TERMS(IJ).NE.1) GO TO 10                                     PHI00140
0012            CALL ATRM(D,IJ,I,J)                                             PHI00150
0013            INDEX=INDEX+1                                                   PHI00160
0014            P(INDEX)=X(I)*X(J)                                              PHI00170
0015         10 CONTINUE                                                        PHI00180
0016            DO 20 II=1,D                                                    PHI00190
0017            II=IJ+L                                                         PHI00200
0018            IF(TERMS(II).NE.1) GO TO 20                                     PHI00210
0019            INDEX=INDEX+1                                                   PHI00220
0020            P(INDEX)=X(IJ)                                                  PHI00230
0021         20 CONTINUE                                                        PHI00240
0022            LL=L+D+1                                                        PHI00250
0023            IF(TERMS(LL).EQ.1) INDEX=INDEX+1                                PHI00260
0024            IF(TERMS(LL).EQ.1) P(INDEX)=1.000                               PHI00270
0025            RETURN                                                          PHI00280
0026            END                                                             PHI00290
```

95

C-2

B.5 SOFTWARE SUBPROGRAM 5 NO. (NP)

Function NP determines the pointer NP to an upper triangular array.

B.5.1 Linkages

The function NP is called by the subroutine PPTA.

B.5.2 Interfaces

NP interfaces with PPTA through a calling sequence and as a function subprogram.

B.5.3 Inputs

Calling sequence

Function. NP(I,J,M)

Parameter	Dimension	In/Out	Description
I	1	In	First rectangular coordinate
J	1	In	Second rectangular coordinate
M	1	In	The size of the PN matrix is M by M.

B.5.4 Outputs

N/A

B.5.5 Storage

Program size = 514 bytes

B.5.6 Description

Function NP determines the pointer NP (the function value) to an upper triangular array using the rectangular coordinates I and J.

B.5.7 Flowchart

N/A

B.5.8 Listings

FILE NP

```
0001            FUNCTION NP(I,J,M)                                          NP 00010
                C THIS SUBPROGRAM DETERMINES THE POINTER NP                 NP 00020
                C TO AN UPPER TRIANGULAR ARRAY USING RECTANGULAR            NP 00030
                C COORDINATES I,J.                                          NP 00040
                C NP(I,J) IS CALLED BY THE PIT-PAT-THADANI PROGRAM.         NP 00050
0002                II=I                                                    NP 00060
0003                JJ=J                                                    NP 00070
0004                IF(II .GT. JJ) GO TO 1                                  NP 00080
0005              3 NP=(M*(II-1)) - ((((II-1)*(II-2))/2) + (JJ-II+1)        NP 00090
0006                GO TO 2                                                 NP 00100
0007              1 K=II                                                    NP 00110
0008                II=JJ                                                   NP 00120
0009                JJ=K                                                    NP 00130
0010                GO TO 3                                                 NP 00140
0011              2 RETURN                                                  NP 00150
0012                END                                                     NP 00160
```

86

B.6 SOFTWARE COMPONENT NO. 6 (PPTBLK)

PPTBLK is a block data subprogram. It is used to initialize several variables.

B.6.1 Linkages

N/A

B.6.2 Interfaces

PPTBLK interfaces with the subprograms in this system through the common blocks, FV, UN, LT and TUN.

B.6.3 Inputs

N/A

B.6.4 Outputs

N/A

B.6.5 Storage

Storage = 80 bytes.

B.6.6 Description

PPTBLK is a block data subprogram which initializes the common blocks FV, UN, LT and TUN.

Common blocks:

COMMON/FV/FEATVC(30), IFMT(20),NDATA

Parameter	Dimension	Description
FEATVC	30	The feature index vector.
IFMT	20	The input format for the input data (see READIT)

FILE PPTBLK

```
0001                  BLOCK DATA                                                        PPT00010
0002                  INTEGER FEATVC                                                    PPT00020
0003                 INTEGER RUNIT,WUNIT                                                PPT00030
0004                  COMMON /FV/FEATVC(30),IFMT(30),NDATA                              PPT00040
0005                  COMMON /LT/LETTER(26)                                             PPT00050
0006                 DATA LETTER/'A','B','C','D','E','F','G','H','I','J','K','L','M',  PPT00060
                    *  'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/           PPT00070
0007                 DATA NDATA/25/                                                     PPT00080
0008                 DATA FEATVC/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,      PPT00090
                    *  20,21,22,23,24,25,26,27,28,29,30/                               PPT00100
0009                  DATA IFMT/'(2I2','.','2A1','.','I1','.','4I1  ','.','A4','.','4(/','.',' BF9','.',4))'PPT00110
                    *  ,17*'    '/                                                     PPT00120
0010                 COMMON /UN/NPDR1,NPDR2,NPRT,RUNIT,WUNIT                           PPT00130
0011                 DATA NPDR1/6/,NPDR2/15/,NPRT/6/                                    PPT00140
0012                  COMMON /GTBK/NR1,NP1,NP2,IV(6),N11,N12,LACPP,NWRF,NOFP            PPT00150
0013                   DATA LACPP/27/                                                   PPT00160
0014                 COMMON /CK/ICK,JCK                                                PPT00170
0015                 DATA ICK/0/,JCK/0/                                                PPT00180
0016                 COMMON /LU/LACGT                                                  PPT00190
0017                 DATA LACGT/23/                                                    PPT00200
0018                 DATA RUNIT/3/,WUNIT/7/                                            PPT00210
0019                 COMMON /TUN/NTRM                                                  PPT00220
0020                 DATA NTRM/3/                                                      PPT00230
0021                 COMMON /PPPTU/ITU                                                 PPT00240
0022                 DATA ITU/12/                                                      PPT00250
0023                 END                                                               PPT00260
```

Parameter	Dimension	Description
NDATA	1	The number of data points per pixel.

The common block UN stores some of the various unit numbers as follows:

NRDL - Card reader for the setup cards or the terminal.

NRDR2 - Card reader for the pixel data.

NPRT - Line printer (or output) unit number.

RUNIT - Utility data set unit number.

WUNIT - Utility data set unit number. (The loss vector is written to this unit)

```
COMMON/LT/LETTER(26)
```

Parameter	Dimension	Description
LETTER	26	The letters of the alphabet.

The common block TUN stores only the terminal output unit number.

B.6.7 Flowchart

N/A

B.6.8 Listings

B.7 SOFTWARE SUBPROGRAM NO. 7 (SETTRM)

Subroutine SETTRM records the terms of the Phi function to be used.

B.7.1 Linkages

SETTRM is called by SPPTA and calls the LARSYS functions NXTCHR, FIND12, and NUMBER.

B.7.2 Interfaces

SETTRM interfaces with SPPTA through the callings sequence and common blocks TRM and UN.

B.7.3 Inputs

Calling sequence: Subr. SETTRM(D)

Parameter	Dimension	In/Out	Description
D	1	In	No. of channels

Input cards (unit NRDR1):

Keyword	Parameter & default values	Function
ALL	-	All term of the phi function are used.
DEFAULT	-	Squared, linear, and the constant terms are used.
LINEAR	ALL	All linear terms (including the constant) are used.

or

Keyword	Parameter & default values	Function
LINEAR	L = N_1, N_2, N_3....	N_i's specify the linear terms used.
SQUARE	ALL	Squared terms are used.
or		
INTERACT	ALL	All interaction terms are used.
or		
INTERACT	I = N_1, N_2	The interaction term between the N_1 term & the N_2 term is used.
CONSTANT	YES	Constant term is used.
or		
CONSTANT	NO	Constant term is not used.
END	—	End of the input information about the Phi function.

B.7.4 Outputs

Common block:

COMMON/TRM/TERMS(300)

Parameter	Dimension	In/Out	Description
TERMS	300	Out	Indicates terms of the Phi function to be used.

B.7.5 Storage

Program Size = 3506

B.7.6 Description

Subroutine SETTRM reads LARSYS type input cards which specify which terms of the phi function are to be used. This information is stored in the array TERMS to be passed back to other subroutines.

B.7.7 Flowchart

N/A

B.7.8 Listing

FILE SETTRM

```
                C       WRITTEN BY C W AHLERS                                   SET00010
                C     SETTRM READS CARDS TO PRODUCE A VECTOR INDICATING THE     SET00020
                C     TERMS IN THE PHI FUNCTION THAT ARE BEING USED             SET00030
0001                  SUBROUTINE SETTRM(D)                                      SET00040
0002                  IMPLICIT INTEGER (A-Z)                                    SET00050
0003                  DIMENSION CODE(10),CARD(62),FOUCOM(3),ACARD(20)           SET00060
0004                  COMMON /IUN/NTRM                                          SET00070
0005                  DIMENSION WORK(30)                                        SET00080
0006                  DIMENSION SLASH(2)                                        SET00090
0007                  DATA SLASH /1H*,1H/                                       SET00100
0008                  DATA CODE/4HLINE,4HSQUA,4HINTE,
                     * 4HTERM,4HALL ,4H,4HDEFA,4H*END/                          SET00120
0009                  DATA  FOUCOM/2H,,2H=,2H*,2H/                              SET00130
0010                  DATA Y/1HY/,BLNK/1H /,LL/1HL/,A/1HA/,S/1HS/,NN/1HN/       SET00140
0011                  DATA II/1HI/                                              SET00150
0012                  COMMON /TRM/TERMS(361)                                    SET00160
0013                  COMMON /UN/NPRT,NRDR,NPRT,RUNIT,WUNIT                     SET00170
                C       WRITE(NTRM,999)                                         SET00180
0014             999    FORMAT(1H ,4HSETTRM)                                    SET00190
0015                  NOLAB = 0                                                 SET00200
0016                  NPUT=7                                                    SET00210
0017                  L=D*(D+1)/2                                               SET00220
0018                  N1=L+D+1                                                  SET00230
                C                                                               SET00240
0019                  WRITE(NPRT,100)                                           SET00250
0020             100  FORMAT(/11X,4HTERMS FOR THE PHI FUNCTION*/)              SET00260
                C                                                               SET00270
                C     SET UP REREAD BUFFER                                      SET00280
                C                                                               SET00290
0021                  RRUNIT = 30                                               SET00300
0022                  CALL REREAD(RRUNIT,90)                                    SET00310
                C                                                               SET00320
                C     PUT CARD IN BUFFER                                        SET00330
                C                                                               SET00340
0023             105  READ(NRDR,103)(ACARD(I),I=1,20)                          SET00350
0024             103  FORMAT(20A4)                                             SET00360
0025                  WRITE(RRUNIT,103)(ACARD(I),I=1,20)                        SET00370
0026                  REWIND RRUNIT                                             SET00380
                C                                                               SET00390
0027                  READ(RRUNIT,110)CODE1,CARD                                SET00400
0028                  REWIND RRUNIT                                             SET00410
0029                  COL = 0                                                   SET00420
0030                  WRITE(NPRT,120)CODE1,CARD                                 SET00430
0031             120  FORMAT(1X,A4,6X,62A1)                                     SET00440
0032             110  FORMAT(A4,6X,62A1)                                        SET00450
0033                  DO 130  I=1,NPUT                                          SET00460
0034                  IF (CODE1.EQ.CODE(I)) GO TO (150,180,210,330,370,         SET00470
                     *                             390,420),I                 SET00480
0035             130     CONTINUE                                               SET00490
0036             135  WRITE(NPRT,140)                                           SET00500
0037             140  FORMAT( * INVALID CONTROL CARD - IGNORED *)               SET00510
0038                  GO TO 105                                                 SET00520
                C                                                               SET00530
                C       LINEAR CARD                                             SET00540
                C                                                               SET00550
0039             150  M = NXTCHR(CARD,COL)                                      SET00560
0040                  NOLAB=0                                                   SET00570
0041                  IF (M.EQ.LL) GO TO 155                                    SET00580
0042                  IF (M.EQ.A) GO TO 156                                     SET00590
                C     IF (M.EQ.BLNK) GO TO 105                                  SET00600
0043             152  WRITE(NPRT,153)                                           SET00610
0044             153  FORMAT(* ERROR ON LINEAR CARD*)                          SET00620
0045                  GO TO 105                                                 SET00630
0046              156    CONTINUE                                               SET00640
0047                  I1=L+1                                                    SET00650
0048                  I2=N1                                                     SET00660
0049                   DO 157 I=I1,I2                                           SET00670
0050                  TERMS(I)=1                                                SET00680
0051             157   CONTINUE                                                 SET00690
0052                  GO TO 105                                                 SET00700
0053             155  J = FIND12(CARD,COL,FOUCOM)                               SET00710
0054                  IF (J .NE. 2) GO TO 152                                   SET00720
0055                  NOLAB = NUMBER(CARD,COL,WORK,NOLAB)                       SET00730
0056                  DO 158 I=1,NOLAB                                          SET00740
0057                  ITRM=WORK(I)+L                                            SET00750
0058                  TERMS(ITRM)=1                                             SET00760
```

FILE SFTTRM

```
0059          158   CONTINUE                                                SFT00770
0060                COL = COL - 1                                           SFT00780
0061                GO TO 105                                               SFT00790
                C                                                           SFT00800
                CC      SQUARED CARD                                        SFT00810
                C                                                           SFT00820
0062          180 M = NXTCHR(CARD,COL)                                      SFT00830
0063                NOLAR=0                                                 SFT00840
0064                IF (M .EQ. BLNK ) GO TO 105                             SFT00850
0065                IF (M.EQ.A) GO TO 190                                   SFT00860
0066                IF (M.EQ.S) GO TO 200                                   SFT00870
0067          185 WRITE(NPRT,187)                                           SFT00880
0068          187 FORMAT(' ERROR ON SQUARED CARD')                          SFT00890
0069                GO TO 105                                               SFT00900
0070          190   CONTINUE                                                SFT00910
0071                DO 191 I=1,D                                            SFT00920
0072                ITRM=NP(I,I,D)                                          SFT00930
0073                TERMS(ITRM)=1                                           SFT00940
0074          191   CONTINUE                                                SFT00950
0075                COL = COL - 1                                           SFT00960
0076                GO TO 105                                               SFT00970
0077          200 J = FIND12(CARD,COL,EQUCOM)                               SFT00980
0078                IF ( J .NE. 2 ) GO TO 185                               SFT00990
0079                NOLAR = NUMBER(CARD,COL,WORK,NOLAR)                     SFT01000
0080                DO 201 I=1,NOLAR                                        SFT01010
0081                II=WORK(I)                                              SFT01020
0082                ITRM=NP(II,II,D)                                        SFT01030
0083                TERMS(ITRM)=1                                           SFT01040
0084          201   CONTINUE                                                SFT01050
0085                COL = COL - 1                                           SFT01060
0086                GO TO 190                                               SFT01070
                C                                                           SFT01080
                CC      INTERACTION CARD                                    SFT01090
                C                                                           SFT01100
0087          210   CONTINUE                                                SFT01110
0088          214 M = NXTCHR(CARD,COL)                                      SFT01120
0089                NOLAR=0                                                 SFT01130
0090                IF (M .EQ. BLNK ) GO TO 105                             SFT01140
0091                IF (M.EQ.A) GO TO 230                                   SFT01150
0092                IF (M.EQ.I) GO TO 240                                   SFT01160
0093          215 WRITE(NPRT,220)                                           SFT01170
0094          220 FORMAT(' ERROR ON INTERACTION CARD')                      SFT01180
0095                GO TO 105                                               SFT01190
0096          230   CONTINUE                                                SFT01200
0097                DO 231 I=1,I                                            SFT01210
0098                CALL NTRM(D,I,I,I2)                                     SFT01220
0099                IF(I1.EQ.I2) GO TO 231                                  SFT01230
0100                TERMS(I)=1                                              SFT01240
0101          231   CONTINUE                                                SFT01250
0102                GO TO 214                                               SFT01260
0103          240 J = FIND12(CARD,COL,EQUCOM)                               SFT01270
0104                IF (J .NE. 2) GO TO 215                                 SFT01280
0105                NOLAR = NUMBER(CARD,COL,WORK,NOLAR)                     SFT01290
0106                IF(NOLAR.GT.2) GO TO 215                                SFT01300
0107                I1=WORK(1)                                              SFT01310
0108                I2=WORK(2)                                              SFT01320
0109                IF(I1.LE.I2) GO TO 241                                  SFT01330
0110                ITEMP=I1                                                SFT01340
0111                I1=I2                                                   SFT01350
0112                I2=ITEMP                                                SFT01360
0113          241   CONTINUE                                                SFT01370
0114                ITRM=NP(I1,I2,D)                                        SFT01380
0115                TERMS(ITRM)=1                                           SFT01390
0116                COL = COL - 1                                           SFT01400
0117                GO TO 214                                               SFT01410
                C                                                           SFT01420
                CC      CONSTANT CARD                                       SFT01430
                C                                                           SFT01440
0118          330 M = NXTCHR(CARD,COL)                                      SFT01450
0119                IF (M .EQ. BLNK ) GO TO 105                             SFT01460
0120                IF (M.EQ.NN) GO TO 341                                  SFT01470
0121                IF (M.EQ.Y) GO TO 340                                   SFT01480
0122          333 WRITE(NPRT,335)                                           SFT01490
0123          335 FORMAT(' ERROR ON CONSTANT CARD')                         SFT01500
0124                GO TO 105                                               SFT01510
0125          340 ITRM = L+D+1                                              SFT01520
```

FILE SETTRM

```
0126                 TERMS(ITRM)=1                                          SET01530
0127                 GO TO 105                                              SET01540
0128           341   ITRM=L+D+1                                             SET01550
0129                 TERMS(ITRM)=0                                          SET01560
0130                 GO TO 105                                              SET01570
             C                                                              SET01580
             C       ALL CARD                                               SET01590
             C                                                              SET01600
0131           370    CONTINUE                                              SET01610
0132                  DO 371 I=1,N1                                         SET01620
0133           371    TERMS(I)=1                                            SET01630
0134                 GO TO 105                                              SET01640
             C                                                              SET01650
             C       DEFAULT CARD                                           SET01660
             C                                                              SET01670
0135           380    CONTINUE                                              SET01680
0136                 DO 381 I=1,D                                           SET01690
0137                 ITRM=NP(I,I,D)                                         SET01700
0138                 TERMS(ITRM)=1                                          SET01710
0139           381   CONTINUE                                               SET01720
0140                 I1=L+1                                                 SET01730
0141                 I2=L1                                                  SET01740
0142                 DO 392 I=I1,I2                                         SET01750
0143           392    TERMS(I)=1                                            SET01760
0144                 GO TO 105                                              SET01770
             C                                                              SET01780
             C                                                              SET01790
             C       *END*                                                  SET01800
             C                                                              SET01810
0145           420   CONTINUE                                               SET01820
0146                 RETURN                                                 SET01830
0147                 END                                                    SET01840
```

B.8 SOFTWARE COMPONENT NO. 8 (NTRM)

Subroutine NTRM computes rectangular coordinates for a location in an upper triangular matrix.

B.8.1 Linkages

Subroutine NTRM is called by SETTRM.

B.8.2 Interfaces

NTRM interfaces with SETTRM through a calling sequence.

B.8.3 Inputs

Calling sequence: Subr. NTRM (D, IJ, I, J)

Parameter	Dimension	In/Out	Description
D	1	In	Dimension of the matrix
IJ	1	In	Upper triangular pointer.
I	1	Out	First rectangular coordinate.
J	1	Out	Second rectangular coordinate.

B.8.4 Outputs

N/A

B.8.5 Storage

Program Size = 544

B.8.6 Description

NTRM computes rectangular coordinates from a pointer indicating a location in an upper triangular matrix.

B.8.7 Flowchart

N/A

B.8.8 Listing

FILE NTRM

```
                C       WRITTEN BY C AHLERS                                                     NTR00010
                C       THIS SUBROUTINE COMPUTES RECTANGULAR COORDINATES FOR A LOCATION         NTR00020
                C       IN AN UPPER TRIANGULAR MATRIX.                                          NTR00030
0001                    SUBROUTINE NTRM(D,IJ,I,J)                                               NTR00040
0002                   INTEGER D                                                                NTR00050
0003                    I =0                                                                    NTR00060
0004                   L1=1                                                                     NTR00070
0005                   L2=D                                                                     NTR00080
0006                   DO 10 K=1,D                                                              NTR00090
0007                    I =I+1                                                                  NTR00100
0008                   IF(IJ.GE.L1.AND.IJ.LE.L2) GO TO 20                                       NTR00110
0009                   L1=L1+(D-K)+1                                                            NTR00120
0010                   L2=L2+(D-K)                                                              NTR00130
0011             10    CONTINUE                                                                 NTR00140
0012             20    CONTINUE                                                                 NTR00150
0013                   J=I+IJ-L1                                                                NTR00160
0014                   RETURN                                                                   NTR00170
0015                   END                                                                      NTR00180
```

B.9 SOFTWARE COMPONENT NO. 9 (ALPHA)

The function ALPHA returns the number indicating the place in the alphabet that cooresponds to a certain letter.

B.9.1 Linkages

ALPHA is called by subroutines PPTA and SPPTA.

B.9.2 Interfaces

ALPHA interfaces with other routines through a calling sequence.

B.9.3 Inputs

Calling sequence: Function ALPHA(S)

Parameter	Dimension	In/Out	Description
S	1	In	A letter of the alphabet.

B.9.4 Outputs

N/A

B.9.5 Storage

Program Size = 596

B.9.6 Description

For an input letter ALPHA returns the number that indicates its order of occurance in the alphabet.

B.9.7 Flowchart

N/A

B.9.8 Listing

· FILE ALPHA

```
C        WRITTEN BY C W AHLERS
C         ALPHA RETURNS AN ORDER NUMBER FOR A GIVEN LETTER OF THE ALPHABET
          FUNCTION ALPHA(S)
          IMPLICIT INTEGER (A-Z)
          DIMENSION A(26)
        DATA A/'A','B','C','D','E','F','G','H','I','J','K','L','M',
     *   'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
           DATA IBLK/' '/
         DO 10 I=1,26
         ALPHA=I
         IF(S.EQ.A(I)) RETURN
   10     CONTINUE
         I=26
         IF(S.EQ.IBLK) RETURN
          WRITE(6,20) S
   20     FORMAT(1H ,5X,'THE SYMBOL ',A1,' CAN NOT BE USED.')
         RETURN
         END
```

112

APPENDIX C

C.1 SOFTWARE COMPONENT NO. 1 (SPPTC)

Subroutine SPPTC reads the input cards and sets option switches.

C.1.1 Linkages

SPPTC is called by the program MPPTC and uses data initialized in PPTBLK.

C.1.2 Interfaces

SPPTC interfaces with MPPTA through a calling sequence and interfaces with MPPTC and PPTBLK through blocks UN, PF, and FV.

C.1.3 Inputs

Calling sequence:

Subr. SPPTC (UNIT, M, D, T, ISGZ, NT, CATREC, NP)

Parameter	Dimension	In/Out	Description
UNIT	1	Out	Unit number for the loss vector matrix data set
M	1	Out	First dimension of the loss vector matrix
D	1	Out	Number of channels
T	1	Out	Number of classes
ISGZ	10	Out	Number of classified into the different classes
NT	1	Out	Total number of pixels
CATREC	26	Out	Array recording occurance of categories
N1	1	Out	Array size used in PPTC
NP	1	Out	Same as NT

Common blocks:

See PPTBLK for information about the common blocks.

Input cards (unit NRDR1);

	Variables	Format	Function
1.	PFLAG	I5	0- for short printout 1- for long printout
2.	IDEF	A1	D- use default data vector input format N- input an input format
3.	(use if IDEF=N) NDATA	I5	Number of data points per pixel
4.	(use if IDEF=N) (IFMT(I), I=1,20)	20A4	Input format
5.	IDEF	A1	D- use default feature index vector
6.	(use if IDEF=N) (FEATVC(I),I=1,D)	30I2	The feature index vector

C.1.4 Outputs

Input information is printed out.

C.1.5 Storage

Program size = 3700

C.1.6 Description

SPPTC is the input subroutine for all except the pixel data. If default options are not used this subroutine inputs the format for the pixel data and the feature index vector.

C.1.7 Flowchart

N/A

C.1.8 Listing

```
FILE SPPTC
```

```
                C   ADAPTED BY C W AHLERS                                                SPP00010
0001                SUBROUTINE SPPTC(UNIT,M,D,T,ISGZ,NT,CATREC,N1,NP)                    SPP00020
0002                INTEGER D,T,CAT,UNIT                                                 SPP00030
0003                INTEGER CATREC(26)                                                   SPP00040
0004                INTEGER TERMS                                                        SPP00050
0005                COMMON /TRM/TERMS(361)                                               SPP00060
0006                COMMON /LT/LETTER(26)                                                SPP00070
0007                DIMENSION ISGZ(10)                                                   SPP00080
0008                INTEGER RUNIT,WUNIT                                                  SPP00090
0009                COMMON /UN/NPOR1,NPOR2,NPRT,RUNIT,WUNIT                              SPP00100
0010                INTEGER PFLAG                                                        SPP00110
0011                COMMON /PF/PFLAG                                                     SPP00120
0012                INTEGER FFATVC,TT,DD                                                 SPP00130
0013                COMMON /FV/FFATVC(30),IFMT(30),NDATA                                 SPP00140
                C   PARAMETER NN1=361,TT=10,DD=30                                       SPP00150
                C DIMENSIONS CHECKED IN THIS SUBROUTINE                                  SPP00160
0014                DATA NN1/361/,TT/10/,DD/30/                                          SPP00170
0015                DATA IY/'Y'/                                                         SPP00180
0016                DOUBLE PRECISION C(10,10)                                            SPP00190
                C                                                                        SPP00200
0017                REWIND NPOR1                                                         SPP00210
0018                REWIND NPOR2                                                         SPP00220
0019                DO 349 I=1,26                                                        SPP00230
0020           349  ISGZ(I)=0                                                           SPP00240
0021                NT=0                                                                 SPP00250
                C READ INPUT DATA.                                                       SPP00260
0022                READ(NPOR1,10) PFLAG                                                 SPP00270
0023                UNIT=WUNIT                                                           SPP00280
0024            10   FORMAT(2I5)                                                        SPP00290
0025                 WRITE(NPRT,20) UNIT                                                 SPP00300
0026            20   FORMAT(//,10X,'LOSS VECTOR MATRIX UNIT NO. = ',I5)                 SPP00310
0027                READ(UNIT,21) M,D,T                                                 SPP00320
0028            21   FORMAT(3I5)                                                        SPP00330
0029                READ(UNIT,375) M,D,T                                                SPP00340
0030            375  FORMAT(3I3)                                                        SPP00350
0031                WRITE(NPRT,376) M                                                   SPP00360
0032            376  FORMAT(//,10X,'M = ',I5)                                           SPP00370
0033                 WRITE(NPRT,22) D,T                                                  SPP00380
0034            22   FORMAT(//,10X,'NO. OF CHANNELS=',I5,//,10X,'NO. OF CLASSES=',      SPP00390
                    *  I5)                                                              SPP00400
0035                 IF(D.GT.DD) WRITE(NPRT,101) DD                                      SPP00410
0036                 IF(T.GT.TT) WRITE(NPRT,102) TT                                      SPP00420
0037            101   FORMAT(//,10X,' ERROR -- THE NO. OF CHANNELS EXCEEDS ',I5)        SPP00430
0038            102   FORMAT(//,10X,' ERROR -- THE NO. OF CLASSES EXCEEDS ',I5)         SPP00440
0039                IF(D.GT.DD) STOP                                                    SPP00450
0040                IF(T.GT.TT) STOP                                                    SPP00460
0041                NP=NT                                                               SPP00470
0042                READ(UNIT,311) (CATREC(I),I=1,26)                                   SPP00480
0043            311   FORMAT(80I1)                                                      SPP00490
0044                WRITE(NPRT,612)                                                     SPP00500
0045            612   FORMAT(//,15X,'CATEGORY',5X,'INDEX')                            SPP00510
0046                DO 610 I =1,26                                                      SPP00520
0047                 IF(CATREC(I).EQ.0) GO TO 610                                       SPP00530
0048                WRITE(NPRT,611) LETTER(I),CATREC(I)                                 SPP00540
0049            611   FORMAT(1H ,15X,A1,10X,I3)                                        SPP00550
0050            610  CONTINUE                                                          SPP00560
0051                N1=0.1+(0.*(0.+1.))/2                                               SPP00570
0052                READ(UNIT,311) (TERMS(I),I=1,N1)                                    SPP00580
0053                WRITE(NPRT,313)                                                     SPP00590
0054            313   FORMAT(//,10X,'THE TERM KEY FOR THE PHI FUNCTION')               SPP00600
0055                 WRITE(NPRT,312) (TERMS(I),I=1,N1)                                  SPP00610
0056            312   FORMAT(1H ,10X,50I1)                                             SPP00620
0057                READ(UNIT,32) ((C(I,J),J=1,T),I=1,T)                                SPP00630
0058             32   FORMAT(2020.10)                                                  SPP00640
0059                WRITE(NPRT,35)                                                      SPP00650
0060             35   FORMAT(//,10X,'THE COST MATRIX')                                 SPP00660
0061                 DO 150 I=1,T                                                       SPP00670
0062                WRITE(NPRT,25) (C(I,J),J=1,T)                                       SPP00680
0063            150  CONTINUE                                                          SPP00690
0064             25   FORMAT(//,10X,10F5.2)                                            SPP00700
0065                READ(NPOR1,506) IDEF                                                SPP00710
0066            506  FORMAT(A1)                                                        SPP00720
0067                 IF(IDEF.EQ.IY) GO TO 507                                           SPP00730
0068                READ(NPOR1,10) NDATA                                                SPP00740
0069            444   FORMAT(//,10X,'NDATA=',I5)                                       SPP00750
                C   READ IN THE FORMAT FOR THE FEATURE VECTOR                          SPP00760
```

C.1

ORIGINAL PAGE IS OF POOR QUALITY

```
FILE SPPTC

0070            READ(NPDR1,450) IFMT                                          SPP00770
0071          450 FORMAT(30A4)                                                SPP00780
0072          507  CONTINUE                                                   SPP00790
0073            WRITE(NPRT,444) NDATA                                         SPP00800
0074            WRITE(NPRT,451)                                               SPP00810
0075            WRITE(NPRT,454) IFMT                                          SPP00820
0076          451 FORMAT(//,10X,'FEATURE VECTOR FORMAT  ')                    SPP00830
0077          454  FORMAT(//,10X,20A4)                                       SPP00840
0078            READ(NPDR1,506) IDEF                                          SPP00850
0079            IF(IDEF.EQ.IY) GO TO 509                                      SPP00860
              C  READ IN THE FEATURE  INDEX VECTOR                          SPP00870
0080            READ(NPDR1,452) (FEATVC(I),I=1,0)                             SPP00880
0081          452 FORMAT(3012)                                                SPP00890
0082          509  CONTINUE                                                   SPP00900
0083            WRITE(NPRT,453) (FEATVC(I),I=1,0)                             SPP00910
0084          453  FORMAT(//,10X,'THE FEATURE INDEX VECTOR ',//,10X,3013)    SPP00920
0085            DO 300 I=1,0                                                  SPP00930
0086            IF(FEATVC(I).LE.0) WRITE(NPRT,301)                            SPP00940
0087            IF(FEATVC(I).LE.0) STOP                                       SPP00950
0088          300  CONTINUE                                                   SPP00960
0089          301  FORMAT(//,10X,'ERROR -- FEATURE VECTOR FORMAT - OR -',    SPP00970
             &' FEATURE INDEX VECTOR IS WRONG')                               SPP00980
0090            WRITE(NPRT,33) N1                                             SPP00990
0091          33   FORMAT(//,10X,'N1=',I5)                                    SPP01000
0092            IF(N1.GT.NN1) WRITE(NPRT,103) NN1                             SPP01010
0093            IF(N1.GT.NN1) STOP                                            SPP01020
0094          103  FORMAT(//,10X,' ERROR -- N1 EXCEEDS ',I10)                SPP01030
0095            REWIND NPDR2                                                  SPP01040
0096            RETURN                                                        SPP01050
0097            END                                                           SPP01060
```

118

C.2 SOFTWARE COMPONENT NO. 2 (PPTC)

Subroutine PPTC is the main computational subroutine of the processor MPPTC. Input from SPPTC is passed to PPTC. PPTC with the aid of other subroutines calculates the classification losses to find the minimum loss.

C.2.1 Linkages

Subroutine PPTC is called by MPPTC and is passed information from SPPTC. PPTC calls subroutines READIT, PHI, PPTSUM, PROBSUM, PPTDTL, PPTTAB, IBETA, FSFMFL, GTDWR.

C.2.2 Interfaces

PPTC interfaces with other routines through a calling sequence and common blocks UN and PF.

C.2.3 Inputs

Calling sequence:

Subr. PPTC(M, D, T, ISGZ, NT, UNIT, N1, A, L, P, CATREC, X, NP)

Parameter	Dimension	In/Out	Description
M	1	In	First dimension of the loss vector matrix
D	1	In	Number of channels
T	1	In	Number of classes
ISGZ	10	In	Number of pixels classified into the different classes.
NT	1	In	Total number of pixels
UNIT	1	In	Unit number for the loss vector matrix data set
N1	1	In	Array size for A and P
A	(N1,T)	In	The loss vector matrix.

Parameter	Dimension	In/Out	Description
L	T	-	The losses for each class
P	N1	-	The phi function vector
CATREC	26	In	Array recording occurance categories
X	D	-	The feature vector
NP	1	In	Same as NT

Common blocks:

See PPTBLK for information about the common blocks.

C.2.4 Outputs

Classification information is printed out.

C.2.5 Storage

Program size = 13624

C.2.6 Description

PPTC takes the interproduct of a loss vector and a phi vector to determine a class loss for a particular feature vector. The minimum of these is used as the classification for a particular set of input data.

C.2.7 Flowchart

N/A

C.2.8 Listing

Parameter	Dimension	In/Out	Description
L	T	-	The losses for each class
P	N1	-	The phi function vector
CATREC	26	In	Array recording occurance categories
X	D	-	The feature vector
NP	1	In	Same as NT

Common blocks:

See PPTBLK for information about the common blocks.

C.2.4 Outputs

Classification information is printed out.

C.2.5 Storage

Program size = 13624

C.2.6 Description

PPTC takes the interproduct of a loss vector and a phi vector to determine a class loss for a particular feature vector. The minimum of these is used as the classification for a particular set of input data.

C.2.7 Flowchart

N/A

C.2.8 Listing

FILE PPTC

```
                C       THE PAT-PIT-THADANI CLASSIFIER.                                          PPT00010
                C       SUBROUTINES REQUIRED...PHI1,READIT.                                      PPT00020
                C       THIS PROGRAM CLASSIFIES THE TEST PATTERN X.                              PPT00030
                C       THE LOSS VECTOR MATRIX IS READ OF A USER SPECIFIED UNIT.                 PPT00040
     0001               SUBROUTINE PPTC(M,D,T,ISGZ,NT,UNIT,N1,A,L,P,CATREC,X,NP1)                PPT00050
                C       PARAMETER N1=361                                                         PPT00060
     0002               INTEGER CAT,D,T,UNIT                                                     PPT00070
     0003               INTEGER DOTLAB(11,19,3)                                                  PPT00080
     0004               INTEGER TYPE                                                             PPT00090
     0005               INTEGER TRNS1,TRNS2,TRNS3                                                PPT00100
     0006               COMMON /TR/TRNS1(256),TRNS2(26),TRNS3(26),MASK(11,19)                    PPT00110
     0007               COMMON /GTRK/NP1,NPRT,TVLR(6),N11,N12,NWRU,NWRF,NOF                      PPT00120
     0008               INTEGER TP1                                                              PPT00130
     0009               DATA IBLNK/'    '/                                                       PPT00140
     0010               DIMENSION THEAD(20)                                                      PPT00150
     0011               DIMENSION IAIVEC(4)                                                      PPT00160
     0012               INTEGER CATREC(26),ISGZ(10)                                              PPT00170
     0013               INTEGER RUNIT,WUNIT                                                      PPT00180
                C       PARAMETER TT=10                                                          PPT00190
     0014               DOUBLE PRECISION A(N1,T),L(T),P(N1),LMIN                                 PPT00200
     0015               DOUBLE PRECISION PROBBP(11,11),PROBAP(11,11)                             PPT00210
     0016               DOUBLE PRECISION PROBBA(11,11)                                           PPT00220
     0017               COMMON /LT/LETTER(26)                                                    PPT00230
     0018               INTEGER SAMPLE,CTCAT,AICAT,BLANK                                         PPT00240
     0019               DATA BLANK/' '/                                                          PPT00250
     0020               INTEGER ALPHA                                                            PPT00260
     0021               DIMENSION IDOT(11,19)                                                    PPT00270
     0022               DOUBLE PRECISION X(D)                                                    PPT00280
     0023               COMMON /UN/NWORK,NBORP,NPRT,RUNIT,WUNIT                                  PPT00290
     0024               INTEGER PFLAG                                                            PPT00300
     0025               COMMON /PF/PFLAG                                                         PPT00310
                C                                                                               PPT00320
                C                                                                               PPT00330
                C       READ LOSS VECTORS OF APPROPRIATE UNIT.                                   PPT00340
                C       COMPUTE CLASS LOSSES L(J) = A(J)*P.                                      PPT00350
                C       ASSIGN X TO CLASS WITH MINIMUM LOSS.                                     PPT00360
                C   READ LOSS VECTOR MATRIX                                                     PPT00370
     0026               WRITE(NPRT,777) PFLAG                                                    PPT00380
     0027         777    FORMAT(//,10X,'PFLAG=',I5)                                            PPT00390
     0028               READ(UNIT,105) ((A(I,J),J=1,T),I=1,M)                                    PPT00400
     0029         105 FORMAT(2020.10)                                                            PPT00410
     0030               WRITE(NPRT,700) ((A(I,J),J=1,T),I=1,M)                                   PPT00420
     0031         700 FORMAT(/5X,'LOSS VECTOR MATRIX '/,(2020.10))                               PPT00430
                C                                                                               PPT00440
                C                                                                               PPT00450
     0032               NWRF=0                                                                   PPT00460
     0033               NOF=30                                                                   PPT00470
     0034           999   CONTINUE                                                             PPT00480
     0035               NWRU=27                                                                  PPT00490
     0036               IGRFLG=0                                                                 PPT00500
     0037               IARFLG=0                                                                 PPT00510
     0038               IGAFLG=0                                                                 PPT00520
     0039               IUNU=0                                                                   PPT00530
     0040               IUNGT=0                                                                  PPT00540
     0041               NZTYP=0                                                                  PPT00550
     0042               NP1=NPRT                                                                 PPT00560
     0043               NWRF=NWRF+1                                                              PPT00570
     0044               WRITE(NPRT,855)                                                          PPT00580
     0045         855      FORMAT(1H1)                                                          PPT00590
     0046               DO 856 I=1,10                                                            PPT00600
     0047         856     ISGZ(I)=0                                                             PPT00610
     0048               DO 850 I=1,11                                                            PPT00620
     0049               DO 850 J=1,19                                                            PPT00630
     0050               DO 851 K=1,3                                                             PPT00640
     0051               DOTLAB(I,J,K)=BLANK                                                      PPT00650
     0052         851    CONTINUE                                                              PPT00660
     0053               MASK(I,J)=0                                                              PPT00670
     0054         850   CONTINUE                                                               PPT00680
     0055               TP1=T+1                                                                  PPT00690
     0056               DO 121 I=1,TP1                                                           PPT00700
     0057               DO 121 J=1,TP1                                                           PPT00710
     0058               PROBAP(I,J)=0.0D0                                                        PPT00720
     0059               PROBBA(I,J)=0.0D0                                                        PPT00730
     0060         121   PROBBP(I,J)=0.0D0                                                      PPT00740
     0061               IF(PFLAG.GE.1) WRITE(NPRT,106)                                           PPT00750
     0062         106    FORMAT(//,10X,'THE INPUT DATA AND CLASSIFICATION RESULTS',//)          PPT00760
```

FILE PPTC

```
0063          KOUNT=0                                                          PPT00770
          C     BEGIN LOOP                                                       PPT00780
0064       707  CONTINUE                                                         PPT00790
0065           DO 464 I=1,2                                                     PPT00800
0066           READ(NPDR2,462) IHEAD                                            PPT00810
0067           IF(IHEAD(1).NE.IBLNK) ISITE=IHEAD(20)                            PPT00820
0068           IF(IHEAD(1).EQ.IBLNK) RETURN                                     PPT00830
0069       462  FORMAT(20A4)                                                     PPT00840
0070           IF(I.EQ.1) WRITE(NPRT,454)                                       PPT00850
0071       454   FORMAT(//,10X,'HEADER RECORDS')                               PPT00860
0072           WRITE(NPRT,463) IHEAD                                            PPT00870
0073       463  FORMAT(//,10X,20A4)                                              PPT00880
0074       464  CONTINUE                                                         PPT00890
0075         1    CONTINUE                                                       PPT00900
0076           CALL READIT(LINE,SAMPLE,GTCAT,AICAT,TYPE,X,D,ICMNT,IAIVEG)       PPT00910
0077           IF(LINE.EQ.0) GO TO 111                                          PPT00920
0078            KOUNT=KOUNT+1                                                   PPT00930
0079           CALL PHI(X,P,0,N1)                                               PPT00940
0080           DO 10 J=1,T                                                      PPT00950
0081           L(J) = 0.0D0                                                     PPT00960
0082           DO 20 I=1,M                                                      PPT00970
0083           L(J) = L(J) + A(I,J)*P(I)                                        PPT00980
0084        20 CONTINUE                                                         PPT00990
0085        10 CONTINUE                                                         PPT01000
          C     DETERMINE MINIMUM LOSS AND CATEGORY.                             PPT01010
0086           CAT = 1                                                          PPT01020
0087           LMIN = L(1)                                                      PPT01030
0088           DO 30 I=1,T                                                      PPT01040
0089           IF(L(I) .GE. LMIN) GO TO 30                                      PPT01050
0090           LMIN=L(I)                                                        PPT01060
0091           CAT = I                                                          PPT01070
0092        30 CONTINUE                                                         PPT01080
0093           IGT=ALPHA(GTCAT)                                                 PPT01090
0094           IGG=CATREC(IGT)                                                  PPT01100
0095           IF(IGG.EQ.0.AND.GTCAT.NE.BLANK) IUNU=IUNU+1                      PPT01110
0096           IF(GTCAT.EQ.BLANK) IUNGT=IUNGT+1                                 PPT01120
0097           ISGZ(CAT)=ISGZ(CAT)+1                                            PPT01130
          C     WRITE(NPRT,3) (X(II),II=1,D)                                     PPT01140
          C     WRITE(NPRT,5) (L(II),II=1,T)                                     PPT01150
0098           IBAD=0                                                           PPT01160
0099           I=LINE                                                           PPT01170
0100           IF(I.LT.1.OR.I.GT.11) IBAD=1                                     PPT01180
0101           J=SAMPLE                                                         PPT01190
0102           IF(J.LT.1.OR.J.GT.19) IBAD=1                                     PPT01200
0103           IF(IBAD.EQ.1) GO TO 211                                          PPT01210
0104           IF(TYPE.EQ.0) NZTYP=NZTYP+1                                      PPT01220
0105           MASK(I,J)=TYPE                                                   PPT01230
0106           IF(GTCAT.EQ.BLANK) GO TO 200                                     PPT01240
0107           K=1                                                              PPT01250
0108           DOTLAB(I,J,K)=GTCAT                                              PPT01260
0109           IF(IGG.EQ.0) GO TO 200                                           PPT01270
0110           IGPFLG=1                                                         PPT01280
0111           PROBGP(IGG,CAT)=PROBGP(IGG,CAT)+1.0D0                            PPT01290
0112       200  CONTINUE                                                         PPT01300
0113           IF(AICAT.EQ.BLANK) GO TO 210                                     PPT01310
0114           K=2                                                              PPT01320
0115           DOTLAB(I,J,K)=AICAT                                              PPT01330
0116           IAPFLG=1                                                         PPT01340
0117           IAI=ALPHA(AICAT)                                                 PPT01350
0118           IAA=CATREC(IAI)                                                  PPT01360
0119           IF(IAA.EQ.0) GO TO 210                                           PPT01370
0120           PROBAP(IAA,CAT)=PROBAP(IAA,CAT)+1.0D0                            PPT01380
0121           IF(GTCAT.EQ.BLANK) GO TO 210                                     PPT01390
0122           IGAFLG=1                                                         PPT01400
0123           PROBGA(IGG,IAA)=PROBGA(IGG,IAA)+1.0D0                            PPT01410
0124       210  CONTINUE                                                         PPT01420
0125           K=3                                                              PPT01430
0126           DOTLAB(I,J,K)=IRETA(CAT,CATREC)                                  PPT01440
0127       211  CONTINUE                                                         PPT01450
          C     WRITE CLASSIFICATION RESULT.                                     PPT01460
0128           IF(PFLAG.GE.1) WRITE(NPRT,3) (X(I),I=1,D)                        PPT01470
0129         3    FORMAT(1H ,2X,'X',2X,15F6.1)                                   PPT01480
0130           IBB=IRETA(CAT,CATREC)                                            PPT01490
0131           IF(PFLAG.GE.1) WRITE(NPRT,4) LINE,SAMPLE,GTCAT,AICAT,IBB         PPT01500
0132           IF(IBAD.EQ.1) WRITE(NPRT,444)                                    PPT01510
0133       444  FORMAT(1H ,10X,'DOT COORDINATES OUT OF BOUNDS')                  PPT01520
```

FILE PPTC

```
0134            IF(PFLAG.GE.1) WRITE(NPRT,44) TYPE                                  PPT01530
0135         44   FORMAT(1H ,10X,'TYPE = ',I2)                                        PPT01540
0136         4    FORMAT(1H ,10X,'LINE=',I4,' SAMPLE=',I4,/,10X,' G. T. LABEL = ',    PPT01550
            *  A1,/,10X,                                                            PPT01560
            *  ' A. I. LABEL = ',A1,/,10X,' P. P. T. LABEL = ',A1)                  PPT01570
0137            IF(PFLAG.GE.1) WRITE(NPRT,4444) ICMNT,(IAIVEG(I),I=1,4)             PPT01580
0138         4444 FORMAT(1H ,10X,'A. I. COMMENT = ',A4,/,10X,'VEGETATION = ',4I1)    PPT01590
0139            IF(PFLAG.GE.1) WRITE(NPRT,5)                                        PPT01600
0140          5 FORMAT(1H ,6X,'  CLASS          LOSS  ')                            PPT01610
0141            IF(PFLAG.EQ.0)  GO TO 555                                           PPT01620
0142            DO 50 I=1,T                                                         PPT01630
0143            IRR=IRETA(I,CATREC)                                                 PPT01640
0144            WRITE(NPRT,51) IRR,L(I)                                             PPT01650
0145         50   CONTINUE                                                         PPT01660
0146         555  CONTINUE                                                         PPT01670
0147         51   FORMAT(1H ,10X,A1,5X,D20.10)                                     PPT01680
0148            IF(PFLAG.GE.2) WRITE(NPRT,802) (P(I),I=1,M)                         PPT01690
0149         802  FORMAT(/1X,'PMI VECTOR',/,(1X,3D20.10))                         PPT01700
0150            IF(PFLAG.GE.1) WRITE(NPRT,56) KOUNT                                 PPT01710
0151         56   FORMAT(1H ,10X,'KOUNT = ',I5)                                    PPT01720
0152            IF(PFLAG.GE.1) WRITE(NPRT,55)                                       PPT01730
0153         55   FORMAT(//)                                                       PPT01740
            C       WRITE(NPRT,4) CAT                                               PPT01750
0154            GO TO 1                                                             PPT01760
            C       END OF LOOP                                                     PPT01770
0155         111  CONTINUE                                                         PPT01780
0156            WRITE(NPRT,222) KOUNT                                               PPT01790
0157         222  FORMAT(//,10X,'THE NO. OF SAMPLES = ',I8)                       PPT01800
0158            WRITE(NPRT,900)                                                     PPT01810
0159         900  FORMAT(/1X,'...OVERALL CLASSIFICATION RESULTS...'/)             PPT01820
0160            DO 510 I=1,T                                                        PPT01830
0161            IRR = IRETA(I,CATREC)                                               PPT01840
0162            WRITE(NPRT,500) ISAZ(I),IRR                                         PPT01850
0163         500  FORMAT(//,10X,I5,'  PIXELS CLASSIFIED AS ',A1)                  PPT01860
0164         510  CONTINUE                                                         PPT01870
0165            IF(IUNGT.NE.0) WRITE(NPRT,511)  IUNGT                               PPT01880
0166         511  FORMAT(//,10X,I5,'  PIXELS WITHOUT GROUND TRUTH')               PPT01890
0167            IF(IUNU.GE.0) WRITE(NPRT,512)  IUNU                                 PPT01900
0168         512    FORMAT(/,10X,I5,'  UNUSED GROUND TRUTH SYMBOLS')               PPT01910
0169            IF(NZTYP.NE.0) WRITE(NPRT,513) NZTYP                                PPT01920
0170         513    FORMAT(/,10X,I5,'  ZERO TYPE NUMBERS FOUND')                   PPT01930
0171            IF(IGPFLG.EQ.1) WRITE(NPRT,710)                                     PPT01940
0172         710  FORMAT(///,10X,'SUMMARY TABLE FOR GT VS. PPT')                   PPT01950
0173            IF(IGPFLG.EQ.1) CALL PPTSUM(PROBGP,T,CATREC)                        PPT01960
0174            IF(IAPFLG.EQ.1) WRITE(NPRT,720)                                     PPT01970
0175         720  FORMAT(///,10X,'SUMMARY TABLE FOR AI VS. PPT')                   PPT01980
0176            IF(IAPFLG.EQ.1) CALL PPTSUM(PROBAP,T,CATREC)                        PPT01990
0177            IF(IGAFLG.EQ.1) WRITE(NPRT,730)                                     PPT02000
0178         730  FORMAT(///,10X,'SUMMARY TABLE FOR GT VS. AI')                    PPT02010
0179            IF(IGAFLG.EQ.1) CALL PPTSUM(PROBGA,T,CATREC)                        PPT02020
0180            IF(IGPFLG.EQ.1) WRITE(NPRT,7710)                                    PPT02030
0181         7710 FORMAT(///,10X,'PROBABILITY TABLE FOR GT VS. PPT')               PPT02040
0182            IF(IGPFLG.EQ.1) CALL PRBSUM(PROBGP,T,CATREC)                        PPT02050
0183            IF(IAPFLG.EQ.1) WRITE(NPRT,7720)                                    PPT02060
0184         7720 FORMAT(///,10X,'PROBABILITY TABLE FOR AI VS. PPT')               PPT02070
0185            IF(IAPFLG.EQ.1) CALL PRBSUM(PROBAP,T,CATREC)                        PPT02080
0186            IF(IGAFLG.EQ.1) WRITE(NPRT,7730)                                    PPT02090
0187         7730 FORMAT(///,10X,'PROBABILITY TABLE FOR GT VS. AI')                PPT02100
0188            IF(IGAFLG.EQ.1) CALL PRBSUM(PROBGA,T,CATREC)                        PPT02110
0189            WRITE(NPRT,610)                                                     PPT02120
0190         610  FORMAT(1H1,//,10X,'CLASSIFIED DOT LABELS')                      PPT02130
0191            CALL PPTDTL(DOTLAB,NSYM,3)                                          PPT02140
0192               NWREM1=NWRE-1                                                    PPT02150
            C       WRITE(NPRT,9966) NWREM1                                         PPT02160
0193         9966 FORMAT(//,10X,'NWREM1=',I5)                                     PPT02170
0194            CALL FSEMFL(NWRU,NWREM1,ISTAT)                                      PPT02180
0195            IF(ISTAT.NE.0) WRITE(NPRT,651) ISTAT                                PPT02190
0196         651  FORMAT(//,10X,'ISTAT = ',I5)                                     PPT02200
0197            IDOO=0                                                              PPT02210
0198            IF(IDOO.EQ.0) GO TO 666                                             PPT02220
0199            DO 870 K=1,3                                                        PPT02230
0200            WRITE(NPRT,880)                                                     PPT02240
0201         880  FORMAT(///)                                                      PPT02250
0202            WRITE(NPRT,871) ((DOTLAB(I,J,K),J=1,19),I=1,11)                     PPT02260
0203         871  FORMAT(1H ,10X,19A1)                                             PPT02270
0204         870   CONTINUE                                                        PPT02280
```

```
FILE PPTC
0205          666  CONTINUE                                                  PPT02290
0206                DO 960 I=1,11                                           PPT02300
0207               DO 860 J=1,19                                            PPT02310
0208          860   IDOT(I,J)=DOTLAB(I,J,3)                                 PPT02320
0209               WRITE(NWRU,861) ISITF                                    PPT02330
0210                CALL GTOWP(IDOT,0,NSYM)                                 PPT02340
                C     ENDFILE NWRU                                          PPT02350
0211                WRITE(NPRT,620)                                         PPT02360
0212          620   FORMAT(1H1,77,10X,'AI DOT LABELS')                      PPT02370
0213                CALL PPTOTL(DOTLAB,NSYM,2)                              PPT02380
0214                DO 630 I=1,11                                           PPT02390
0215                DO 630 J=1,19                                           PPT02400
0216          630   IDOT(I,J)=DOTLAB(I,J,2)                                 PPT02410
0217                 NWRU=28                                                PPT02420
0218                NWRU1=NWRU-1                                            PPT02430
0219               CALL FSFMFL(NWRU,NWRU1,ISTAT)                            PPT02440
0220               IF(ISTAT.NE.0) WRITE(NPRT,651) ISTAT                     PPT02450
0221               WRITE(NWRU,644) ISITF                                    PPT02460
0222          644  FORMAT('SITF = ',A4,' TYPE = AI ')                      PPT02470
0223               CALL GTOWP(IDOT,0,NSYM)                                  PPT02480
                C     ENDFILE NWRU                                          PPT02490
0224          861   FORMAT('SITF =  ',A4,' TYPE = PPT ')                    PPT02500
0225               IF(IGAFLG.EQ.1) CALL PPTTAB(1,2,DOTLAB)                  PPT02510
0226               IF(IRFFLG.EQ.1) CALL PPTTAB(1,3,DOTLAB)                  PPT02520
0227               IF(IAFFLG.EQ.1) CALL PPTTAB(2,3,DOTLAB)                  PPT02530
0228                GO TO 999                                               PPT02540
0229                END                                                     PPT02550
```

125

C.3 SOFTWARE COMPONENT NO. 3 (IBETA)

The function IBETA returns a letter corresponding to a category index.

C.3.1 Linkages

IBETA is called by subroutines PPTC and PRBSUM.

C.3.2 Interfaces

IBETA interfaces with other routines through a calling sequence and the common block LT.

C.3.3 Inputs

Calling sequence: Function IBETA (CAT, CATREC)

Parameter	Dimension	In/Out	Description
CAT	1	In	Category index.
CATREC	26	In	Array loader with index numbers corresponding to the letters of the alphabet.

Common block:

COMMON /LT/LETTER(26)

Parameter	Dimension	In/Out	Description
LETTER	26	In	An array containing the letters of the alphabet.

C.3.4 Outputs

N/A

C.3.5 Storage

Program size = 430

C.3.6 Description

IBETA returns a letter corresponding to a category index.

C.3.7 Flowchart

N/A

C.3.8 Listing

C.3.6 Description

IBETA returns a letter corresponding to a category index.

C.3.7 Flowchart

N/A

C.3.8 Listing

```
C        WRITTEN BY C W AHLERS                                       THE00010
C        CATREC IS A RECORD OF THE ORDER NOS. OF CATEGORIES          THE00020
C        THETA  RETURNS A LETTER FOR AN ORDER NO.                    THE00030
         FUNCTION THETA (CAT,CATREC)                                 THE00040
         IMPLICIT INTEGER (A-Z)                                      THE00050
         DIMENSION CATREC(26)                                        THE00060
         COMMON /LET/LETTER(26)                                      THE00070
         DO 10 I=1,26                                                THE00080
         IH=I                                                        THE00090
         IF(CAT.EQ.CATREC(I)) GO TO 20                               THE00100
   10    CONTINUE                                                    THE00110
   20    THETA=LETTER(IH)                                            THE00120
         RETURN                                                      THE00130
         END                                                         THE00140
```

124

C.4 SOFTWARE COMPONENT NO. 4 (PPTSUM)

Subroutine PPTSUM partially compiles and prints tables of joint occurances.

C.4.1 Linkages

PRBSUM is called by PPTC and calls IBETA.

C.4.2 Interfaces

PRBSUM interfaces other routines through calling sequences and the common block UN.

C.4.3 Inputs

Calling sequence: Subr. PPTSUM (PROBT, T, CATREC)

Parameter	Dimension	In/Out	Description
PROB	(11,11)	In	A matrix of joint counts.
T	1	In	The no. of categories.
CATREC	26	In	An array indicating the category index that corresponds to a category.

C.4.4 Outputs

PPTSUM prints out joint occurrance tables.

C.4.5 Storage

Program size = 1256.

C.4.6 Description

PPTSUM fills in the totals and prints out tables of joint occurances.

C.4.7 Flowchart

N/A

C.4.8 Listing

FILE PPTSUM

```
                C       WRITTEN BY C W AHLERS                                   PPT00010
                C       PPTSUM PRINTS JOINT OCCURANCE TABLES                    PPT00020
                C       COLUMN & ROW SUMS ARE COMPUTED                          PPT00030
    0001              SUBROUTINE PPTSUM(PROB,T,CATREC)                          PPT00040
    0002              INTEGER CATREC(24)                                        PPT00050
    0003                 DIMENSION LABEL(10)                                    PPT00060
    0004              DOUBLE PRECISION PROB(11,11)                              PPT00070
    0005              INTEGER T,TP1                                             PPT00080
    0006              INTEGER PUNIT,WUNIT                                       PPT00090
    0007              COMMON /UNI/NPROB1,NPROB2,NPRT,PUNIT,WUNIT                PPT00100
    0008              DO 500 I=1,T                                              PPT00110
    0009     500      LABEL(I)=LBETA(I,CATREC)                                  PPT00120
    0010              TP1=T+1                                                   PPT00130
    0011              DO 610 I=1,T                                              PPT00140
    0012              DO 610 J=1,T                                              PPT00150
    0013              PROB(I,TP1)=PROB(I,TP1)+PROB(I,J)                         PPT00160
    0014               PROB(TP1,TP1)=PROB(TP1,TP1)+PROB(I,J)                    PPT00170
    0015     610       CONTINUE                                                 PPT00180
    0016              IF(PROB(TP1,TP1).EQ.0.0D0) RETURN                         PPT00190
    0017              DO 710 J=1,T                                              PPT00200
    0018              DO 710 I=1,T                                              PPT00210
    0019              PROB(TP1,J)=PROB(TP1,J)+PROB(I,J)                         PPT00220
    0020     710      CONTINUE                                                  PPT00230
    0021              WRITE(NPRT,720)                                           PPT00240
    0022     720       FORMAT(//)                                               PPT00250
    0023              WRITE(NPRT,721) (LABEL(I),I=1,T)                          PPT00260
    0024     721      FORMAT(1H ,8X,10(9X,A1),/)                                PPT00270
    0025              DO 800 I=1,T                                              PPT00280
    0026              WRITE(NPRT,801) LABEL(I),(PROB(I,J),J=1,TP1)              PPT00290
    0027     800      CONTINUE                                                  PPT00300
    0028     801      FORMAT(/,4X,A1,5X,10F10.0)                                PPT00310
    0029              WRITE(NPRT,802) (PROB(TP1,J),J=1,TP1)                     PPT00320
    0030     802      FORMAT(/,10X,10F10.0)                                     PPT00330
    0031              RETURN                                                    PPT00340
    0032              END                                                       PPT00350
```

C.5 SOFTWARE COMPONENT NO. 5 (PRBSUM)

Subroutine PRBSUM compiles and prints tables of joint probabilities.

C.5.1 Linkages

PRBSUM is called by PPTC and calls IBETA.

C.5.2 Interfaces

PRBSUM interfaces with other routines through calling sequences and the common block UN.

C.5.3 Inputs

Calling sequence: Subr. PRBSUM (PROBT, T, CATREC)

Parameter	Dimension	In/Out	Description
PROB	(11,11)	In	A matrix of joint probabilities.
T	1	In	The no. of categories.
CATREC	26	In	An array indicating the category index that corresponds to a category.

C.5.4 Outputs

PRBSUM prints out joint occurrance tables.

C.5.5 Storage

Program size = 1220

C.5.6 Description

PPTSUM fills in the totals and prints out tables of joint occurances.

C.5.7 Flowchart

N/A

C.5.8 Listing

FILE PRASUM

```
                C       WRITTEN BY C W AHLERS                                   PRB00010
                C       PRASUM PRINTS JOINT PROBABILITIES TABLES                PRB00020
0001                    SUBROUTINE PRASUM(PROB,T,CATREC)                        PRB00030
0002                    INTEGER CATREC(25)                                      PRB00040
0003                      DIMENSION LABEL(10)                                   PRB00050
0004                    DOUBLE PRECISION PROB(11,11)                            PRB00060
0005                    DOUBLE PRECISION PCC                                    PRB00070
0006                    INTEGER T,TP1                                           PRB00080
0007                    INTEGER RUNIT,WUNIT                                     PRB00090
0008                    COMMON /UN/NRDR1,NRDR2,NPRT,RUNIT,WUNIT                 PRB00100
0009                    DO 500 I=1,T                                            PRB00110
0010                500 LABEL(I)=IBETA(I,CATREC)                                PRB00120
0011                    PCC=0.000                                               PRB00130
0012                    TP1=T+1                                                 PRB00140
0013                    IF(PROB(TP1,TP1).EQ.0.0D0) RETURN                       PRB00150
0014                    DO 610 I=1,TP1                                          PRB00160
0015                    DO 610 J=1,TP1                                          PRB00170
0016                      PROB(I,J)=PROB(I,J)/PROB(TP1,TP1)                     PRB00180
0017                    IF(I.EQ.J) PCC=PCC+PROB(I,I)                            PRB00190
0018                610   CONTINUE                                            PRB00200
0019                    PCC=PCC-PROB(TP1,TP1)                                   PRB00210
0020                    WRITE(NPRT,720)                                         PRB00220
0021                720   FORMAT(//)                                          PRB00230
0022                    WRITE(NPRT,721) (LABEL(I),I=1,T)                        PRB00240
0023                721   FORMAT(1H ,8X,10(9X,A1),/)                          PRB00250
0024                    DO 800 I=1,T                                            PRB00260
0025                    WRITE(NPRT,801) LABEL(I),(PROB(I,J),J=1,TP1)            PRB00270
0026                800   CONTINUE                                            PRB00280
0027                801   FORMAT(/,4X,A1,5X,10F10.5)                          PRB00290
0028                    WRITE(NPRT,802) (PROB(TP1,J),J=1,TP1)                   PRB00300
0029                802   FORMAT(/,10X,10F10.5)                               PRB00310
0030                    WRITE(NPRT,803) PCC                                     PRB00320
0031                803   FORMAT(//,10X,'THE ESTIMATE OF THE PROBABILITY',    PRB00330
                     *    ' OF CORRECT LABELING IS ',F10.5)                    PRB00340
0032                    RETURN                                                  PRB00350
0033                    END                                                     PRB00360
```

FILE PRASUM

COMMON BLOCK /UN / MAP SIZE 14

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NPRT	0	NRAR2	4	NPRT	8	RUNIT	C	WUNIT	10

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IRETA	88	IACOM#	AC						

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
PCC	CA	I	D0	T	D4	TP1	D8	J	DC

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CATREC	F0	LAREL	E4	PROB	10C				

FORMAT STATEMENT MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
720	110	721	114	801	123	802	131	803	138

OPTIONS IN EFFECT ID,EBCDIC,SOURCE,NOLIST,DECK,NOLOAD,MAP
OPTIONS IN EFFECT NAME = PRASUM , LINECNT = 60
STATISTICS SOURCE STATEMENTS = 33,PROGRAM SIZE = 1220
STATISTICS NO DIAGNOSTICS GENERATED

C.6 SOFTWARE COMPONENT NO. 6 (PPTDTL)

Subroutine PPTDTL counts the number of categories in a specified 209 dot gride as well as storing the category types found.

C.6.1 Linkages

PPTDTL is called by PPTC and must be called before GTDWR.

C.6.2 Interfaces

PPTDTL interfaces with other routines through a calling sequence and the common block TR.

C.6.3 Inputs

Calling sequence:
Subr. PPTDTL(DMTX, NSYM, K)

Parameter	Dimension	In/Out	Description
DTMTX	(11,19,3)	In	A matrix containing three 209 dot grids 1. ground truth 2. A.I. 3. PPT
NSYM	1	Out	Number of categories found
K	1	In	An index indicating which of the 209 dot grids to examine.

Common block:
COMMON/TR/TRNS1(256), TRNS2(26), TRNS3(26), TY(11,19)

Parameter	Dimension	In/Out	Description
TRNS1	256	-	Not used
TRNS2	26	-	Category symbols are stored in this vector
TRNS3	26	Out	This vector is a compressed version of TRNS2
TY.	(11,19)	-	Type mask.

C.6.4 Outputs

N/A

C.6.5 Storage

Program size = 832

C.6.6 Description

PPTDTL stores category Labels in TRNS3 and records the number of categories in NSYM.

C.6.7 Flowchart

N/A

C.6.8 Listings

```
C        WRITTEN C W AHLERS                                          PPT00010
C        GIVEN A MATRIX OF LETTERS  THIS ROUTINE LOADS TRNS3         PPT00020
C        WITH THE LETTERS FOUND & COUNTS THE NUMBER OF DIFFERENT     PPT00030
C          LETTERS FOUND (NSYM)                                      PPT00040
         SUBROUTINE PPTDTL(DMTX,NSYM,K)                              PPT00050
          IMPLICIT INTEGER (A-Z)                                     PPT00060
        COMMON /TR/TRNS1(256),TRNS2(26),TRNS3(26),TY(11,19)          PPT00070
        DATA B/' '/                                                  PPT00080
        DIMENSION DMTX(11,19,3)                                      PPT00090
        DO 5  I=1,26                                                 PPT00100
         TRNS2(I)=B                                                  PPT00110
        TRNS3(I)=B                                                   PPT00120
    5    CONTINUE                                                    PPT00130
        DO 10 I=1,11                                                 PPT00140
        DO 10  J=1,19                                                PPT00150
        SYM=DMTX(I,J,K)                                              PPT00160
        IF(SYM.EQ.B) GO TO 10                                        PPT00170
        NUM=ALPHA(SYM)                                               PPT00180
        TRNS2(NUM)=SYM                                               PPT00190
   10    CONTINUE                                                    PPT00200
C        WRITE(6,60) (TRNS2(I),I=1,26)                               PPT00210
   60     FORMAT(1H ,5X,26A1)                                        PPT00220
        NSYM=0                                                       PPT00230
        DO 100 I=1,26                                                PPT00240
        IF(TRNS2(I).NE.B) NSYM=NSYM+1                                PPT00250
        IF(TRNS2(I).NE.B) TRNS3(NSYM)=TRNS2(I)                       PPT00260
  100   CONTINUE                                                     PPT00270
        WRITE(6,110) (TRNS3(I),I=1,26)                               PPT00280
  110    FORMAT(///,5X,'CATEGORIES FOUND: ',26A1)                    PPT00290
        RETURN                                                       PPT00300
        END                                                          PPT00310
```

C.7 SOFTWARE COMPONENT NO. 7 (PPTTAB)

Subroutine PPTTAB prints joint label tables for the 209 dots.

C.7.1 Linkages

PPTTAB is called by PPTC.

C.7.2 Interfaces

PPTTAB interfaces with other routines through a calling sequence and common blocks UN/and TR.

C.7.3 Inputs

Calling sequence:
Subr. PPTTAB(I1, I2, DOTLAB)

Parameter	Dimension	In/Out	Description
I1	1	In	Index for the first label type.
I2	1	In	Index for the second label type.
DOTLAB	(11,19,3)	In	A matrix containing three 209 dot label grids. 1. ground truth 2. A.I. 3. PPT

Common blocks:
COMMON/TR/TRNS1(256), TRNS2(26), TRNS3(26), MASK(11,19)

Parameter	Dimension	In/Out	Description
TRNS1	256	-	not used
TRNS2	26	-	not used
TRNS3	26	-	not used
MASK	(11,19)	In	Dot Type Mask.

COMMON/UN/NRDR1, NRDR2, NPRT, RUNIT, WUNIT

Parameter	Dimension	In/Out	Description
NRDR1	1	-	not used
NRDR2	1	-	not used
NPRT	1	-	Printer unit number
RUNIT	1	-	not used
WUNIT	1	-	not used

C.7.4 Outputs

Joint label tables are written to the line printer. These tables indicate such things as the ground truth label for a dot vs the AI label for a dot.

C.7.5 Storage

Program size = 1636.

C.7.6 Description

Joint label tables are written by PPTTAB to the line printer for the type one and type two dots.

C.7.7 Flowchart

N/A

C.7.8 Listings

FILE PPTTAB

```
                C       WRITTEN BY C W AHLERS                                   PPT00010
                C THIS SUBROUTINE PRINTS LABEL TABELS                           PPT00020
                C       DOTLAB(I,J,1)     GROUND TRUTH LABELS                   PPT00030
                C       DOTLAB(I,J,2)     A. I. LABELS                          PPT00040
                C       DOTLAB(I,J,3)     P. P. I. LABELS                       PPT00050
0001                    SUBROUTINE PPTTAB(I1,I2,DOTLAB)                         PPT00060
0002                  IMPLICIT INTEGER (A-Z)                                    PPT00070
0003                  DATA SLASH/'/'/,BLANK/' '/                                PPT00080
0004                  DIMENSION LINE(57)                                        PPT00090
0005                  COMMON /UN/NRDR1,NRDR2,NPRT,RUNIT,WUNIT                   PPT00100
0006                  COMMON /TH/TRNS1(256),TRNS2(26),TRNS3(26),MASK(11,19)     PPT00110
0007                    DIMENSION DOTLAB(11,19,3)                               PPT00120
                C       NTYPE=NO. OF TYPES                                      PPT00130
0008                    NTYPE=2                                                 PPT00140
0009                  DO 100 ITYPE=1,NTYPE                                      PPT00150
0010                    WRITE(NPRT,10)                                          PPT00160
0011             10     FORMAT(1H1)                                             PPT00170
0012                    IF(I1.EQ.1.AND.I2.EQ.2)WRITE(NPRT,120)                  PPT00180
0013            120     FORMAT(//,50X,'GROUND TRUTH VS A. I. LABELS')           PPT00190
0014                    IF(I1.EQ.1.AND.I2.EQ.3) WRITE(NPRT,130)                 PPT00200
0015            130     FORMAT(//,50X,'GROUND TRUTH VS CLASSIFIED LABELS')      PPT00210
0016                  IF(I1.EQ.2.AND.I2.EQ.3) WRITE(NPRT,230)                   PPT00220
0017            230     FORMAT(//,50X,'A. I. LABELS VS CLASSIFIED LABELS')      PPT00230
0018                  WRITE(NPRT,20) ITYPE                                      PPT00240
0019             20     FORMAT(//,50X,'TYPE ',I1,' DOT CLASSIFICATION')         PPT00250
0020                  WRITE(NPRT,30) (I,I=10,190,10)                            PPT00260
0021             30     FORMAT(//,12X,19I5)                                     PPT00270
0022                    DO 200 I=1,11                                           PPT00280
0023                    DO 220 J=1,19                                           PPT00290
0024                    DO 210 K=1,3                                            PPT00300
0025                    POINT=3*(J-1)+K                                         PPT00310
0026                    LINE(POINT)=BLANK                                       PPT00320
0027            210     CONTINUE                                                PPT00330
0028                    IF(MASK(I,J).NE.ITYPE)  GO TO 220                       PPT00340
0029                    LL=3*(J-1)                                              PPT00350
0030                    LINE(LL+1)=DOTLAB(I,J,I1)                               PPT00360
0031                    LINE(LL+3)=DOTLAB(I,J,I2)                               PPT00370
0032                  IF(DOTLAB(I,J,I1).NE.BLANK.OR.DOTLAB(I,J,I2).NE.BLANK)    PPT00380
                     *  LINE(LL+2)=SLASH                                        PPT00390
0033            220     CONTINUE                                                PPT00400
0034                    II0=I*10                                                PPT00410
0035                    WRITE(NPRT,300)II0,(LINE(KK),KK=1,57)                   PPT00420
0036            300     FORMAT(//,9X,I3,19(2X,3A1))                             PPT00430
0037            200     CONTINUE                                                PPT00440
0038            100     CONTINUE                                                PPT00450
0039                    RETURN                                                  PPT00460
0040                    END                                                     PPT00470
```

C.8 SOFTWARE COMPONENT NO. 8 (NLOW)

Function NLOW returns a pointer to a lower triangular matrix given rectangular components.

C.8.1 Linkages

NLOW is called by PPTA.

C.8.2 Interfaces

NLOW interfaces with other routines through a calling sequence.

C.8.3 Inputs

Parameter	Dimension	In/Out	Description
I	1	In	First rectangular coordinate
J	1	In	Second rectangular coordinate
M	1	In	Size of a symetric matrix.

C.8.4 Outputs

N/A

C.8.5 Storage

Program size = 490.

C.8.6 Description

NLOW returns a pointer to a lower triangular matrix (size m by m) given the rectangular components I an J.

C.8.7 Flowchart

N/A

C.8.8 Listings

```
FORTRAN IV G LEVEL 21          NLOW          DATE = 79003          16/59/28          PAGE 0001
FILE *LOW

                C       WRITTEN BY C AHLERS                                          NL000010
                C       NLOW RETURNS A POINTER TO A LOWER TRIANGULAR MATRIX (M BY M) NL000020
                C       GIVEN THE RECTANGULAR COMPONENTS I & J                       NL000030
0001                    FUNCTION NLOW(I,J,M)                                         NL000040
0002                    NLOW=0                                                       NL000050
0003                    DO 10 KI=1,M                                                 NL000060
0004                    DO 10 KP=1,KI                                                NL000070
0005                    NLOW=NLOW+1                                                  NL000080
0006                    IF(KI.EQ.I.AND.KP.EQ.J) RETURN                               NL000090
0007                 10 CONTINUE                                                     NL000100
0008                    RETURN                                                       NL000110
0009                    END                                                          NL000120
```

C.3

44

APPENDIX D

APPENDIX D

APPENDIX D

The following calculations are made in computing the LIST variables.

BRIET (i,j) = (B(i, j) - MEANB)/SDG

where i = pixel (1-209)

j = index to acquisition number

B(i,j) = value extracted from 16 tape using the KAUTH transformation

SDB is calculated as described in Table 3-3

(BIØ = biostage from second AI header card Table 3-4)

MEANB is calculated as described in Table 3-3

GREEN (i,j) = (G(i,j)-MEANB)/SDG

where i = pixel (1-209)

j = index to acquisition number

G (i,j) = value extracted from 16-channel tape

SDG is calculated as described in Table 3-3

(BIO = biostage from second AI header card Table 3-4)

MEANG is calculated as described in Table 3-3

ABRIET (i,j) = /BRIET(i,j)/

AGREEN(i,j) = /GREEN(i,j)/

$$SQAIRB(i) = \sum_{j=1}^{4} (BRIET(i,j)^2)$$

$$SQAIRG(i) = \sum_{j=1}^{4} (GREEN(i,j)^2)$$

$$PIEB(i) = \prod_{j=1}^{4} (1 + ABRIET(i,j))$$

$$PIEG(i) = \prod_{j=1}^{4} (1 + AGREEN(i,j))$$

CANKY(i,j) = 0 if PCR(j) = first class response

5 if PCR(j) = second class response

10 if PCR(j) = any other response

PCR (j) = Pixel class responses (Table 3-4).

Class responses defined in Table 3-2 are indexed by biostage from header card (Table 3-4).

$$\text{CANTJ}(i) = 0 \text{ if } \sum_{j=1}^{4} (\text{CANKY}(i,j) = 0 \text{ or } 5$$

$$1 \text{ if } \sum_{j=1}^{4} (\text{CANKY}(i,j) > 5$$

APPENDIX E

E.1 SOFTWARE SUBPROGRAM NO. 1 (REDIF3)

Subrpogram REDIF3 reads and decodes all of the input control card images (except for the MAPTAP card) when the DISPLAY processor is invoked. It also reads and interprets field cards (TEST or designated).

E.1.1 Linkages

Subprogram REDIF3 is called from SETUP3 after SETUP3 has read the MAPTAP control card and the first two records of the MAPTAP file. REDIF3 calls subroutine CHIN, EOD-LARSYS utility routines FLTNUM, NXTCHR, LAREAD, FIND12, NUMBER, and CMERR, and the Fortran re-read routine REREAD.

E.1.2 Interfaces

REDIF3 includes labeled common blocks /GLOBAL/ and /DISPL/.

E.1.3 Inputs

o Calling Sequence

Keyword	Dimension	In/Out	Description
TSTSAV	(4,200)	Out	Test or Designated field information - class ID, # vertices per field
TSTFLD	(5,200)	Out	Test or Designated field surrounding rectangle line and sample numbers, pointer to field in TSTVER
TSTVER	(VDIM)	Out	field vertices from LAREAD
VDIM	1	In	Number of words of available storage in ARRAY buffer, computed in SETUP3
LISTSW	1	Out	Switch to indicate whether or not LIST processing has been selected

Keyword	Dimension	In/Out	Description
GTUNIT	1	Out	Ground truth dot file unit number (LIST processing)
GTFILE	1	Out	Ground truth dot file relative file number (LIST processing)
AIUNIT	1	Out	AI dot file unit number (LIST)
AIFILE	1	Out	AI dot file relative file number (LIST)
PPUNIT	1	Out	PPTC classifier dot file unit number (LIST)
PPFILE	1	Out	PPTC classifier dot file relative file number (LIST)
NAMECT	1	Out	Name of selected category (LIST processing)
ALP	2	Out	Bias correction alpha values (LIST processing)

E.1.4 Outputs

N/A

E.1.5 Storage Requirements

9962

E.1.6 Description

REDIF3 has been expanded to accept the new control cards described in Section 3.2.8.3, in support of LIST processing. This extra information is passed to SETUP3 and in turn to DSPLY2 by means of an expnaded calling sequence. The variables LISTSW to ALP described above constitute this expansion.

E.1.7 Flowchart

N/A

E.1.8 Listing

FILE REDIF3

```
0001          SUBROUTINE REDIF3(TSTSAV,TSTFLD,TSTVER,VDIM,
             *  LISTSW,GTUNIT,GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,
             *  NAMECT,ALP)
0002          IMPLICIT INTEGER (A-H,O-Z)
        C
        C*** CODE ADDED TO INCLUDE LIST PROCESSING
        C
0003               REAL ALP(2)
        CI----------------------------------------------------------------I
        CI----------------------------------------------------------------I
        CI                                                                I
        CI    PURPOSE.. READS AND ANALYZES SUPERVISOR CONTROL CARDS       I
        CI              FOR ' DISPLAY'                                    I
        CI                                                                I
        CI                                                                I
        CI----------------------------------------------------------------I
        CI----------------------------------------------------------------I
        C
        C
        C
        C
0004          EQUIVALENCE     (HED1(1),HEAD(41)), (DATE(1),HEAD(22)),
             1                (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
        C
        C     ----------------------------------------------------------------
        C     INCLUDE COMBK6
        C     INCLUDE CMBK10,LIST
0005          COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
             *              HISFIL,HISKEY,TPFORM,FRIPTP,ERPKEY,MAPUNT,NOFILE,
             *   DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
             *  ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
             *  ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
             *   CRDUNT,PRTUNT,RANDIO
        C$END
0006          COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),
             *              SUBCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,
             *              PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,
             *              PLTKEY,BMFLG,BMCOMB,BMFFAT,CDATE(2),
             *              FLDSV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,
             *              THRES(60),SYMMTX(66),HIGH(60),CON(60)
             *              ,FLDKEY,NOFLD2,NOFLD3,NOFFT2,FETVC2(30)
             *              ,NOSUB2,NOTRFD,TOTVT2,NOCLS2
             *              ,KATNO(60),NOCAT,FILTER,MAPFMT
             *              ,DESKEY,DESUNI,DESOTH,CROP ,ACROP,AOTHER,ATOTAL
             *              ,SITE(6),ANALYS(5),CAM(15),CRPKEY,KEPPTS(60)
             *              ,DOTKEY,DOTERR
0007          REAL HIGH,CHIN
0008          REAL ACROP,AOTHER,ATOTAL,X
        C
0009          REAL THRES
0010          DIMENSION TSTSAV(4,200),TSTFLD(5,200),TSTVER(VDIM),
             *          OPT(20),COMVEC(2),INF(7),ACARD(20)
             3,CODHCD(10),EQUCOM(3),CARD2(62),CARD1(80), SLASH(2)
             4 ,        COMENT(15), DATE(3), HED1(15), HED2(15)
        C
0011          DATA OPT/'SYMB','SITE','OPTI','THRE','ANAL',
             *         'COMM','HED1','HED2','DATE','PROC','ACPE',
             *         '$END','FORM','CROP','DOTF',
             *         'GTUN','AIUN','PPUN','NAME','ALPH'/
0012          DATA OPTNUM/12/,ENDHCD/'$END'/,THSHCD/'THRS'/,BLANK/' '/ ,
             1  COMMA/','/,CODHCD/' ','S','O','P','N','E','C','T','F'/
             2,EQUCOM/2,'=',','/,COMVEC/1,','/, SLASH/1,'/'/
        C
0013          DATA LO/'O'/,LI/'I'/,LF/'F'/,LU/'U'/
0014          DATA LL/'L'/,LS/'S'/,LT/'T'/,LC/'C'/,LN/'N'/,OTHER/'OTHE'/,
             *UNIDEN/'UNIO'/
0015          DIMENSION SYMMT(66),EQUVEC(2)
0016          DATA SYMMT/'1','2','3','4','5','6','7','8','9','A','B','C','D',
             1           'E','F','G','H','I','J','K','L','M','N','O','P','Q',
             2           'R','S','T','U','V','W','X','Y','Z','_','|','¬','/',
             3           '-','*','+','$','#','=','0',' ','=','∓',')','(',':',
             4           '&','>','%','?','@',',','.',' ' /
0017          DATA TRNSYM,TSTSYM,DUPSYM,THRSYM/'*','+','$',' '/
0018          DATA DESSYM/'#'/,EQUVEC/1,'='/
0019          DATA C,L/'C','L'/
0020          LOGICAL*1 LM(4),LCROP(4)
```

153

ORIGINAL PAGE IS OF POOR QUALITY

FILE REDIF3

```
0001          SUBROUTINE REDIF3(TSTSAV,TSTFLO,TSTVER,VDIM,                    RED00010
             *  LISTS#,GTUNIT,GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,             RED00020
             *  NAMECT,ALP)                                                   RED00030
0002          IMPLICIT INTEGER (A-H,O-Z)                                      RED00040
        C                                                                     RED00050
        C*** CODE ADDED TO INCLUDE LIST PROCESSING                            RED00060
        C                                                                     RED00070
0003                REAL ALP(2)                                               RED00080
        CI-------------------------------------------------------------------IRED00090
        CI-------------------------------------------------------------------IRED00100
        CI                                                                   IRED00110
        CI    PURPOSE.. READS AND ANALYZES SUPERVISOR CONTROL CARDS          IRED00120
        CI              FOR ' DISPLAY'                                       IRED00130
        CI                                                                   IRED00140
        CI                                                                   IRED00150
        CI-------------------------------------------------------------------IRED00160
        CI-------------------------------------------------------------------IRED00170
        C                                                                     RED00180
        C                                                                     RED00190
        C                                                                     RED00200
        C                                                                     RED00210
0004          EQUIVALENCE     (HED1(1),HEAD(4)), (DATE(1),HEAD(22)),          RED00220
             1                (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))         RED00230
        C                                                                     RED00240
        C     ----------------------------------------------------------------RED00250
        C     INCLUDE COMRK6                                                  RED00260
        C     INCLUDE CMRK10,LIST                                             RED00270
0005          COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,       RED00280
             *             HISFIL,HISKEY,TRFORM,FRIPTP,ERRKEY,MAPUNT,NOFILE,  RED00290
             *   DSUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL               RED00300
             * ,NHSTUN,NHSTFI,SCTRUN,MAPFIL                                   RED00310
             * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,             RED00320
             *  CRDUNT,PPTUNT,RANDIO                                          RED00330
        C$END                                                                 RED00340
0006          COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60), RED00350
             *             SUBCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,         RED00360
             *             PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,  RED00370
             *             PLTKEY,BMFLG,BMCOMB,BMFEAT,CDATE(2),               RED00380
             *             FLDSV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,  RED00390
             *             THRES(60),SYMMTX(66),HIGH(60),CON(60)              RED00400
             *             ,FLDKEY,NOFLD2,NOFLD3,NOFFT2,FETVC2(30)            RED00410
             *             ,NOSUB2,NOTRFD,TOTVT2,NOCLS2                       RED00420
             *             ,KATNO(60),NOCAT,FILTER,MAPFMT                     RED00430
             *             ,DFSKEY,DFSUNI,DFSOTH,CROP ,ACROP,AOTHER,ATOTAL    RED00440
             *             ,SITE(6),ANALYS(5),CAM(15),CRPKEY,KEPPTS(60)       RED00450
             *             ,DOTKEY,DOTERR                                     RED00460
0007          REAL HIGH,CHIN                                                  RED00470
0008          REAL ACROP,AOTHER,ATOTAL,X                                      RED00480
        C                                                                     RED00490
0009          REAL THRES                                                      RED00500
0010          DIMENSION TSTSAV(4,200),TSTFLO(5,200),TSTVER(VDIM),             RED00510
             *            OPT(20),COMVEC(2),INF(7),ACARD(20)                  RED00520
             3,CODBCD(10),EQUCOM(3),CARD2(62),CARD1(80), SLASH(2)             RED00530
             4 ,            COMENT(15), DATE(3), HED1(15), HED2(15)           RED00540
        C                                                                     RED00550
0011          DATA OPT/'SYMB','SITE','OPTI','THRE','ANAL',                    RED00560
             *          'COMM','HED1','HED2','DATE','PROC','ACRE',            RED00570
             *          '*END','FORM','CROP','DOTF',                          RED00580
             *          'GTUN','AIUN','PPUN','NAME','ALPH'/                   RED00590
0012          DATA OPTNUM/12/,ENDBCD/'*END'/,THSBCD/'THRS'/,BLANK/' '/ ,      RED00600
             1  COMMA/','/,CODBCD/' ','S','O','P','N','E','C','T','F'/        RED00610
             2,EQUCOM/2,'=',','/,COMVEC/1,','/, SLASH/1,'/'/                  RED00620
        C                                                                     RED00630
0013          DATA LO/'O'/,LI/'I'/,LF/'F'/,LU/'U'/                            RED00640
0014          DATA LL/'L'/,LS/'S'/,LT/'T'/,LC/'C'/,LN/'N'/,OTHER/'OTHE'/,     RED00650
             *UNIDEN/'UNID'/                                                  RED00660
0015          DIMENSION SYMMT(66),EQUVEC(2)                                   RED00670
0016          DATA SYMMT/'1','2','3','4','5','6','7','8','9','A','B','C','D', RED00680
             1             'E','F','G','H','I','J','K','L','M','N','O','P','Q', RED00690
             2             'R','S','T','U','V','W','X','Y','Z','_','|','¬','/', RED00700
             3             '-','*','+','$','"','=','0',' ','=','±',')','(',':', RED00710
             4             '&','>','%','?','#',',',',',',',' '/               RED00720
0017          DATA TRNSYM,TSTSYM,DUPSYM,THRSYM/'*','+','$',' '/               RED00730
0018          DATA DFSSYM/'#'/,EQUVEC/1,'='/                                  RED00740
0019          DATA C,L/'C','L'/                                               RED00750
0020          LOGICAL*1 LM(4),LCROP(4)                                        RED00760
```

```
FILE REDIF3
0021                EQUIVALENCE  (CARD2(1),CARD1(11)),(M,LM(1)),(LCROP(1),KCROP)      RED00770
0022                DO 5 I=1,6A                                                       RED00780
0023              5 SYMMTX(I)=SYMMT(I)                                                RED00790
                                                                                      RED00800
C                   ------------------------------------------------------------------RED00810
C                   ------------------------------------------------------------------RED00820
C                                                                                     RED00830
C                   INIT                                                              RED00840
C                   ------------------------------------------------------------------RED00850
C                                                                                     RED00860
C                                                                                     RED00870
C***** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING                            RED00880
C                                                                                     RED00890
0024                      LISTSW = 0                                                  RED00900
0025                      GTOUTT = 0                                                  RED00910
0026                      GTFILE = 0                                                  RED00920
0027                      AIOUTT = 0                                                  RED00930
0028                      AIFILE = 0                                                  RED00940
0029                      PPUNIT = 0                                                  RED00950
0030                      PPFILE = 0                                                  RED00960
0031                      NAMECT = LS                                                 RED00970
0032                     ALP(1) = 0.                                                  RED00980
0033                     ALP(2) = 0.                                                  RED00990
0034                DOTERR=0                                                          RED01000
0035                DOTKEY = 0                                                        RED01010
0036                TSTKEY = 0                                                        RED01020
0037                TRUKEY = 0                                                        RED01030
0038                STATKY = 0                                                        RED01040
0039                PCEDKY = 0                                                        RED01050
0040                PLTKEY = 0                                                        RED01060
0041                MOFFMT = 0                                                        RED01070
0042                NO-OP=1                                                           RED01080
0043                FILTER=0                                                          RED01090
0044                TRNSVA = 0                                                        RED01100
0045                ERRTRS = 0                                                        RED01110
0046                SYSOUT = 0                                                        RED01120
0047                TRNSKY=0                                                          RED01130
0048                STOPER = 0                                                        RED01140
0049                GRPIN = 0                                                         RED01150
0050                NOGRP = 0                                                         RED01160
0051                NOTREO = 0                                                        RED01170
0052                TYPE=0                                                            RED01180
0053                CARKEY=0                                                          RED01190
0054                TRSOUT=0                                                          RED01200
0055                DFSKEY=0                                                          RED01210
0056                TSTOUT = 0                                                        RED01220
0057                LOGSWT = 0                                                        RED01230
0058                PCCLKY = 1                                                        RED01240
0059                SYMMAX = 60                                                       RED01250
0060                DO 6 I=1,SYMMAX                                                   RED01260
0061                TYPES(I) = 0                                                      RED01270
0062              6 CONTINUE                                                          RED01280
0063                NBUFL03=200                                                       RED01290
                                                                                      RED01300
C                   ------------------------------------------------------------------RED01310
C                   READ IN SUPERVISOR CARDS                                        RED01320
C                   ------------------------------------------------------------------RED01330
C                                                                                     RED01340
C                   SET UP REREAD BUFFER                                            RED01350
C                                                                                     RED01360
0064                RRUNIT=30                                                         RED01370
0065                CALL REREAD(RRUNIT,80)                                            RED01380
C                                                                                     RED01390
C                    NOW READ THE CARD INTO THE BUFFER                                RED01400
C                                                                                     RED01410
0066             10 CONTINUE                                                          RED01420
0067             15 READ(21,180)(ACARD(I),I=1,20)                                     RED01430
0068            180 FORMAT(20A4)                                                      RED01440
0069                WRITE(RRUNIT,180)(ACARD(I),I=1,20)                                RED01450
0070                REWIND RRUNIT                                                     RED01460
0071                READ(30,104)CODE,CARD2                                            RED01470
0072            104 FORMAT(A4,6X,62A1)                                                RED01480
0073                REWIND RRUNIT                                                     RED01490
0074             17 WRITE(6,304)      CODE,CARD2                                      RED01500
0075            304 FORMAT(T7,A4,6X,62A1)                                             RED01510
0076             19 COL = 0                                                           RED01520
```

FILE -EDIF3

```
0077            DO 20 I=1,20
0078            IF(OPT(I).EQ.CODE)GO TO (100,200,300,400,600,710,720,730,
               *     740,760,770,800,750,780,785,210,211,212,230,240), I
0079       20   CONTINUE
0080            GOTO 1500
          C
          C     GET SYMBOLS
          C     ------------------------------------------------------------
          C
0081      100 IF ( SYMCNT .GE. SYMMAX ) GOTO 10
0082            SYMCNT = SYMCNT+1
0083            SYMNTX(SYMCNT) = BLANK
0084            M = NXTCHR(CARD2,COL)
0085            IF ( M .EQ. BLANK ) GO TO 10
0086            IF ( M .EQ. COMMA) GO TO 100
0087            SYMNTX(SYMCNT) = CARD2(COL)
0088      110 M = NXTCHR(CARD2,COL)
0089            IF ( M .EQ. BLANK ) GO TO 10
0090            IF ( M .NE. COMMA ) GO TO 110
0091            GO TO 100
          C*
          C*    SITE NAME
          C*
0092      200 READ(30,201)SITE
0093      201 FORMAT(10X,6A4)
0094            REWIND RHUNIT
0095            GO TO 10
          C
          C***  CODE ADDED NOV 13,1978, TO INCLUDE LIST PROCESSING
          C
          C***   READ GT AI OR PP UNIT AND FILE NUMBERS
0096      210          IPAT = 16
0097                   GO TO 214
0098      211          IPAT = 17
0099                   GO TO 214
0100      212          IPAT = 18
0101      214          M = NXTCHR(CARD2,COL)
0102                   IF (M.EQ.BLANK) GO TO 216
0103                   IF (M.NE.LU) GO TO 215
0104                   M = FIND12(CARD2,COL,EQUVEC)
0105                   IF(M.NE.2) GO TO 216
0106                   ISTART = 0
0107                   M = NUMBER(CARD2,COL,IPATT,ISTART)
0108                   M = FIND12(CARD2,COL,EQUVEC)
0109                   IF(M.NE.2) GO TO 216
0110                   ISTART = 0
0111                   M = NUMBER(CARD2,COL,IPATTT,ISTART)
0112                   GO TO 218
0113      215          IF(M.NE.LF) GO TO 216
0114                   M = FIND12(CARD2,COL,EQUVEC)
0115                   IF (M.NE.2) GO TO 216
0116                   ISTART = 0
0117                   M = NUMBER(CARD2,COL,IPATTT,ISTART)
0118                   M = FIND12(CARD2,COL,EQUVEC)
0119                   IF(M.NE.2) GO TO 216
0120                   ISTART = 0
0121                   M = NUMBER(CARD2,COL,IPATT,ISTART)
0122                   GO TO 218
0123      216          WRITE(6,217) OPT(IPAT)
0124      217          FORMAT('  ERROR ON ',A4,'  CONTROL CARD ')
0125                   GO TO 10
0126      218          IF(IPAT.NE.16)  GO TO 219
0127                   GTUNIT = IPATT
0128                   GTFILE = IPATTT
0129                   GO TO 221
0130      219          IF(IPAT.NE.17) GO TO 220
0131                   AIUNIT = IPATT
0132                   AIFILE = IPATTT
0133                   GO TO 221
0134      220          PPUNIT = IPATT
0135                   PPFILE = IPATTT
0136      221          LISTSW = 1
0137                   GO TO 10
          C*** SELECTED LIST CLASS NAME
0138      230           NAMECT = NXTCHR(CARD2,COL)
0139                   IF (NAMECT.NE.BLANK) GO TO 10
```

158

FILE REDIF3

```
0140              WRITE(6,235)
0141      235     FORMAT('   NO NAME APPEARS ON SELECTED CATEGORY CARD ')
0142              NAMECT = LS
0143              GO TO 10
          C***    VALUES FOR ALPHA IN BIAS CORRECTION
          C
0144      240     M = FLTNUM(CARD2,COL,ALP,2)
0145              IF (M.EQ.2) GO TO 10
0146              WRITE(6,245)
0147      245     FORMAT('  ERROR ON ALPHA CARD. DEFAULTING TO ZERO   ')
0148              ALP(1) = 0.
0149              ALP(2) = 0.
0150              GO TO 10
          C
          C****
          C
          C
          C       SCAN  OPTION CARD
          C       ------------------------
          C
0151        300 M = NXTCHR(CARD2,COL)
0152            DO 310   I=1,9
0153            IF(M.EQ.CODACD(I))GO TO (10,340,360,380,385,395,
               *     397,398,399),I
0154        310 CONTINUE
0155        315 WRITE(6,3152)  CODE, CARD2
0156       3152 FORMAT(//// 5X,'**** DSPLAY/REDIF3 --- ERROR IN ''OPTION'' CA
               *RD ...'//5X,2H'',A4,6X,62A1,2H''//5X,'**** SCAN OF THIS CARD DIS
               2CONTINUED --- PROCEEDING TO NEXT CARD ***** ///)
0157            GO TO 10
          C     ---------
          C
0158        340 STATKY = 1
0159            GO TO 390
          C     ---------
          C
0160        360 THMKEY = 1
0161            GO TO 390
          C     ---------
          C
          C
          C
0162        380 M=NXTCHR(CARD2,COL)
0163            IF(M.EQ.C)PCFDKY=1
0164            IF(M.EQ.L)PLTKEY=1
0165            GO TO 390
          C     -----
          C
0166        385 NOMAP=0
          C     ---------
          C
0167        390 M = FIND12(CARD2,COL,EQUCOM)
0168            IF ( M .LE. 0 ) GO TO 10
0169            GO TO 300
          C
          C-   -- SET THRESHOLD KEYS FOR EMPERICAL THRESHOLDING,TURN OTHERS OFF
0170        395 THRSVA=0
0171            EMPTRS=2
0172            THRSKY=2
0173            GO TO 390
          C
          C     ---------------
          C
          C
          C-   -- SET THRESHOLD KEYS FOR CHI SQUARE THRESHOLDING,TURN OTHERS OFF
0174        397 EMPTRS=0
0175            THRSVA=0
0176            THRSKY=1
0177            GO TO 390
          C
          C     --------------
          C
          C-   -- SET THRESHOLD KEYS FOR INPUT-VALUE THRESHOLDING,TURN OTHERS OFF
0178        398 EMPTRS=0
0179            THRSVA=3
0180            THRSKY=3
0181            GO TO 390
          C-   CHECK FOR FISHER OR FILTER
```

FILE REDIF3

```
0182        399 COL=COL+1
0183            M=NATCHR(CARD2,COL)
0184            IF(M.EQ.LL)GO TO 3990
0185            IF(M.EQ.LS)GO TO 3991
0186            GO TO 315
0187       3990 FILTHR=1
0188            GO TO 390
          C-
0189       3991 IX=0
0190            DO 3995 I=1,NOSUP2
0191            IF(KEPPTS(I).GT.NOFET2)GO TO 3995
0192            IF(IX.NE.0)GO TO 3993
0193            WRITE(6,3992)
0194       3992 FORMAT(1H0,////)
0195            IX=1
0196       3993 WRITE(6,3994)  I , KEPPTS(I) , NOFET2
0197       3994 FORMAT(' *****   FISHER THRESHOLD REQUESTED-NOT PERFORMED',/,8X,
               1'... NO. SAMPLES FOR SUBCLASS',I6,'(=',I6,') IS LESS THAN OR EQUAL
               2 TO NUMBER OF CHANNELS (=', I6,')',/)
0198       3995 CONTINUE
0199            IF(IX.EQ.1)GO TO 3996
0200            WRITE(6,3992)
0201            GO TO 390
          C-
          C-    -- SET THRESHOLD KEYS FOR  FISHER   THRESHOLDING,TURN OTHERS OFF
0202       3996 FLPTHS=0
0203            THRSVA=0
0204            THRSKY=4
          C-
          C-
0205            GO TO 390
          C
          C     READ IN THRESHOLDS
          C     ----------------------
          C
0206        400 CONTINUE
0207            I = SYMMAX-THSCNT
0208            THSCNT = THSCNT+FLTNUM(CARD2,COL,THRES(THSCNT+1),I)
0209            GO TO 10
          C
          C*
          C*    ANALYST NAME
          C*
0210        600 READ(30,601)ANALYS
0211        601 FORMAT(10X,5A4)
0212            REWIND RRUNIT
0213            GO TO 10
          C
          C     COMMENT
          C     ----------------------
          C
0214        710 READ (30,9998) COMENT
0215       9998 FORMAT (10X,15A4)
0216            REWIND RRUNIT
0217            GOTO 10
          C
          C     HED1
          C     ----------------------
          C
0218        720 READ (30,9998) HED1
0219            REWIND RRUNIT
0220            GOTO 10
          C
          C     HED2
          C     ----------------------
          C
0221        730 READ (30,9998) HED2
0222            REWIND RRUNIT
0223            GOTO 10
          C
          C     DATE
          C     ----------------------
          C
0224        740 READ (30,9998) DATE
0225            REWIND RRUNIT
0226            GO TO 10
```

158

FILE REDIF3

```
                C
                C       MAP TAPE FORMAT
                C       ---------------------
0227            C 750   M = NXTCHR(CARD2,COL)
0228                    IF(M.EQ.LU)MAPFMT=1
0229                    IF(M.EQ.LL)MAPFMT=2
0230                    IF(M.EQ.BLANK)MAPFMT=1
0231                    GO TO 10
                C
                C
                C*
                C*      PROCEDURE CONFIGURATION TITLE
                C*
0232              760   READ(30,9998)CAMS
0233                    REWIND RPUNIT
0234                    GO TO 10
                C*
                C*      ACREAGE
                C*
0235              770   M=NXTCHR(CARD2,COL)
0236                    IF(M.EQ.BLANK)GO TO 10
0237                    IF(M.EQ.COMMA)GO TO 770
0238                    J=FIND12(CARD2,COL,EQUCOM)
0239                    IF(J.EQ.2)GO TO 773
0240                    WRITE(6,772)
0241              772   FORMAT(' *ERROR IN ACREAGE CARD - CARD IGNORED')
0242                    GO TO 10
0243              773   J = FLNUM(CARD2,COL,X,1)
0244                    IF(M.EQ.LT)ATOTAL=X
0245                    IF(M.EQ.LC)ACROP=X
0246                    IF(M.EQ.LO)AOTHER=X
0247                    GO TO 770
                C*
                C*      INTENSIVE STUDY CROP NAME
                C*
0248              780   M=NXTCHR(CARD2,COL)
0249                    IC=1
0250                    KCROP=BLANK
0251                    CRPKEY=1
0252              781   LCROP(IC)=LM(1)
0253                    CROP=KCROP
0254                    IC=IC+1
0255                    IF(IC.GT.4)GO TO 10
0256                    M=NXTCHR(CARD2,COL)
0257                    IF(M.EQ.BLANK)GO TO 10
0258                    GO TO 781
                C*
                C*      DOT DATA PROCESSING
                C*
                C*
                C*      DOTFILE'S USER ASSIGNED UNIT AND/OR FILE -- DEFAULT UNIT=8,DEFAULT
                C*      FILE = 1  --  IF CLASSIFICATION NOT BY CATEGORY, TURN OFF DOT
                C*       PROCESSING .
                C*
                C        DOTFILE CARD
0259              785   CONTINUE
0260                    IF(NOCAT.EQ.0) GO TO 794
0261                    DOTKEY=1
0262                    M=NXTCHR(CARD2,COL)
0263              786   IF(M.EQ.BLANK) GO TO 794
0264                    IF(M.NE.LU)GO TO 788
0265                    M=FIND12(CARD2,COL,EQUVEC)
0266                    IF(M.NE.2)GO TO 794
0267                    ISTART=0
0268                    M=NUMBER(CARD2,COL,DOTUNT,ISTART)
0269                    M=FIND12(CARD2,COL,EQUVEC)
0270                    IF(M.NE.2) GO TO 794
0271                    ISTART=0
0272                    M=NUMBER(CARD2,COL,DOTFIL,ISTART)
0273                    DOTFIL=DOTFIL-1
0274                    IF(DOTFIL.LT.0) DOTFIL=0
0275                    GO TO 10
0276              788   IF(M.NE.LI) GO TO 789
0277                    M=FIND12(CARD2,COL,SLASH)
0278                    IF(M.NE.2) GO TO 794
0279                    M=NXTCHR(CARD2,COL)
```

FILE REDIF3

```
0280            IF(M.EQ.LE) GO TO 790
0281            IF(M.EQ.LU) GO TO 786
0282            GO TO 794
0283      789   IF(M.NE.LE) GO TO 794
0284      790   M=FIND12(CARD2,COL,EQUVEC)
0285            IF(M.NE.2) GO TO 794
0286            ISTART=0
0287            M=NUMBER(CARD2,COL,DOTFIL,ISTART)
0288            DOTFIL=DOTFIL-1
0289            M=FIND12(CARD2,COL,EQUVEC)
0290            IF(M.NE.2) GO TO 794
0291            ISTART=0
0292            M=NUMBER(CARD2,COL,DOTUNT,ISTART)
0293            GO TO 10
0294      794   DOTERR=1
0295            THRSKY=0
0296            WRITE(6,4000)
0297     4000   FORMAT(//5X,' ERROR ON DOTFIL CARD')
0298            GO TO 10
           C*
           C*
0299       800  CONTINUE
           C*
           C*   *END*   END OF CONTROL CARDS - NOW FIND CHI-SQUARE THRESHOLDS,
           C*              THEN READ IN TEST FIELDS
0300            IF (THSCNT .EQ. 0)  GO TO 830
           C*   IF THRESHOLDS WERE INPUT WITHOUT OPTION - ASSUME CHI-SQUARE
0301            IF(THRSKY.EQ.0)THRSKY=1
0302            NOFGR = NOFFT2
0303            IF (RMFLG .GT. 0) NOFGR = RMCOMB
           C-
           C-   BYPASS IF NOT EQUAL TO 1 OR 2.
           C-
0304            IF(THRSKY.GT.2) GO TO 830
           C-
0305            DO 825 MZ=1,NOSUB2
0306            HIGH(MZ) = 1. - THRES(MZ)
0307            THRES(MZ) = CHIN(HIGH(MZ),NOFGR,FLAG)
0308            IF(FLAG.EQ.1) GOTO 826
0309            GO TO 825
0310       826  WRITE(6,827) MZ,THRES(MZ)
0311       827  FORMAT(// ' *** A THRESHOLD VALUE IS OUTSIDE THE ALLOWABLE RANGE 0
               * - 1, THEREFORE'/'         NO THRESHOLDING HAS BEEN DONE IN THIS RUN
               3***' ,I5,F15.5//)
0312            THRSKY = 6
0313            GO TO 830
0314       825  CONTINUE
0315       830  CONTINUE
           C*
           C*   NOW READ TEST OR DESIGNATED FIELDS
           C*
0316            TOTVT3=0
0317            NOFLD3=1
0318            IPT=1
0319       840  ICK=LAREAD(TSTSAV(1,NOFLD3),TSTVER(IPT),INF,NV)
0320            IF(ICK.EQ.-3)GO TO 865
0321            IF(ICK.EQ.-2)GO TO 850
0322            IF(ICK.EQ.-1)GO TO 860
0323            IF(ICK.EQ. 0)GO TO 870
0324            TSTSAV(2,NOFLD3)=CLSIND
0325            TSTSAV(3,NOFLD3)=SUBIND
0326            TSTSAV(4,NOFLD3)=NV
0327            TSTFLD(1,NOFLD3)=INF(1)
0328            TSTFLD(2,NOFLD3)=INF(2)
0329            TSTFLD(3,NOFLD3)=INF(4)
0330            TSTFLD(4,NOFLD3)=INF(5)
0331            TSTFLD(5,NOFLD3)=IPT
0332            IPT=IPT + 2*NV
0333            NOFLD3 = NOFLD3 + 1
0334            TOTVT3=TOTVT3+NV
0335              GO TO 840
           C*   SUBCLASS NAME
           C*
0336       850  READ(30,851)NAME
0337            REWIND RRUNIT
0338            IF(TYPE.EQ.2)GO TO 1400
```

160

FILE REDIF3

```
0339            TYPE=1
0340        851 FORMAT(10X,A4)
0341            DO 852 I=1,NOSUB2
0342            IF(NAME.EQ.SUBNAM(I))GO TO 854
0343        852 CONTINUE
0344            WRITE(6,853)NAME
0345        853 FORMAT(' *ERROR ON SUBCLASS NAME CARD -',A4,' DOES NOT MATCH A SUB
               *CLASS FROM THE MAPTAP FILE *')
0346            CALL CYERR
0347        854 SUBIND=I
0348            CLSIND=CLSSUB(I)
0349            GO TO 840
          C*
          C*    CLASSNAME CARD
          C*
0350        860 READ(30,851)NAME
0351            REWIND BRUNIT
0352            IF(TYPE.EQ.2)GO TO 1400
0353            TYPE=1
0354            DO 861 I=1,NOCLS2
0355            IF(NAME.EQ.CLSNAM(I))GO TO 863
0356        861 CONTINUE
0357            WRITE(6,862)NAME
0358        862 FORMAT(' *ERROR ON CLASSNAME CARD -',A4,' DOES NOT MATCH A CLASS N
               *AME FROM THE MAPTAP FILE *')
0359            CALL CYERR
0360        863 CLSIND=I
0361            SUBIND=0
0362          . GO TO 840
          C*
          C*    DESIGNATED FIELDS
          C*
0363        865 READ(30,851)TEST
0364            REWIND BRUNIT
0365            IF(TYPE.EQ.1)GO TO 1400
0366            TYPE=2
0367            SUBIND=1
0368              CLSIND = NOSUB3 + 4
0369            IF(TEST.EQ.OTHER)SUBIND=2
0370            IF(TEST.EQ.OTHER)CLSIND=NOSUB3+5
          C***
          C***
          C***   THIS CODE ADDED AUG 31,1978 TO ALLOW CLASSNAME TO APPEAR
          C***   ON DESIGNATED CARD THIS FORCES RECLASSIFICATION OF
          C***   DESIGNATED PIXELS INTO THE FIRST SUBCLASS ASSIGNED
          C***   TO THE CLASS NAMED ON THE DESIG CARD STARTING AT COL 11
          C***
          C***
0371                IF (TEST.EQ.OTHER.OR.TEST.EQ.UNIDEN) GO TO 840
0372                DO 866 I = 1,60
0373                IF (TEST.NE.CLSNAM(I)) GO TO 866
0374                II = I
0375                GO TO 867
0376        866     CONTINUE
0377                WRITE(PRTUNT,4005) TEST
0378       4005     FORMAT('  DESIGNATED FIELD OF CLASSNAME',A4,'DOES NOT
               *  MATCH A CLASSNAME ON MAPTAP---DEFAULTING TO UNIDEN')
0379                GO TO 840
0380        867     DO 868 I =1,60
0381                IF (II.NE.CLSSUB(I)) GO TO 868
0382                SUBIND = 0
0383                CLSIND = I
0384                GO TO 869
0385        868     CONTINUE
0386                WRITE(PRTUNT,4006) TEST
0387       4006     FORMAT('  DESIGNATED FIELD',A4,'CANNOT BE MATCHED
               *   DEFAULTING TO UNIDENTIFIABLE')
0388        869     GO TO 840
          C*
          C*    $END$  - END OF TEST OR DESIGNATED FIELDS
          C*
0389        970 NOFLD3 = NOFLD3-1
0390            IF(TYPE.EQ.1 .AND. NOFLD3.GT.0)TSTKEY=1
          C*
          C*  NEITHER TEST NOR TRAINING FIELDS ARE ALLOWED,WITH DOT DATA
          C*    PROCESSING  --  THE PERFORMANCE TABLE , PCTAB , IS UTILIZED FOR
```

FILE REDIF3

```
      C*  DOT DATA CLASSIFICATION  PERFORMANCE SUMMARIES
      C*
           IF ( DOTKEY .GT. 0  .OR.  DOTERR .GT. 0 )  TSTKEY = 0
      C
      C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
      C
                     IF (LISTSW.EQ.0) GO TO 900
                     TSTKEY = 0
                     DOTKEY = 0
                     TRNKEY = 0
                     DOTERR = 0
        900          CONTINUE
           IF(TYPE.EQ.2 .AND. NOFLD3.GT.0)DESKEY=1
      C*
      C*   SET THRESHOLD AND OUTLINE SYMBOLS
      C*
           SYMMTX(NOSUB3)=THRSYM
           SYMMTX(NOSUB3+1)=TRNSYM
           SYMMTX(NOSUB3+2)=TSTSYM
           SYMMTX(NOSUB3+3)=DOPSYM
           SYMMTX(NOSUB3+4)=DESSYM
           SYMMTX(NOSUB3+5)=DESSYM
      C
      C    GO HOME
      C    ----------------------------------------------------------------
      C
           RETURN
      C
      C    ERROR ROUTINES
      C    ----------------------------------------------------------------
      C
       1400 WRITE(6,14000)
      14000 FORMAT(' TEST FIELDS AND DESIGNATED FIELDS CANNOT BE INPUT TOGETHE
          *R'/)
       1500 WRITE (6,15002) CODE, CARD2
      15002 FORMAT(/1X,A4,6X,62A1/' INVALID CONTROL CARD-CHECK SPELLING OF KEY
          * WORD')
           GO TO 10
           END
```

E.2 SOFTWARE SUBPROGRAM NO. 2 (SETUP3)

Subprogram SETUP3, in conjunction with REDIF3, co-ordinates the initialization function for the DISPLAY processor. In particular, it reads the MAPTAP control card, reads header information from the MAPTAP file (classification map), and reads the dot data file in the case of a P1 (not LIST) application.

E.2.1 Linkages

SETUP3 is called by subroutine DSPLAY, the driver for the DISPLAY processor. It calls REDIF3, EOD-LARSYS utility routines CMERR, NXTCHR, FIND12, WRTFLD, NUMBER, RDDOTS, FSBSFL, and standard Fortran functions.

E.2.2 Interfaces

SETUP3 has included labeled common blocks /GLOBAL/ and /DISPL/.

E.2.3 Inputs

Calling Sequence:

Keyword	Dimension	In/Out	Description
ARRAY	TOP	In/Out	blank common storage for use in buffering and data transfer
TOP	1	In	Limit of ARRAY size, currently set = 10600 in EOD-LARSYS.
LISTSW	1	In	See description of these LIST processing variables in subsection E.2.3
GTUNIT	1	In	
GTFILE	1	In	
AIUNIT	1	In	
AIFILE	1	In	

Keyword	Dimension	In/Out	Description
PPUNIT	1	In	
PPFILE	1	In	
NAMECT	1	In	
ALP	1	In	

E.2.4 Outputs

N/A

E.2.5 Storage Requirements

13234

E.2.6 Description

Subroutine SETUP3 has been expanded to incorporate the LIST processing option. If the LIST switch LISTSW has been turned on in REDIF3, all other dot-related P1 switches are turned off. The P1 dot file is not read. The calling sequence has been expanded in order to pass LIST information to DSPLY2 upon return of SETUP3 to DSPLAY.

E.2.7 Flowchart

N/A

E.2.8 Listing

```
FILE SETUP3
0001                  SUBROUTINE SETUP3(ARRAY,TOP,LISTSW,GTUNIT,GTFILE,                SET00010
                     *     ATUNIT,ATFILE,PPUNIT,PPFILE,NAMFCT,ALP)                     SET00020
0002                   IMPLICIT INTEGER (A-Z)                                         SET00030
                C                                                                     SET00040
                C***  CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING               SET00050
                C                                                                     SET00060
0003                     REAL ALP(2)                                                  SET00070
0004                  LOGICAL OKAY                                                   SET00080
                C      INCLUDE COMMKA,LIST                                            SET00090
                C      INCLUDE COMK10,LIST                                            SET00100
0005                   COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,      SET00110
                     *          HISFIL,HISKEY,THFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,    SET00120
                     *    DRUM1D,DRUM2S,PAGST7,DATFIL,STAFIL,ASAV,ASAVFL               SET00130
                     *  ,NMSTUP,NMSTFI,SCTPUN,MAPFIL                                   SET00140
                     *  ,DOTUNT,DOTFIL,UCMPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,             SET00150
                     *  CRDUNT,PRTUNT,BUNDIO                                           SET00160
0006                   COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),SET00170
                     *            SUBCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,         SET00180
                     *            PCFOKY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,  SET00190
                     *            PLTKEY,BMFLG,BMCOMS,BMFEAT,CDATE(2),               SET00200
                     *            FLDSV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,  SET00210
                     *            THRES(60),SYMBTX(66),MIGM(60),COU(60)              SET00220
                     *            ,FLDKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)            SET00230
                     *            ,NOSUB2,NOTREU,TOTVT2,NUCLS2                       SET00240
                     *          ,KATNO(60),NOCAT,FILTER,MAPFMT                       SET00250
                     *            ,DESKEY,DESUNI,DESUTH,CROP  ,ACROP,AOTHER,ATOTAL   SET00260
                     *            ,SITE(6),ANALYS(5),CAM(15),CRMKEY,KEPPTS(60)       SET00270
                     *            ,DOTKEY,DOTERR                                     SET00280
                C$END                                                                 SET00290
                C                                                                     SET00300
0007                   DATA A/'A'/,B/'NO'/                                            SET00310
0008                   DATA BLANK/'    '/                                             SET00320
                CI-------------------------------------------------------------------ISET00330
                CI-------------------------------------------------------------------ISET00340
                CI                                                                    ISET00350
                CI     PURPOSE.. LOCATES FILE ON 'MAPTAP' AND COORDINATES             ISET00360
                CI               ROUTINES TO ANALYSE 'DISPLAY' CONTROL CARDS          ISET00370
                CI                                                                    ISET00380
                CI                                                                    ISET00390
                CI-------------------------------------------------------------------ISET00400
                CC                                                                    SET00410
                C                                                                     SET00420
0009                   DIMENSION NUMVEC(30),FILVEC(2)                                 SET00430
0010                   DIMENSION CARD(62),ARRAY(1)                                    SET00440
                C*                                                                    SET00450
0011                   DIMENSION  DOTCAT(62)                                          SET00460
                C*                                                                    SET00470
0012                   EQUIVALENCE ( DOTCAT(1) , CARD(1)  )                           SET00480
                C*                                                                    SET00490
0013                   DATA  YHCD/ 'Y' /, NHCD/ 'N' /, FILVEC/ 1 , 'F' /              SET00500
                C*                                                                    SET00510
0014                    DIMENSION EQUVEC(2)                                           SET00520
0015                   DATA   DRUM/'DRUM' /                                           SET00530
0016                    DATA UHCD/'U'/,FHCD/'F'/,IHCD/'I'/,EQUVEC/1,'='/              SET00540
0017                    DIMENSION SLASH(2)                                            SET00550
0018                    DATA SLASH/1,'/'/                                             SET00560
                C*                                                                    SET00570
                C      INIZ                                                           SET00580
                C      ---------------------------------------------------------------SET00590
                CCCCC                                                                 SET00600
                CCCCC                                                                 SET00610
                CCCCC                                                                 SET00620
                CCCCC                                                                 SET00630
                CCCCC  READ FIRST CONTROL CARD FOR MAPTAP UNIT AND FILE NUMBER        SET00640
                CCCCC                                                                 SET00650
                C                                                                     SET00660
0019                     READ(CRDUNT,1) CARD                                          SET00670
0020                1    FORMAT(10X,62A1)                                             SET00680
0021                     NFILE = 1                                                    SET00690
0022                     COL = 0                                                      SET00700
0023                     J = NXTCHR(CARD,COL)                                         SET00710
0024                2    IF (J.EQ.BLANK) GO TO 6                                      SET00720
0025                     IF (J.NE.UHCD) GO TO 3                                       SET00730
0026                     J = FIND12(CARD,COL,EQUVEC)                                  SET00740
0027                     IF (J.NE.2) GO TO 6                                          SET00750
0028                     ISTART = 0                                                   SET00760
```

FILE SETUP3

```
0029                J = NUMBER(CARD,COL,MAPTAP,ISTART)                                  SET00770
0030                J = FIND12(CARD,COL,EQUVEC)                                         SET00780
0031                IF (J.NE.2) GO TO 6                                                 SET00790
0032                ISTART = 0                                                          SET00800
0033                J = NUMBER(CARD,COL,NFILE,ISTART)                                   SET00810
0034                GO TO 10                                                            SET00820
0035           3    IF (J.LE.1HCO) GO TO 4                                              SET00830
0036                J = FIND12(CARD,COL,SLASH)                                          SET00840
0037                IF (J.NE.2) GO TO 6                                                 SET00850
0038                J = MATCH(CARD,COL)                                                 SET00860
0039                IF (J.EQ.1HCO) GO TO 5                                              SET00870
0040                IF (J.EQ.1H00) GO TO 2                                              SET00880
0041                GO TO 6                                                             SET00890
0042           4    IF (J.GE.1HCO) GO TO 6                                              SET00900
0043           5    J = FIND12(CARD,COL,EQUVEC)                                         SET00910
0044                IF (J.NE.2) GO TO 6                                                 SET00920
0045                ISTART = 0                                                          SET00930
0046                J = NUMBER(CARD,COL,NFILE,ISTART)                                   SET00940
0047                J = FIND12(CARD,COL,EQUVEC)                                         SET00950
0048                IF (J.NE.2) GO TO 6                                                 SET00960
0049                ISTART = 0                                                          SET00970
0050                J = NUMBER(CARD,COL,MAPTAP,ISTART)                                  SET00980
0051                GO TO 10                                                            SET00990
0052           6    WRITE(6,7)                                                          SET01000
0053           7     FORMAT(' ERROR ON MAPTAP CARD')                                    SET01010
0054          10     CONTINUE                                                           SET01020
0055                 IF (NFILE.LE.0)  NFILE = 1                                        SET01030
0056                SERIAL = NFILE                                                      SET01040
               C                                                                       SET01050
               C     GET TAPE READY                                                    SET01060
               C                                                                       SET01070
0057          40 REWIND MAPTAP                                                         SET01080
0058             IF ( NFILE .EQ. 1 ) GO TO 50                                         SET01090
0059             NF = NFILE - 1                                                       SET01100
0060             CALL FSRSFL(MAPTAP,NF,ISTAT)                                         SET01110
0061             IF (ISTAT .EQ. 0) GO TO 50                                           SET01120
0062             WRITE(6, 45)  NF, ISTAT                                              SET01130
0063          45  FORMAT(//// 5X,'***** OSPLAY/SETUP3 ... ERROR CONDITION ON ATTEN   SET01140
                  1PT TO POSITION MAPTAP OVER',I8,3X,'FILES'// 5X,                    SET01150
                  2 '***** FSRSFL STATUS CODE =',I4,3X,'... ABORTING RUN *****'/1H1)  SET01160
0064             REWIND MAPTAP                                                        SET01170
0065             CALL CMERR                                                           SET01180
               C                                                                       SET01190
               CCC   READ  MAPTAP                                                    SET01200
               CCC   ------------------------------------------------------------    SET01210
               C                                                                       SET01220
               C                                                                       SET01230
0066          50 CONTINUE                                                           SET01240
0067             CATFLG=1                                                           SET01250
0068             READ(MAPTAP) CDATE(1),CDATE(2),NMFLG,BMCOMB,BMFEAT,NOCLS2,         SET01260
                *              NOFLD2,NOSUB2,NOFET2,TOTVT2,NUCAT,VARSZ2,           SET01270
                *             (FFTVC2(I),I=1,NOFET2)                               SET01280
0069             NCAT=NUCAT                                                         SET01290
0070             IF(NOCAT.GT.0)GO TO 55                                             SET01300
0071             CATFLG=0                                                           SET01310
0072             NCAT=NOCLS2                                                        SET01320
               C*    SET BASE ADDRESSES FOR TRAINING FIELD INFORMATION              SET01330
0073          55 CONTINUE                                                           SET01340
0074             NOSUB3=NOSUB2+1                                                    SET01350
0075             FLDSV2=1                                                           SET01360
0076             VERTX2=FLDSV2 + NOFLD2*4                                           SET01370
0077             FIFLD2=VERTX2 + TOTVT2*2                                           SET01380
0078             TOP1= FIELD2 + NOFLD2*5                                            SET01390
0079             NV=TOTVT2*2                                                        SET01400
0080             NF=NOFLD2*4                                                        SET01410
0081             READ(MAPTAP)(CATNAM(I),I=1,NCAT),(CLSNAM(I),I=1,NOCLS2),           SET01420
                *             (SUBNO(I),I=1,NOCLS2),(SUBNAM(I),I=1,NOSUB2),       SET01430
                *             (ARRAY(FLDSV2-1+I),I=1,NF),                         SET01440
                *             (ARRAY(VERTX2-1+I),I=1,NV),                         SET01450
                *             (SUBCAT(I),I=1,NOSUB2),(CLSSUB(I),I=1,NOSUB2)       SET01460
                *            ,(KATNO(I),I=1,NOCLS2),(KEPPTS(I),I=1,NOSUB2)        SET01470
               C*                                                                     SET01480
               C*    GO READ CONTROL CARDS AND TEST FIELDS                          SET01490
               C*                                                                     SET01500
0082             FLDSV3=TOP1                                                        SET01510
0083             FIFLD3= FLDSV3 + 800                                               SET01520
```

16 6

```
FILE SETUP3

                C*     RESERVE ROOM FOR 200 TEXT FIELDS                                  SET01530
 00B4                  VERTX3=FIELD3 + 1000                                              SET01540
 00B5                  VDIM = TOP - VERTX3                                               SET01550
 00B6                  CALL MEDIF3(ARRAY(FLUSV3),ARRAY(FIELD3),ARRAY(VERTX3),VDIM,       SET01560
                      *   LISTS,GTUNIT,GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,              SET01570
                      *   NAMECI,ALP)                                                   SET01580
 00B7                  TOP2=VERTX3+TOTVT3*2                                              SET01590
 00B8                  IF(DOFLD3.LE.0)TOP2=TOP1                                          SET01600
                C*     WRITE OUT SAVED TRAINING FIELDS AND TEST FIELDS                   SET01610
 00B9                  IF ( STOPFG .NE. 0 ) GOTO 200                                     SET01620
                C                                                                        SET01630
                C      PRINT OUT SUPERVISOR INFORMATION                                  SET01640
                C      ------------------------------------------------------------------SET01650
                C                                                                        SET01660
 0090                  IF(THRSKY .EQ. 1)  GO TO 80                                       SET01670
 0091                  IF(EXPTRS .EQ. 2)  GO TO 80                                       SET01680
 0092                  IF(THRSVA .EQ. 3)  GO TO 80                                       SET01690
                C-                                                                       SET01700
                C-     TEST FOR FISHER                                                   SET01710
 0093                  IF(THRSKY.EQ.4) GO TO 90                                          SET01720
                C-                                                                       SET01730
 0094                  OUTHRS = 4                                                        SET01740
 0095             80 CONTINUE                                                           SET01750
 0096                  WRITE(6,810)                                                      SET01760
                C*                                                                       SET01770
 0097                  WRITE(6,800)                                                      SET01780
 0098              800 FORMAT(15,'YOU HAVE SELECTED THE FOLLOWING OPTIONS:' / )         SET01790
                C*                                                                       SET01800
                C*     CALL TAPLAB(MUNIT , MTAPE)                                        SET01810
 0099                  MUNIT='MAPTAP                                                     SET01820
 0100                  MTAPE = 'UNIT                                                     SET01830
                C*                                                                       SET01840
 0101                  WRITE(6,801)  MTAPE , MUNIT , NFILE                               SET01850
                C*                                                                       SET01860
 0102              801 FORMAT( T10, 'PROCESS THE CLASSIFICATION RESULTS FROM MAPTAP ( ' ,SET01870
                      *  A6, ' ), UNIT' , I5, ' , FILE' , I5 )                          SET01880
 0103                  IF (OUTHRS .EQ. 4) WRITE(6,803)                                   SET01890
 0104                  IF (THRSKY .EQ. 1) WRITE(6,804)                                   SET01900
 0105                  IF(EXPTRS .EQ. 2) WRITE(6,805)                                    SET01910
 0106                  IF (THRSVA .EQ. 3) WRITE(6,816)                                   SET01920
                C-                                                                       SET01930
                C-     TEST FOR FISHER                                                   SET01940
 0107                  IF(THRSKY.EQ.4) WRITE(6,817)                                      SET01950
                C-                                                                       SET01960
                C*                                                                       SET01970
                C*                                                                       SET01980
                C*     IF DOT DATA PROCESSING WAS REQUESTED, THE DOT DATA FILE,UNIT,     SET01990
                C*     AND TAPE NUMBER WILL BE PRINTED OUT                               SET02000
                C*                                                                       SET02010
                C*** CODE ADDED NOV 13,1976 TO INCLUDE LIST PROCESSING                   SET02020
                C*                                                                       SET02030
 0108                     IF (LISTS.NE.0) GO TO 81                                      SET02040
                C*                                                                       SET02050
 0109                  IF ( DOTKEY .LE. 0 )  GO TO 81                                    SET02060
                C*     CALL TAPLAB(DOTUNIT,DTAPE)                                        SET02070
 0110                  DTAPE = DOTUNT                                                    SET02080
                C*                                                                       SET02090
 0111                   IPAT = DOTFIL + 1                                               SET02100
 0112                  WRITE(6,825) DOTUNT,IPAT,DTAPE                                    SET02110
                C*                                                                       SET02120
 0113                  IF ( TRNKEY .GT. 0 )  WRITE(6,826 )                               SET02130
                C*                                                                       SET02140
                C*     DO NOT PRINT FORMAT STMT. 806 IF DOTKEY .GT.0                     SET02150
                C*                                                                       SET02160
 0114              81       CONTINUE                                                    SET02170
                C*                                                                       SET02180
 0115                  IF (TRNKEY.EQ.1.AND.DOTKEY.LE.0) WRITE (6,806)                    SET02190
 0116                  IF ( TSTKEY  .EQ. 1 )  WRITE(6,808)                               SET02200
 0117                  IF ( STATKY  .EQ. 1 )  WRITE(6,810)                               SET02210
 0118                  IF ( PCFDKY  .EQ. 1 )  WRITE(6,812)                               SET02220
 0119                  IF(NOMAP.EQ.0)WRITE(6,820)                                        SET02230
 0120                  IF(PLTKEY.NE.0)WRITE(6,821)                                       SET02240
 0121                  IF(FILTER.EQ.1)WRITE(6,822)                                       SET02250
 0122                  IF(DESKEY.EQ.1)WRITE(6,823)                                       SET02260
 0123                  IF(CRPKEY.EQ.1)WRITE(6,824)CROP                                   SET02270
                C                                                                        SET02280
```

FILE SETUP3

```
                C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING                 SET02290
                C                                                                       SET02300
0124                      IF (LISTS*.EQ.0) GO TO 82                                     SET02310
0125                      WRITE(6,827)                                                  SET02320
0126                      WRITE(6,828)                                                  SET02330
0127                      WRITE(6,829) GTUNIT,GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE        SET02340
0128                      WRITE(6,830)  NAMECT                                          SET02350
0129                      WRITE(6,831) ALP(1),ALP(2)                                    SET02360
0130               827    FORMAT('0     LIST PROCESSING OPTION SELECTED  ')             SET02370
0131               828    FORMAT(///,5X,' NUMBERS OF GROUND TRUTH ,  AI, AND            SET02380
                     * DISCRIMINATOR UNITS AND FILES ARE AS FOLLOWS ')                  SET02390
0132               829    FORMAT(///,5X,6(2X,I5))                                       SET02400
0133               830    FORMAT(///,5X,'  SELECTED CATEGORY NAME FOR LIST IS  ',A4)    SET02410
0134               831    FORMAT(///,5X,'  BIAS CORRECTION  ALPHAS ARE  ',2F10.6)       SET02420
0135                803 FORMAT(T10,'APPLY BAYES THRESHOLDING')                          SET02430
0136                804 FORMAT(T10,'APPLY CHI SQUARE THRESHOLDS')                       SET02440
0137                805 FORMAT(T10,'APPLY EMPIRICAL THRESHOLDS')                        SET02450
0138                806 FORMAT(T10,'OUTLINE THE TRAINING FIELDS')                       SET02460
0139                807 FORMAT(T10,'OUTLINE THE TEST FIELDS')                           SET02470
0140                810 FORMAT(T10,'PRINT OUT THE STATISTICS')                          SET02480
0141                812 FORMAT(T10,'PRINT THE GROUND TRUTH PERFORMANCE SUMMARIES BY FIELD'SET02490
                      * )                                                               SET02500
                C-                                                                      SET02510
0142                817 FORMAT(T10,'APPLY FISHER F-DISTRIBUTION THRESHOLDS')            SET02520
                C-                                                                      SET02530
0143                818 FORMAT(T10, 'APPLY USER INPUT THRESHOLD VALUES')                SET02540
0144                819 FORMAT(////)                                                    SET02550
0145                820 FORMAT(T10,'DO NOT DISPLAY A CLASSIFICATION MAP')               SET02560
0146                821 FORMAT(T10,'DISPLAY THE HISTOGRAMS OF THE QUADRATIC FORM FOR ALL SSET02570
                     *UNCLASSES')                                                       SET02580
0147                822 FORMAT(T10,'PERFORM SPATIAL FILTERING')                         SET02590
0148                823 FORMAT(T10,'EXCLUDE PIXELS IN THE DESIGNATED AREAS FROM CLASSIFICASET02600
                     *TION SUMMARIES')                                                  SET02610
0149                824 FORMAT(T10,'PRINT THE INTENSIVE TEST SITE SUMMARY REPORT FOR ',A6)SET02620
0150                825  FORMAT (T10,'PRINT DOT DATA PERFORMANCE SUMMARIES FOR DOT DATA FROSET02630
                     *M FORTRAN UNIT',I3,' ,FILE NO. ',I3, ' ,TAPE (OR FILE)  ', A4)    SET02640
                C*                                                                      SET02650
0151                826  FORMAT( T10, 'OUTLINE THE DOTS ON THE CLASSIFICATION MAP' )    SET02660
                C                                                                       SET02670
0152                 82       CONTINUE                                                 SET02680
                C                                                                       SET02690
                C                                                                       SET02700
                C                                                                       SET02710
0153                      CALL WRTFLD(ARRAY(FLDSV2),ARRAY(VERTX2),NOFLD2,1,CLSNAM,SUBNAM)SET02720
0154                      IF(NOFLD3.LE.0)GO TO 85                                       SET02730
0155                      IF(TSTKEY.EQ.1)IK=2                                           SET02740
0156                      IF(DESKEY.EQ.1)IK=3                                           SET02750
0157                      CALL WRTFLD(ARRAY(FLDSV3),ARRAY(VERTX3),NOFLD3,IK,CLSNAM,SUBNAM)SET02760
0158                 85 CONTINUE                                                        SET02770
                C*                                                                      SET02780
                C*    A COMPARISON OF CATEGORY NAMES FROM MAPTAP AND RDDOTS WILL BE     SET02790
                C*    PERFORMED IF DOTKEY = 1                                         SET02800
                C                                                                       SET02810
                C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING                  SET02820
                C                                                                       SET02830
0159                      IF (LISTS*.NE.0) GO TO 87                                     SET02840
                C*                                                                      SET02850
0160                IF ( DOTKEY .LE. 0 .OR. DOTFRR .GT. 0 )  GO TO 86                   SET02860
                C*                                                                      SET02870
                C*     MAKE SPACE AVAILABLE IN  ARRAY  FOR DOT DATA INFORMATION,        SET02880
                C*     INCLUDING 1000 SCRATCH LOCATIONS FOR TEMPORARY STORAGE OF DOT    SET02890
                C*     DATA RETURNED FROM  SUBR.  RDDOTS.                              SET02900
                C*                                                                      SET02910
                C*     MOVE THE TEST/DESIG STORAGE ( TSTSAV,TSTFLD,TSTVER ) TO OVERLAY  SET02920
                C*     THE INPUT ( MAPTAP )  TRAINING FIELD STORAGE .                  SET02930
                C*                                                                      SET02940
                C*                                                                      SET02950
0161                IF ( NOCAT .LE. 0 )  GO TO 108                                      SET02960
                C*                                                                      SET02970
0162                 87       CONTINUE                                                 SET02980
0163                      NTSAV = FIELD3 - FLDSV3                                       SET02990
0164                      NTFLD = VERTX3 - FIELD3                                       SET03000
0165                      NTVER = TOTVT3 * 2                                            SET03010
0166                      NMOVE = NTSAV + NTFLD + NTVER                                 SET03020
0167                      FROMAD = FLDSV3                                               SET03030
0168                      DO  90   I=1,NMOVE                                            SET03040
```

FILE SETUP3

```
            II = I
            ARRAY(I) = ARRAY(FROMAD)
       90   FROMAD = FLDSV3 + II
      C*
      C*     RESET THE TEST/DESIG FIELDS STORAGE BASE ADDRESSES IN ARRAY
      C*
            FLDSV3 = 1
            FIFLD3 = FLDSV3 + NTSAV
            VERTX3 = FIFLD3 + NTFLD
            TOP1 = VERTX3 + NTVER
      C
      C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
                IF (LISTSW.EQ.0) GO TO 88
                FLDSV2 = TOP1
                FIFLD2 = TOP1 + 4
                VERTX2 = TOP1 + 5
                TOP2 = VERTX2 + 500
                NOFLD2 = 209
                NOTRFD = 209
                PCTID3 = TOP2
                RETURN
       88       CONTINUE
      C*
      C*     SET THE BASE ADDRESS  IN ARRAY FOR THE DOT 'TRAINING FIELDS'
      C*     STORAGE ---  ALLOW FOR A MAX. OF 250  DOTS
      C*
      C*    BEGIN TRNSAV
            FLDSV2 = TOP1
      C*    BEGIN TRNFLD
            FIFLD2=FLDSV2 + 1000
      C*    BEGIN TRNVER
            VERTX2= FIFLD2 + 1250
      C*    BEGIN SCRATCH AREA FOR DOT(4,NDOTS)
            TOP2 = VERTX2 + 500
            DIFF = TOP - TOP2
            IF (DIFF.LT.1000) GO TO 508
            TYPSWT = 2
            DO 91 I=1,62
            DOTCAT(I) = BLANK
       91   CONTINUE
      C*
      C*     TO  RDDOTS :  TYPSWT = 2
      C*
      C*     RECEIVED FROM RDDOTS/DOTFILE :
      C*  (1)  DOTS(4,NDOTS), PLACED IN ARRAY(TOP2) TO ARRAY(TOP2+ 4*NDOTS-1)
      C* WITH DOTS(1,I)=SAMPLE NO.  ,  DOTS(2,I)=LINE NO.  ,  DOTS(3,I)=TYPE ,
      C*   DOTS(4,I)=DOT CATEGORY NO.
      C*  (2) SIZES = 4  ,  (3) TOTVT2 = TOTAL NO. OF DOTS IN THE DOT FILE ,
      C*  (4) NDCAT = NO. OF DOT CATEGORIES IN THE DOT FILE ,
      C*  AND  (5)  DOTCAT(1), ... , DOTCAT(NDCAT) = DOT CATEGORY NAMES
      C*
      C*  ( DOTCAT IS EQUIVALENCED TO CARD(1) , ... , CARD(62)
      C*
      C*
            SIZES = 0
            ZERO = 0
            CALL RDDOTS(ARRAY(TOP2),DUMMY,ZERO,TYPSWT,SIZES,TOTVT2,NDCAT,
           *  DOTCAT,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,
           2  DUMMY , DUMMY )
      C*
      C*
      C*     TEST FOR THE MAX. NO. OF DOTS
      C*     SET  NOFLD2 = TOTVT2 = TOTAL NO. OF DOTS
      C*
            IF ( TOTVT2 .GT. 250 )   GO TO 1091
            NOFLD2 = TOTVT2
      C*
      C*    STORAGE ALLOCATION FOR THE DOT DATA INFORMATION
      C*
      C*    TRNSAV(1,I)  = ARRAY(FLDSV2)
      C*    TRNVER(1,I)  = ARRAY(VERTX2)
      C*    TRNFLD(1,I)  = ARRAY(FIELD2)
      C*
      C*    SUBROUTINE DTCHK WILL COMPARE THE CATEGORY NAMES. IF EACH DOT
      C*    CATEGORY NAME MATCHES WITH A CATEGORY NAME FROM MAPTAP, THE DOTS
      C*    WILL BE ORDERED BY ASCENDING LINE NUMBER, AND ASCENDING SAMPLES
```

169

FILE SETUP3

```
                C*    FOR EACH LINE NUMBER                                                    SET03810
                C*                                                                            SET03820
                C*    THE DOTS IN DOTFILE NEED TO BE CLASSIFIED BY CATEGORY                   SET03830
                C*                                                                            SET03840
                C*                                                                            SET03850
                C*                                                                            SET03860
                C* TO  DTCHK :  DOTS(4,NDOTS)  AT ARRAY(TOP2)                                 SET03870
                C*                                                                            SET03880
                C*   FROM  DTCHK :                                                          SET03890
                C*                 TRNSAV(4,NDOTS) ,BEGINNING AT ARRAY(FLDSV2) ,            SET03900
                C*  WITH  TRNSAV(1,I) = UNUSED (SCRATCH AREA)  ,                            SET03910
                C*        TRNSAV(2,I) = DOT CATEGORY NO.                                    SET03920
                C*        TRNSAV(3,I) = DOT TYPE  ( =1 OR =2 )                              SET03930
                C*        TRNSAV(4,I) = 1                                                   SET03940
                C*                                                                            SET03950
                C*                                                                            SET03960
                C*                 TRNVER(2,NDOTS) , BEGINNING AT ARRAY(VERTX2) ,           SET03970
                C*  WITH TRNVER(1,I) = DOT SAMPLE NO.  ,                                    SET03980
                C*       TRNVER(2,I) = DOT LINE NO.                                         SET03990
      0201          CONTINUE                                                                SET04000
                C*                                                                            SET04010
                C*                                                                            SET04020
      0202          IF( NDCAT .LE. 0 )   GO TO 110                                          SET04030
      0203          GO TO 500                                                               SET04040
      0204       84 CONTINUE                                                                SET04050
                C*                                                                            SET04060
                C*                                                                            SET04070
                C*    WILL PCTAB FIT IN ARRAY                                                 SET04080
      0205          PCTSZ = NOFLD2                                                          SET04090
      0206          NOTRFD = NOFLD2                                                         SET04100
      0207          GO TO 105                                                               SET04110
                C*                                                                            SET04120
                C*                                                                            SET04130
      0208       85 NOTRFD=NOFLD2                                                           SET04140
      0209          IF(TSTKEY.EQ.1)NOTRFD=NOFLD3                                            SET04150
      0210          PCTSZ=NOTRFD*NOSUB2                                                     SET04160
      0211          IF ( PCTSZ .LE. ( TOP-TOP2) )  GO TO 106                                SET04170
                C*    MOVE TEST FIELD INFO SO NO GAPS ARE IN ARRAY (IF STORAGE IS NEEDED      SET04180
      0212          IF(TSTKEY.EQ.0) GO TO 105                                               SET04190
                C*    MUST TRAILING FIELD INFO BE KEPT IN CORE                                SET04200
      0213          IAD=FLDSV2-1                                                            SET04210
      0214          IF(TRNKEY.EQ.1)IAD=TOP1-1                                               SET04220
      0215          IHD=FLDSV3-1                                                            SET04230
      0216          NF=4*NOFLD3                                                             SET04240
      0217          TIME=0                                                                  SET04250
      0218       99 DO 101 I=1,NF                                                           SET04260
      0219      101 ARRAY(IAD+I)=ARRAY(IHD+I)                                               SET04270
      0220          IF(TIME.GT.0)GO TO 102                                                  SET04280
      0221          IAD=IAD+NF                                                              SET04290
      0222          IHD=FIELD3-1                                                            SET04300
      0223          NF=5*NOFLD3                                                             SET04310
      0224          TIME=TIME+1                                                             SET04320
      0225          GO TO 99                                                                SET04330
      0226      102 IF(TIME.EQ.2)GO TO 103                                                  SET04340
      0227          IAD=IAD+NF                                                              SET04350
      0228          IHD=VERTX3-1                                                            SET04360
      0229          NF=TOTVT3*2                                                             SET04370
      0230          TIME=TIME+1                                                             SET04380
      0231          GO TO 99                                                                SET04390
      0232      103 CONTINUE                                                                SET04400
      0233          FLDSV3=TOP1                                                             SET04410
      0234          IF(TRNKEY.EQ.0)FLDSV3=1                                                 SET04420
      0235          FIELD3=FLDSV3 + 4*NOFLD3                                                SET04430
      0236          VERTX3=FIELD3 + 5*NOFLD3                                                SET04440
      0237          TOP2 = VERTX3 + 2*TOTVT3                                                SET04450
                C*                                                                            SET04460
      0238      105 IF ( PCTSZ .GT. ( TOP-TOP2) )   GO TO 508                               SET04470
                C*                                                                            SET04480
      0239      106 PCTID3 = TOP2                                                           SET04490
                C*    SET UP FIELD ARRAY FOR TRAINING FIELDS                                  SET04500
                C*                                                                            SET04510
      0240          IF( DOTERR .GT. 0 )   RETURN                                            SET04520
                C*                                                                            SET04530
      0241      107 IF(TSTKEY.EQ.0.OR.TRNKEY.EQ.1)GO TO 400                                 SET04540
      0242      401 CONTINUE                                                                SET04550
                C*                                                                            SET04560
```

FILE SETUP3

```
               C*     SET FLOKEY                                                  SET04570
               C*                                                                 SET04580
               C*                                                                 SET04590
               C*     IF DOTKEY=1,SETKEY WILL NOT BE CALLED AND THE RESETTING OF THE  SET04600
               C*     KEYS WILL TAKE PLACE                                        SET04610
               C*                                                                 SET04620
0243                  IF( DOTKEY .GT. 0 )   GO TO 201                             SET04630
0244                  IF(TSTKEY.NE.1)GO TO 340                                    SET04640
0245                  FLOKEY=1                                                    SET04650
0246                  DO 310 I=1,NOFLD3                                           SET04660
0247                  NS = FLDS3-1+3+(I-1)*4                                      SET04670
0248                  IF(ARRAY(NS).EQ.0)GO TO 320                                 SET04680
0249             310  CONTINUE                                                    SET04690
0250                  GO TO 200                                                   SET04700
0251             340  FLOKEY=1                                                    SET04710
0252                  DO 350 I=1,NOFLD2                                           SET04720
0253                  NS = FLDS2-1+3+(I-1)*4                                      SET04730
0254                  IF(ARRAY(NS).EQ.0)GO TO 320                                 SET04740
0255             350  CONTINUE                                                    SET04750
0256                  RETURN                                                      SET04760
0257             320  FLOKEY=0                                                    SET04770
0258             200  RETURN                                                      SET04780
0259             201  DOTKEY = NOCAT                                              SET04790
0260                  FLOKEY = 0                                                  SET04800
0261                  RETURN                                                      SET04810
               C                                                                  SET04820
               C      ERROR ROUTINES                                              SET04830
               C      -----------------------------------------------------------SET04840
               C                                                                  SET04850
               C                                                                  SET04860
0262             508  WRITE (6,5044) DIFF                                         SET04870
0263            5044  FORMAT (/////5X,'****** DISPLAY/SETUP3 - CORE OVERFLOW (TOP-TOP2) SET04880
                     1 BY',I6,'-- EXECUTION TERMINATED ******'/1H1)               SET04890
0264                  CALL CYERR                                                  SET04900
0265             108  WRITE (6,109)                                               SET04910
0266             109  FORMAT(/////5X,'****  CLASSIFICATION BY CATEGORY (ON MAPTAP ) IS  SET04920
                     1 REQUIRED IN ORDER TO PROCESS THE DOT DATA  ****' / 5X, '**** DOT PESET04930
                     2RFORMANCE SUMMARIES WILL NOT BE OUTPUT  ****' //// )        SET04940
               C*                                                                 SET04950
0267                  GO TO 112                                                   SET04960
               C*                                                                 SET04970
               C*  ERROR RETURN -  MAX. NO. OF DOTS EXCEEDED                      SET04980
               C*                                                                 SET04990
0268            1091  WRITE(6,1092)  TOTVT2                                       SET05000
               C*                                                                 SET05010
0269            1092  FORMAT(////5X,'***** DSPLAY/DSPLY2 ****  NO. OF DOTS =',I6,'   --- SET05020
                     * EXCEEDS THE MAX. ALLOWABLE (250)' / 5X,'*****  DOT PERFORMANCE SUMSET05030
                     *MARIES WILL NOT BE PRODUCED *****' /// )                    SET05040
               C*                                                                 SET05050
0270                  GO TO 112                                                   SET05060
               C*                                                                 SET05070
0271             110  WRITE(6,111)   NOCAT                                        SET05080
               C*                                                                 SET05090
0272             111  FORMAT(//////5X,'***** SETUP3**  FROM DOTFILE, THE NO. OF DOT CATEGSET05100
                     *ORIES =',I6 /5X,'*****  DOT PERFORMANCE SUMMARIES WILL NOT BE PROVISET05110
                     *DED  *****' ////// )                                        SET05120
               C*                                                                 SET05130
               C*     ERROR IN DOTFILE - RESET DOTKEY AND TRNKEY -- TURN ON DOTERR  SET05140
               C*                                                                 SET05150
0273             112  DOTKEY = 0                                                  SET05160
0274                  TRNKEY=0                                                    SET05170
0275                  DOTERR=1                                                    SET05180
0276                  RETURN                                                      SET05190
               C*                                                                 SET05200
               C*                                                                 SET05210
               C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*SET05220
               C*                                                                 SET05230
               C*                                                                 SET05240
               C*     INTERNAL ROUTINE TO FIND RECTANGULAR COORDINTES FOR TRAINING FIELDSET05250
               C*                                                                 SET05260
0277             400  CONTINUE                                                    SET05270
0278                  IP=1                                                        SET05280
0279                  IPT=1                                                       SET05290
0280                  DO 20 I=1,NOFLD2                                            SET05300
0281                  SAVSTR = 100000                                             SET05310
0282                  SAVEND = 0                                                  SET05320
```

171

FILE SETUP3

```
0283              LINSTR = 100000
0284              LINEND = 0
0285              NS = FLDSV2-1+4+(I-1)*4
0286              NV = ARRAY(NS)
0287              NS = FIELD2-1+5+(I-1)*5
0288              ARRAY(NS) = IPT
0289              IPT=IPT+NV*2
0290              IE=IB+NV-1
0291              DO 410 J=IB,IE
0292              NS = VERTX2-1+1+(J-1)*2
0293              SAMSTR = MIN0(SAMSTR,ARRAY(NS))
0294              SAMEND = MAX0(SAMEND,ARRAY(NS))
0295              NS = VERTX2-1+2+(J-1)*2
0296              LINSTR=MIN0(LINSTR,ARRAY(NS))
0297              LINEND=MAX0(LINEND,ARRAY(NS))
0298          410 CONTINUE
0299              NS=FIELD2-1+1+(I-1)*5
0300              ARRAY(NS)=LINSTR
0301              NS=FIELD2-1+2+(I-1)*5
0302              ARRAY(NS)=LINEND
0303              NS=FIELD2-1+3+(I-1)*5
0304              ARRAY(NS)=SAMSTR
0305              NS=FIELD2-1+4+(I-1)*5
0306              ARRAY(NS)=SAMEND
0307              IB=IE+1
0308           20 CONTINUE
0309              GO TO 401
            C*
            C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
            C*
            C*
            C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
            C*
            C*    INTERNAL ROUTINE TO CHECK THE CATEGORY NAMES AND TO ORDER THE DOTS
            C*    IF THE CATEGORY NAMES MATCH
            C*
0310          500 CONTINUE
            C*
            C*
            C*    COMPUTATION TO FIGURE OUT HOW MANY LINES ARE NEEDED TO WRITE OUT
            C*    THE DOT CATEGORY NAMES AND THE CATEGORY NAMES FROM MAPTAP
            C*
            C*  NCAT = NO. OF CLASSIFICATION (MAPTAP) CATEGORIES
            C*  NDCAT = NO. OF DOT FILE CATEGORIES
            C*
0311              NDOTS=NOFLD2
0312              NCAT=NOCAT
0313              NDLIN = ((NDCAT*7)/119) + 1
0314              NMLIN = ((NCAT*7)/119) + 1
            C*
            C*     USE TRNSAV(1,I) AS SCRATCH AREA
            C*
0315              DO 51 I=1,NDOTS
0316              NS=FLDSV2-1+1+(I-1)*4
0317              ARRAY(NS)=0
0318           51 CONTINUE
            C*
            C*
            C*    CATEGORY NAMES CHECK
            C*
0319              OKAY = .TRUE.
            C*
            C*    PRINT OUT DOT FILE INFORMATION
            C*
0320              WRITE(6,9000)NDCAT,NDOTS,(DOTCAT(I),I=1,NDCAT)
            C*
0321         9000 FORMAT(//////////2X,'DOT FILE INFORMATION :' //2X,'NO. DOT CATEGOR
                 *IES =', I6/ 2X, 'TOTAL NO. OF DOTS =' , I6/ 2X, 'DOT CATEGORY NAME
                 *S :' // ( 14A9 )  )
            C*
0322              WRITE(6,9001)
0323              DO 205 KK=1,NDOTS
0324              NS = TOP2-1+(KK-1)*4
0325              WRITE(6,9002)KK,(ARRAY(NS+K),K=1,4)
0326          205 CONTINUE
            C*
```

172

FILE SETUP3

```
0327          9001 FORMAT(//2X,'DOT RECORD :' // T10,'DOT NO.' , T22,'SAMPLE' , T32, SET06090
                  *'LINE' , T40, 'TYPE' , T49, 'CATEGORY' / T9, '----------' , T21 , SET06100
                  *  '----------' , T31, '-------' , T40, '----' , T48, '----------' // SET06110
                  *)                                                                SET06120
0328          9002 FORMAT(T12,I3,T23,I4,T32,I4,T41,I2,T51,I3)                       SET06130
0329                M = 0                                                           SET06140
0330                DO 510 I=1,NDCAT                                                SET06150
0331                DO 520 J=1,NCAT                                                 SET06160
0332                IF(DOTCAT(I).EQ.CATNAM(J))GO TO 11                              SET06170
0333             520 CONTINUE                                                       SET06180
0334                IF ( M .NE. NDCAT )  OKAY = .FALSE.                             SET06190
0335                GO TO 2101                                                      SET06200
0336              11  M=M+1                                                        SET06210
                  C*                                                              SET06220
                  C*     SETTING THE LABELED DOT CATEGORY NUMBER TO THE MAPTAP'S CATEGORY SET06230
                  C*     NUMBER THAT CORRESPONDS TO THE DOT CATEGORY NAME          SET06240
                  C*                                                              SET06250
0337                DO 515 NSET=1,NDOTS                                           SET06260
0338                NS = TOP2-1+4+(NSET-1)*4                                      SET06270
0339                NSS = FLOSV2-1+1+(NSET-1)*4                                   SET06280
0340                IF(ARRAY(NS).EQ.I)ARRAY(NSS)=J                                SET06290
0341             515 CONTINUE                                                     SET06300
0342             510 CONTINUE                                                     SET06310
                  C*                                                              SET06320
0343                DO  100  I=1,NDOTS                                            SET06330
0344                NS = FLOSV2-1+1+(I-1)*4                                       SET06340
0345                CHK=ARRAY(NS)                                                 SET06350
0346                IF ( CHK .EQ. 0 )   GO TO 100                                 SET06360
0347                NS = TOP2-1+4+(I-1)*4                                         SET06370
0348                NSS = FLOSV2-1+1+(I-1)*4                                      SET06380
0349                ARRAY(NS)=ARRAY(NSS)                                          SET06390
0350          100   CONTINUE                                                      SET06400
                  C*                                                              SET06410
                  C*     FINDING THE MAX AND MIN LINE NUMBERS                     SET06420
0351          2101  IF(.NOT.OKAY)GO TO 12                                         SET06430
                  C*                                                              SET06440
0352                MNLIN = ARRAY(TOP2-1+2)                                       SET06450
0353                MXLIN = ARRAY(TOP2-1+2)                                       SET06460
0354                DO 1000 NSET=1,NDOTS                                          SET06470
0355                NS = TOP2+1+(NSET-1)*4                                        SET06480
0356                IF(ARRAY(NS).LT.MNLIN)MNLIN=ARRAY(NS)                         SET06490
0357                IF(ARRAY(NS).GT.MXLIN)MXLIN=ARRAY(NS)                         SET06500
0358          1000  CONTINUE                                                      SET06510
0359                IF( MXLIN .LE. 0  .OR. MNLIN .LE. 0                           SET06520
                  *    .OR. MNLIN .GT. MXLIN )   OKAY = .FALSE.                   SET06530
0360                IF(.NOT.OKAY)GO TO 540                                        SET06540
                  C*                                                              SET06550
                  C*                                                              SET06560
                  C*     CHECK TO SEE WHAT KIND OF CORRESPONDENCE EXISTS BETWEEN THE TWO SET06570
                  C*     SETS OF CATEGORY NAMES                                   SET06580
                  C*                                                              SET06590
0361                NT=0                                                          SET06600
0362                IF (NL.EQ.LCAT) GO TO 13                                      SET06610
0363                WRITE (6,14) M                                                SET06620
0364             14   FORMAT(//////////////////2X,'THERE IS ',A2,' 1-1 CORRESPONDENCE BETWSET06630
                  *EEN THE CATEGORY NAMES FROM DOTFILE --')                       SET06640
0365             20   K=1                                                        SET06650
0366                  M=17                                                       SET06660
0367                  DO 26 I=1,NDLIN                                            SET06670
0368                  IF (M.GT.NDCAT) M=NDCAT                                    SET06680
0369                  WRITE(6,27)(DOTCAT(NO),NO=K,M)                             SET06690
0370             27   FORMAT(/1X,17(3X,A4))                                      SET06700
0371                  K=M+1                                                      SET06710
0372                  M=M+17                                                     SET06720
0373             26   CONTINUE                                                   SET06730
0374                  WRITE (6,30)                                               SET06740
0375             30   FORMAT (/' AND THE CATEGORY NAMES FROM MAPTAP -- ')        SET06750
0376                  K=1                                                        SET06760
0377                  M=17                                                       SET06770
0378                  DO 28 J=1,NMLIN                                            SET06780
0379                  IF (M.GT.NCAT) M=NCAT                                      SET06790
0380                  WRITE (6,27) (CATNAM(NM),NM=K,M)                           SET06800
0381                  K=M+1                                                      SET06810
0382                  M=M+17                                                     SET06820
0383             28   CONTINUE                                                   SET06830
0384                  IF (NT.EQ.1) GO TO 21                                      SET06840
```

FILE SETUP3

```
      IF (NT.EQ.2) GO TO 18
      GO TO 21
 13   WRITE (6,14) A
      NT=1
      GO TO 29
C*
C*    ORDERING THE DOT DATA
C*
 21   CNTR = 0
      DO  37  I=MNLIN,MXLIN
      M = CNTR + 1
      DO 32 J=1,NDOTS
      NS=TOP2-1+2+(J-1)*4
      IF(ARRAY(NS).NE.1)GO TO 31
      CNTR = CNTR + 1
      NS = VERT(2-1+2+(CNTR-1)*2
      ARRAY(NS)=I
      NS = FLDSV2-1+4+(CNTR-1)*4
      ARRAY(NS)=1
      NS = VERT(2-1+1+(CNTR-1)*2
      NSS = TOP2+(J-1)*4
      ARRAY(NS) = ARRAY(NSS)
      NS = FLDSV2+2+(CNTR-1)*4
      NSS = TOP2+2+(J-1)*4
      ARRAY(NS )=ARRAY(NSS)
      NS = FLDSV2+1+(CNTR-1)*4
      NSS = TOP2-1+4+(J-1)*4
      ARRAY(NS)=ARRAY(NSS)
   31 CONTINUE
   32 CONTINUE
C*
 37   CONTINUE
C*
      GO TO 84
 12   WRITE (6,25)
 25   FORMAT (1H1//5X,'*** DOT DATA PERFORMANCE SUMMARIES WILL NOT BE
     1 PRODUCED - THE CATEGORY NAMES FROM MAPTAP AND DOTFILE DO NOT MATC
     2H'//2X,'CATEGORY NAMES FROM DOTFILE ARE -')
      NT=2
      GO TO 29
C*
 540  WRITE(6,41)MNLIN,MXLIN
C*
 41   FORMAT(//////5X,'**** SETUP3/DTCHK  ---  FROM DOTFILE, THE MIN. LI
     *NE NO. =',I6,'  ,MAX. LINE NO.=',I6/ 5X,'*****  DOT PERFORMANCE SU
     *MMARIES WILL NOT BE PRODUCED  *****' //////)
 18   DOTKEY = 0
C*
      TRNKEY=0
      DOTERR=1
C*
      GO TO 64
      END
```

E.3 SOFTWARE SUBPROGRAM NO. 3 (DSPLY2)

Subprogram DSPLY2 performs the bulk of the summary information for the DISPLAY processor. It proceeds on a field-by-field basis, using MAPTAP machine classification information. It writes a formatted display map (Universal or LARSYS II).

E.3.1 Linkages

Subprogram DSPLY2 is called by the DISPLAY processor driver DSPLAY. It calls LISTSM, MAPHD, DESIG, PCT, FLDBOR, and PRTSUM, and EOD-LARSYS utility routines RWRITE, SETMRG, WRTHED, WRTLN, BNI4A1, RREAD.

E.3.2 Interfaces

DSPLY2 has included labeled common blocks /GLOBAL/ and /DISPL/.

E.3.3 Inputs

Calling Sequence:

Keyword	Dimension	In/Out	Description
TRNSAV	(4, NOFLD2)	In	Generally, used for dot storage in P1 runs, not used except for extraction of machine classification information in LIST processing.
TRNFLD	(5, NOFLD2)	In	
TRNVER	(2, TOTVT2)	In	
TSTSAV	(4, NOFLD3)	In	In LIST processing DO/DU field information (otherwise, TEST or DO/DU field information).
TSTFLD	(5, NOFLD3)	In	

Keyword	Dimension	In/Out	Description
TSTVER	(2, TOTVT3)	In	
PCTAB	(NOTRFD, 1)	Out	Subclass number of machine classification for 209 dots
LISTSW	1	In	For description of these LIST variables, see subsection E.2.3
GTUNIT	1	In	
GTFILE	1	In	
AIUNIT	1	In	
AIFILE	1	In	
PPUNIT	1	In	
PPFILE	1	In	
NAMECT	1	In	
ALP	2	In	

E.3.4 Outputs

A Universal or LARSYS II formatted classification display map may be produced.

E.3.5 Storage Requirements

68206

E.3.6 Description

Subprogram DSPLY2 has been expanded to accomodate LIST processing. If LISTSW is on, an extraction is made of the machine classification results for the entire 209 dot grid. This extraction is done after modification of classification results by DO/DU input fields and/or thresholding, both regular options of EOD-LARSYS. Machine

classification results are stored in PCTAB. Within the major field loop, a call is made to LISTSM for CCIT report generation and proportion estimation using Type 2 dots from any or all of ground truth, PPTC, and AI dot files.

E.3.7 Flowchart

N/A

E.3.8 Listing

```
FILE DSPLY2
0001            SUBROUTINE DSPLY2(TRNSAV,TRNFLD,TRNVER,TSTSAV,TSTFLD,          DSP00010
               *TSTVER,PCTAB,LISTSW,GTUNIT,GTFILE,AIUNIT,AIFILE,               DSP00020
               *PPUNIT,PPFILE,NAMECT,ALP)                                      DSP00030
0002                 IMPLICIT INTEGER(A-Z)                                     DSP00040
C                                                                              DSP00050
C***  CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING                        DSP00060
C                                                                              DSP00070
0003            REAL ALP(2)                                                    DSP00080
CI-------------------------------------------------------------------------IDSP00090
CI-------------------------------------------------------------------------IDSP00100
CI                                                                         IDSP00110
CI     DSPLY2 CAN  PERFORM SPATIAL FILTERING ON THE CLASSIFIED DATA.       DSP00120
CI        (I.E. - THE FOUR NEAREST NEIGHBORS OF EACH PIXEL ARE             DSP00130
CI                TESTED FOR 'SAMENESS'. IF THE FOUR NEIGHBORS ARE         DSP00140
CI                CLASSIFIED AS ONE TYPE AND THE PIXEL IN QUESTION         DSP00150
CI                WAS CLASSIFIED AS A DIFFERENT TYPE, THE CLASSIFICATION   DSP00160
CI                IS CHANGED TO THE SAME AS THE NEIGHBORS.)                DSP00170
CI                EXAMPLE:                                                 DSP00180
CI                   LINE N   -     C     -                                DSP00190
CI                        N+1 -    CXC    -  X IS CHANGED TO C             DSP00200
CI                        N+2 -     C     -                                DSP00210
CI                                                                         DSP00220
CI     DSPLY2 ALSO PERFORMS THRESHOLDING, PRINTS THE CLASSIFICATION        DSP00230
CI     MAP, AND CALLS THE APPROPIATE ROUTINES TO BUILD AND THEN PRINT      DSP00240
CI     THE CLASSIFICATION PERFORMANCE TABLES.                              DSP00250
C*                                                                         DSP00260
C*     THE DOT DATA PERFORMANCE SUMMARIES ARE PROVIDED FROM THE INTERNAL   DSP00270
C*     SUBROUTINE,DOTSUM                                                   DSP00280
C*                                                                         DSP00290
C*     LIST PROCESSING ADDED NOV 13, 1978 LISTSW IS THE                    DSP00300
C*     NAME OF THE SWITCH                                                  DSP00310
CI                                                                         DSP00320
CI                                                                         IDSP00330
CI-------------------------------------------------------------------------IDSP00340
CI-------------------------------------------------------------------------IDSP00350
C                                                                              DSP00360
C                                                                              DSP00370
0004            REAL CON                                                       DSP00380
0005            DIMENSION TRNSAV(4,NOFLD2),TRNFLD(5,NOFLD2),TRNVER(2,TOTVT2)   DSP00390
               *          ,TSTSAV(4,NOFLD3),TSTFLD(5,NOFLD3),TSTVER(2,TOTVT3)  DSP00400
               *     ,COL(3,110),SCRAT(330),FLDINF(6)                         DSP00410
               *          ,IR(1000,3),VR(1000),OUT(1000),ILINE(3),BUF(110,20)  DSP00420
               *          ,JSTAT(20),PCTAB(NOTRFD,NOSUB3)                      DSP00430
               2, CLTOT(61) , DOTLBL(61,2),PUNC(21)                            DSP00440
               3 , LAMCLS(60,61), LBLTOT(61), TOTLBL(60), PCORR(61) ,          DSP00450
               4  ALPHA(60,60), TOTTHR(60),DOTORD(61,2)                        DSP00460
C      INCLUDE CMBK10,LIST                                                     DSP00470
C      INCLUDE COMBK6                                                          DSP00480
0006           COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,HMFILE,BMKEY,       DSP00490
               *             HISFIL,HISKEY,TRFORM,FRIPTP,FRPKEY,MAPUNT,NOFILE,DSP00500
               *    DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL             DSP00510
               *  ,NHSTOU,NHSTFI,SCTRUN,MAPFIL                                 DSP00520
               *  ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,           DSP00530
               *  CRDUNT,PRTUNT,RANDIO                                         DSP00540
0007           COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),  DSP00550
               *             SUBCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,       DSP00560
               *             PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTHS,THRSVA,DSP00570
               *             PLTKEY,BMFLG,BMCONR,BMFEAT,CDATE(2),             DSP00580
               *             FLDSV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,DSP00590
               *             THRES(60),SYMTX(66),HIGH(60),COM(60)             DSP00600
               *             ,FLDKEY,NOFLD2,NOFLD3,NOFFT2,FFTVC2(30)          DSP00610
               *             ,NOSUB2,NOTRFD,TOTVT2,NUCLS2                     DSP00620
               *             ,KATNO(60),NOCAT,FILTER,MAPFMT                   DSP00630
               *             ,DESKEY,DESUNI,DESOTH,CROP  ,ACROP,AOTHER,ATOTAL DSP00640
               *             ,SITE(6),ANALYS(5),CAM(15),CMPKEY,KEPPTS(60)     DSP00650
               *             ,DOTKEY,DOTERR                                   DSP00660
C      END                                                                     DSP00670
0008           LOGICAL START,FULL                                              DSP00680
0009           DATA AST/'*****'/                                               DSP00690
0010           DATA THRESH/'THRE'/,BLANK/'    '/                               DSP00700
0011           EQUIVALENCE      (FLDINF(1), LINSTR), (FLDINF(2),LINEND),       DSP00710
              1                 (FLDINF(3),LININC), (FLDINF(4), SAMSTR),       DSP00720
              2                 (FLDINF(5),SAMEND),(FLDINF(6),SAMINC),         DSP00730
              3                 (COL,SCRAT),(IR,HUF)                           DSP00740
C-------------------------------------------------------------------------     DSP00750
C      FOLLOWING EQUIVALENCED ARRAYS WILL ONLY BE ACCESSED IN INTERNAL          DSP00760
```

FILE DSPLY2

```
                C          SUBROUTINE DOTSUM
  0012                 EQUIVALENCE (CLTOT(1),IR(1,1)),(PUNC(1),IR(62,1)),
                      *(POTLHL(1,1),IR(124,1)),
                      *  ( PCORR(1) , IR(246,1) ) , ( TOTLHL(1) , IR(307,1) ) ,
                      *    ( TOTTHR(1) , IR(367,1) ) , ( LHLTOT(1) , IR(427,1) )  ,
                      *    ( DATORD(1,1) , IR(488,1) )
  0013                 REAL ALPHA,PCORR,PUNC
  0014                  DIMENSION FMT1(19),FMT2(24),FMT3(8),FMT4(8),FMT5(9),TEMP(5,3)
                C      DATA FMT1/'(T9,','      ','4    ','(8HC','ATEG','ORY ','25X)','/T2,',
                C     *  '    ','4    ','(26H','----','----','----','----','----','----',
                C     *    '--,7','X)) '/
  0015                 DATA FMT2/'(T2,','      ','4    ','(26H','LAHE','LED ',' CLA','SSIF',
                      *    'IED(','SUBC','L),/','X)/T','2,  ','4    ','(7H-','----','--,2',
                      *    'X,17','H---','----','----','----','--,7','X)) '/
  0016                 DATA FMT3/'(T40','      ','3    ','(1H(','F7.','3,1H',')',4X','))/) '/
  0017                 DATA FMT4/'(/T2','4,   ','8    ','(1H(','F7.','3,1H',')',4X','))  '/
                C      DATA FMT5/'(//T','8,   ','4    ','(1H(','I3,','IH,','I3,I','H),2',
                C     *    '4)))'/
  0018                 DIMENSION BLANKS(4)
  0019                 DATA BLANKS/'    ','    ','    ','    '/
  0020                  REAL PCC,PCTOT,PIXTOT
                C-----------------------------------------------------------------------
                C
                C
                C      -----------------------------------------------------------------
                C
                C
  0021                 DIMENSION FORMAT(3,2)
  0022                 DATA FORMAT/'UNIV','ERSA','L   ','LARS','YS I','I   '/
  0023                 REAL      TOTALS(66),VR,THRES
  0024                 DIMENSION TTOL(66)
                C
                C      -----------------------------------------------------------------
                C      -----------------------------------------------------------------
                C
  0025                 TRUNO=NOSUB3+1
  0026                 TSTNO=NOSUB3+2
                C*
                C*     SET POINTERS FOR SYMBOLS ARRAY AND TOTALS ARRAY
                C*
  0027                 DUPNO = NOSUB3+3
  0028                 DESUNI= NOSUB3+4
  0029                 DESOTH= NOSUB3+5
                C*
                C*     FLAG USED IN DOTPCT TO INITIALIZE PCTAB=0
                C*
  0030                 PCTKEY=0
                C
                C***   CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
                C
  0031                         FLDCNT = 0
  0032                         IF (LISTSW.EQ.0) GO TO 17
  0033                         DO 16 I = 1,209
  0034                         LR = (I - 1)/19
  0035                         LR = (LR + 1)*10
  0036                         LS = (LR - 1)/10
  0037                         LS = 10*(I - (LS*19))
  0038                         TRNVER(1,I) = LS
  0039                         TRNVER(2,I) = LR
  0040           16            CONTINUE
  0041                         GO TO 20
  0042           17            CONTINUE
  0043                 IF ( DOTKEY .GT. 0  .OR.  DOTERR .GT. 0 )  GO TO 11
  0044                 DO 10 I=1,NOSUB3
  0045                 DO 10 J=1,NOTPFD
  0046              10 PCTAB(J,I) = 0
                C*
  0047                 GO TO 20
                C*
  0048           11    IF( DOTERR .GT. 0 )  GO TO 20
                C*
                C*     INITIALIZE  TRNSAV FOR OUTLINING DOTS IN SUBR. FLDBOR
                C*
  0049                 DO  12  I=1,NOFLD2
  0050           12    TRNSAV(4,I) = 2
                C*
```

FILE 'SPLY2

```
                C*                                                                      DSP01530
                C                                                                       DSP01540
                C                                                                       DSP01550
                C       PRINT OUT HEADING                                               DSP01560
                C       ---------------------------------------------------------------DSP01570
                C                                                                       DSP01580
0051                20 READ(MAPTAP)FLDINF,PTS,LINES,FLDESC                              DSP01590
0052                   IF (PTS .GT. 1000) WRITE(6,22)                                   DSP01600
0053                   IF (PTS .GT. 1000) STOP                                          DSP01610
0054                22 FORMAT(' DISPLAY WILL ACCEPT ONLY 1000 PTS/SCAN LINE')           DSP01620
0055                   ISTRT=SAMSTR                                                     DSP01630
0056                   IEND=SAMFND                                                      DSP01640
0057                   IF (PTS.EQ.0) GO TO 310                                          DSP01650
0058                   NOREC = 1                                                        DSP01660
0059                   DO  25  I=1,NDESOTH                                              DSP01670
0060                   TTOL(I)=0                                                        DSP01680
0061                25 TOTALS(I)=0.0                                                    DSP01690
                                                                                        DSP01700
                C       PRINT OUT THE COLUMN NUMBERS                                    DSP01710
                C       ---------------------------------------------------------------DSP01720
                C                                                                       DSP01730
0062                30 J = 0                                                            DSP01740
0063                   CALL SETMRG(68,0,68)                                             DSP01750
0064                   IF(NOMAP.EQ.0)GO TO 85                                           DSP01760
0065                    PRTFG=1                                                         DSP01770
0066                    GO TO 370                                                       DSP01780
0067             31     PRTFG=1                                                         DSP01790
0068                    GO TO 510                                                       DSP01800
0069                85 CONTINUE                                                         DSP01810
0070                   SPKNT=0                                                          DSP01820
0071                   COUNTR=0                                                         DSP01830
0072                   NUMLN=0                                                          DSP01840
0073                   LAST=0                                                           DSP01850
0074                   START=.FALSE.                                                    DSP01860
0075                   FULL =.FALSE.                                                    DSP01870
0076                   I1=1                                                             DSP01880
0077                   I2=2                                                             DSP01890
0078                   I3=3                                                             DSP01900
0079                   J=1                                                              DSP01910
0080                   ADRES=DRUMAD                                                     DSP01920
0081                91 READ(MAPTAP) ILINE(J),(IR(I,J),I=1,PTS),(VR(I),I=1,PTS)         DSP01930
0082                   IF(ILINE(J).EQ.0)GO TO 105                                       DSP01940
                C                                                                       DSP01950
                C*      IF DESIGNATED FIELDS HAVE BEEN INPUT, SET THE IR ARRAY FOR PIXELS DSP01960
                C*      IN THOSE FIELDS.                                                DSP01970
0083                   IF(DESKEY.EQ.1)CALL DESIG(ILINE(J),IR(1,J),TSTSAV,TSTFLD,TSTVER,DSP01980
                     *                    NOFLDS,SAMSTR,SAMFND,SAMINC)                  DSP01990
0084                   DO 100 I=1,PTS                                                   DSP02000
0085                   L = IR(I,J)                                                      DSP02010
0086                   IF(L.EQ.0)GO TO 100                                              DSP02020
0087                   IF(L.EQ.NOSUB3)GO TO 92                                          DSP02030
0088                   IF(L.GT.NOSUB3)GO TO 95                                          DSP02040
                C                                                                       DSP02050
                C*      THRESHOLDING                                                    DSP02060
                C*                                                                      DSP02070
                C*      THE VALUE OF THE QUADRATIC FORM MUST BE EXTRACTED FROM VR FOR THRDSP02080
                C*                                                                      DSP02090
                C*              Q = -2*VR - CON                                         DSP02100
                C*                                                                      DSP02110
                C*                                                                      DSP02120
0089                   IF(THRSKY.EQ.0)GO TO 95                                          DSP02130
0090                   IF((-2.*VR(I)-CON(L)).LT. THRES(L))GO TO 95                      DSP02140
0091                92 CONTINUE                                                         DSP02150
                C                                                                       DSP02160
                C*      TTOL = TOTAL PIXELS THRESHOLDED, BY SUBCLASS                    DSP02170
                C*                                                                      DSP02180
0092                   TTOL(L) = TTOL(L) + 1                                            DSP02190
0093                   IR(I,J) = NOSUB3                                                 DSP02200
0094                   L = NOSUB3                                                       DSP02210
                C                                                                       DSP02220
                C*     TOTALS = TOTAL NO. PIXELS CLASSIFIED INTO EACH SUBCLASS,         DSP02230
                C*      INCLUDING THRESHOLDED AND DESIGNATED OTHER , DESIGNATED UNIDENT DSP02240
                C*                                                                      DSP02250
0095                95  TOTALS(L) = TOTALS(L) + 1                                      DSP02260
0096               100 CONTINUE                                                         DSP02270
                C*                                                                      DSP02280
```

FILE DSPLY2

```
                C*     HAVE 3 LINES BEEN READ
                C*
 0097                  IF(START)GO TO 105
 0098                  J=J+1
 0099                  IF(J.LT.3)GO TO 91
 0100                  START=.TRUE.
 0101                  GO TO 91
                C*
                C*     SPATIAL FILTERING
                C*
 0102             105  IF(FILTER.EQ.0)GO TO 115
 0103                  I=2
 0104             106  IF(IR(I-1,I2) .NE. IR(I+1,I2)) GO TO 110
 0105                  IF(IR(I,I1)    .NE. IR(I,I3))    GO TO 110
 0106                  IF(IR(I,I1)    .NE. IR(I-1,I2))  GO TO 110
 0107                  IF(IR(I,I2) .EQ. IR(I,I1))GO TO 110
 0108                  IF(IR(I,I1) .EQ. NOSUB3)GO TO 110
 0109                  ICC=IR(I,I1)
 0110                  ICK=IR(I,I2)
 0111                  IF(ICK.EQ.NOSUB3)GO TO 110
 0112                  TOTALS(ICC)=TOTALS(ICC)+1
 0113                  TOTALS(ICK)=TOTALS(ICK)-1
 0114                  SPKNT=SPKNT+1
 0115                  IR(I,I2) = IR(I,I1)
 0116             110  I=I+1
 0117                  IF(I.LE.PTS-1)GO TO 106
                C*
                C*     GET PERFORMANCE FOR LINE I1
                C*
                C
                C***   CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
                C
 0118            115            IF (LISTSW.NE.0) GO TO 116
 0119                  IF (DOTERR.EQ.1) GO TO 114
 0120                  IF (DOTKEY.EQ.0) GO TO 40
 0121            116          CONTINUE
                C*
                C*     TEST TO SEE IF THE CURRENT LINE CONTAINS ANY DOTS
                C*
 0122                  IF (ILINE(I1).GT.TRNVER(2,NOFLD2)) GO TO 114
 0123                  BCNT = 0
 0124                  ECNT = 0
 0125                  DO 41 I=1,NOFLD2
 0126                  IF (ILINE(I1).NE.TRNVER(2,I)) GO TO 42
 0127                  IF (BCNT.EQ.0) BCNT = I
 0128                  GO TO 41
 0129             42   IF (BCNT.EQ.0) GO TO 41
 0130                  ECNT = I-1
 0131                  GO TO 43
 0132             41   CONTINUE
 0133                  IF (ILINE(I1).NE.TRNVER(2,NOFLD2)) GO TO 114
 0134                  ECNT = NOFLD2
                C*
                C*     FOR DOT DATA PROCESSING, CALL THE INTERNAL SUBROUTINE DOTPCT TO
                C*     BUILD THE CLASSIFICATION PERFORMANCE TABLE (PCTAB)
                C*
 0135             43    GO TO 432
 0136             40   IF (TSTKEY.EQ.0)CALL PCT(ILINE(I1),IR(1,I1),TRNFLD,TRNVER,
                     *                       TRNSAV,PCTAB,NOFLD2,SAMSTR,SAMEND,SAMINC)
 0137                  IF(TSTKEY.EQ.1)CALL PCT(ILINE(I1),IR(1,I1),TSTFLD,TSTVER,
                     *                       TSTSAV,PCTAB,NOFLD3,SAMSTR,SAMEND,SAMINC)
 0138            114   IF (NOMAP.EQ.0) GO TO 135
                C*
                C
                C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
                C
 0139                          IF (LISTSW.NE.0) GO TO 117
 0140                          IF (DOTERR.GT.0) GO TO 117
                C*
                C*
                C*     OUTLINE TRAINING AND/OR TEST FIELDS
                C*
 0141                  IF(TRNKEY.EQ.1)CALL FLDBOR(TRNNO,ILINE(I1),IR(1,I1),NOFLD2,
                     *            TRNFLD,TRNSAV,TRNVER,NOSUB3,SAMSTR,SAMEND,
                     *            SAMINC,LININC)
 0142                  IF(TSTKEY.EQ.1)CALL FLDBOR(TSTNO,ILINE(I1),IR(1,I1),NOFLD3,
```

FILE DSPLY2

```
                     *              TSTFLD,TSTSAV,TSTVER,NOSUR3,SAMSTR,SAMEND,          DSP03050
                     *              SAMINC,LININC)                                      DSP03060
                C*    SET UP SYMBOLS FOR THIS LINE. FIRST MAKE SURE I/O FROM LAST LINE  DSP03070
                C*    IS COMPLETED.                                                     DSP03080
     0143        117      CONTINUE                                                      DSP03090
     0144                 DO 120 I=1,PTS                                               DSP03100
     0145                 L = IN(I,I1)                                                 DSP03110
     0146                 IF(L.EQ.0)OUT(I)=BLANK                                       DSP03120
     0147                 IF(L.NE.0)OUT(I)=SYMMTX(L)                                   DSP03130
     0148       120 CONTINUE                                                           DSP03140
                C*                                                                     DSP03150
                C*    WRITE FIRST 110 SAMPLES ON LINE PRINTER AND THE REST ON DRUM     DSP03160
                C*                                                                     DSP03170
     0149                 IPTS=PTS                                                     DSP03180
     0150                 IF(IPTS.GT.110)IPTS=110                                      DSP03190
     0151                 IF(PTS.LE.110)GO TO 125                                      DSP03200
     0152                 IPD=PTS-110                                                  DSP03210
     0153                 IF(FULL)GO TO 125                                            DSP03220
     0154                 CALL WRITE(ADDRS,OUT(111),IPD,LSTAT)                         DSP03230
     0155                 ADDRS=ADDRS+IPD                                              DSP03240
     0156                 DRUMLN=DRUMLN+1                                              DSP03250
     0157                 IF(ADDRS+IPD .LE. DRUMAD+DRMWDS)GO TO 125                    DSP03260
     0158                 FULL=.TRUE.                                                  DSP03270
     0159       125 WRITE(6,240)(LINE(I1),(OUT(I),I=1,IPTS)                          DSP03280
                C*                                                                     DSP03290
                C*    IS CLASSIFICATION MAP TO BE OUTPUT IN UNIVERSAL OR LARSYS        DSP03300
                C*    FORMAT                                                           DSP03310
                C*                                                                     DSP03320
     0160        135 IF (MAPFMT .LE. 0)  GO TO 200                                    DSP03330
                C*                                                                     DSP03340
                C*    CHECK TO SEE IF LAST WRITE IS COMPLETED                          DSP03350
                C*                                                                     DSP03360
     0161                 GO TO (155,160), NOREC                                       DSP03370
                C*                                                                     DSP03380
                C*    WRITE HEADER RECORD                                              DSP03390
                C*                                                                     DSP03400
     0162        155 NC = 1                                                           DSP03410
     0163                 LNES = 0                                                     DSP03420
     0164                 FEAT = 1                                                     DSP03430
     0165                 LSTLIN = 0                                                   DSP03440
     0166                 NOREC = 2                                                    DSP03450
     0167                 NOFILE = NOFILE + 1                                          DSP03460
     0168                 CALL WRTHED(NC,FEAT,PTS,MAPFMT,MAPUNT)                       DSP03470
                C*                                                                     DSP03480
                C*    WRITE DATA RECORD                                                DSP03490
                C*                                                                     DSP03500
     0169        160 LNES = LNES + 1                                                  DSP03510
     0170                 IF (LNES .EQ. LINES)  LSTLIN = -1                            DSP03520
     0171                 CALL WRTLN(IP(1,I1),LSTLIN)                                  DSP03530
     0172       200 CONTINUE                                                          DSP03540
     0173                 IF(ILINE(I3).EQ.0)GO TO 201                                  DSP03550
                C*                                                                     DSP03560
                C*    SET INDICES AND GO READ NEXT LINE                                DSP03570
                C*                                                                     DSP03580
     0174                 J=I1                                                         DSP03590
     0175                 I1=I2                                                        DSP03600
     0176                 I2=I3                                                        DSP03610
     0177                 I3=J                                                         DSP03620
     0178                 GO TO 91                                                     DSP03630
                C*                                                                     DSP03640
                C*    LAST LINE IN THIS FIELD HAS BEEN READ. MAKE SURE LAST 2 LINES    DSP03650
                C*    ARE PRINTED.                                                     DSP03660
                C*                                                                     DSP03670
     0179       201 LAST=LAST+1                                                       DSP03680
     0180                 IF(LAST.EQ.2)GO TO 203                                       DSP03690
     0181                 I1=I2                                                        DSP03700
     0182                 I2=I3                                                        DSP03710
     0183                 GO TO 115                                                    DSP03720
                C*                                                                     DSP03730
                C*    NOW FINISH PRINTING MAP FOR THIS FIELD.                          DSP03740
                C*                                                                     DSP03750
     0184       203 CONTINUE                                                          DSP03760
     0185                 IF (MAPFMT .GT. 0) WRITE(6,2200)NOFILE,FLDESC,(FORMAT(I,MAPFMT),  DSP03770
                    *      I=1,3),LNES                                                DSP03780
     0186        2200 FORMAT(///T55,'FILE NO.    - ',I6,/T55,'FIELD NAME  - ',A4,/    DSP03790
                     ,  T55,'FORMAT      - ',3A4,/T55,'NO. RECORDS - ',I6)              DSP03800
```


182

```
FILE DSPLY2
0187            IF (NOMAP .LE. 0 ) GO TO 230                                    DSP03810
                C                                                               DSP03820
0188            IF(PTS.LE.110)GO TO 230                                         DSP03830
0189            AD=0                                                            DSP03840
0190            NRUFS=20                                                        DSP03850
0191            IF(NRUFS.GT.LINES)NRUFS=LINES                                   DSP03860
0192            KPTS=PTS-110                                                    DSP03870
0193            LPTS=110                                                        DSP03880
0194            NWDTHS=KPTS/110                                                 DSP03890
0195            IF(MOD(KPTS,110).NE.0)NWDTHS=NWDTHS+1                           DSP03900
0196            LASTRC=MOD(KPTS,110)                                            DSP03910
0197            IF(LASTRC.EQ.0)LASTRC=110                                       DSP03920
0198            I=0                                                             DSP03930
0199     219    I=I+1                                                           DSP03940
0200            IF (I.GT.NWDTHS) GO TO 220                                      DSP03950
0201            LINCNT=0                                                        DSP03960
0202              WRITE=2                                                       DSP03970
0203             GO TO 370                                                      DSP03980
0204     221     PRTFG=2                                                       DSP03990
0205              GO TO 510                                                     DSP04000
0206     222     CONTINUE                                                      DSP04010
0207            IF(I.EQ.NWDTHS)LPTS=LASTRC                                      DSP04020
0208            ADRES=ORIMAD + AD                                               DSP04030
0209            DO 205 J=1,NRUFS                                                DSP04040
0210            CALL READ(ADRES,BUF(1,J),LPTS,JSTAT(J))                         DSP04050
0211            ADRES=ADRES+KPTS                                                DSP04060
0212            LINCNT=LINCNT + 1                                               DSP04070
0213     205    CONTINUE                                                        DSP04080
0214            LINE=LINSTR                                                     DSP04090
0215            IBUF=1                                                          DSP04100
                C*      FINISHED READING                                        DSP04110
0216     210    IF(JSTAT(IBUF).EQ.1)GO TO 210                                   DSP04120
0217            WRITE(6,240)LINE,(BUF(IK,IBUF),IK=1,LPTS)                       DSP04130
0218            LINE = LINE + LININC                                            DSP04140
0219            LINCNT=LINCNT+1                                                 DSP04150
0220            IF(LINCNT .GT. 60)GO TO 215                                     DSP04160
0221            CALL READ(ADRES,BUF(1,IBUF),LPTS,JSTAT(IBUF))                   DSP04170
0222            ADRES=ADRES + KPTS                                              DSP04180
0223     215    IBUF=IBUF+1                                                     DSP04190
0224            IF(IBUF.GT.NRUFS)IBUF=1                                         DSP04200
0225            IF(LINE.LE.LINEND)GO TO 210                                     DSP04210
0226            AD=AD + LPTS                                                    DSP04220
0227            GO TO 214                                                       DSP04230
0228     220    CONTINUE                                                        DSP04240
0229     240    FORMAT(' ',I5,2X,110A1)                                         DSP04250
0230            IF(FILTER.EQ.0)GO TO 230                                        DSP04260
0231            WRITE(6,305)SPKNT                                               DSP04270
0232     305    FORMAT(//' THE CLASSIFICATION OF',I7,' PIXELS WAS CHANGED AS A RESUDSP04280
                *LT OF SPATIAL FILTERING'/)                                     DSP04290
0233     230    CONTINUE                                                        DSP04300
0234     250    CONTINUE                                                        DSP04310
                C*                                                              DSP04320
                C*      PRINT CLASSIFICATION SUMMARY FOR THIS FIELD             DSP04330
                C*                                                              DSP04340
0235            CALL SETMRG(68,4,62)                                            DSP04350
0236            CALL PRTSUM(TOTALS,TTOL,FLDESC)                                 DSP04360
0237                  IF (LISTS*.NE.0) GO TO 255                                DSP04370
0238            IF ( DOTKEY .EQ. 0  .OR.  DOTERR .GT. 0 )  GO TO 500            DSP04380
                C*                                                              DSP04390
                C*      DOTS IN THE DESIGNATED AREA OR NOT IN THE CLASSIFIED AREA WILL  DSP04400
                C*      HAVE PCTAB=0                                            DSP04410
                C                                                               DSP04420
                C*** CODE ADDED NOV 13 ,1978 TO INCLUDE LIST PROCESSING         DSP04430
                C                                                               DSP04440
0239             255          CONTINUE                                          DSP04450
                C*                                                              DSP04460
                C                                                               DSP04470
0240            WRITE (6,5)                                                     DSP04480
0241             5      FORMAT (1H1)                                            DSP04490
0242                    DO 7 CHPCT=1,NOFLD2                                     DSP04500
0243                    IF (PCTAB(CHPCT,1).LE.NOSUB3) GO TO 8                   DSP04510
0244                    WRITE (6,4) TRNVER(1,CHPCT),TRNVER(2,CHPCT)             DSP04520
0245              4     FORMAT(//3X, 'DOT (' , I4, ' ,' , I4, ' )  IS IN THE DESIGNATED ARDSP04530
                       *EA' )                                                   DSP04540
0246                    PCTAB(CHPCT,1) = 0                                      DSP04550
0247                    GO TO 7                                                 DSP04560
```

FILE DSPLY2

```
0248      8    IF (PCTAB(CHPCT,1).GT.0) GO TO 7
0249           WRITE (6,2) TRNVER(1,CHPCT),TRNVER(2,CHPCT)
0250      2    FORMAT(//3X, 'DOT (' , I4 , ' ,' , I4, ' )  IS NOT IN THE CLASSIFI
              *ED AREA' )
0251      7    CONTINUE
0252           WRITE (6,5)
          C
          C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
          C
0253                 IF (LISTSW.EQ.0) GO TO 600
0254               FLDCNT = FLDCNT + 1
0255                             CALL LISTSM(TOTALS,TTOL,
              *      PCTAB,LISTSW,GTUNIT,GTFILE,AIUNIT,AIFILE,
              *      PPUNIT,PPFILE,NAMECT,ALP,FLDCNT,
              *      NOCAT,CATNAM,SUBCAT,NOFLD2,NOSUB2)
          C
          C
          C
0256        600 CONTINUE
0257            GO TO 20
0258        310 CONTINUE
0259            RETURN
          C
          C         SELF-CONTAINED SUBROUTINE TO PRINT HEADERS
          C         -----------------------------------------------------------------
          C
0260        370 WRITE (6,HEAD)
0261            WRITE(6,380)FLDESC,CDATE
0262        380 FORMAT(' DISPLAY OF CLASSIFIED FIELD............',A4/
              *        '                CLASSIFICATION  DATE.......',2A4)
0263            IF(RMFLG.GT.0)WRITE(6,390)RMCOMB,(FFTVC2(L),L=1,RMFEAT)
0264            IF(RMFLG.EQ.0)WRITE(6,400) (FFTVC2(L),L=1,NOFFT2)
0265            CALL MAPHD (NOCAT,SYMMTX,CATNAM,KATNO,CLSNAM,SUBNO,SUBNAM,
              *              CLSSUB,NOCLS2,NOSUB2,THRSKY,THRES)
0266            IF(DESKEY.EQ.1)WRITE(6,410) SYMMTX(DESUNI)
0267        410 FORMAT(/5X,'** DESIGNATED FIELDS SYMBOL IS ',A1/)
0268        390 FORMAT(12X,'CLASSIFICATION CHANNELS...',I2,' LINEAR COMBINATIONS O
              *F CHANNELS'/39X, 30I3)
0269        400 FORMAT(12X,'CLASSIFICATION CHANNELS...',30I3)
0270             GO TO (31,221),WRTFG
          C*
          C*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
          C*
          C*        INTERNAL ROUTINE TO PRINT COLUMN NUMBERS
          C*
0271       510    J=0
0272            DO 50 IJ=ISTRT,IEND,SAMINC
0273            J=J+1
0274            COL(1,J) = IJ/100
0275            COL(2,J)=MOD(IJ,100)/10
0276            COL(3,J)=MOD(IJ,10)
0277            IF(J.EQ.110)GO TO 60
0278         50 CONTINUE
0279         60 SAMEN=IJ
0280            ISTRT=SAMEN + SAMINC
0281            IPTS=J
0282             WRITE(6,96)
0283            DO 80 IJ=1,3
0284         80 WRITE(6,90)(COL(IJ,J),J=1,IPTS)
0285         90 FORMAT(9X,110I1)
0286             WRITE(6,96)
0287      96     FORMAT(/)
0288             GO TO (85,222),PRTFG
          C
          C*****************************************************************
          C
          C         INTERNAL SUBROUTINE DOTPCT
          C
          C*        PURPOSE - TO SET PCTAB TO EACH DOT'S RESPECTIVE SUBCLASS
          C*         (OR THRESHOLD) NUMBER WHEN THE DOTKEY FLAG IS ON
          C
          C*****************************************************************
          C*
          C*        TEST FOR THE POSITION OF THE DOT IN THE CLASSIFICATION
          C*        RECORD
          C*
```

FILE DSPLY2

```
                C*     IF THE DOT'S POSITION IS NOT COMPATIBLE WITH THE CLASSIFICATION   DSP05330
                C*     RECORD, PCTAB(JJ,I) = 0                                           DSP05340
                C*                                                                       DSP05350
0289             432    IF (PCTKEY.NE.0) GO TO 430                                       DSP05360
0290                    DO 431 I=1,NOFLD2                                                DSP05370
0291             431    PCTAB(I,1)=0                                                     DSP05380
0292                    PCTKEY = 1                                                       DSP05390
0293             430    DO 411 K=SCNT,ECNT                                               DSP05400
0294                    SAMOF=TRNVER(1,K) - SAMSTR                                       DSP05410
0295                   SAMPS=SAMOF/SAMINC + 1                                            DSP05420
0296                   CHSPS=(SAMPS-1)*SAMINC                                            DSP05430
0297                   SAMOF = SAMOF + SAMSTR                                            DSP05440
0298                   CHSPS = CHSPS + SAMSTR                                            DSP05450
0299                    IF (CHSPS.NE.SAMOF) GO TO 421                                    DSP05460
0300                    PCTAB(K,1) = IR(SAMPS,11)                                        DSP05470
0301                    GO TO 411                                                        DSP05480
0302             421    PCTAB(K,1) = 0                                                   DSP05490
0303             411    CONTINUE                                                         DSP05500
0304                    GO TO 114                                                        DSP05510
                C                                                                        DSP05520
                CC                                                                       DSP05530
                C*************************************************************************DSP05540
                C                                                                        DSP05550
                CC                                                                       DSP05560
                CC     INTERNAL SUBROUTINE DOTSUM                                        DSP05570
                CC                                                                       DSP05580
                C* PURPOSE - TO COMPUTE BOTH UNCORRECTED PROPORTIONS AND BIAS-CORRECTED  DSP05590
                C*  PROPORTIONS OF THE ANALYST-LABELED CATEGORIES IN THE CLASSIFIED      DSP05600
                C*  AREA.  ALSO COMPUTES THE ALPHA TABLE AND OUTPUTS TWO CATEGORY       DSP05610
                C*  CLASSIFICATION PERFORMANCE SUMMARIES -                              DSP05620
                C*          (1)  SUMMARY OF ANALYST-LABELED CATEGORIES VERSUS TOTAL     DSP05630
                C*               NUMBER OF DOTS CLASSIFIED INTO EACH CATEGORY.          DSP05640
                C*          (2)  CLASSIFICATION SUMMARY FOR EACH INDIVIDUAL DOT.        DSP05650
                C                                                                        DSP05660
                C*************************************************************************DSP05670
                C*                                                                       DSP05680
                C*     INITIALIZE ARRAYS                                                 DSP05690
                C*                                                                       DSP05700
0305             600    TYPE=TRNSAV(3,1)                                                 DSP05710
                C*     ERROR FLAG FOR INVALID SUBCLASS                                   DSP05720
0306                    NOGOOD=-54321                                                    DSP05730
                C*     NUMBER OF DOT LABLED CATAGORIES                                   DSP05740
0307                    NOCAT=DOTKEY                                                     DSP05750
                C*     NUMBER OF CLASSIFICATION DOT CATAGORIES                           DSP05760
0308                    NCLAT=NOCAT                                                      DSP05770
                C*     NO. OF CLASSIFIED CAT. + 1 FOR THRESHOLDED DOTS                   DSP05780
0309                    NCLCAT = NOCAT + 1                                               DSP05790
                C*                                                                       DSP05800
                C*      ON INPUT TO DOTSUM,                                              DSP05810
                C*      TRNSAV(1,I) =  SCRATCH USAGE                                     DSP05820
                C*      TRNSAV(2,I) = MAPTAP CATEGORY NO. CORRESPONDING TO THE DOT'S     DSP05830
                C*      TRNSAV(3,I) = DOT TYPE  =1  OR  =2                               DSP05840
                C*      TRNSAV(4,I) = NO. OF VERTICES FOR THE DOT  =2                    DSP05850
                C*                                                                       DSP05860
                C*      TRNVER(1,I) = DOT SAMPLE NO.                                     DSP05870
                C*      TRNVER(2,I) = DOT LINE NO.                                       DSP05880
                C*                                                                       DSP05890
                C*      FOR I = 1,2, ... , NO. OF DOTS                                   DSP05900
                C*                                                                       DSP05910
                C*                                                                       DSP05920
0310                    DO 610 I=1,NOFLD2                                                DSP05930
                C*     LABLED DOT CATAGORY                                               DSP05940
0311                   TRNSAV(3,I)=TRNSAV(2,I)                                           DSP05950
                C*     VERTEX SAMPLE OF DOT                                              DSP05960
0312                   TRNSAV(1,I)=TRNVER(1,I)                                           DSP05970
                C*     LINE NUMBER OF DOT                                                DSP05980
0313                   TRNSAV(2,I)=TRNVER(2,I)                                           DSP05990
0314             610    CONTINUE                                                         DSP06000
0315                    DO 630 I=1,NCLCAT                                                DSP06010
0316                   CLTOT(I)=0                                                        DSP06020
0317             630    CONTINUE                                                         DSP06030
0318                   DO 130 I=1,NOCAT                                                  DSP06040
0319                   LALTOT(I) = 0                                                     DSP06050
0320                   TOTLAL(I)=0                                                       DSP06060
0321                   TOTTMP(I)=0                                                       DSP06070
0322                    DO 690 J=1,NCLCAT                                                DSP06080
```

FILE : SPLY2

```
0323            LABCLS(I,J) = 0
0324      690    CONTINUE
0325      130   CONTINUE
          C*
          C*    COMPUTE TOTAL NUMBER OF PIXELS CLASSIFIED IN TOTAL CLASSIFIED
          C*    AREA WITHOUT THRESHOLDED PIXELS
          C*
0326            PIXTOT=0.0
0327             DO 620 I=1,NOSUB2
0328             PIXTOT=PIXTOT + TOTALS(I)
0329      620    CONTINUE
          C*
          C*    CALCULATE TOTAL NO. PIXELS FOR EACH ANALYST-LABELED CATEGORY
          C*
0330             DO 631 I=1,NOSUB2
0331            CAT = SUBCAT(I)
0332              LBLTOT(CAT) = LBLTOT(CAT) + TOTALS(I)
0333      631    CONTINUE
          C*
          C*    COMPUTE DOT TOTALS CLASSIFIED BY CATEGORY
          C*
0334             DO 640 I=1,NOFLD2
0335            SUBCL=PCTAB(I,1)
0336              IF (SUBCL.LE.0.OR.SUBCL.GT.NOSUB3) GO TO 641
0337            CAT=SUBCAT(SUBCL)
0338            CLTOT(CAT)=CLTOT(CAT) + 1
0339              GO TO 643
0340      641    CONTINUE
0341            PCTAB(I,1) = NOGOOD
0342      643    CONTINUE
0343      640    CONTINUE
          C*
          C*    BUILD AN ORDERED SET OF DOT CATEGORY LABELS FOR ANALYST-LABELED
          C*     CATEGORIES
          C*
0344            J=1
0345            LBL=TRNSAV(3,1)
0346            DOTORD(J,1) = CATNAM(LBL)
0347            DOTORD(J,2) = LBL
0348             DO 650 I=2,NOFLD2
0349            LBL=TRNSAV(3,I)
0350             DO 660 K=1,J
0351            IF(LBL.EQ.DOTORD(K,2)) GO TO 49
0352      660    CONTINUE
0353            J=J+1
0354            DOTORD(J,1) = CATNAM(LBL)
0355            DOTORD(J,2) = LBL
0356       49   CONTINUE
0357      650    CONTINUE
          C*
0358            IF ( J .NE. NDCAT )  WRITE(6,3003)  J , NDCAT
          C*
0359      3003 FORMAT(//// 5X, '**** DSPLY2/DOTSUM -- DISCREPANCY IN DOTFILE INFO
               *RMATION ****' // 27X,'NO. OF DOT CATEGORY LABELS MATCHING MAPTAP
               *CATEGORY NAMES =' , I6// 27X, 'NO. OF DOT CATEGORIES IS GIVEN AS =
               * ' , I6 //// )
          C*
0360            DO 53 I=1,NDCAT
0361            K = I
0362            DO 52 J=1,NDCAT
0363            JK = DOTORD(J,2)
0364            IF(JK.NE.K) GO TO 51
0365            DOTLBL(K,1) = DOTORD(J,1)
0366            DOTLBL(K,2) = DOTORD(J,2)
0367       51   CONTINUE
0368       52   CONTINUE
0369       53   CONTINUE
          C*
          C*    COMPUTE UNCORRECTED PROPORTIONS OF DOT CATEGORIES OVER THE TOTAL
          C*     AREA CLASSIFIED
          C*
0370            DO 70 I=1,NDCAT
0371            J=DOTLBL(I,2)
0372            PUNC(I) =  FLOAT(LBLTOT(J))/PIXTOT
0373       70   CONTINUE
          C*
```

```
FILE : SPLY2
                C*     INITIALIZE TRNSAV(4,I),I=1,NOFLD2 (NO. OF DOTS) TO CONTAIN          DSP06850
                C*     CLASSIFICATION CATEGORY NUMBER OF DOT I.                            DSP06860
                C*                                                                         DSP06870
    0374              DO 680 I=1,NOFLD2                                                    DSP06880
    0375              SUBCL = PCT48(I,1)                                                   DSP06890
    0376              IF(SUBCL.EQ.NOGOOD) CAT = NOGOOD                                     DSP06900
    0377              IF (SUBCL.EQ.NOSUB3) CAT=NOCAT + 1                                   DSP06910
    0378              IF(SUBCL.GE.1.AND.SUBCL.LE.NOSUB2)CAT=SUBCAT(SUBCL)                  DSP06920
    0379              TRNSAV(4,I)=CAT                                                      DSP06930
    0380       680    CONTINUE                                                             DSP06940
                C*                                                                         DSP06950
                C*                                                                         DSP06960
                C*     TABLE OF TOTALS OF ANALYST-LABELED DOT CATEGORIES VERSUS            DSP06970
                C*      CLASSIFIED DOT CATEGORIES.  LABCLS(LABELED CAT,CLASSIFIED CAT)     DSP06980
                C*                                                                         DSP06990
                C*                                                                         DSP07000
    0381              DO 615 J=1,NOFLD2                                                    DSP07010
    0382              CLS=TRNSAV(4,J)                                                      DSP07020
    0383               IF (CLS.EQ.NOGOOD) GO TO 616                                        DSP07030
    0384              LB=TRNSAV(3,J)                                                       DSP07040
    0385              LABCLS(LB,CLS) = LABCLS(LB,CLS) + 1                                  DSP07050
    0386       616    CONTINUE                                                             DSP07060
    0387       615    CONTINUE                                                             DSP07070
                C*                                                                         DSP07080
                C*     COMPUTE ALPHA TABLE FOR LABELED VERSUS CLASSIFIED DOTS              DSP07090
                C*                                                                         DSP07100
    0388                DO 161 I=1,NOCAT                                                   DSP07110
    0389               DO 156 J=1,NOCAT                                                    DSP07120
    0390              IF(CLTOT(I).LE.0) GO TO 153                                          DSP07130
    0391              ALPHA(I,J) =  FLOAT(LABCLS(J,I))/ FLOAT(CLTOT(I))                    DSP07140
    0392              GO TO 154                                                            DSP07150
    0393        153   ALPHA(I,J) = 0.0                                                     DSP07160
    0394        154   CONTINUE                                                             DSP07170
    0395        156    CONTINUE                                                            DSP07180
    0396        161    CONTINUE                                                            DSP07190
                C*                                                                         DSP07200
                C*     COMPUTE CORRECTED PROPORTIONS FOR LABELED DOT CATEGORIES            DSP07210
                C*                                                                         DSP07220
    0397              DO 180 I=1,NOCAT                                                     DSP07230
    0398              PCTOT=0.0                                                            DSP07240
    0399              DO 170 J=1,NOCAT                                                     DSP07250
    0400              PCTOT=PCTOT + ALPHA(J,I)*PUNC(J)                                     DSP07260
    0401       170    CONTINUE                                                             DSP07270
    0402              PCORR(I)=PCTOT*100                                                   DSP07280
    0403       180    CONTINUE                                                             DSP07290
                C*                                                                         DSP07300
                C*     MAKE PUNC AND ALPHA TO REPRESENT PERCENTAGES                        DSP07310
                C*                                                                         DSP07320
    0404              DO 185 J=1,NOCAT                                                     DSP07330
    0405              PUNC(J)=PUNC(J) * 100.0                                              DSP07340
    0406              DO 175 I=1,NOCAT                                                     DSP07350
    0407              ALPHA(I,J) = ALPHA(I,J) * 100.0                                      DSP07360
    0408        175   CONTINUE                                                             DSP07370
    0409        185   CONTINUE                                                             DSP07380
                C*                                                                         DSP07390
                C*     CALCULATE TOTAL NO. LABELED DOTS IN EACH LABELED DOT CATEGORY       DSP07400
                C*                                                                         DSP07410
    0410               DO 199 J=1,NOCAT                                                    DSP07420
    0411              LBL=DOTLBL(J,2)                                                      DSP07430
    0412              DO 190 I=1,NOFLD2                                                    DSP07440
    0413              IF (TRNSAV(3,I).EQ.LBL) TOTLBL(J)=TOTLBL(J)+1                        DSP07450
    0414       190    CONTINUE                                                            DSP07460
    0415       199    CONTINUE                                                            DSP07470
                C*                                                                         DSP07480
                C*     TOTTHR WILL CONTAIN TOTAL THRESHOLDED DOTS IN EACH ANALYST-LABELEDDSP07490
                C*      CATEGORY                                                           DSP07500
                C*                                                                         DSP07510
    0416               DO 251  I=1,NOCAT                                                   DSP07520
    0417              TOTTHR(I) = LABCLS(I,NCLCAT)                                         DSP07530
    0418       251    CONTINUE                                                            DSP07540
                C      PRINT SUMMARY TABLES                                                DSP07550
    0419              K1=1                                                                 DSP07560
    0420              K2=NOCAT                                                             DSP07570
    0421              IF(NOCAT.GT.3)K2=3                                                   DSP07580
    0422              WX=1                                                                 DSP07590
    0423              NN = NOCAT                                                           DSP07600
```

L81

FILE DSPLY2

```
      IF(NOCAT.GT.30)NN=30
 400  WRITE(6,1000)
1000  FORMAT(1H1,T35,'DOT DATA PERFORMANCE SUMMARY')
      WRITE(6,1002)
1002  FORMAT(/ T25,'(SUMMARY OF ANALYST LABELED VERSUS CLASSIFIED CATEGO
     *RIES)')
      WRITE(6,1004)
1004  FORMAT(//,T2,'ANALYST',T14,'TOTAL NUMBER',T31,'PERCENT',T46,'NUMBE
     *R',T59,'UNCORRECTED',T76,'BIAS',T92,'DOT CLASSIFICATION SUMMARY')
      WRITE(6,1005)
1005  FORMAT(T2,'LABELED',T15,'ANALYST',T30,'CORRECTLY',T43,'THRESHOLDED
     *',T59,'CATEGORY',T74,'CORRECTED',T102,'(ALPHA)')
      WRITE(6,1006)
1006  FORMAT(T4,'DOT',T14,'LABELED DOTS',T29,'CLASSIFIED',T58,'PROPORTIO
     *N',T74,'CATEGORY')
      WRITE(6,1008)(CATNAM(I),I=1,K2)
1008  FORMAT(T2, 'CATEGORY',T62,'%',T73,'PROPORTION',1X,3(9X,A4)/T77,'%'
     *  // )
      DO 1010 I=NX,NN
      PCC = (FLOAT(LABCLS(I,I))/ FLOAT(TOTLBL(I))) * 100.0
      WRITE(6,2001)DOTLBL(I,1),TOTLBL(I),PCC,TOTTHR(I),PUNC(I),PCORR(I),
     *(LABCLS(I,J),J=K1,K2)
2001  FORMAT(/T4,A4,T15,I4,T30,F7.3,T45,I4,T60,F7.3,T74,F7.3,T91,3(I6,7X
     *)/)
      CALL BNI4A1(FMT3(3),1,K2)
      WRITE(6,FMT3)(ALPHA(J,I),J=K1,K2)
1010  CONTINUE
      IF(NOCAT.LE.3)GO TO 2005
      K3=K2
1200  WRITE(6,3000)
3000  FORMAT(1H1,T2,'ANALYST LABELED',T50,'DOT CLASSIFICATION SUMMARY')
      WRITE(6,3001)
3001  FORMAT(T3,'DOT CATEGORY',T59,'(ALPHA)')
      K3=K3 + 1
      KNN = K3 + 8- 1
      IF(NOCAT.LE.KNN)KNN=NOCAT
      WRITE(6,3002)(CATNAM(I),I=K3,KNN)
3002  FORMAT(/T25,8(A4,9X))
      DO 4000 II=NX,NN
      WRITE(6,3004) DOTLBL(II,1), (LABCLS(II,J),J=K3,KNN)
3004  FORMAT(//T8,A4,T25,8(I6,7X))
      KN=KNN-K3+1
      CALL BNI4A1(FMT4(3),1,KN)
      WRITE(6,FMT4)(ALPHA(J,II),J=K3,KNN)
4000  CONTINUE
      IF(KNN.LE.NOCAT)GO TO 2005
      K3=KNN
      GO TO 1200
2005  CONTINUE
      IF(NN.GE.NOCAT) GO TO 4999
      NX=NN +1
      NN= NX + 7
      IF(NN.GT.NOCAT) NN=NOCAT
      GO TO 400
4999  SUMMAY(NOSUB3)=THRESH
      PG = 0
      WRITE(6,5001)
5001  FORMAT(1H1,T39,'DOT DATA PERFORMANCE SUMMARY')
      WRITE(6,5003)
5003  FORMAT(/T39,'(SUMMARY OF INDIVIDUAL DOTS)')
      WRITE(6,5004)
5004  FORMAT(T43,'DOT ID = (SAMPLE,LINE)',/)
      SAMP2 = 0
      NWRITE = NOFLD2/4
      IF(MOD(NOFLD2,4).GT.0)NWRITE=NWRITE+1
      DO 7000 I=1,NWRITE
      PG = PG + 1
      SAMP1 = SAMP2 + 1
      SAMP2=SAMP1 + 3
      IF ( SAMP2 .GT. NOFLD2)  SAMP2= NOFLD2
      NSAMP = SAMP2 - SAMP1 + 1
      CALL BNI4A1(FMT5(3),1,NSAMP)
      WRITE(6,7005)(BLANKS(1),TRNSAV(1,S),TRNSAV(2,S),S=SAMP1,SAMP2)
7005  FORMAT(//T7,3(A1,1H(,I3,1H,,I3,1H),23X),A1,1H(,I3,1H,,I3,1H))
      CALL BNI4A1(FMT1(3),1,NSAMP)
      CALL BNI4A1(FMT1(10),1,NSAMP)
```

FILE DSPLY2

```
0493                CALL HNI4A1(FMT2(3),1,NSAMP)                                    DSP0A370
0494                CALL HNI4A1(FMT2(14),1,NSAMP)                                   DSP0A380
                C   WRITE(6,FMT)                                                    DSP0A390
0495                WRITE(6,7001) (BLANKS(Z),Z=1,NSAMP)                             DSP0A400
0496             7001 FORMAT (T7,3(A1,8HCATEGORY,24X),A1,8HCATEGORY)                DSP0A410
0497                WRITE (6,8001) (BLANKS(Z),Z=1,NSAMP)                            DSP0A420
0498             8001 FORMAT (1,3(A1,26H--------  ----------,6X),A1,               DSP0A430
                    *  26H--------  ----------)                                     DSP0A440
0499                WRITE(6,FMT2)                                                   DSP0A450
0500                J=0                                                             DSP0A460
0501                DO 6000 S=SAMP1,SAMP2                                           DSP0A470
0502                J=J+1                                                           DSP0A480
0503                XYZ=FMUSAV(3,S)                                                 DSP0A490
0504                ABC=FMUSAV(4,S)                                                 DSP0A500
0505                DEF = FCTAB(S,1)                                                DSP0A510
0506                TEMP(J,1)=CATNAM(XYZ)                                           DSP0A520
0507                IF(DEF.EQ.NOGOOD) GO TO 5998                                    DSP0A530
0508                TEMP(J,2)=CATNAM(ABC)                                           DSP0A540
0509                IF(DEF.EQ.NOSUCH)TEMP(J,2)=AST                                  DSP0A550
0510                TEMP(J,3)=SUBNAM(DEF)                                           DSP0A560
0511                GO TO 5999                                                      DSP0A570
0512             5998 TEMP(J,3)=AST                                                 DSP0A580
0513                TEMP(J,2)=AST                                                   DSP0A590
0514             5999 CONTINUE                                                      DSP0A600
0515             6000 CONTINUE                                                      DSP0A610
0516                WRITE(6,5020)(TEMP(P,1),TEMP(P,2),TEMP(P,3),P=1,J)              DSP0A620
0517             5020 FORMAT(/T4,3(A4,54,A4,6X,'(',A4,')',8X),A4,5X,A4,6X,'(',A4,')')DSP0A630
0518                IF(P0.LT.5)GO TO 6001                                           DSP0A640
0519                P0 = 0                                                          DSP0A650
0520                WRITE(6,6002)                                                   DSP0A660
0521             6002 FORMAT(1H1)                                                   DSP0A670
0522             6001 CONTINUE                                                      DSP0A680
0523                W = 1                                                           DSP0A690
0524             7000 CONTINUE                                                      DSP0A700
0525                  GO TO 500                                                     DSP0A710
0526                  END                                                           DSP0A720
```

E.4 SOFTWARE SUBPROGRAM NO. 4 (DSPLAY)

Subprogram DSPLAY is the driver for the DISPLAY processor of the EOD-LARSYS system.

E.4.1 Linkages

DSPLAY is called by MONTOR. It calls SETUP3, DSPLY1, DSPLY2, PRTPCT, EMTHRS, and FDIST.

E.4.2 Interfaces

DSPLAY has included labeled common block /DISPL/.

E.4.3 Inputs

Calling Sequence:

Keyword	Dimension	In/Out	Description
ARRAY	Top	Out	blank common used for data transfer
TOP		In	Size of ARRAY, currently = 10600.

E.4.4 Outputs

N/A

E.4.5 Storage Requirements

1532

E.4.6 Description

Subprogram DSPLAY has been modified to incorporate LIST processing. The calling sequences for DSPLY2 and SETUP3 have been expanded. If LISTSW is turned on, a transfer around the EMTHRS calls is made. After the DSPLY2 call is completed, a transfer around the PRTPCT calls is made.

(These transfers generally follow those made for normal P1 processing).

E.4.7 Flowchart

N/A

E.4.8 Listing

FILE DSPLAY

```
0001                SUBROUTINE DSPLAY(ARRAY,TOP)                                        DSP00010
                    C                                                                   DSP00020
0002                IMPLICIT INTEGER (A-H,O-Z)                                          DSP00030
                    C                                                                   DSP00040
                    C                                                                   DSP00050
0003                DIMENSION ARRAY(1)                                                  DSP00060
                    C                                                                   DSP00070
                    CC                                                                  DSP00080
                    CI--------------------------------------------------------------------IDSP00090
                    CI--------------------------------------------------------------------IDSP00100
                    CI                                                                  IDSP00110
                    CI    CALL..    CALL DSPLAY(ARRAY,TOP)                              DSP00120
                    CI                                                                  IDSP00130
                    CI    ARGS..    ARRAY =  SEE 'MONTOR'                             IDSP00140
                    CI              TOP   =  SEE 'MONTOR'                             IDSP00150
                    CI                                                                  IDSP00160
                    CI    PURPOSE..COORDINATES ROUTINES FOR DISPLAYING CLASSIFICATION   DSP00170
                    CI            MAP AND PERFORMANCE TABLES.                           DSP00180
                    CI                                                                  IDSP00190
                    CI--------------------------------------------------------------------IDSP00200
                    CI--------------------------------------------------------------------IDSP00210
                    CC                                                                  DSP00220
                          INCLUDE CHEK10,LIST                                           DSP00230
                          INCLUDE COMT10                                                DSP00240
0004                      COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),  DSP00250
                    *                 SUBCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,      DSP00260
                    *                 PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,  DSP00270
                    *                 PLTKEY,BMFLG,BMCOMB,BMFEAT,CDATE(2),            DSP00280
                    *                 FLDSV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,  DSP00290
                    *                 THRES(60),SYMBTX(66),HIGH(60),CON(60)           DSP00300
                    *                 ,FLDKEY,NOFLD2,NOFLD3,NOFFT2,FFTVC2(30)         DSP00310
                    *                 ,NOSUB2,NOTRED,TOTVT2,NOCLS2                    DSP00320
                    *                 ,KAT00(60),NOCAT,FILTER,MAPFMT                  DSP00330
                    *                 ,DESKEY,DESUNI,DESOTH,CHOP  ,ACHOP,AOTHER,ATOTAL  DSP00340
                    *                 ,SITE(6),ANALYS(5),CAM(15),CRMKEY,KEPPTS(60)    DSP00350
                    *                 ,DOTKEY,DOTERR                                  DSP00360
                    C*                                                                  DSP00370
                    C*    COMMON BLOCK DISPL IS USED ONLY IN THE DISPLAY PROCESSOR      DSP00380
                    C*                                                                  DSP00390
                    C*                                                                  DSP00400
                    C*    DEFINITIONS                                                   DSP00410
                    C*                                                                  DSP00420
                    C*      CATFLG - FLAG INDICATING WHETHER OR NOT CATEGORY PERFORMANCE DSP00430
                    C*               REPORTS MUST BE GENERATED.                         DSP00440
                    C*      CATNAM - NAMES OF CATEGORIES. READ FROM MAPTAP.             DSP00450
                    C*      CLSNAM - NAMES OF CLASSES. READ FROM MAPTAP.                DSP00460
                    C*      SUBNAM - NAMES OF SUBCLASSES. READ FROM MAPTAP.             DSP00470
                    C*      SUBCAT - SUBCLASS-CATEGORY CORRESPONDENCE VECTOR            DSP00480
                    C*               (SUBCAT(I)=M  MEANS SUBCLASS I BELONGS TO CATEGORY M)  DSP00490
                    C*      CLSSUB - SUBCLASS-CLASS CORRESPONDENCE VECTOR.              DSP00500
                    C*               (CLSSUB(I)=M  MEANS SUBCLASS I BELONGS TO CLASS M) DSP00510
                    C*      NOMAP  - TRIGGER INDICATING WHETHER OR NOT A MAP IS TO BE PRINTEDSP00520
                    C*      TOTVT3 - TOTAL NO. OF VERTICES IN INPUT TEST FIELDS.        DSP00530
                    C*      NOSUB3 - NO. OF SUBCLASSES USED IN CLASSIFY PLUS ONE,FOR THE  DSP00540
                    C*               THRESHOLD CLASS.                                   DSP00550
                    C*      PCFDKY - KEY INDICATING WHETHER OR NOT GROUND TRUTH PERFORMANCE DSP00560
                    C*               REPORTS ARE TO BE PRINTED ON A PER FIELD BASIS.    DSP00570
                    C*      TSTKEY - KEY INDICATING WHETHER OR NOT TEST FIELDS WERE INPUT.  DSP00580
                    C*      TRNKEY - KEY INDICATING WHETHER OR NOT TRAINING FIELDS ARE TO  DSP00590
                    C*               BE OUTLINED.                                       DSP00600
0005                      CONTINUE                                                      DSP00610
                    C*      THRSKY - THRESHOLD KEY                                      DSP00620
                    C*               =1 APPLY CHI-SQUARE THRESHOLDS                     DSP00630
                    C*               =2 APPLY EMPIRICAL  THRESHOLDS                     DSP00640
                    C*               =3 APPLY USER-INPUT THRESHOLDS                     DSP00650
                    C*               =4 APPLY FISHER DISTRIBUTION THRESHOLD             DSP00660
                    C*               =0 NO THRESHOLDING                                 DSP00670
                    C*      STATKY - KEY FOR PRINTING STATS FROM MAPTAP                 DSP00680
                    C*      EMPTRS - EMPIRICAL THRESHOLDING FLAG                        DSP00690
                    C*      THRSVA - USER-INPUT THRESHOLD VALUE FLAG                    DSP00700
                    C*      PLTKEY - FLAG FOR PRINTING CUMMULATIVE HISTOGRAMS OF QUADRATIC  DSP00710
                    C*               FORM.                                              DSP00720
                    C*      BMFLG  - FLAG INDICATING WHETHER OR NOT A B-MATRIX WAS      DSP00730
                    C*               APPLIED IN CLASSIFY.                               DSP00740
                    C*      BMCOMB - NO. OF LINEAR COMBINATIONS IN B-MATRIX             DSP00750
                    C*      BMFEAT - NO. OF CHANNELS USED IN COMPUTING B-MATRIX         DSP00760
```

C-3

194

```
FILE DSPLAY
C*      CDATE  - DATE OF CLASSIFICATION                                        DSP00770
C*      FLDSV2 - ADDRESS IN 'ARRAY' FOR TRAINING FIELD INFORMATION.            DSP00780
C*               FOR EACH TRAINING FIED 4 PIECES OF INFORMATION ARE            DSP00790
C*               STORED - 1=FIELD NAME                                         DSP00800
C*                        2=CLASS NO.                                          DSP00810
C*                        3=SUBCLASS NO.                                       DSP00820
C*                        4=NO. OF VERTICES                                    DSP00830
C*      FIELD2 - ADDRESS IN 'ARRAY' FOR RECTANGULAR AREA SURROUNDING           DSP00840
C*               EACH TRAINING FIELD. FOR EACH TRAINING FIELD 5 PIECES         DSP00850
C*               OF INFORMATION ARE STORED.                                    DSP00860
C*                        1=LINE START                                         DSP00870
C*                        2=LINE END                                           DSP00880
0006            CONTINUE                                                       DSP00890
C*                        3=SAMPLE START                                       DSP00900
C*                        4=SAMPLE END                                         DSP00910
C*                        5=POINTER INTO VERTEX ARRAY FOR VERTICES             DSP00920
C*                          OF THIS FIELD.                                     DSP00930
C*      VERTX2 - ADDRESS IN 'ARRAY' FOR TRAINING FIELD VERTICES.               DSP00940
C*      FLDSV3 - SAME AS FLDSV2 FOR TEST FIELDS                                DSP00950
C*      FIELD3 - SAME AS FIELD2 FOR TEST FIELDS                                DSP00960
C*      VERTX3 - SAME AS VERTX2 FOR TEST FIELDS                                DSP00970
C*      PCTID3 - ADDRESS IN 'ARRAY' FOR PERFORMANCE TABLE.                     DSP00980
C*      THRES  - THRESHOLD VALUES                                              DSP00990
C*      SYMBTX - SYMBOLS FOR EACH SUBCLASS, PLUS THRESHOLD SYMBOL              DSP01000
C*               AND OUTLINE SYMBOLS.                                          DSP01010
C*      HIGH   - THRESHOLD REJECTION PERCENTAGE - EMPIRICAL OPTION             DSP01020
C*      CON    - CONSTANT FACTOR FROM PROBABILITY DENSITY FUNCTION             DSP01030
C*               FROM CLASSIFY. ONE FOR EACH SUBCLASS.                         DSP01040
C*      FLDKEY - KEY INDICATING WHETHER GROUND TRUTH FIELDS ARE                DSP01050
C*               ASSOCIATED WITH CLASSES OR SUBCLASSES.                        DSP01060
C*      NOFLD2 - NO. OF TRAINING FIELDS                                        DSP01070
C*      NOFLD3 - NO. OF TEST FIELDS                                            DSP01080
C*      NOCHT2 - NO. OF CHANNELS USED IN CLASSIFICATION.                       DSP01090
C*      FETVC2 - CHANNELS USED IN CLASSIFICATION.                              DSP01100
C*      NOSUB2 - NO. OF SUBCLASSES USED IN CLASSIFICATION.                     DSP01110
C*      NOTAFD - NO. OF GROUND TRUTH FIELDS FOR WHICH PERFORMANCE              DSP01120
C*               TABLES WILL BE MADE. EQUALS NOFLD3 OR NOFLD2.                 DSP01130
C*      TOTVT2 - TOTAL NO. OF VERTICES FOR TRAINING FIELDS.                    DSP01140
C*      NOCLS2 - NO. OF CLASSES USED IN CLASSIFICATION.                        DSP01150
C*      KATNO  - CLASS - CATEGORY CORRESPONDENCE VECTOR                        DSP01160
0007            CONTINUE                                                       DSP01170
C*               (KATNO(I)=M  MEANS CLASS I IS IN CATEGORY M)                  DSP01180
C*      NOCAT  - NO. OF CATEGORIES.                                            DSP01190
C*      FILTER - FLAG FOR SPATIAL FILTERING OPTION.                            DSP01200
C*    MAPFMT - FORMAT FOR OUTPUT MAP TAPE                                      DSP01210
C*      DESKEY - KEY INDICATING WHETHER OR NOT DESIGNATED FIELDS WERE INDSP01220
C*      DESUNI - NO. FOR DESIGNATED UNIDENTIABLE  (NOSUB2+5)                   DSP01230
C*      DESOTH - NO. FOR DESIGNATED OTHER  (NOSUB2+6)                          DSP01240
C*      CROP   - NAME OF CROP FOR WHICH INTENSIVE TEST SITE SUMMARY            DSP01250
C*               REPORT IS TO BE PRINTED. CROP IS TO BE COMPARED WITH OTDSP01260
C*      ACROP  - ACRES OF 'CROP'  - USER INPUT                                 DSP01270
C*      AOTHER - ACRES OF 'OTHER' - USER INPUT                                 DSP01280
C*      ATOTAL - TOTAL ACRES IN CLASSIFIED SEGMENT                             DSP01290
C*      SITE   - NAME OF SITE (CLASSIFIED SEGMENT)                             DSP01300
C*      ANALYS - NAME OF ANALYST PERFORMING STUDY                              DSP01310
C*      CANS   - NAME OF PROCEDURE CONFIGURATION USED IN STUDY                 DSP01320
C*      CHPKEY - KEY FOR GENERATING INTENSIVE TEST SITE SUMMARY REPORT         DSP01330
C*      KFPPTS - TOTAL NUMBER PIXELS IN EACH SUBCLASS                          DSP01340
C*                                                                             DSP01350
C*      DOTKEY - KEY INDICATING WHETHER OR NOT DOT DATA CLASSIFICATION         DSP01360
C*    PERFORMANCE SUMMARIES ARE TO BE PROCESSED:  DOTKEY = 0 , NO DOT          DSP01370
C*    DATA PROCESSING ;  DOTKEY .GT. 0 , DOT PERFORMANCE SUMMARIES            DSP01380
C*    ARE PROVIDED ( IF DOTERR = 0 ) .                                         DSP01390
C*                                                                             DSP01400
C*      DOTERR - FLAG INDICATING THAT THE DOT DATA PROCESSING WILL BE          DSP01410
C*    DISCONTINUED DUE EITHER TO DISAGREEMENT OF LABELED  VS                   DSP01420
C*    CLASSIFICATION CATEGORY NAMES, OR ERRONEOUS PARAMETERS ON                DSP01430
C*    DOTFILE CONTROL CARD .                                                   DSP01440
0008            CONTINUE                                                       DSP01450
C*                                                                             DSP01460
C*                                                                             DSP01470
C$END                                                                          DSP01480
C*                                                                             DSP01490
C*    SETUP3 WILL READ FIRST 2 RECORDS FROM MAPTAP, AND CALL READF3            DSP01500
C*    TO READ IN CONTROL CARDS.  ALL OF THE PARAMETERS IN COMMON BLOCK         DSP01510
C*    DISPL ARE INITIALIZE BEFORE RETURNING TO THIS ROUTINE IN ADDITION        DSP01520
```

```
FILE DSPLAY
C*    TRAINING AND/OR TEST FIELD DEFINITIONS WILL BE STORED IN 'ARRAY'                DSP01530
C*                                                                                    DSP01540
C                                                                                     DSP01550
C***      THIS ADDED OR CHANGED NOV.13,1978 TO INCLUDE LIST PROS.                     DSP01560
C                                                                                     DSP01570
0009            REAL  ALP(2)                                                          DSP01580
0010            CALL SETUP3(ARRAY,TOP,LISTSW,GTUNIT,GTFILE,AIUNIT,AIFILE,             DSP01590
     *    PPUNIT,PPFILE,NAMECT,ALP)                                                   DSP01600
C*                                                                                    DSP01610
C*    DSPLY1 WILL READ NEXT 2 RECORDS FROM MAPTAP AND PRINT THE                       DSP01620
C*    STATISTICS IF REQUESTED.                                                        DSP01630
C*                                                                                    DSP01640
0011        CALL DSPLY1                                                             DSP01650
C                                                                                     DSP01660
C***         CODE ADDED NOV. 13,1978 TO INCLUDE LIST PROCESSING                       DSP01670
C                                                                                     DSP01680
0012            IF (LISTSW.NE.0) GO TO 30                                           DSP01690
0013        IF(EMPTRS.NE.2.AND. PLTKEY.NE.1)GO TO 30                                DSP01700
C*                                                                                    DSP01710
C*    EMTHRS COMPUTES AND PLOTS THE HISTOGRAM OF THE QUADRATIC FORM                   DSP01720
C*    FOR THE CORRECTLY CLASSIFIED PIXELS WITHIN THE TRAINING OR TEST                 DSP01730
C*    FIELDS.                                                                         DSP01740
C*                                                                                    DSP01750
0014        IF ( DOTKEY .GT. 0 .OR. DOTERR .GT. 0 )  GO TO 30                       DSP01760
0015        IF(PLOKEY.EQ.1)GO TO 10                                                 DSP01770
0016        WRITE(6,100)                                                            DSP01780
0017        GO TO 30                                                                DSP01790
0018     10 CONTINUE                                                                DSP01800
0019        IF(TSTKEY.EQ.1)CALL EMTHRS(ARRAY(FLDSV3),ARRAY(FIELD3),                 DSP01810
     *                                   ARRAY(VERTX3),NOFLD3)                      DSP01820
0020        IF(TSTKEY.NE.1)CALL EMTHRS(ARRAY(FLDSV2),ARRAY(FIELD2),                 DSP01830
     *                                   ARRAY(VERTX2),NOFLD2)                      DSP01840
0021     30 CONTINUE                                                                DSP01850
C-                                                                                    DSP01860
C--   TEST THRSKY = 4  FOR FISHER F-DISTRIBUTION THRESHOLDS                          DSP01870
C--   CALL FDIST TO COMPUTE AND STORE THRESHOLDS                                      DSP01880
C-                                                                                    DSP01890
0022        IF(THRSKY.EQ.4) CALL FDIST                                              DSP01900
C-                                                                                    DSP01910
C*                                                                                    DSP01920
C*    DSPLY2 PRINTS THE MAP AND CALLS PCT TO BUILD PERFORMANCE TABLES.                DSP01930
C*                                                                                    DSP01940
0023        CALL DSPLY2(ARRAY(FLDSV2),ARRAY(FIELD2),ARRAY(VERTX2),                  DSP01950
     *              ARRAY(FLDSV3),ARRAY(FIELD3),ARRAY(VERTX3),                      DSP01960
     *          ARRAY(PCTID3),LISTSW,GTUNIT,GTFILE,                                 DSP01970
     *          AIUNIT,AIFILE,PPUNIT,PPFILE,NAMECT,ALP)                             DSP01980
C*                                                                                    DSP01990
C*    IF DOT DATA PROCESSING WAS REQUESTED, THE PERFORMANCE                           DSP02000
C*    TABLES WERE PERFORMED IN DSPLY2                                                 DSP02010
C*                                                                                    DSP02020
C                                                                                     DSP02030
C***      CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING                          DSP02040
C                                                                                     DSP02050
0024            IF (LISTSW.NE.0) GO TO 99                                           DSP02060
0025        IF (DOTERR.EQ.1) GO TO 99                                               DSP02070
0026        IF (DOTKEY.GT.0) GO TO 99                                               DSP02080
C*                                                                                    DSP02090
C*    PRTPCT PRINTS THE PERFORMANCE TABLES                                            DSP02100
C*                                                                                    DSP02110
0027        IF(TSTKEY.EQ.1)CALL PRTPCT(ARRAY(FLDSV3),ARRAY(PCTID3),NOFLD3)          DSP02120
0028        IF(TSTKEY.NE.1)CALL PRTPCT(ARRAY(FLDSV2),ARRAY(PCTID3),NOFLD2)          DSP02130
0029     99 WRITE(6,6000)                                                           DSP02140
0030   6000 FORMAT(1H1,5X,' ***** DISPLAY COMPLETED *****'///)                      DSP02150
0031    100 FORMAT(/'  **DISPLAY** - FIELDS MUST BE DEFINED FOR SUBCLASSES FOR     DSP02160
     * EMPIRICAL THRESHOLDS'/)                                                      DSP02170
0032        RETURN                                                                  DSP02180
0033        END                                                                     DSP02190
```

E.5 SOFTWARE SUBPROGRAM NO. 5 (LISTSM)

Subprogram LISTSM has been added to the EOD-LARSYS DISPLAY processor in support of LIST processing. It reads the ground truth, AI and/or PPTC dot files, prepares CCIT outputs of AI, PPTC, and/or ground truth vs machine classification results for Type 1 and Type 2 dots, and computes proportion estimates with correction of the selected category for all these cases.

E.5.1 Linkages

LISTSM is called by DSPLY2. It calls LISTLC, LISTPR, and the EOD-LARSYS utility error routine CMERR.

E.5.2 Interfaces

LISTSM has included labeled common blocks /LISTMM/, INFORM/, and /DOTVEC/.

E.5.3 Inputs

Calling Sequence:

Keyword	Dimension	In/Out	Description
TOTALS	66	In	Total number of pixels in each subclass including DO, DU and thresholded, in the classified field
TTOL	66	In	Total numbers of thresholded pixels from each classification subclass
PCTAB	(209,1)	In.	Subclass number of each of the 209 pixels in dot position by machine classification after thresholding and DO/DU editing

Keyword	Dimension	In/Out	Description
LISTSW	1	In	See Subsection E.1.3 for description of these LIST variables
GTUNIT	1	In	
GTFILE	1	In	
AIUNIT	1	In	
PPUNIT	1	In	
PPFILE	1	In	
NAMECT	1	In	
ALP	2	In	
FLDCNT	1	In	Count of fields from MAPTAP. If =1, read all dot files for initialization
NCAT	1	In	Corresponds to NOCAT in /DISPL/, number of machine categories
CATNM	(61)	In	Corresponds to CATNUM (61) in /DISPL/, category names
SUBCAT	(60)	In	Corresponds to SUBCAT (60) in /DISPL/, map of subclasses to categories
NFLD2	1	In	Corresponds to NOFLD2 in /DISPL/, set = 209 in LIST
NSUB2	1	In	Corresponds to NOSUB2 in /DISPL/ number of regular subclasses (not including DO/DU and thresholded).

E.5.4 Outputs

N/A

E.5.5 Storage Requirements

13744

E.5.6 Description

Subprogram LISTSM has been added to the DISPLAY processor in support of LIST processing. It reads the various dot files, prepares CCIT output subsequent to printing, and compiles proportion estimates for Type 2 dot sets.

E.5.7 Flowchart

N/A

E.5.8 Listing

```
      SUBROUTINE LISTSM(TOTALS,TTOL,PCTAH,LISTSW,GTUNIT,
     * GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,NAMFCT,ALP,
     * FLDCNT,NCAT,CATNM,SUBCAT,NFLD2,NSUB2)
      IMPLICIT INTEGER (A-Z)
C
C*****     SUBROUTINE WRITTEN NOV 1978 TO INCLUDE LIST PROCESSING
C****      M AND N ARE BOUNDARY DOT NAMES
C****      D IS THE DESIGNATED NAME
C****      NAMFCT IS THE NAME OF THE 'SMALL GRAINS ' CATEGORY
C****      THE CATEGORY CLASSIFIER MUST HAVE BEEN INVOKED
C
      COMMON /LISTSM/ NPGA(3,2),NAMPGA(209,3,2),LINPGA(209,3,2),
     * SAMPGA(209,3,2),DOTLAB(209,4,2),VPGA(3),IPGA
C2345678
C     INCLUDE CMBK14
C     INCLUDE COMBK1
      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARS72,TOTVT2,NOFLD2,
     * AVAR2,COVAR2,CLSID2,SUMNO2,SUMDS2,FLDSV2,VERTX2,
     * FETVC2(30),SUMVC2(75),SUPPTR(75),CLSVC2(60),
     * KEPPFS(60),NOGRP,GRPNAM(60),GRPOFX(61),
     * GRPCHS(61),GROUPS(124)
      COMMON/OUTPUT/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),
     * PPTKFY,SIZE,LACIF
C$END
      DIMENSION ALPPSG(3)
         DATA SYATHK/'#'/,SYMOFS/' '/, SYMOUT/' '/,
     * MRGD/'M'/,PRCD/'R'/,OBCD/'O'/,BLANK/' '/,
     * ALPPSG/'PPC ','GT  ','AI  '/
C
C
      DIMENSION ITTOL(1),PCTAH(1),CATNM(1),SUBCAT(1),
     * FIELDS(9,250),VERTEX(1000),INFUNT(3),INFFIL(3)
      REAL TOTALS(1),ALP(2)
C
C
      REAL PMACH(2),PIXTOT,P11,P12,PB1,PB2,P,CATTOT(61)
      CONTINUE
      IF (FLDCNT.GT.1) GO TO 400
C
C*****     INITIALIZE IF FIRST FIELD TO BE SUMMARIZED
C
      DO 10 I = 1,3
      DO 10 II = 1,2
   10 NPGA(I,II) = 0
      DO 20 I = 1,2
      DO 20 II = 1,3
      DO 20 III = 1,209
      NAMPGA(III,II,I) = BLANK
      DOTLAB(III,II,I) = BLANK
   20 CONTINUE
C
C***      READ IN PPC GT AI FILES
C*** ASSUME TYPE 1 AND 2 ON SAME UNIT   BACK-TO-BACK
C
      IPGA = 0
      IF (PPUNIT.EQ.0) GO TO 25
      IPGA = IPGA + 1
      VPGA(IPGA) = 1
   25 IF (GTUNIT.EQ.0) GO TO 30
      IPGA = IPGA + 1
      VPGA(IPGA) = 2
   30 IF (AIUNIT.EQ.0) GO TO 35
      IPGA = IPGA + 1
      VPGA(IPGA) = 3
   35 IF (IPGA.E).0) CALL CMERR
C
C
      INFUNT(1) = PPUNIT
      INFFIL(1) = PPFILE - 1
      INFUNT(2) = GTUNIT
      INFFIL(2) = GTFILE - 1
      INFUNT(3) = AIUNIT
      INFFIL(3) = AIFILE - 1
C
C
      DO 100 I = 1,IPGA
         ISIT = VPGA(I)
      IPT =0
      STAMNT = 1
```

861

```
      TYPE = 1                                                          LIS00800
      NOCAT = 0                                                         LIS00810
      NOFLD2 = 0                                                        LIS00820
      TOTVT2 = 0                                                        LIS00830
      TOTVEC = 0                                                        LIS00840
      SWCHG = 0                                                         LIS00850
      INIT = 0                                                          LIS00860
  110 CALL LISTLC(FIELDS,STAMNT,&130,&140,&150,SWCHG,                   LIS00870
     *  INIT,INFUNT(ISIT),INFFIL(ISIT),IPT,VERTEX)                     LIS00880
C                                                                       LIS00890
C                                                                       LIS00900
  130 NPRA(ISIT,TYPE) = NOFLD2                                          LIS00910
      NAMPGA(NOFLD2,ISIT,TYPE) = FIELDS(1,NOFLD2)                       LIS00920
      LIMPGA(NOFLD2,ISIT,TYPE) = FLDINF(1)                              LIS00930
      SAMPGA(NOFLD2,ISIT,TYPE) = FLDINF(4)                              LIS00940
      I10 = FLDINF(1)/10                                                LIS00950
      I110 = FLOAT(F(4)/10                                              LIS00960
      J = (I10 - 1)*16 + I10                                            LIS00970
      DOTLAB(J,ISIT,TYPE) = FIELDS(1,NOFLD2)                            LIS00980
      GO TO 110                                                         LIS00990
C                                                                       LIS01000
C***      DOT TYPE CHANGE                                               LIS01010
C                                                                       LIS01020
  140 NOFLD2 = 0                                                        LIS01030
      GO TO 110                                                         LIS01040
C                                                                       LIS01050
C***    SEND CARD IMAGE DETECTED                                        LIS01060
C                                                                       LIS01070
  150 CONTINUE                                                          LIS01080
  100   CONTINUE                                                        LIS01090
C                                                                       LIS01100
C***    FCTAB STORED IN ORDER OF FIRST LINE---SECOND LINE ETC.          LIS01110
C                                                                       LIS01120
  400 NSUB3 = NSUB2 + 1                                                 LIS01130
      NSUB6 = NSUB2 + 4                                                 LIS01140
      NSUB7 = NSUB2 + 5                                                 LIS01150
      DO 430 I = 1,NFLD2                                                LIS01160
      SUBCL = FCTAB(I)                                                  LIS01170
      IF (SUBCL.GT.0) GO TO 405                                         LIS01180
      DOTLAB(I,4,1) = SYMOUT                                            LIS01190
      GO TO 420                                                         LIS01200
  405 IF (SUBCL.LE.NSUB2) GO TO 410                                     LIS01210
      IF (SUBCL.EQ.NSUB3)  DOTLAB(I,4,1) = SYMTHR                       LIS01220
      IF (SUBCL.EQ.NSUB6.OR.SUBCL.EQ.NSUB7) DOTLAB(I,4,1) =SYMDES       LIS01230
      GO TO 420                                                         LIS01240
  410 CAT = SUBCAT(SUBCL)                                               LIS01250
      DOTLAB(I,4,1) = CATNM(CAT)                                        LIS01260
  420 DOTLAB(I,4,2) = DOTLAB(I,4,1)                                     LIS01270
  430 CONTINUE                                                          LIS01280
C                                                                       LIS01290
C***    COMPUTE TOTAL NUMBER OF CLASSIFIED DOTS                         LIS01300
C                                                                       LIS01310
      PIXTOT = 0.                                                       LIS01320
      DO 440 I = 1,NSUB2                                                LIS01330
  440 PIXTOT = PIXTOT + TOTALS(I)                                       LIS01340
C                                                                       LIS01350
C***      COMPUTE TOTAL NO. OF PIXELS IN EACH MACHINE CATEGORY          LIS01360
C***      COUNT UP PIXELS OF CHOSEN AND OTHER CATEGORIES                LIS01370
C                                                                       LIS01380
      PMACH(1) = 0.                                                     LIS01390
      PMACH(2) = 0.                                                     LIS01400
      DO 445 I = 1,61                                                   LIS01410
  445 CATTOT(I) = 0.                                                    LIS01420
      DO 450  I = 1,NSUB2                                               LIS01430
      CAT = SUBCAT(I)                                                   LIS01440
      CATTOT(CAT) = CATTOT(CAT) + TOTALS(I)                             LIS01450
      DUM = CATNM(CAT)                                                  LIS01460
      IF ( DUM.NE.NAMECT) GO TO 448                                     LIS01470
      ICAT = CAT                                                        LIS01480
      PMACH(1) =              CATTOT(CAT)                               LIS01490
      GO TO 450                                                         LIS01500
  448 PMACH(2) =              CATTOT(CAT)                               LIS01510
  450 CONTINUE                                                          LIS01520
C23456789                                                               LIS01530
C***    NCAT = NO. OF MACHINE CLASSIFIED CATEGORIES                     LIS01540
C***    ICAT = CATEGORY NUMBER OF PREFERRED  CATEGORY                   LIS01550
C23456789                                                               LIS01560
      PMACH(1) = PMACH(1)/PIXTOT                                        LIS01570
      PMACH(2) = PMACH(2)/PIXTOT                                        LIS01580
```

```
C                                                                       LIS01590
C*** MAJOR LOOP                                                         LIS01600
C                                                                       LIS01610
      DO 600 I = 1,IPGA                                                 LIS01620
      ISIT = VPGA(I)                                                    LIS01630
      IF (LPGA(ISIT,1).EQ.0) GO TO 505                                  LIS01640
      ITYPE = 1                                                         LIS01650
      CALL LISTPR(ISIT,DOTLAB,ITYPE)                                    LIS01660
  505 IF (SPGA(ISIT,2).EQ.0) GO TO 600                                  LIS01670
      ITYPE = 2                                                         LIS01680
      CALL LISTPR(ISIT,DOTLAB(1,1,2),ITYPE)                             LIS01690
C                                                                       LIS01700
C*** COMPUTE N11,N12,N21,N22,NN1,NN2 FOR TYPE 2 DOTS                    LIS01710
C                                                                       LIS01720
      N11 = 0                                                           LIS01730
      N12 = 0                                                           LIS01740
      N21 = 0                                                           LIS01750
      N22 = 0                                                           LIS01760
      NN1 = 0                                                           LIS01770
      NN2 = 0                                                           LIS01780
      DO 540 II = 1,209                                                 LIS01790
      DUM = SAMPGA(II,ISIT,2)                                           LIS01800
      IF (DUM.NE.NAMECT) GO TO 530                                      LIS01810
C                                                                       LIS01820
C*** DOT LABEL IS PREFERRED CATEGORY                                    LIS01830
C                                                                       LIS01840
      DUMS = SAMPGA(II,ISIT,2)/10                                       LIS01850
      DUML = LINPGA(II,ISIT,2)/10                                       LIS01860
      DO 515 III = 1,11                                                 LIS01870
      DO 514 IIII = 1,19                                                LIS01880
      IF(IIII.NE.DUMS) GO TO 514                                        LIS01890
      IF(III.NE.DUML)  GO TO 514                                        LIS01900
C*** FOUND ANOTHER CLASSIFIED DOT                                       LIS01910
      J = (III - 1)*19 + IIII                                           LIS01920
      DUMA = DOTLAB(J,4,2)                                              LIS01930
      IF (DUMA.EQ.DUMA) NN1 = NN1 + 1                                   LIS01940
      IF (DUMA.EQ.SYMOFS) GO TO 514                                     LIS01950
C234567890                                                              LIS01960
      IF (DUMA.EQ.SYMTHR) GO TO 514                                     LIS01970
      IF (DUMA.EQ.SYMOUT) GO TO 514                                     LIS01980
      IF (DUMA.NE.DUMA) N12 = N12 + 1                                   LIS01990
  514 CONTINUE                                                          LIS02000
  515 CONTINUE                                                          LIS02010
      GO TO 540                                                         LIS02020
C                                                                       LIS02030
C***                                                                    LIS02040
C                                                                       LIS02050
  530 IF (DUMA.NE.MACD.AND.DUM.NE.RACD) GO TO 550                       LIS02060
C                                                                       LIS02070
C***    THIS PIXEL WAS LABELED BOUNDARY                                 LIS02080
C                                                                       LIS02090
      DUMS = SAMPGA(II,ISIT,2)/10                                       LIS02100
      DUML = LINPGA(II,ISIT,2)/10                                       LIS02110
      DO 535 III = 1,11                                                 LIS02120
      DO 534 IIII = 1,19                                                LIS02130
      IF(IIII.NE.DUMS) GO TO 534                                        LIS02140
      IF (III.NE.DUML) GO TO 534                                        LIS02150
      J = (III - 1)*19 + IIII                                           LIS02160
      DUMA = DOTLAB(J,4,2)                                              LIS02170
      IF (DUMA.EQ.NAMECT) NB1 = NB1 + 1                                 LIS02180
      IF (DUMA.EQ.SYMOFS)  GO TO 534                                    LIS02190
      IF (DUMA.EQ.SYMTHR) GO TO 534                                     LIS02200
      IF (DUMA.EQ.SYMOUT) GO TO 534                                     LIS02210
      IF (DUMA.NE.NAMECT) NB2 = NB2 + 1                                 LIS02220
  534 CONTINUE                                                          LIS02230
  535 CONTINUE                                                          LIS02240
      GO TO 580                                                         LIS02250
C                                                                       LIS02260
C***    THIS PIXEL IS LABELED DESIGNATED OR IS IN THE OTHER CATEGORY    LIS02270
C                                                                       LIS02280
  550 IF (DUM.EQ.DRCD) GO TO 580                                        LIS02290
C*** IT'S IN THE OTHER CATEGORY                                         LIS02300
      DUMS = SAMPGA(II,ISIT,2)/10                                       LIS02310
      DUML = LINPGA(II,ISIT,2)/10                                       LIS02320
      DO 555 III = 1,11                                                 LIS02330
      DO 554 IIII = 1,19                                                LIS02340
      IF (IIII.NE.DUMS) GO TO 554                                       LIS02350
      IF (III.NE.DUML)  GO TO 554                                       LIS02360
      J = (III - 1)*19 + IIII                                           LIS02370
```

```
      DUMA = OUTLAR(J,4,2)                                              LIS02380
      IF (DUMA.EQ.NAMECT) N21 = N21 + 1                                 LIS02390
      IF (DUMA.EQ.SYMDES) GO TO 554                                     LIS02400
      IF (DUMA.EQ.SYMTHR) GO TO 554                                     LIS02410
      IF (DUMA.EQ.SYMOUT) GO TO 554                                     LIS02420
      IF (DUMA.NE.NAMECT) N22 = N22 + 1                                 LIS02430
  554 CONTINUE                                                          LIS02440
  555 CONTINUE                                                          LIS02450
  500 CONTINUE                                                          LIS02460
C                                                                       LIS02470
C*** COMPUTE PROPORTION OF PREFERRED CLASS                              LIS02480
C                                                                       LIS02490
C234567890                                                              LIS02500
      P11 = FLOAT(N11)/FLOAT(N11 + N21 + NB1)                           LIS02510
      P12 = FLOAT(N12)/FLOAT(N12 + N22 + NB2)                           LIS02520
      PB1 = FLOAT(NB1)/FLOAT(N11 + N21 + NB1)                           LIS02530
      PB2 = FLOAT(NB2)/FLOAT(N12 + N22 + NB2)                           LIS02540
      P = PMACH(1)*(P11 + ALP(1)*PB1)                                   LIS02550
      P = P + PMACH(2)*(P12 + ALP(2)*PB2)                               LIS02560
C                                                                       LIS02570
      WRITE(6,990)  ALPMSG(ISIT)                                        LIS02580
      WRITE(6,1000)                                                     LIS02590
      WRITE(6,1050) PMACH(1),PMACH(2)                                   LIS02600
      WRITE(6,1010)                                                     LIS02610
      WRITE(6,1020) N11,N12,N21,N22,NB1,NB2                             LIS02620
      WRITE(6,1030)                                                     LIS02630
      WRITE(6,1040) P11,P12,PB1,PB2,P,ALP(1),ALP(2)                     LIS02640
C                                                                       LIS02650
  990 FORMAT(1H1,' FOR LIST PROCESSING',' ',A4,'  VS MACHINE CLASS')    LIS02660
 1000 FORMAT(1H0,' PROPORTION SUMMARY  ')                               LIS02670
 1010  FORMAT(1H0,'  N11      N12      N21      N22      NB1      NB2')  LIS02680
 1020 FORMAT(1H0,6(2X,I5))                                              LIS02690
 1030  FORMAT (1H0,'  P11     P12      PB1      PB2      P    ALP   ')  LIS02700
 1040 FORMAT(1H0,7(2X,F6.4))                                            LIS02710
 1050 FORMAT(1H0,'  P(1) = ',F7.4,'     P(2) = ',F7.4)                 LIS02720
  600 CONTINUE                                                          LIS02730
      RETURN                                                            LIS02740
      END                                                               LIS02750
```

E.6 SOFTWARE SUBPROGRAM NO. 6 (LISTPR)

Subprogram LISTPR has been added to the DISPLAY processor of the EOD-LARSYS system in support of LIST processing. It outputs the CCIT-type reports of GT, AI, and/or PPTC dots vs machine classification.

E.6.1 Linkages

LISTPR is called by LISTSM. It calls no other routines.

E.6.2 Interfaces

No common blocks are included.

E.6.3 Inputs

Calling Sequence:

Keyword	Dimension	In/Out	Description
ISIT	1	In	Type of CCIT report =1 PPTC vs machine =2 GT vs machine =3 AI vs machine
DOTLAB	(19,11,4)	In	Dot category names 19 ∿ sample # 11 ∿ line # 4 ∿ PPTC GT AI Machine
ITYPE	1	In	Dot type, 1 or 2, to be used for this output of CCIT report.

E.6.4 Outputs

E.6.5 Storage Requirements

1444

E.6.6 Description

Subprogram LISTPR outputs the CCIT-type report comparing PPTC, GT, or AI category name vs machine classified name for Type 1 or Type 2 dots.

E.6.7 Flowchart

N/A

E.6.8 Listing

```
C THIS SUBROUTINE PRINTS LABEL TABELS                                   LIS00010
      SUBROUTINE LISTPR(ISIT,OUTLAB,ITYPE)                              LIS00020
      IMPLICIT INTEGER (A-Z)                                            LIS00030
      DATA SLASH/'/'/,BLANK/' '/                                        LIS00040
      DIMENSION LINE(57)                                                LIS00050
            DIMENSION OUTLAB(14,11,4)                                   LIS00060
            I1 = ISIT                                                   LIS00070
            I2 = 4                                                      LIS00080
            NPRT = 5                                                    LIS00090
      WRITE(NPRT,10)                                                    LIS00100
   10 FORMAT(1H1)                                                       LIS00110
         IF ( ISIT.EQ.1) WRITE(NPRT,120)                                LIS00120
  120       FORMAT(//,50(' '),'PRIC  VS  CLASSIFIED LABELS ')           LIS00130
         IF (ISIT.EQ.2) WRITE(NPRT,130)                                 LIS00140
  130  FORMAT(//,50X,'MOUTH VS CLASSIFIED LABELS')                       LIS00150
         IF (ISIT.EQ.3) WRITE(NPRT,230)                                 LIS00160
  230  FORMAT(//,50X,'H. I. LABELS VS CLASSIFIED LABELS')                LIS00170
      WRITE(NPRT,20) ITYPE                                              LIS00180
   20 FORMAT(//,50X,'TYPE ',I1,' DOT CLASSIFICATION')                   LIS00190
      WRITE(NPRT,30) (I,I=10,190,10)                                    LIS00200
   30 FORMAT(//,12X,19I5)                                               LIS00210
      DO 200 I=1,11                                                     LIS00220
         DO 215 J = 1,57                                                LIS00230
  215    LINE(J) = BLANK                                                LIS00240
      DO 220 J=1,14                                                     LIS00250
      LL=3*(J-1)                                                        LIS00260
      LINE(LL+1)=OUTLAB(J,I,I1)                                         LIS00270
      LINE(LL+3)=OUTLAB(J,I,I2)                                         LIS00280
        IF (OUTLAB(J,I,I1).NE.BLANK)  LINE(LL + 2) = SLASH              LIS00290
      LL2 = LINE(LL + 2)                                                LIS00300
      IF(LL2.NE.SLASH) LINE(LL + 3) = BLANK                             LIS00310
  220  CONTINUE                                                         LIS00320
      I10=I*10                                                          LIS00330
      WRITE(NPRT,300)I10,(LINE(KK),KK=1,57)                             LIS00340
  300 FORMAT(//,9X,I3,1X(2X,3A1))                                       LIS00350
  200  CONTINUE                                                         LIS00360
  100  CONTINUE                                                         LIS00370
      RETURN                                                            LIS00380
      END                                                               LIS00390
```

E.7 SOFTWARE SUBPROGRAM NO. 7 (LISTLC)

Subprogram LISTLC has been added to the DISPLAY and DOTDATA processors of the EOD-LARSYS system in support of LIST processing.

E.7.1 Linkages

LISTLC is called by subprograms LISTSM and DOTS. It calls subprogram NUMBR, EOD-LARSYS utility routines FSFMSL, and NXTCHR, and the Fortran re-read routine REREAD.

E.7.2 Interfaces

Subprogram LISTLC has included labeled common blocks /INFORM/ and /DOTVEC/.

E.7.3 Inputs

Calling Sequence:

Keyword	Dimension	In/Out	Description
FIELDS	(4,__)	Out	Information concerning dots indexed by NOFLDL (1,__) = category name (4,__) = 2 (# vertices)
STAMNT	1	In/Out	Initially set =1, reset to 2 if dot card has been read and is being processed.
*	-		
*	-		
*	-		
SWCHG	1	In/Out	Initially set =0, counts no. times dot type changes

Keyword	Dimension	In/Out	Description
INIT	1	In/Out	Set =0 initially, reset to 1 after dot files read in
IUNIT	1	In	Unit number of input dot file
IFILE	1	In	Relative file number of input dot file
IPT	1	In/Out	Pointer in VERTEX, incremented by 4
VERTEX	(_)	Out	Dot vertices (Sample, line numbers repeated)

E.7.4 Outputs

N/A

E.7.5 Storage Requirements

3658

E.7.6 Description

Subprogram LISTLC is similar to FLDLAC except dot files are read in on "initial" entry and SWCHG is included in the calling sequence rather than in an internal DATA statement.

E.7.7 Flowchart

N/A

E.7.8 Listing

```
C      FIELDS - CATEGORY NAME AND DOT TYPE FOR DOT I STORED IN          LIS00010
C               FIELD(1,I) AND FIELD(4,I)                               LIS00020
C      STAMNT - INITIALLY SET TO 1. SWITCHED TO INDICATE DOTS BEING     LIS00030
C               TAKEN FROM CURRENTLY READ CARD.                         LIS00040
C      IPT - INITIALLY SET TO 1.INDEX NUMBER FOR FIELD VERTEX INFORMATION LIS00050
C      VERTEX - VERTEX INFORMATION FOR EACH DOT.                        LIS00060
C                                                                       LIS00070
      SUBROUTINE LISTLC(FIELDS,STAMNT,*,*,*,SWCHG,INIT,IUNIT,IFILE,IPT, LIS00080
     *VERTEX)                                                           LIS00090
      IMPLICIT INTEGER (A-Z)                                           LIS00100
      REAL NUM                                                         LIS00110
      DIMENSION FIELDS(4,1),VERTEX(1),CARD(75),NDOTS(30)               LIS00120
      DIMENSION SCARD(80)                                              LIS00130
      LOGICAL SWITCH                                                   LIS00140
           DATA SWITCH/.TRUE./,ENDHCD/'$EN'/,                          LIS00150
     *CATNM1/' '/                                                      LIS00160
C     INCLUDE COMMON                                                   LIS00170
CC    INCLUDE COMMON                                                   LIS00180
      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,         LIS00190
     *              AVAR2,COVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTX2,   LIS00200
     *              FETVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60),       LIS00210
     *              KEPRTS(60),NOGRP,GRPNAM(60),GRPDEX(61),            LIS00220
     *              GRPCHK(61),GRDUPS(24)                              LIS00230
      COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY     LIS00240
     *              ,SIZE ,LACTF                                       LIS00250
           DIMENSION IBUF(80)                                          LIS00260
C$FOO                                                                 LIS00270
        IF (IUNIT.NE.0) GO TO 5                                       LIS00280
        READ 10,IT                                                    LIS00290
        CALL FSFMFL(IUNIT,IFILE,ISTAT)                                LIS00300
        READ (IUNIT,1010) (IBUF(I), I=1,80)                           LIS00310
1010      FORMAT(80A1)                                                LIS00320
      WRITE(6,1020) (IBUF(I), I=1,80)                                 LIS00330
1020  FORMAT(1H0,80A1)                                                LIS00340
5       IUNIT = 1                                                     LIS00350
      IF(STAMNT.EQ.2)GO TO 30                                         LIS00360
      IF(.NOT.SWITCH)GO TO 20                                         LIS00370
        CALL REREAD(30,*0)                                            LIS00380
10    READ(IUNIT,103)(ACARD(I),I=1,80)                                LIS00390
103   FORMAT(80A1)                                                    LIS00400
      WRITE(30,103)(ACARD(I),I=1,80)                                  LIS00410
      REWIND 30                                                       LIS00420
      READ(30,1000)TO,TYPES,CARD                                      LIS00430
      REWIND 30                                                       LIS00440
1000  FORMAT(A3,1X,I1,75A1)                                           LIS00450
      IF(TYPE.EQ.TYPES )GO TO 20                                      LIS00460
      IF(SWCHG.NE.0)GO TO 40                                          LIS00470
      TYPE = TYPES                                                    LIS00480
C                                                                     LIS00490
CC     READ CARD                                                      LIS00500
C                                                                     LIS00510
 20   COL = 0                                                         LIS00520
      CATLA = NXTCHR(CARD,COL)                                        LIS00530
C*    IF NEXT CHAR IS NOT A CAT. NAME, CORRECT COL COUNT TO READ NUM  LIS00540
      IF(CATLA.GT.0)GO TO 21                                          LIS00550
      IF(CATLA.EQ.CATNM1)GO TO 23                                     LIS00560
      NOCAT=NOCAT + 1                                                 LIS00570
      CATNAM(NOCAT)=CATLA                                             LIS00580
      CATNM1 = CATLA                                                  LIS00590
      GO TO 23                                                        LIS00600
 21   COL=COL - 1                                                     LIS00610
 23   NDCARD=0                                                        LIS00620
      CALL NUMBR(NDOTS,NDCARD,CARD,COL)                               LIS00630
      IF(NDCARD.EQ.0)GO TO 10                                         LIS00640
      ICNT = 0                                                        LIS00650
      STAMNT = 2                                                      LIS00660
      SWITCH = .TRUE.                                                 LIS00670
      GO TO 100                                                       LIS00680
C                                                                     LIS00690
CC     TEST FOR END OF DOTS TO BE PROCESSED ON CARD                   LIS00700
C                                                                     LIS00710
 30   IF(ICNT.LT.NDCARD)GO TO 100                                     LIS00720
C                                                                     LIS00730
CC     READ NEXT CARD                                                 LIS00740
C                                                                     LIS00750
      STAMNT = 1                                                      LIS00760
      ICNT = 0                                                        LIS00770
      READ(IUNIT,103)(ACARD(I),I=1,80)                                LIS00780
      WRITE(30,103)(ACARD(I),I=1,80)                                  LIS00790
```

```
      NEXT 0 3)
      READ(30,1000 )ID,TYPES,CARD
      NEXT 1 30
      IF(ID.EQ.ENDHCD)RETURN 3
      IF(TYPE.EQ.TYPES)GO TO 20
      SWITCH = .FALSE.
      SWCHG = SWCHG + 1
      IF(SWCHG.GT.1)GO TO 40
      TYPE = TYPES
       IPT = 0
C***** CHANGED JUNE 28 1978
      RETURN 2
C
C
 100  ICNT = ICNT + 1
      NOFLD2 = NOFLD2 + 1
C
C     COMPUTE LINE INCREMENT
C
      N1 = NDOTS(ICNT)
      NI =IABS( N1) / 100000000
      LI = IABS(N1) - NI * 100000000
      IF(LI.GE.10000000)NI = NI +1
C
C     COMPUTE SAMPLE INCREMENT
C
      KK=1
      IF(N1.LT.0)KK=-1
      LI = NI * KK
      N2 = N1 - LI * 100000000
      N3 = IABS( N2)/10000
      SI = IABS(N2)-N3 * 10000
      IF(SI.GE.1000)N3 = N3 + 1
      KK=1
      IF(N2.LT.0)KK=-1
      SI = N3 *KK
      LACI = N2 - SI * 10000
      LR = (LACI-1)/19
      LR = (LR+1) * 10
      LS = LR - 1
      LS = LS /10
      LS = 10 * (LACI - (LS*19))
      L = LR -LI
      S = LS + SI
C
C     STORE DOT INFO
C
      FIELDS(1,NOFLD2) = CATNM
      FIELDS(4,NOFLD2) = 2
      FLDINF(1) = L
      FLDINF(2) = L
      FLDINF(3) = 1
      FLDINF(4) = S
      FLDINF(5) = S
      FLDINF(6) = 1
      IF(IPT.NE.0)GO TO 35
      IPT = -3
 35   IPT = IPT + 4
      VERTEX(IPT) = S
      VERTEX(IPT+1)=L
      VERTEX(IPT+2)=S
      VERTEX(IPT+3)=L
      RETURN 1
 40   WRITE(6,2000)
2000  FORMAT(//5X,'ERROR HAS OCCURRED IN READING LACIE FORMATTED DOT CAR
     *DS - SUBROUTINE FLDLAC - EXIT TAKEN')
      RETURN 3
      END
```

E.8 SOFTWARE SUBPROGRAM NO. 8 (DOTS)

Subprogram DOTS is the principal program in the DOTDATA processor. It reads the input dot information, looks for spectral data from the MSS data file, and writes the Type 1 and Type 2 dot files. In LIST processing, it has been augmented to allow the user to input a LACIE formatted dot file in lieu of in-line dot descriptors in the control card/field card stream.

E.8.1 Linkages

DOTS is called by DOTDAT, the driver for the DOTDATA processor of EOD-LARSYS. It calls FLDLAC, LISTLC, FLDTYP, WRTDOT, and EOD-LARSYS utility routines TAPHDR, LINERD, FDLINT, WRTFLD, and FLDINT.

E.8.2 Interfaces

DOTS has included labeled common blocks /INFORM/, /DOTVEC/, /GLOBAL/, and /ISOLNK/.

E.8.3 Inputs

Calling Sequence:

Keyword	Dimension	In/Out	Description
DATA	(SIZE, __) max 5000	Out	SIZE = 4 + NOFET2 storage for writing dot data file
FIELDS	(4,__) max 1000	Out	Field information for each dot (1, _) category name (4, _) 2 ∿ # vertices
VERTEX	(_) max TOP - 6001	Out	Vertex information (line + sample number) for Type 1 or Type 2 dots
TOP	1	In	ARRAY blank common storage size, currently = 10600

E.8.4 Outputs

E.8.5 Storage Requirements

43690

E.8.6 Description

DOTS has been modified to support LIST processing in the following manner. The /DOTVEC/ variable LACIE can now be a unit number (>1) as well as a P1 switch (=1). If LACIE >1 is encountered, the new routine LISTLC will be called to read a dot file P1 in-line LACIE dot cards or FLDTYP in the case of field-formatted dot cards). Two new variables INIT and SWCHG have been added to help with determination of the initial call to LISTLC.

E.8.7 Flowchart

N/A

E.8.8 Listing

```
FILE DOTS

                C                                                                       DOT00010
                C     DOTS IS THE CO-ORDINATOR FOR CREATING THE DOT DATA FILE           DOT00020
                C                                                                       DOT00030
0001                  SUBROUTINE DOTS(DATA,FIELDS,VERTEX,TOP)                           DOT00040
0002                  IMPLICIT INTEGER (A-Z)                                            DOT00050
0003                  DIMENSION DATA(SIZE,1),IDATA(10000)                               DOT00060
0004                  DIMENSION FIELDS(4,1),          FL(12),VERTEX(1)                  DOT00070
                C     INCLUDE COMRK1,LIST                                               DOT00080
                C     INCLUDE COMRK6,LIST                                               DOT00090
                C     INCLUDE CMRK14,LIST                                               DOT00100
                C     INCLUDE CMRK16,LIST                                               DOT00110
                C                                                                       DOT00120
0005                  COMMON/INFORM/NUCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,          DOT00130
                     *              AVAR2,COVAR2,CLSID2,SUBUD2,SUBDS2,FLDSV2,VERTX2,  DOT00140
                     *              FETVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60),      DOT00150
                     *              KEPTS(60),NUGRP,GRPNAM(60),GRPDEX(61),            DOT00160
                     *              GRPCHR(61),GROUPS(124)                            DOT00170
0006                  COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,HMFILE,HMKEY,         DOT00180
                     *       HISFIL,HISKEY,TWFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,        DOT00190
                     *    DRUMAD,DRMXDS,PAGST7,DATFIL,STAFIL,ASAV,ASAVFL              DOT00200
                     *  ,NHSTUN,NHSTFI,SCTRUN,DAPFIL                                  DOT00210
                     *  ,DOTUNT,DOTFIL,UCMPAS,TPNSFL,BMTRFL,HISTFL,PCHUNT,            DOT00220
                     *  CKOUNT,PRTUNT,RANDIO                                          DOT00230
0007                  COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEL,FLDINF(6),PRTKEY      DOT00240
                     *                ,SIZE ,LACIE                                   DOT00250
0008                  COMMON/ISUBNK/SUNAME(9),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP      DOT00260
                C$END                                                                   DOT00270
                C                                                                       DOT00280
0009                  DATA BLANK/' '/                                                   DOT00290
0010                  NOSUN = 8                                                         DOT00300
0011                  ISUNT = 1                                                         DOT00310
0012                  ISUNC = 0                                                         DOT00320
0013                  STAMNT = 1                                                        DOT00330
0014                  SWICH = 0                                                         DOT00340
0015                  IPT = 1                                                           DOT00350
0016                  IF(LACIE.NE.0)IPT=0                                               DOT00360
                C                                                                       DOT00370
                C***   CODE ADDED NOV 21, 1978 TO SUPPORT LIST PROCESSING               DOT00380
                C                                                                       DOT00390
0017                       INIT = 0                                                     DOT00400
0018                      ZERO = 0                                                      DOT00410
0019                        SWCHG = 0                                                   DOT00420
0020                  NOCAT = 0                                                         DOT00430
0021                  NOFLD2=0                                                          DOT00440
0022                  TYPE = 1                                                          DOT00450
                C                                                                       DOT00460
                C     INITIALIZE  IMAGE DATA TAPE                                       DOT00470
                C                                                                       DOT00480
0023            90    CALL TAPHDR(DATAPE,DATFIL)                                        DOT00490
                C                                                                       DOT00500
0024                  DO A0 I=1,SIZE                                                    DOT00510
0025                  DO A0 J=1,TOTVEC                                                  DOT00520
0026             A0   DATA(I,J) = 0                                                     DOT00530
0027                  TOTVT2 = 0                                                        DOT00540
0028                  TOTVEC = 0                                                        DOT00550
0029             H5     IF(LACIE.EQ.1)CALL FLDLAC(FIELDS,STAMNT,&100,&510,&520,IPT,    DOT00560
                     *VERTEX)                                                           DOT00570
                C                                                                       DOT00580
                C*** ADDED NOV 21, 1978 IN SUPPORT OF LIST PROCESSING                   DOT00590
                C                                                                       DOT00600
0030                       IF (LACIE.GT.1) CALL LISTLC(FIELDS,STAMNT,&100,&510,        DOT00610
                     *     &520,SWCHG,INIT,LACIE,ZERO,IPT,VERTEX)                     DOT00620
0031                  CALL FLDTYP(FIELDS,STAMNT,&100,&510,&520,IPT,VERTEX)              DOT00630
                C                                                                       DOT00640
0032             100  LINSTR = FLDINF(1)                                               DOT00650
0033                  LINEND = FLDINF(2)                                                DOT00660
0034                  LININC = FLDINF(3)                                                DOT00670
0035                  SAMSTR = FLDINF(4)                                                DOT00680
0036                  SAMEND = FLDINF(5)                                                DOT00690
0037                  SAMINC = FLDINF(6)                                                DOT00700
0038                  FIELDS(2,NOFLD2) = NOCAT                                          DOT00710
0039                  TOTVT2 = FIELDS(4,NOFLD2) + TOTVT2                                DOT00720
                C                                                                       DOT00730
0040                  ILINE = (LINEND-LINSTR)/LININC + 1                                DOT00740
0041                  NSAMP = (SAMEND-SAMSTR)/SAMINC + 1                                DOT00750
                C                                                                       DOT00760
```

FILE DOTS

```
               C      POSITION IMAGE TAPE FOR THIS FIELD
               C
  0042                CALL FLDINT(FLDINF(1),FETVC2,NOFET2)
               C
               C      READ A SCAN  LINE OF DATA AND PROCESS IT
               C
  0043                DO 500 LINE=LINSTR,LINEND,LININC
  0044                NLINE = NLINE + 1
               C
  0045                CALL LINERD(IDATA,ENDTAP)
  0046                IF (ENDTAP .EQ. -1) GO TO 600
               C
               C      FIND INTERSECTIONS FOR N-R FIELDS
               C
  0047                CALL FDLINT(VERTEX(IPT),FIELDS(4,NOFLD2),FL,LINE, SAMP,NI)
               C
  0048                DO 400 J=1,NI,2
  0049                IB = (FL(J)-SAMSTR)/SAMINC + 1
  0050                IE = (FL(J+1)- SAMSTR)/SAMINC + 1
  0051                IF (MOD(SAMSTR,SAMINC) .NE. MOD(FL(J),SAMINC)) IB = IB + 1
  0052                IF (IB .GT. IE) GO TO 400
               C
               C
               C      COLLECTING INFO FOR DATA REC FOR DOTFIL
               C
  0053                DO 350 K=IB,IE
               C
  0054                TOTVEC = TOTVEC + 1
  0055                IF(TOTVEC.LE.250) GO TO 110
  0056                TOTVEC=250
               C
  0057                WRITE(6,108)
  0058          108   FORMAT( ////)
  0059                WRITE(6,109)
  0060          109   FORMAT(25X,'**** NOTE - TOTVEC WAS GREATER THAN 250, THEREFORE TOT
                     1VEC WAS SET TO 250 ****')
  0061                GO TO 600
  0062          110   CONTINUE
  0063                DATA(1,TOTVEC) = SAMSTR + SAMINC * (K-1)
  0064                DATA(2,TOTVEC) = LINE
  0065                DATA(3,TOTVEC) = TYPE
  0066                DATA(4,TOTVEC) = NOCAT
  0067                DO 150 I=1,NOFET2
  0068                KK = (I-1)*NSAMP + K
  0069          150   DATA(4+I,TOTVEC) = IDATA(KK)
  0070          350   CONTINUE
               C
  0071          400   CONTINUE
               C
  0072          500   CONTINUE
  0073          600   CONTINUE
  0074                GO TO 85
               C
               C      WRITE DOT DATA FILE
               C
  0075          510   CALL WRTFLD(FIELDS,VERTEX,NOFLD2,2,CATNAM,DUMMY)
  0076                CALL WRTDOT(TOTVEC,NOSUN,FIELDS,VERTEX,SUNANG,DATA,NOCAT,
                     *            CATNAM,SIZE,NOFET2,FETVC2,TOTVT2,NOFLD2,
                     *   DOTUNT,DOTFIL)
  0077                DOTFIL = DOTFIL + 1
  0078                    NOCAT = 0
  0079                SWTCH = 1
  0080                IF(LACIE.NE.0)NOFLD2=0
  0081                IF(PRTKEY.EQ.1)GO TO 530      <-- GO TO 90
               C
               C      %END CARD FOUND
               C
  0082          520   CALL WRTFLD(FIELDS,VERTEX,NOFLD2,2,CATNAM,DUMMY)
  0083                CALL WRTDOT(TOTVEC,NOSUN,FIELDS,VERTEX,SUNANG,DATA,NOCAT,
                     *        CATNAM,SIZE,NOFET2,FETVC2,TOTVT2,NOFLD2 ,
                     *   DOTUNT,DOTFIL)
  0084                SWTCH =  0
  0085                IF(PRTKEY.EQ.1)GO TO 530
               C
               C
               C      ROUTINE TO PRINT DOT DATA RECORD
```

FILE DOTS

```
              C
              C
0086            530 CONTINUE
0087            700 FORMAT(//)
0088            690 FORMAT(1X,' NO.',2X,'SAMPLE',2X,'LINE',2X,'TYPE',2X,'CATEGORY',
                   1     30X,'DATA'/)
0089                ISTART=1
0090                IEND=10
0091            799 CONTINUE
0092                IKT=0
0093                DO 800 II=1,TOTVEC
0094                IKT=IKT+1
0095                IF(IEND.GT.NOFFT2) IEND=NOFFT2
0096                IF(II.NE.1.AND.IKT.EQ.1)WRITE(6,810)
0097            810 FORMAT(1H1,5(/))
0098                IF(IKT.NE.1) GO TO 820
0099                WRITE(6,700)
0100                WRITE(6,690)
0101                WRITE(6,720)(BLANK,FETVC2(I),I=ISTART,IEND)
0102            720 FORMAT(37X,10(A1,'CH(',I2,')'))
0103            820 CONTINUE
0104                WRITE(6,710)II,(DATA(I,II),I=1,4),(DATA(4+JJ,II),JJ=ISTART,IEND)
0105           710  FORMAT(1X,I3,1H.,3X,I4,3X,I4,2X,I2,6X,I2,8X,10(I3,4X))
0106                WRITE(6,712)
0107            712 FORMAT( )
0108            800 CONTINUE
0109                IF(NOFFT2.GT.10) GO TO 830
0110                GO TO 840
0111            830 CONTINUE
0112                IF(ITWO.EQ.1) GO TO 840
0113                ITWO=1
0114                ISTART=IEND+ISTART
0115                IEND=NOFFT2
0116                GO TO 799
0117            840 CONTINUE
0118                IF (SWTCH .EQ. 1) GO TO 90
0119                RETURN
0120                END
```

E.9 SOFTWARE SUBPROGRAM NO. 9 (RDDOTS)

RDDOTS is a utility subprogram in the EOD-LARSYS system which reads the dot files written by the DOTDATA processor.

E.9.1 Linkages

RDDOTS is called by subprograms FILERD, ISOCLS and SETUP3. It calls RDDOT1 and the EOD-LARSYS utility routine FSBSFL.

E.9.2 Interfaces

RDDOTS has included the labeled common block /GLOBAL/.

E.9.3 Inputs

(See the Procedure 1 manual of EOD-LARSYS for description of calling sequence variables).

E.9.4 Outputs

N/A

E.9.5 Storage Requirements

22860

E.9.6 Description

RDDOTS has been modified to support LIST processing in the following way. In the past, if the calling program requested more dot information than was available on the DOT file, an error would be called and execution discontinued. This has been changed. TOTDT3 is compared with TOTDOT, the number of dots on the file. If TOTDT3 is greater than TOTDOT, TOTDT3 is reduced to be TOTDOT.

In some LIST applications, all Type 1 dots will be used as starting dots to ISOCLS. ISOCLS can accept up to 60 starting dots. A standard input dot card to ISOCLS will be set up with 60 dots.

Then, if fewer than 60 Type 1 dots are on the DOTFILE, the number of starting dots will be reduced accordingly.

E.9.7 Flowchart

N/A

E.9.8 Listing

FILE RDDOTS

```
                      C                                                                 RDD00010
                      C     UTILITY ROUTINE THAT READS THE DOTFIL                       RDD00020
                      C                                                                 RDD00030
0001                        SUBROUTINE RDDOTS(DOTS,DOTVEC,TOTDT3,TYPSWT,SIZES,TOTDT2,   RDD00040
                     *        NOCAT,CATNAM,NOFET2,FETVC2,NOFEAT,FETVEC,NOSUN,ANGLE,    RDD00050
                     *  NOFLD,TOTVRT,FLDSAV,VERTEX,/KVAR/)                              RDD00060
0002                        IMPLICIT INTEGER(A-Z)                                       RDD00070
0003                        DIMENSION KVAR(SIZES,1)                                     RDD00080
0004                        REAL KVAR                                                   RDD00090
                      C                                                                 RDD00100
                      C     TYPSWT = 1 -RETURNS SPECTRAL INFO                           RDD00110
                      C     TYPSWT = 2 -RETURNS SPATIAL INFO                            RDD00120
                      C     TYPSWT = 3 -RETURNS BOTH SPATIAL AND SPECTRAL INFO          RDD00130
                      C                                                                 RDD00140
                      C                                                                 RDD00150
0005                        DATA BLANK/' '/                                             RDD00160
                      C     INCLUDE COMPK6,LIST                                         RDD00170
0006                        COMMON /GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,  RDD00180
                     *                HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,  RDD00190
                     *     DUMMAG,DUMMOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL              RDD00200
                     *   ,NMSTUN,NMSTFI,SCTPUN,MAPFIL                                  RDD00210
                     *   ,DOTUNT,DOTFIL,NCMPAS,TRNSFL,RMTRFL,HISTFI,PCHUNT,            RDD00220
                     *   CROUNT,ERTUNT,RANDIO                                          RDD00230
                      C$END                                                             RDD00240
                      C                                                                 RDD00250
0007                        DIMENSION CATNAM(1),DOTVEC(1),DOTS(SIZES,1)                 RDD00260
0008                        DIMENSION FETVEC(30),FETVC3(30),FETVC2(1)                   RDD00270
0009                        DIMENSION FLDSAV(4,1),VERTEX(2,1),ANGLE(1)                  RDD00280
0010                        DIMENSION TEMDOT(5000)                                      RDD00290
                      C                                                                 RDD00300
                      C     READ REC NO. 1 FOR INDICES                                  RDD00310
                      C                                                                 RDD00320
0011                        REWIND DOTUNT                                               RDD00330
0012                        CALL FSASFL(DOTUNT,DOTFIL,ISTAT)                            RDD00340
                      C                                                                 RDD00350
0013                        READ(DOTUNT)NOCAT,NOFEAT,NOFLD,TOTVRT,TOTDOT,NOSUN,(CATNAM(I),  RDD00360
                     *             I=1,NOCAT),SIZE                                    RDD00370
                      C                                                                 RDD00380
                      C     COMPUTE ADDRESSES FOR ARRAY                                 RDD00390
                      C                                                                 RDD00400
0014                        DOTS1  = 1                                                  RDD00410
                      C                                                                 RDD00420
                      C                                                                 RDD00430
                      C     READ REC. NO. 2                                             RDD00440
                      C                                                                 RDD00450
0015                        IF (TYPSWT .EQ. 1) READ(DOTUNT)(FETVEC(I),I=1,NOFEAT)       RDD00460
0016                        IF (TYPSWT .EQ. 2) READ(DOTUNT) DUMMY                       RDD00470
0017                        IF (TYPSWT .EQ. 3) READ(DOTUNT)(FETVEC(I),I=1,NOFEAT)       RDD00480
                     *     ,((FLDSAV(I,J),I=1,4),J=1,NOFLD),((VERTEX(I,J),I=1,2),      RDD00490
                     *     J=1,TOTVRT),(ANGLE(I),I=1,NOSUN)                            RDD00500
                      C                                                                 RDD00510
0018                        IF (TYPSWT .EQ. 2) GO TO 87                                 RDD00520
0019                        IF (NOFET2 .NE. 0) GO TO 60                                 RDD00530
                      C                                                                 RDD00540
                      C     SET DEFAULT CHANNELS                                        RDD00550
                      C                                                                 RDD00560
0020                        DO 50 I=1,NOFEAT                                            RDD00570
0021                        FETVC2(I) = I                                               RDD00580
0022                   50   FETVC3(I) = I                                               RDD00590
0023                        NOFET2 = NOFEAT                                             RDD00600
0024                        GO TO 87                                                    RDD00610
0025                   60   DO 80 J=1,NOFET2                                            RDD00620
0026                        DO 70 K=1,NOFEAT                                            RDD00630
0027                        IF( FETVEC(K) .EQ. FETVC2(J)) GO TO 75                      RDD00640
0028                   70   CONTINUE                                                    RDD00650
0029                        WRITE(6,85)FETVC2(J),(FETVEC(I),I=1,NOFEAT)                 RDD00660
0030                   85   FORMAT(//' CHANNEL ',I2,' IS NOT ON DOTFIL'/ ' CHANNELS ARE',30I3)  RDD00670
0031                        CALL CMERR                                                  RDD00680
0032                   75   FETVC3(J) = K                                               RDD00690
                      C                                                                 RDD00700
0033                   80   CONTINUE                                                    RDD00710
0034                   87   IF (TOTDT3 .EQ. 0) GO TO 96                                 RDD00720
                      C                                                                 RDD00730
                      C***                                                              RDD00740
                      C***    CODE ADDED NOV 21, 1978 FOR LIST PROCESSING               RDD00750
                      C                                                                 RDD00760
```

```
FILE RDDOTS
0035                    IF (TOTDT3.GT.TOTDOT) TOTDT3 = TOTDOT                   RDD00770
                C                                                               RDD00780
0036                    DO 95  J=1,TOTDT3                                       RDD00790
0037                    IF (DOTVEC(J).LE.TOTDOT) GO TO 95                       RDD00800
0038                    WRITE(6,90)DOTVEC(J),TOTDOT                             RDD00810
0039                 90 FORMAT(/ ' DOT NO. ',I3,' IS NOT ON DOTFIL'/ ' FILE CONTAINS ',I3,RDD00820
                       *     ' DOTS' )                                          RDD00830
0040                    CALL CMERR                                              RDD00840
0041                 95 CONTINUE                                                RDD00850
0042                 96 CONTINUE                                                RDD00860
                C                                                               RDD00870
                C                                                               RDD00880
                C       READ REC  NO.3 -- DOTS                                  RDD00890
                C                                                               RDD00900
0043                    TOTAL = SIZE * TOTDOT                                   RDD00910
0044                    READ(DOTUNT)(TEMDOT(DOTS1-1+I),I=1,TOTAL)               RDD00920
                C                                                               RDD00930
0045                    IF (TYPSWT .EQ. 1 ) GO TO 130                           RDD00940
0046                    IF (TYPSWT .EQ. 2 ) GO TO 150                           RDD00950
0047                    IF (TYPSWT .EQ. 3 ) GO TO 180                           RDD00960
                C                                                               RDD00970
                C       RETRIEVE SPECTRAL INFO                                  RDD00980
                C                                                               RDD00990
0048                 130 CONTINUE                                               RDD01000
0049                    SIZES = NOFET2                                          RDD01010
0050                    TOTDT2 = TOTDT3                                         RDD01020
0051                    CALL RDDOT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3,RDD01030
                       * SIZE,TOTDOT,TOTDT3,NOFET2,TYPSWT)                      RDD01040
0052                    RETURN                                                  RDD01050
                C                                                               RDD01060
                C       RETRIEVE SPATIAL INFO                                   RDD01070
                C                                                               RDD01080
0053                 150 CONTINUE                                               RDD01090
0054                    SIZES = 4                                               RDD01100
0055                    TOTDT2 = TOTDOT                                         RDD01110
                C                                                               RDD01120
0056                    CALL RDDOT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3,RDD01130
                       * SIZE,TOTDOT,TOTDT3,NOFET2,TYPSWT)                      RDD01140
0057                    RETURN                                                  RDD01150
                C                                                               RDD01160
0058                 180 CONTINUE                                               RDD01170
                C                                                               RDD01180
                C       RETRIEVE SPECTRAL AND SPATIAL INFO                      RDD01190
                C                                                               RDD01200
0059                    SIZES = SIZE                                            RDD01210
0060                    TOTDT2 = TOTDOT - TOTDT3                                RDD01220
0061                    CALL RDDOT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3,RDD01230
                       * SIZE,TOTDOT,TOTDT3,NOFET2,TYPSWT)                      RDD01240
0062                    RETURN                                                  RDD01250
                C                                                               RDD01260
0063                    END                                                     RDD01270
```

217

E.10 SOFTWARE SUBPROGRAM NO. 10 (SET13)

Subprogram SET13 reads and decodes control card images for the DOTDATA processor.

E.10.1 Linkages

Subprogram SET13 is called by DOTDATA. It calls EOD-LARSYS utility routines ORDER, NXTCHR, FIND12, and NUMBER, and the Fortran re-read routine REREAD.

E.10.2 Interfaces

Subprogram SET13 has included labeled common blocks /INFORM/, /GLOBAL/, and /DOTVEC/.

E.10.3 Inputs

SET13 has no calling sequence provided.

E.10.4 Outputs

N/A

E.10.5 Storage Requirements

3648

E.10.6 Description

Subprogram SET13 has been expanded to read a new OPTION card ___ U unit # in order to specify a Fortran unit number for a LACIE formatted dot file. This file replaces the in-line LACIE formatted dot cards following the *END card in the control card stream. The variable LACIE in /DOTVEC/ then is set to the unit number.

E.10.7 Flowchart

N/A

E.10.8 Listing

```
FILE SET13
                                                                        SET00010
0001                SUBROUTINE SET13                                    SET00020
0002                IMPLICIT INTEGER (A-Z)                              SET00030
0003                DIMENSION CODE(9),CARD(62),EQUCOM(3),ACARD(20)      SET00040
0004                DIMENSION SLASH(2)                                  SET00050
0005                DATA SLASH /1H*,1H*/                                SET00060
0006                DATA CODE/4HCHAN,4HDATA,4HDOTF,                     SET00070
                   * 4HOPTI,4HDATE,4HCOMM,4HEDI ,4HED2 ,4HEND/          SET00080
0007                DATA  EQUCOM/2,1H=,1H,/                             SET00090
0008                DATA D/1HD/,BLNK/1H /,U/1HU/,FF/1HF/,OO/1H0/,P/1HP/ SET00100
0009                DATA L/1HL/                                         SET00110
                C       INCLUDE COMMK1,LIST                             SET00120
                C       INCLUDE COMMK4,LIST                             SET00130
                C       INCLUDE COMMK5,LIST                             SET00140
                C       INCLUDE CMBK14,LIST                             SET00150
0010                COMMON/IFO~M/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2, SET00160
                   *                AVAR2,COVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTX2, SET00170
                   *                FFTVC2(30),SUBVC2(75),SUBPTR(75),CLSVC2(60), SET00180
                   *                KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),  SET00190
                   *                GRPCHK(61),GROUPS(124)                   SET00200
0011                DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)      SET00210
0012                EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),   SET00220
                   2            (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)) SET00230
0013                COMMON/GLOBAL/HEAD(64),MAPTAP,DATAMF,SAVTAP,MAPFILE,RMKEY, SET00240
                   *             HISFIL,HISKEY,TAFORM,ERRPTP,ERRKEY,MAPUNT,NOFILE, SET00250
                   *    DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL   SET00260
                   *  ,NHSTUN,NHSTFI,SCTRUN,MAPFIL                       SET00270
                   *  ;DOTUNT,DOTFIL,NCHPAS,TMASFL,HMTRFL,HISTFL,PCHUNT, SET00280
                   *  CRDUNT,PRTUNT,RANDID                               SET00290
0014                COMMON /DOTVEC/TYPE,CATNAM(50),NOCAT,TOTVEC,FLDINF(6),PRTKEY SET00300
                   *                ,SIZE , LACIE                        SET00310
                C%END                                                   SET00320
0015                ZERO = 0                                            SET00330
0016                NOFET2 = 0                                          SET00340
0017                FIELD  = 1                                          SET00350
0018                PRTKEY = 0                                          SET00360
0019                INPUT=5                                             SET00370
0020                LACIE = 0                                           SET00380
                C                                                       SET00390
0021                WRITE(6,100)                                        SET00400
0022              100 FORMAT(/11X,*INPUT SUMMARY*//)                    SET00410
                C                                                       SET00420
                C       SET UP REREAD BUFFER                            SET00430
                C                                                       SET00440
0023                RRUNIT = 30                                         SET00450
0024                CALL REREAD(RRUNIT,80)                              SET00460
                C                                                       SET00470
                C     PUT CARD IN BUFFER                                SET00480
                C                                                       SET00490
0025              105 READ(21,103)(ACARD(I),I=1,20)                     SET00500
0026              103 FORMAT(2044)                                      SET00510
0027                  WRITE(30,103)(ACARD(I),I=1,20)                    SET00520
0028                  REWIND RRUNIT                                     SET00530
                C                                                       SET00540
0029                  READ(30,110)CODE1,CARD                            SET00550
0030                  REWIND RRUNIT                                     SET00560
0031                  COL = 0                                           SET00570
0032                  WRITE(6,120)CODE1,CARD                            SET00580
0033              120 FORMAT(1X,A4,6X,62A1)                             SET00590
0034              110 FORMAT(A4,6X,62A1)                                SET00600
0035                  DO 130  I=1,9PUT                                  SET00610
0036                  IF (CODE1.EQ.CODE(I)) GO TO (150,180,210,430,370, SET00620
                   *                           390,400,410,420),I       SET00630
0037              130   CONTINUE                                        SET00640
0038              135 WRITE(6,140)                                      SET00650
0039              140 FORMAT( * INVALID CONTROL CARD - IGNORED *)       SET00660
0040                  GO TO 105                                         SET00670
                C                                                       SET00680
                C       CHANNEL CARD                                    SET00690
                C                                                       SET00700
0041              150 M = NXTCHR(CARD,COL)                              SET00710
0042                  IF (M.EQ.0) GO TO 155                             SET00720
0043                  IF (M.EQ.BLNK) GO TO 105                          SET00730
0044              152 WRITE(6,153)                                      SET00740
0045              153 FORMAT(* ERROR ON DATA CARD*)                     SET00750
0046                  GO TO 105                                         SET00760
```

```
FILE SET13

0047          155 J = FIND12(CARD,COL,EQUCOM)                                   SET00770
0048              IF (J .NE. 2) GO TO 152                                       SET00780
0049              NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)                       SET00790
0050              CALL ORDER(FETVC2,NOFET2)                                     SET00800
0051              GO TO 105                                                     SET00810
                C                                                               SET00820
                CC    DATA FILE CARD                                            SET00830
                C                                                               SET00840
0052          180 M = NXTCHR(CARD,COL)                                          SET00850
0053              IF (M .EQ. BLNK ) GO TO 105                                   SET00860
0054              IF (M.EQ.U) GO TO 190                                         SET00870
0055              IF (M.EQ.FF) GO TO 200                                        SET00880
0056          185 WRITE(6,187)                                                  SET00890
0057          187 FORMAT(' ERROR ON DATA FILE CARD')                            SET00900
0058              GO TO 105                                                     SET00910
0059          190 J = FIND12(CARD,COL,EQUCOM)                                   SET00920
0060              IF (J .NE. 2) GO TO 185                                       SET00930
0061              M = NUMBER(CARD,COL,DATAPF,ZERO)                              SET00940
0062              COL = COL - 1                                                 SET00950
0063              GO TO 180                                                     SET00960
0064          200 J = FIND12(CARD,COL,EQUCOM)                                   SET00970
0065              IF ( J .NE. 2 ) GO TO 185                                     SET00980
0066              M = NUMBER(CARD,COL,DATFIL,ZERO)                              SET00990
0067              DATFIL = DATFIL - 1                                           SET01000
0068              COL = COL - 1                                                 SET01010
0069              GO TO 180                                                     SET01020
                C                                                               SET01030
                CC    DOT FILE CARD                                             SET01040
                C                                                               SET01050
0070          210 M = NXTCHR(CARD,COL)                                          SET01060
0071              IF (M.EQ.DD) GO TO 213                                        SET01070
0072              IF (M.EQ.BLNK) GO TO 105                                      SET01080
0073              GO TO 215                                                     SET01090
0074          213 J = FIND12(CARD,COL,SLASH)                                    SET01100
0075              IF (J .EQ. -1) GO TO 215                                      SET01110
0076          214 M = NXTCHR(CARD,COL)                                          SET01120
0077              IF (M .EQ. BLNK ) GO TO 105                                   SET01130
0078              IF (M.EQ.U) GO TO 230                                         SET01140
0079              IF ( M.EQ.FF) GO TO 240                                       SET01150
0080          215 WRITE(6,220)                                                  SET01160
0081          220 FORMAT(' ERROR ON DOT FILE CARD')                             SET01170
0082              GO TO 105                                                     SET01180
0083          230 J = FIND12(CARD,COL,EQUCOM)                                   SET01190
0084              IF ( J .NE. 2)  GO TO 215                                     SET01200
0085              M = NUMBER(CARD,COL,DOTUNT,ZERO)                              SET01210
0086              COL = COL - 1                                                 SET01220
0087              GO TO 214                                                     SET01230
0088          240 J = FIND12(CARD,COL,EQUCOM)                                   SET01240
0089              IF (J .NE.  2) GO TO 215                                      SET01250
0090              M = NUMBER(CARD,COL,DOTFIL,ZERO)                              SET01260
0091              DOTFIL = DOTFIL - 1                                           SET01270
0092              COL = COL - 1                                                 SET01280
0093              GO TO 214                                                     SET01290
                C                                                               SET01300
                CC    OPTION CARD                                               SET01310
                C                                                               SET01320
0094          330 M = NXTCHR(CARD,COL)                                          SET01330
0095              IF (M .EQ. BLNK ) GO TO 105                                   SET01340
0096              IF (M.EQ.P) GO TO 340                                         SET01350
0097              IF(M.EQ.L)GO TO 345                                           SET01360
                C                                                               SET01370
                C*** CODE ADDED NOV 21, 1978 IN SUPPORT OF LIST PROCESSING      SET01380
                C                                                               SET01390
0098                    IF (M.EQ.U) GO TO 350                                   SET01400
                C                                                               SET01410
0099          333 WRITE(6,335)                                                  SET01420
0100          335 FORMAT(' ERROR ON OPTION CARD')                               SET01430
0101              GO TO 105                                                     SET01440
0102          340 PRTKEY = 1                                                    SET01450
0103              GO TO 105                                                     SET01460
0104          345   LACIE = 1                                                   SET01470
0105              GO TO 105                                                     SET01480
0106          350       M = NUMBER(CARD,COL,LACIE,ZERO)                       SET01490
0107              GO TO 105                                                     SET01500
                C                                                               SET01510
                C     DATE CARD                                                 SET01520
```

FILE SFT13

```
                C
0104              370 M = NXTCHR(CARD,COL)                                          SET01530
0109                  IF ( M .EQ. BLNK ) GO TO 105                                  SET01540
0110                  READ(30,390)DATE                                              SET01550
0111              380   FORMAT(10X,15A4)                                           SET01560
0112                  REWIND RRUNIT                                                 SET01570
0113                  GO TO 105                                                     SET01580
                C                                                                   SET01590
                C     COMMENT CARD                                                  SET01600
                C                                                                   SET01610
0114              390 M = NXTCHR(CARD,COL)                                          SET01620
0115                  IF (M .EQ. BLNK ) GO TO 105                                   SET01630
0116                  READ(30,380)COMENT                                            SET01640
0117                  REWIND RRUNIT                                                 SET01650
0118                  GO TO 105                                                     SET01660
                C                                                                   SET01670
                C                                                                   SET01680
                C     HED1                                                          SET01690
0119              400 M = NXTCHR(CARD,COL)                                          SET01700
0120                  READ(30,380) HED1                                             SET01710
0121                  REWIND RRUNIT                                                 SET01720
0122                  GO TO 105                                                     SET01730
                C                                                                   SET01740
                C     HED2                                                          SET01750
                C                                                                   SET01760
0123              410 M = NXTCHR(CARD,COL)                                          SET01770
0124                  READ(30,380) HED2                                             SET01780
0125                  REWIND RRUNIT                                                 SET01790
0126                  GO TO 105                                                     SET01800
                C                                                                   SET01810
                C     *END*                                                         SET01820
                C                                                                   SET01830
0127              420 CONTINUE                                                      SET01840
0128                  IF (NOFET2 .NE. 0) GO TO 440                                  SET01850
0129                  DO 430 I=1,30                                                 SET01860
0130                  FETVC2(I) = I                                                 SET01870
0131              430     CONTINUE                                                  SET01880
0132                  NOFET2 = I                                                    SET01890
                C                                                                   SET01900
0133              440 SIZF = 4 + NOFET2                                             SET01910
                C                                                                   SET01920
                C                                                                   SET01930
                C                                                                   SET01940
0134                  WRITE(6,1000)                                                 SET01950
0135                  IF (NOFET2 .NE. 0) WRITE(6,1010) (FETVC2(I),I=1,NOFET2)       SET01960
0136                  IF (PRTKEY .EQ. 1) WRITE(6,1030)                              SET01970
0137           1040 FORMAT(' LACIE FORMATTED DOT CARDS USED AS EOD-LARSYS FIELD CARDS'SET01980
                     :)                                                             SET01990
0138                  IF(LACIE.EQ.1)WRITE(6,1040)                                   SET02000
0139           1000 FORMAT(//' USER HAS REQUESTED THE FOLLOWING OPTIONS :'/)        SET02010
0140           1010 FORMAT(' SELECTED CHANNELS ARE',3013)                           SET02020
0141           1030 FORMAT(' PRINT DATA VECTORS')                                   SET02030
                C                                                                   SET02040
0142                  RETURN                                                        SET02050
                C                                                                   SET02060
0143                  END                                                           SET02070
```

APPENDIX F

PROGRAM PREPPT

AT PRESENT ALL SOIL LINE NUMBERS ARE SET TO 13.5
SEE SUBROUTINE CALCUL
READAI-COMPARISON OF SEG.NO. 1ST CARD. SEG.NO. REQUESTED 1851 1021
READAI-COMPARISON OF SEG.NO. 1ST CARD. SEG.NO. REQUESTED 1851 1021
EXCESSIVE PROCESSING TIME DUE TO ERROR IN ORDER OF AI FILES
IN PROCEDURE FILE

INPUT IMAGE DATA TAPE INFORMATION

FORMAT	UNIVERSAL
NO. OF CHANNELS	16
NO. OF PIXELS/LINE	196
FIRST SCAN LINE NO.	1
FIRST PIXEL REFERENCE PT	1

RAW DATA TAPE SEG.NO.= 1851

INPUT IMAGE DATA TAPE INFORMATION

FORMAT	UNIVERSAL
NO. OF CHANNELS	16
NO. OF PIXELS/LINE	196
FIRST SCAN LINE NO.	1
FIRST PIXEL REFERENCE PT	1

RAW DATA TAPE SEG.NO.= 1851
SKIPPING FILE 1

INPUT IMAGE DATA TAPE INFORMATION

FORMAT	UNIVERSAL
NO. OF CHANNELS	16
NO. OF PIXELS/LINE	196
FIRST SCAN LINE NO.	1
FIRST PIXEL REFERENCE PT	1

RAW DATA TAPE SEG.NO.= 1021
EXCESSIVE PROCESSING TIME DUE TO ERROR IN ORDER OF RAW DATA FILES
IN PROCEDURE FILE

THE SOIL LINE NUMBERS ARE 0 0 0 0

		1	1	1	IDATA	34	INDEX	1
		1	2	1	IDATA	47	INDEX	20
		1	3	1	IDATA	62	INDEX	39
		1	4	1	IDATA	27	INDEX	58
		1	1	2	IDATA	32	INDEX	77
		1	2	2	IDATA	51	INDEX	96
		1	3	2	IDATA	41	INDEX	115
		1	4	2	IDATA	24	INDEX	134
		1	1	3	IDATA	44	INDEX	153
		1	2	3	IDATA	58	INDEX	172
		1	3	3	IDATA	71	INDEX	191
		1	4	3	IDATA	26	INDEX	210
		1	1	4	IDATA	44	INDEX	229
		1	2	4	IDATA	54	INDEX	248
		1	3	4	IDATA	63	INDEX	267
		1	4	4	IDATA	23	INDEX	286
	LDATA	38	47	62	27	SOILL	0	
1	1	1	8BIT	89.9716	MEANB		75.3179 SDB	15.1790
1	1	1	GRN	8.9691	MEANG		12.6880 SDG	7.6490
	IDATA	32	51	41	24	SOILL	0	
1	1	2	8BIT	75.4100	MEANB		75.5519 SDB	16.4190
1	1	2	GRN	-5.2638	MEANG		12.7360 SDG	8.9150
	LDATA	46	58	71	26	SOILL	0	
1	1	3	8BIT	105.0843	MEANB		82.0000 SDB	21.0600
1	1	3	GRN	4.2443	MEANG		8.7400 SDG	6.7200
	LDATA	44	54	63	23	SOILL	0	
1	1	4	8BIT	95.8122	MEANB		89.8000 SDB	27.9000
1	1	4	GRN	1.8695	MEANG		5.4600 SDG	7.4000

EXCESSIVE PROCESSING TIME DUE TO ERROR IN SPECIFYING ORDER OF FILES IN PROCEDURE FILE

GROUND TRUTH HEADER

SITE = 1021 TYPE = GT

A. I. HEADER RECORDS
1021 1 6255 6361 7159 7194A 1021
7194 14 24 58 60H 1021

EXCESSIVE PROCESSING TIME DUE TO ERROR IN ORDER OF AI FILES
IN PROCEDURE FILE

INPUT IMAGE DATA TAPE INFORMATION

FORMAT UNIVERSAL
NO. OF CHANNELS 16
NO. OF PIXELS/LINE 196
FIRST SCAN LINE NO. 1
FIRST PIXEL OFFSET OF 1
RAW DATA TAPE SEG. ID.= 1851
THE SCAN LINE NUMBERS ARE 16 20 16 15

	1	1	1	IDATA	14	INDEX	1
	1	2	1	IDATA	12	INDEX	20
	1	3	1	IDATA	49	INDEX	39
	1	4	1	IDATA	25	INDEX	58
	1	1	2	IDATA	11	INDEX	77
	1	2	2	IDATA	14	INDEX	96
	1	3	2	IDATA	21	INDEX	115
	1	4	2	IDATA	9	INDEX	134
	1	1	3	IDATA	27	INDEX	153
	1	2	3	IDATA	31	INDEX	172
	1	3	3	IDATA	39	INDEX	191
	1	4	3	IDATA	18	INDEX	210
	1	1	4	IDATA	31	INDEX	229
	1	2	4	IDATA	28	INDEX	248
	1	3	4	IDATA	58	INDEX	267
	1	4	4	IDATA	26	INDEX	286

IDATA 14 12 49 25 SUILL 16

1	1	1	3211	51.5744	MEANB	75.6700 SDB	17.0400
1	1	1	3211	30.6061	MEANG	12.7600 SDG	9.5500

IDATA 11 14 21 9 SUILL 20

1	1	2	3211	28.6667	MEANB	75.6700 SDB	17.0400
1	1	2	3211	7.7576	MEANG	12.7600 SDG	9.5500

IDATA 27 31 39 18 SUILL 15

1	1	3	3211	57.0934	MEANB	75.6700 SDB	17.0400
1	1	3	3211	7.2424	MEANG	12.7600 SDG	9.5500

IDATA 31 28 58 26 SUILL 15

1	1	4	3211	73.2193	MEANB	89.8000 SDB	27.9000
1	1	4	3211	20.6061	MEANG	5.4600 SDG	7.4000

EXCESSIVE PROCESSING TIME DUE TO ERROR IN SPECIFYING ORDER OF FILES IN PROCEDURE FILE

GROUND TRUTH HEADER
SITE = 1851 TYPE = GT

A. I. HEADER RECORDS
1851 1 6255 6361 7087 7193A 1851
7193 23 24 26 60H 1851

END OF PROGRAM 0 AI ERRORS DETECTED
0 RAW DATA SEGMENTS MISSING
0 GROUND TRUTH ERRORS

32K

```
 THE PATTERSON-PITT-THADANI ALGORITHM PROGRAM

JSC740       C AHLERS              DEC  11,1978    10 38 23 AM

PFLAG=     0

RUNIT=     3      WUNIT=     7

NO. OF CHANNELS=    25
NO. OF CLASSES=     2


  CATEGORY         INDEX
      S                1
      2                2
      7                0

THE COST MATRIX

 0.0  1.00

 1.00 0.0

NDATA=    25

FEATURE VECTOR FORMAT

(2I2.2A1.I1.4I1 ,A4,4(/, 8F9.4))


THE FEATURE INDEX VECTOR
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

N1=  351      N2=62127
 TERMS FOR THE PHI FUNCTION
LINE                 ALL
*END

 1021 1 6255 6363 7159 7194A                                          1021

 7194      18   20   58   608                                        1021

 1851 1 6289 6361 7067 7193A                                          1851

 7193      23   24   26   608                                        1851

THE NO. OF SAMPLES =       414


 THE Q MATRIX

-0.20863356220-01     0.20863356220-01
-0.28845603090-01     0.28845603090-01
-0.25431066300-01     0.25431066300-01
 0.38897522750 00    -0.38897522750 00
-0.33067156800-01     0.33067156800-01
 0.31103474720 00    -0.31103474720 00
```

0.2862384802D-01	-0.2862384802D-01
0.3945958645D-01	-0.3945958645D-01
0.5636819017D-02	-0.5636819017D-02
-0.2700422932D-02	0.2700422932D-02
-0.7623586598D-02	0.7623586598D-02
-0.1950007050D-02	0.1950007050D-02
0.9153701453D-02	-0.9153701453D-02
0.4160882415D-02	-0.4160882415D-02
0.6046002611D-02	-0.6046002611D-02
-0.3144986432D-01	0.3144986432D-01
-0.6009464851D-01	0.6009464851D-01
0.1341371273D-01	-0.1341371273D-01
0.6104045957D-01	-0.6104045957D-01
0.2117696142D 00	-0.2117696142D 00
0.1949607172D 00	-0.1949607172D 00
0.7966901089D-01	-0.7966901089D-01
0.3059353812D 00	-0.3059353812D 00
0.3217423361D-01	-0.3217423361D-01
-0.2329875748D-01	0.2329875748D-01
0.5217017031D-01	0.9478298297D 00

CATFGORY	OCCURANCES
N	290
S	105

4 PIXELS WITHOUT GROUND TRUTH

4 UNUSED GROUND TRUTH SYMBOLS

THF LOSS VECTOR MATRIX

0.2086335622D-01	-0.2086335622D-01
0.2884560309D-01	-0.2884560309D-01
0.2543106630D-01	-0.2543106630D-01
-0.3889752275D 00	0.3889752275D 00
0.3306715880D-01	-0.3306715880D-01
-0.3110347472D 00	0.3110347472D 00
-0.2862384802D-01	0.2862384802D-01
-0.3945958645D-01	0.3945958645D-01
-0.5636819017D-02	0.5636819017D-02
0.2700422932D-02	-0.2700422932D-02
0.7623586598D-02	-0.7623586598D-02
0.1950007050D-02	-0.1950007050D-02
-0.9153701453D-02	0.9153701453D-02
-0.4160882415D-02	0.4160882415D-02
-0.6046002611D-02	0.6046002611D-02
0.3144986432D-01	-0.3144986432D-01
0.6009464851D-01	-0.6009464851D-01
-0.1341371273D-01	0.1341371273D-01
-0.6104045957D-01	0.6104045957D-01
-0.2117696142D 00	0.2117696142D 00
-0.1949607172D 00	0.1949607172D 00
-0.7966901089D-01	0.7966901089D-01
-0.3059353812D 00	0.3059353812D 00
-0.3217423361D-01	0.3217423361D-01
0.2329875748D-01	-0.2329875748D-01
0.9478298297D 00	0.5217017031D-01

TRAINING OVER

LOSS VECTOR MATRIX RESIDES ON UNIT 7

M,D,T 26 25 2

UPPER BOUND ON BAYES RISK 0.8170022914D 00

TIME FOR PPTA 0.749

```
 THE PATTERSON-PITT-THADANI CLASSIFIER PROGRAM

JSC740       C AHLERS             DEC  11,1978    11 21 32 AM

LOSS VECTOR MATRIX UNIT NO. =     7

M =    26

NO. OF CHANNELS=   25
NO. OF CLASSES=    2

     CATEGORY     INDEX
       N            2
       S            1

THE TERM KEY FOR THE PHI FUNCTION
 00000000000000000000000000000000000000000000000000
 00000000000000000000000000000000000000000000000000
 00000000000000000000000000000000000000000000000000
 00000000000000000000000000000000000000000000000000
 00000000000000000000000000000000000000000000000000
 00000000000000000000000000000000000000000000000000
 00000000000000000000000011111111111111111111111111
 1

THE COST MATRIX

 0.0  1.00

 1.00 0.0

NDATA=   25

FEATURE VECTOR FORMAT

(2I2,2A1,I1,4I1 ,A4,4(/, 8F9.4))


THE FEATURE INDEX VECTOR
  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

N1=  351

PFLAG=    0

 LOSS VECTOR MATRIX
 0.2086335622D-01   -0.2086335622D-01
 0.2884560309D-01   -0.2884560309D-01
 0.2543106630D-01   -0.2543106630D-01
-0.3889752275D 00    0.3889752275D 00
 0.3306715688D-01   -0.3306715688D-01
-0.3110347472D 00    0.3110347472D 00
-0.2862384802D-01    0.2862384802D-01
-0.3945958645D-01    0.3945958645D-01
-0.5636819017D-02    0.5636819017D-02
 0.2700422932D-02   -0.2700422932D-02
 0.7623586598D-02   -0.7623586598D-02
 0.1950007050D-02   -0.1950007050D-02
-0.9153701453D-02    0.9153701453D-02
-0.4160882415D-02    0.4160882415D-02
-0.6046002611D-02    0.6046002611D-02
 0.3144986432D-01   -0.3144986432D-01
 0.6009464851D-01   -0.6009464851D-01
-0.1341371273D-01    0.1341371273D-01
-0.6104046997D-01    0.6104046997D-01
```

-0.2117696142D 00	0.2117696142D 00
-0.1949607172D 00	0.1949607172D 00
-0.7966901089D-01	0.7966901089D-01
-0.3059353812D 00	0.3059353812D 00
-0.3217423361D-01	0.3217423361D-01
0.2329875748D-01	-0.2329875748D-01
0.9478798297D 00	0.5217017031D-01

HEADER RECORDS

1021 1 6255 6363 7159 7194A

7194 18 20 58 608

THE NO. OF SAMPLES = 209

...OVERALL CLASSIFICATION RESULTS...

183 PIXELS CLASSIFIED AS S

26 PIXELS CLASSIFIED AS N

4 PIXELS WITHOUT GROUND TRUTH

3 UNUSED GROUND TRUTH SYMBOLS

SUMMARY TABLE FOR GT VS. PPT

	S	N	
S	59.	8.	67.
N	117.	18.	135.
	176.	26.	202.

SUMMARY TABLE FOR AI VS. PPT

	S	N	
S	45.	5.	50.
N	114.	17.	131.
	159.	22.	181.

SUMMARY TABLE FOR GT VS. AI

	S	N	
S	45.	13.	58.
N	2.	116.	118.
	47.	129.	176.

PROBABILITY TABLE FOR GT VS. PPT

	S	N	
S	0.29208	0.03960	0.33168
N	0.57921	0.08911	0.66832
	0.87129	0.12871	1.00000

1021

1021

229

THE ESTIMATE OF THE PROBABILITY OF CORRECT LABELING IS 0.38119

PROBABILITY TABLE FOR AI VS. PPT

	S	N	
S	0.24862	0.02762	0.27624
N	0.62983	0.09392	0.72376
	0.87845	0.12155	1.00000

THE ESTIMATE OF THE PROBABILITY OF CORRECT LABELING IS 0.34254

PROBABILITY TABLE FOR GT VS. AI

	S	N	
S	0.25568	0.07386	0.32955
N	0.01196	0.65909	0.67045
	0.26765	0.73295	1.00000

THE ESTIMATE OF THE PROBABILITY OF CORRECT LABELING IS 0.91477

CLASSIFIED DOT LABELS

CATEGORIES FOUND: NS

NwRFm1= 0

LACIE FORMAT DOT LABELS
WRITTEN TO UNIT 27

DOT	1	N	4	6	8	10	12	14	16	18	20	22	42	52	80		
DOT	1	S	2	24	26	28	30	32	34	36	38	40	44	46	48	50	54
DOT	1	S	56	58	60	62	64	66	68	70	72	74	76	78	82	84	86
DOT	1	S	88	90	92	94	96	98	100	102	104	106	108	110	112	114	116
DOT	1	S	120	122	124	126	128	130	132	134	136	138	140	142	144	146	148
DOT	1	S	150	152	154	156	158	160	162	164	166	168	170	172	174	176	178
DOT	1	S	180	182	184	186	188	190	192	194	196	198	200	202	204	206	208
DOT	2	N	3	5	7	9	11	13	15	17	19	21	43	51	57		
DOT	2	S	1	23	25	27	29	31	33	35	37	39	41	45	47	49	53
DOT	2	S	55	59	61	63	65	67	69	71	73	75	77	79	81	83	85
DOT	2	S	87	89	91	93	95	97	99	101	103	105	107	109	111	113	115
DOT	2	S	116	117	119	121	123	125	127	129	131	133	135	137	139	141	143
DOT	2	S	145	147	149	151	153	155	157	159	161	163	165	167	169	171	173
DOT	2	S	175	177	179	181	183	185	187	189	191	193	195	197	199	201	203
DOT	2	S	205	207	209												

AI DOT LABELS

CATEGORIES FOUND: D4NRS

LACIE FORMAT DOT LABELS
#RITTEN TO UNIT 28

DOT	1	D	6	84	112	124	132	136	200	206	208					
DOT	1	M	204													
DOT	1	N	2	8	12	14	16	22	24	26	30	32	34	36	38	42	48
DOT	1	N	50	52	54	56	58	62	64	66	68	70	76	78	80	82	88
DOT	1	N	90	94	100	104	106	108	110	118	122	130	134	138	140	142	144
DOT	1	N	146	148	150	152	154	156	160	162	164	166	168	174	176	180	182
DOT	1	N	184	186	190	192	196										
DOT	1	R	4	40	202												
DOT	1	S	10	18	20	28	44	46	60	72	74	86	92	96	98	102	114
DOT	1	S	120	126	128	158	170	172	178	188	194	198					
DOT	2	D	43	67	85	97	133	207									
DOT	2	M	11	99	107	116	143	165	197								
DOT	2	N	1	3	5	7	9	13	15	21	23	25	29	31	33	35	37
DOT	2	N	49	51	53	55	57	61	63	65	69	71	75	79	83	87	89
DOT	2	N	93	101	103	105	111	113	115	117	119	121	123	125	135	137	139
DOT	2	N	141	147	149	151	153	155	159	163	167	173	175	179	181	183	185
DOT	2	N	187	191	195	201	205	209									
DOT	2	R	95	109													
DOT	2	S	17	19	27	39	41	45	47	59	73	77	81	91	127	129	131
DOT	2	S	145	157	161	169	171	177	189	193	199	203					

GROUND TRUTH VS A. I. LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		S/N		S/R		N/D		N/N		S/S		N/N		N/N		N/N		S/S	
20	S/S		N/N		N/N		N/N		S/S		N/N		N/N		N/N		N/N		S/N
30		S/R		N/N		S/S		S/S		N/N		S/N		N/N		N/N		S/N	
40	N/N		S/S		N/N		S/N		N/N		N/N		N/N		S/S		S/S		N/N
50		N/N		N/N		N/N		N/D		S/S		N/N		N/N		S/S		N/N	
60	S/S		S/S		N/N		S/S		N/N		N/N		N/N		N/N		N/D		S/S
70				N/N		/S		N/N		S/D		S/S		S/S		N/N		N/D	
80	N/N		N/D		N/N		S/N		S/N		N/N		N/N		N/N		N/N		N/N
90		S/N		N/N		S/S		N/N		N/N		S/N		N/N		N/N		S/S	
100	S/S		N/N		/N		S/S		N/N		N/N		N/N		N/N		S/S		N/N
110		N/N		/S		N/N		S/S		N/D		N/R		S/M		N/D		N/D	

GROUND TRUTH VS A. I. LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/N		N/N		N/N		N/N		S/N		N/M		N/N		N/N		S/S		S/S
20		N/N		N/N		N/N		S/S		N/N		N/N		N/N		N/N		N/N	
30	S/S		S/S		N/D		S/S		S/S		N/N		S/N		N/N		S/N		N/N
40		S/S		N/N		N/N		N/N		N/D		N/N		N/N		N/S		/N	
50	M/S		N/N		S/S		N/N		N/D		N/N		N/N		S/S		N/N		S/R
60		N/D		S/M		N/N		N/N		N/N		N/M		S/R		N/N		N/N	
70	N/N	M/M	N/N		N/N		N/N		N/N		N/N		S/S		S/S		S/S		N/D
80		N/N		N/N		S/N		N/N		S/M		S/S		N/N		N/N		N/N	
90	N/N		N/N		S/S		N/N		S/S		N/N		M/M		N/N		S/S		N/S
100		N/N		N/N		S/S		N/N		N/N		N/N		N/N		N/N		S/S	
110	N/N		S/S		N/N		S/M		S/S		N/N		S/S		N/N		N/D		N/N

GROUND TRUTH VS CLASSIFIED LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		S/S		S/N		N/N		N/N		S/N		N/N		N/N		N/N		S/N	
20	S/N		N/N		N/S		N/S		S/S		N/S		N/S		N/S		N/S		S/S
30		S/S		N/N		S/S		S/S		N/S		S/S		N/N		N/S		S/S	
40	N/S		S/S		N/S		S/S		N/S		N/S		N/S		S/S		S/S		N/S
50		N/S		N/N		N/S		N/S		S/S		N/S		N/S		S/S		N/S	
60	S/S		S/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S		S/S
70				N/S		/S		N/S		S/S		S/S		S/S		N/S		N/S	
80	N/S		N/S		N/S		S/S		S/S		N/S		N/S		N/S		N/S		N/S
90		S/S		N/S		S/S		N/S		N/S		S/S		N/S		N/S		S/S	
100	S/S		N/S		/S		S/S		N/S		N/S		N/S		N/S		S/S		N/S
110		N/S		/S		N/S		S/S		N/S		N/S		S/S		N/S		N/S	

GROUND TRUTH VS CLASSIFIED LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/S		N/N		N/N		N/N		S/N		N/N		N/N		N/N		S/N		S/N
20		N/N		N/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S	
30	S/S		S/S		N/N		S/S		S/S		N/S		S/N		N/S		S/S		N/N
40		S/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S		/S	
50	N/S		N/S		S/S		N/S		N/S		N/S		N/S		S/S		N/S		S/S
60		N/S		S/S		N/S		N/S		N/S		N/S		S/S		N/S		N/S	
70	N/S	N/S	N/S		N/S		N/S		N/S		N/S		S/S		S/S		S/S		N/S
80		N/S		N/S		S/S		N/S		S/S		S/S		N/S		N/S		N/S	
90	N/S		N/S		S/S		N/S		S/S		N/S		N/S		N/S		S/S		N/S
100		N/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S		S/S	
110	N/S		S/S		N/S		S/S		S/S		N/S		S/S		N/S		N/S		N/S

A. I. LABELS VS CLASSIFIED LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		N/S		R/N		D/N		N/N		S/N		N/N		N/N		N/N		S/N	
20	S/N		N/N		N/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S
30		R/S		N/N		S/S		S/S		N/S		N/S		N/N		N/S		N/S	
40	N/S		S/S		N/S		N/S		N/S		N/S		N/S		S/S		S/S		N/S
50		N/S		N/N		N/S		D/S		S/S		N/S		N/S		S/S		N/S	
60	S/S		S/S		N/S		S/S		N/S		N/S		N/S		N/S		D/S		S/S
70				N/S		S/S		N/S		D/S		S/S		S/S		N/S		D/S	
80	N/S		D/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S
90		N/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S		S/S	
100	S/S		N/S		N/S		S/S		N/S		N/S		N/S		N/S		S/S		N/S
110		N/S		S/S		N/S		S/S		D/S		R/S		H/S		D/S		D/S	

A. I. LABELS VS CLASSIFIED LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/S		N/N		N/N		N/N		N/N		M/N		N/N		N/N		S/N		S/N
20		N/N		N/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S	
30	S/S		S/S		D/N		S/S		S/S		N/S		N/N		N/S		N/S		N/N
40		S/S		N/S		N/S		N/S		D/S		N/S		N/S		S/S		N/S	
50	S/S		N/S		S/S		N/S		D/S		N/S		N/S		S/S		N/S		R/S
60		D/S		M/S		N/S		N/S		N/S		M/S		R/S		N/S		N/S	
70	N/S	M/S	N/S		N/S		N/S		N/S		N/S		S/S		S/S		S/S		D/S
80		N/S		N/S		N/S		N/S		M/S		S/S		N/S		N/S		N/S	
90	N/S		N/S		S/S		N/S		S/S		N/S		M/S		N/S		S/S		S/S
100		N/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S		S/S	
110	N/S		S/S		N/S		M/S		S/S		N/S		S/S		N/S		D/S		N/S

HEADER RECORDS

1851 1 6289 6361 7067 7193A 1851

7193 23 24 26 608 1851

THE NO. OF SAMPLES = 205

...OVERALL CLASSIFICATION RESULTS...

197 PIXELS CLASSIFIED AS S

8 PIXELS CLASSIFIED AS N

12 UNUSED GROUND TRUTH SYMBOLS

SUMMARY TABLE FOR GT VS. PPT

	S	N	
S	37.	1.	38.
N	148.	7.	155.
	185.	8.	193.

SUMMARY TABLE FOR AI VS. PPT

	S	N	
S	54.	1.	55.
N	47.	2.	49.
	101.	3.	104.

SUMMARY TABLE FOR GT VS. AI

	S	N	
S	30.	1.	31.
N	21.	47.	68.
	51.	48.	99.

PROBABILITY TABLE FOR GT VS. PPT

	S	N	
S	0.19171	0.00518	0.19689
N	0.76684	0.03627	0.80311
	0.95855	0.04145	1.00000

THE ESTIMATE OF THE PROBABILITY OF CORRECT LABELING IS 0.22798

PROBABILITY TABLE FOR AI VS. PPT

	S	N	
S	0.51923	0.00962	0.52885
N	0.45192	0.01923	0.47115
	0.97115	0.02885	1.00000

THE ESTIMATE OF THE PROBABILITY OF CORRECT LABELING IS 0.53846

PROBABILITY TABLE FOR GT VS. AI

	S	N	
S	0.30303	0.01010	0.31313
N	0.21212	0.47475	0.68687
	0.51515	0.48485	1.00000

THE ESTIMATE OF THE PROBABILITY OF CORRECT LABELING IS 0.77778

CLASSIFIED DOT LABELS

CATEGORIES FOUND: NS

NWKFM1= 1

LACIE FORMAT DOT LABELS
WRITTEN TO UNIT 27

DOT	1	N	148	166												
DOT	1	S	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
DOT	1	S	34	40	42	44	46	48	50	52	54	56	58	60	62	64	66
DOT	1	S	68	70	72	74	76	78	80	82	84	86	88	90	92	94	96
DOT	1	S	98	100	102	104	106	110	112	114	116	118	120	122	124	126	130
DOT	1	S	132	134	136	138	140	142	144	146	150	152	154	156	158	160	162
DOT	1	S	168	170	172	174	176	178	180	182	184	186	188	190	192	194	196
DOT	1	S	198	204	206												
DOT	2	N	135	137	141	145	147	153									
DOT	2	S	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29
DOT	2	S	31	32	33	35	39	41	43	45	47	49	51	53	55	57	59
DOT	2	S	61	63	65	67	69	71	73	75	77	79	81	83	85	87	89
DOT	2	S	91	93	95	97	99	101	103	105	107	108	109	111	113	115	117
DOT	2	S	119	121	123	125	127	128	129	131	133	139	143	149	151	155	157
DOT	2	S	159	161	163	165	167	169	171	173	175	177	179	181	183	185	187
DOT	2	S	189	191	193	195	197	199	200	201	202	203	205	207	208	209	

AI DOT LABELS

CATEGORIES FOUND: DMNRS

LACIE FORMAT DOT LABELS
WRITTEN TO UNIT 28

DOT 1 D	2	4	8	10	14	16	18	20	22	34	40	42	50	54	58
DOT 1 U	74	94	98	136	138	144	152	166	168	170	178	186	190	206	
DOT 1 M	6	28	44	48	60	66	76	78	80	100	110	118	124	130	172
DOT 1 M	174	176	180	182											
DOT 1 N	26	52	62	64	72	82	88	104	106	112	132	140	148	150	154
DOT 1 N	188	192	194	196											
DOT 1 W	12														
DOT 1 S	24	30	46	56	68	70	84	86	90	92	96	102	114	116	120
DOT 1 S	122	126	134	142	146	156	158	160	162	184	198	204			
DOT 2 D	1	5	9	11	13	15	17	19	21	33	35	41	43	51	53
DOT 2 D	55	73	75	107	131	133	139	141	145	149	151	155	157	159	163
DOT 2 D	165	167	169	179	181	187	205	207							
DOT 2 M	7	29	59	61	63	71	77	109	128	137	147	197	200	208	
DOT 2 N	3	23	39	49	57	65	67	79	81	85	87	91	95	108	111
DOT 2 N	113	119	125	127	129	135	143	171	177	185	191	193	195	203	209
DOT 2 S	25	27	31	32	45	47	69	83	89	93	97	99	101	103	105
DOT 2 S	115	117	121	123	153	161	173	175	183	189	199	201	202		

GROUND TRUTH VS A. I. LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		N/D		N/D		N/M		N/D		N/D		S/R		N/D		N/D		N/D	
20	N/D		N/D		N/S		N/N		N/M		N/S				N/D				
30		S/D		N/D		N/M		N/S		N/M		N/D		N/N		N/D		S/S	
40	N/D		N/M		N/N		N/N		N/M		S/S		S/S		N/N		N/D		N/M
50		N/M		N/M		N/N		S/S		S/S		N/N		N/S		S/S		N/D	
60	N/S		N/D		N/M		S/S		N/N		N/N				N/M		N/N		S/S
70		S/S		N/M		S/S		S/S		N/M		N/S				N/M		N/N	
80	S/S		S/D		N/D		N/N		S/S		N/D		S/S		N/N		N/N		N/D
90		N/N		S/S		S/S		S/S		S/S				N/D		N/D		N/D	
100	S/M		N/M		N/M		N/D		N/M		N/M		S/S		N/D		N/N		N/D
110		N/N		N/N		N/N		N/S						N/S		N/D			

GROUND TRUTH VS A. I. LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/D		N/N		N/D		N/M		N/D		N/D		N/D		N/D		N/D		N/D
20		N/D		N/N		N/S		N/S		N/M		M/S	M/S	N/D		N/D			
30	N/N		N/D		N/D		N/S		N/S		N/N		N/D		N/D		N/D		N/N
40		R/M		N/M		S/M		N/N		N/N		S/S		M/M		N/D		N/D	
50	N/M		N/N		N/N		S/S		N/N		N/N		M/S		N/N		S/S		N/N
60		N/S		N/S		S/S		N/S		N/S		N/D	M/N	S/M		N/N		N/N	
70	N/S		S/S		N/N		S/S		N/S		N/N		N/N	M/M	S/N		N/D		M/D
80		N/N		N/M		N/D		N/D		N/N		S/D		N/M		N/D		N/D	
90	N/S		N/D		N/D		N/D		S/S		N/D		N/D		N/D		N/D		N/N
100		N/S		S/S		N/N		N/D		N/D		S/S		N/N		N/D		S/S	
110	N/N		N/N		N/N		N/M		S/S	R/M	N/S	M/S	N/N		N/D		M/D	M/M	N/N

GROUND TRUTH VS CLASSIFIED LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		N/S		N/S		N/S		N/S		N/S		S/S		N/S		N/S		N/S	
20	N/S		N/S		N/S		N/S		N/S		N/S				N/S				
30		S/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S		S/S	
40	N/S		N/S		N/S		N/S		N/S		S/S		S/S		N/S		N/S		N/S
50		N/S		N/S		N/S		S/S		S/S		N/S		N/S		S/S		N/S	
60	N/S		N/S		N/S		S/S		N/S		N/S				N/S		N/S		S/S
70		S/S		N/S		S/S		S/S		N/S		N/S				N/S		N/S	
80	S/S		S/S		N/S		N/S		S/S		N/S		S/S		N/N		N/S		N/S
90		N/S		S/S		S/S		S/S		S/S				N/N		N/S		N/S	
100	S/S		N/S		N/S		N/S		N/S		N/S		S/S		N/S		N/S		N/S
110		N/S		N/S		N/S		N/S						N/S		N/S			

GROUND TRUTH VS CLASSIFIED LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S
20		N/S		N/S		N/S		N/S		N/S		M/S	M/S	N/S		N/S			
30	N/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S		N/S
40		R/S		N/S		S/S		N/S		N/S		S/S		M/S		N/S		N/S	
50	N/S		N/S		N/S		S/S		N/S		N/S		M/S		N/S		S/S		N/S
60		N/S		N/S		S/S		N/S		N/S		N/S	M/S	S/S		N/S		N/S	
70	N/S		S/S		N/S		S/S		N/S		N/S		N/S	M/S	S/S		N/S		M/S
80		N/N		N/N		N/S		N/N		N/S		S/N		N/N		N/S		N/S	
90	N/N		N/S		N/S		N/S		S/S		N/S		N/S		N/S		N/S		N/S
100		N/S		S/S		N/S		N/S		N/S		S/S		N/S		N/S		S/S	
110	N/S		N/S		N/S		N/S		S/S	R/S	N/S	M/S	N/S		N/S		M/S	M/S	N/S

A. I. LABELS VS CLASSIFIED LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		D/S		D/S		M/S		D/S		D/S		R/S		D/S		D/S		D/S	
20	D/S		D/S		S/S		N/S		M/S		S/S				D/S				
30		D/S		D/S		M/S		S/S		M/S		D/S		N/S		D/S		S/S	
40	D/S		M/S		N/S		N/S		M/S		S/S		S/S		N/S		D/S		M/S
50		M/S		M/S		N/S		S/S		S/S		N/S		S/S		S/S		D/S	
60	S/S		D/S		M/S		S/S		N/S		N/S				M/S		N/S		S/S
70		S/S		M/S		S/S		S/S		M/S		S/S				M/S		N/S	
80	S/S		D/S		D/S		N/S		S/S		D/S		S/S		N/N		N/S		D/S
90		N/S		S/S		S/S		S/S		S/S				D/N		D/S		D/S	
100	M/S		M/S		M/S		D/S		M/S		M/S		S/S		D/S		N/S		D/S
110		N/S		N/S		N/S		S/S						S/S		D/S			

A. I. LABELS VS CLASSIFIED LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	D/S		N/S		D/S		M/S		D/S		D/S		D/S		D/S		D/S		D/S
20		D/S		N/S		S/S		S/S		M/S		S/S	S/S	D/S		D/S			
30	N/S		D/S		D/S		S/S		S/S		N/S		D/S		D/S		D/S		N/S
40		M/S		M/S		M/S		N/S		N/S		S/S		M/S		D/S		D/S	
50	M/S		N/S		N/S		S/S		N/S		N/S		S/S		N/S		S/S		N/S
60		S/S		S/S		S/S		S/S		S/S		D/S	N/S	M/S		N/S		N/S	
70	S/S		S/S		N/S		S/S		S/S		N/S		N/S	M/S	N/S		D/S		D/S
80		N/N		M/N		D/S		D/N		N/S		D/N		M/N		D/S		D/S	
90	S/N		D/S		D/S		D/S		S/S		D/S		D/S		D/S		D/S		N/S
100		S/S		S/S		N/S		D/S		D/S		S/S		N/S		D/S		S/S	
110	N/S		N/S		N/S		M/S		S/S	M/S	S/S	S/S	N/S		D/S		D/S	M/S	N/S

TIME FOR PPTC 0.432

```
[illegible]             1
[illegible]TA
[illegible]FILE   [illegible] IT=11,FILE=1
[illegible]FILE   [illegible]UT/UNIT=19,FILE=1
[illegible]       [illegible]=1,2,3,4
[illegible]       [illegible]
[illegible]T      [illegible] 1 RUN
[illegible]       [illegible]
[illegible]
[illegible] 1  [illegible] 24 30 34 [illegible] 98 100 106 108 110 150
[illegible] 1  [illegible] 172 [illegible]
[illegible] 1  [illegible] 20 22 26 28 32 34 36 58 60 62 64 70 72 76
[illegible] 1  [illegible] [illegible] 112 114 134 138 140 142 144
[illegible] 1  [illegible] 152 174 176 178 186 188 190
[illegible] 2  [illegible] [illegible] 11 15 17 43 45 49 57 79 83 89 95 117
[illegible] 2  [illegible] 119 127 133 155 159 193 195 197 201
[illegible] 2  [illegible] [illegible] 13 [illegible] 39 41 47 51 53 55 77 81 85 87 91
[illegible] 2  [illegible] [illegible] 115 121 123 125 129 131 153 157 161 163
[illegible] 2  [illegible] 16 [illegible] 167 169 171 191 199 203 205 207 209
[illegible]
[illegible]
[illegible]FILE   [illegible] IT=11,FILE=1
[illegible]FILE   [illegible] IT=19,FILE=1
ST[illegible]FILE  [illegible]IT=20,FILE=1
[illegible]       [illegible]=1,2,3,4
[illegible]       [illegible]=1,2,3,4
[illegible]       1,[illegible]
ST[illegible]     1[illegible]
[illegible]       [illegible]
[illegible]       2[illegible]
[illegible]       2[illegible]
[illegible]       [illegible]
[illegible]       S[illegible]TS
[illegible]T      [illegible] 1 RUN
[illegible]          1,10,15,20,30,35,40
[illegible]
[illegible]
[illegible] T     (1[illegible],10),(10,10),(196,10),(196,117),(10,117)
[illegible]
[illegible]
[illegible]FILE   [illegible] IT=19,FILE=1
ST[illegible]FILE  [illegible] IT=20,FILE=1
[illegible]FILE   [illegible]IT=20,FILE=2
[illegible]       [illegible]EST
THRESHOLD[illegible] [illegible]
[illegible]IT     [illegible]EDURE 1 RUN
[illegible]
[illegible]
[illegible]TEY
[illegible]       [illegible]UT/U[illegible]IT=2,FILE=1
[illegible]FILE   [illegible] IT=11,FILE=1
STATEFILE  [illegible]IT=[illegible],FILE=2
[illegible] Y     FIL[illegible]
[illegible]T      FI[illegible]
[illegible]T      [illegible] 1 RUN
[illegible]
[illegible]T      (1[illegible],10),(10,10),(196,10),(196,117),(10,117)
[illegible]
[illegible]LAY
[illegible]TAP    [illegible]PUT/UNIT=2,FILE=1
[illegible]IT             UNIT=23,FILE=1
[illegible]IT             UNIT=27,FILE=1
[illegible]IT             UNIT=28,FILE=1
[illegible]                 5
[illegible]               ,5,,5
~~[illegible]    [illegible]IT=1[illegible],FILE=1~~
[illegible]
[illegible]
[illegible]
```


254

FILE: FILE FT23F001 A


```
SIZE =  1851  TYPE = GT
DOT 1 S 12 40 56 68 70 84 86 92 102 114 116 120 122 134 136 142 146
DOT 1 S 156 158 160 162 164 172 184
DOT 1 N 2 4 6 8 10 14 16 18 20 22 24 26 28 30 34 36 42 44 46 48 50
DOT 1 N 52 54 58 60 62 64 66 72 74 76 78 80 82 88 90 94 96 98
DOT 1 N 100 104 106 110 112 118 124 126 130 132 138 140 144 148 150
DOT 1 N 152 154 166 168 170 174 176 178 180 182 186 188 190 192
DOT 1 N 194 196 198 204 206
DOT 2 S 63 69 83 93 101 109 117 121 129 145 161 175 183 189 199
DOT 2 N 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 33 35 37 39 41 43 45
DOT 2 N 47 49 51 53 55 57 61 65 67 73 75 77 79 81 85 87 91 95 97
DOT 2 N 99 103 105 107 111 113 115 119 123 125 127 131 135 137 139
DOT 2 N 141 143 147 149 151 153 155 157 159 163 165 167 169 171
DOT 2 N 173 177 179 181 185 187 191 193 195 197 201 203 205 209
DOT 2 N 31 32 38 71 89 108 128 133 202 207 208
DOT 2 B 200 60
$ESD
$END
```

FILE: FILE FT27F001 A


```
SITE =  1021 TYPE = PPT
DOT 1 N      2   4   6   8  12  14  16  22  24  26  28  30  32  34  36
DOT 1 N     38  40  42  48  52  54  56  58  62  66  68  70  76  78  80
DOT 1 N     82  84  88  90  94 100 106 108 110 118 120 122 124 130 132
DOT 1 N    134 136 138 142 144 146 148 150 154 156 160 162 166 168 174
DOT 1 S     10  18  20  44  46  50  60  64  72  74  86  92  96  98 102
DOT 1 S    104 112 114 126 128 140 152 158 164 170 172 178 188 190 196
DOT 2 N      1   3   5   7  11  13  15  21  23  25  27  29  31  33  37
DOT 2 N     39  43  49  51  53  55  57  61  63  65  67  69  71  75  79
DOT 2 N     87  93  95 103 105 107 115 116 117 119 121 123 125 131 133
DOT 2 N    135 137 141 143 147 149 151 153 157 159 161 163 165 167 173
DOT 2 S      9  17  19  35  41  45  47  59  73  77  81  83  85  89  91
DOT 2 S     93  99 101 109 111 113 127 129 139 145 155 169 171 177 189
$EOD
```

```
SITE = 1021 TYPE = A1
DOT 1 0      6  64 112 124 132 136 200 206 208
DOT 1 E    204
DOT 1 N      2   6  12  14  16  22  24  26  30  32  34  36  38  42  48
DOT 1 N     50  52  54  56  58  62  64  66  68  70  76  78  80  82  88
DOT 1 N     90  94 100 104 106 108 110 118 122 130 134 138 140 142 144
DOT 1 N    146 148 150 152 154 156 160 162 164 166 168 174 176 180 182
DOT 1 N    184 186 190 192 196
DOT 1 R      4  40 202
DOT 1 S     10  18  20  28  44  46  60  72  74  86  92  96  98 102 114
DOT 1 S    120 126 128 158 170 172 178 188 194 198
DOT 2 D     43  67  85  97 133 207
DOT 2 E     11  99 147 116 143 165 197
DOT 2 N      1   3   5   7   9  13  15  21  23  25  29  31  33  35  37
DOT 2 N     49  51  53  55  57  61  63  65  69  71  75  79  83  87  89
DOT 2 N     93 101 103 105 111 113 115 117 119 121 123 125 135 137 139
DOT 2 N    141 147 149 151 153 155 159 163 167 173 175 179 181 183 185
DOT 2 N    187 191 195 201 205 209
DOT 2 R     95 104
DOT 2 S     17  19  27  39  41  45  47  59  73  77  81  91 127 129 131
DOT 2 S    145 157 161 169 171 177 189 193 199 203
$END
```

PLEASE BE ADVISED OF THE CHANGE TO CLASSIFY AND DISPLAY PROCESSOR CARD IMAGES
MAPTAP FILE AND UNIT NUMBER NOW APPEAR AS ORDINARY CONTROL CARDS FIRST ONES AFTER PROCESSOR CARD

1 LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0

FORM

$DDTDATA

INPUT SUMMARY

```
DATA    UNIT=11,FILE=1
DDTF    OUTPUT/UNIT=19,FILE=1
CHAN    DATA=1,2,3,4
OPTI    PRINT
COMM    PROCEDURE 1 RUN
OPTI    LACIE
$END
```

USER HAS REQUESTED THE FOLLOWING OPTIONS :

SELECTED CHANNELS ARE 1 2 3 4
PRINT DATA VECTORS
LACIE FORMATTED DDT CARDS USED AS EOD-LARSYS FIELD CARDS
SUN ANGLES : 34 60 60 60 60 60 60 60

INPUT IMAGE DATA TAPE INFORMATION

FORMAT	UNIVERSAL
NO. OF CHANNELS	4
NO. OF PIXELS/LINE	196
FIRST SCAN LINE NO.	1
FIRST PIXEL REFERENCE PT	1

1 LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0 PROCEDURE 1 RUN

INPUT FIELDS

FIELD	CLASS	SUBCLASS	VERTICES (SAMPLE,LINE)
1 S	1 S		(50, 20)
2 S	1 S		(110, 20)
3 S	1 S		(190, 20)
4 S	1 S		(110, 40)
5 S	1 S		(30, 60)
6 S	1 S		(50, 60)
7 S	1 S		(110, 60)
8 S	1 S		(130, 60)
9 S	1 S		(150, 60)
10 S	1 S		(170, 60)
11 S	1 S		(10, 100)
12 S	1 S		(90, 100)
13 N	2 N		(10, 20)
14 N	2 N		(30, 20)
15 N	2 N		(70, 20)
16 N	2 N		(90, 20)
17 N	2 N		(130, 20)
18 N	2 N		(150, 20)
19 N	2 N		(170, 20)
20 N	2 N		(10, 40)
21 N	2 N		(30, 40)
22 N	2 N		(50, 40)
23 N	2 N		(70, 40)
24 N	2 N		(130, 40)
25 N	2 N		(150, 40)
26 N	2 N		(190, 40)
27 N	2 N		(10, 60)
28 N	2 N		(90, 60)
29 N	2 N		(170, 60)
30 N	2 N		(190, 60)
31 N	2 N		(10, 80)
32 N	2 N		(50, 80)

33	N		2	N			(70,	80)
34	N		2	N			(90,	80)
35	N		2	N			(110,	80)
36	N		2	N			(190,	80)
37	N		2	N			(30,	100)
38	N		2	N			(50,	100)
39	N		2	N			(70,	100)
40	Y		2	N			(150,	100)
41	N		2	N			(170,	100)
42	N		2	N			(190,	100)

NO.	SAMPLE	LINE	TYPE	CATEGORY	DATA			
					CH(1)	CH(2)	CH(3)	CH(4)
1.	50	20	1	1	22	28	30	10
2.	110	20	1	1	21	25	27	11
3.	190	20	1	1	28	35	34	13
4.	110	40	1	1	26	27	30	12
5.	30	60	1	1	20	30	29	12
6.	50	60	1	1	21	27	26	11
7.	110	60	1	1	24	30	29	14
8.	130	60	1	1	27	35	31	13
9.	150	60	1	1	21	27	26	12
10.	170	80	1	1	25	30	30	11
11.	10	100	1	1	26	33	33	13
12.	50	100	1	1	37	55	64	25
13.	10	20	1	2	28	38	40	16
14.	30	20	1	2	25	26	25	11
15.	70	20	1	2	25	35	37	14
16.	90	20	1	2	25	30	30	13
17.	130	20	1	2	22	28	30	12
18.	150	20	1	2	28	32	31	13
19.	170	20	1	2	25	32	31	13
20.	10	40	1	2	28	33	38	14
21.	30	40	1	2	26	36	36	14
22.	50	40	1	2	24	28	30	12
23.	70	40	1	2	21	27	24	11
24.	130	40	1	2	26	27	28	11
25.	150	40	1	2	24	27	28	12
26.	190	40	1	2	24	28	26	12
27.	10	60	1	2	21	27	31	14
28.	90	60	1	2	21	27	29	11
29.	170	60	1	2	29	40	40	17
30.	190	60	1	2	24	32	31	13
31.	10	80	1	2	21	26	23	9
32.	50	80	1	2	28	35	34	14
33.	70	80	1	2	22	28	31	16
34.	90	80	1	2	22	32	31	12
35.	110	80	1	2	21	26	27	12
36.	190	80	1	2	25	30	31	13

37.	30	100	1	2	26	33	28	13
38.	50	100	1	2	24	25	27	12
39.	70	100	1	2	28	36	45	20
40.	150	100	1	2	26	31	33	13
41.	170	100	1	2	26	33	33	14
42.	190	100	1	2	21	25	27	11

SUM ANGLES : 34 60 60 60 60 60 60 60

INPUT IMAGE DATA TAPE INFORMATION

FORMAT UNIVERSAL
NO. OF CHANNELS 4
NO. OF PIXELS/LINE 196
FIRST SCAN LINE NO. 1
FIRST PIXEL REFERENCE PT 1

1

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0 PROCEDURE 1 RUN

INPUT FIELDS

FIELD	CLASS	SUBCLASS	VERTICES (SAMPLE,LINE)
1 S	1 S		(10, 10)
2 S	1 S		(30, 10)
3 S	1 S		(90, 10)
4 S	1 S		(110, 10)
5 S	1 S		(150, 10)
6 S	1 S		(170, 10)
7 S	1 S		(50, 30)
8 S	1 S		(70, 30)
9 S	1 S		(110, 30)
10 S	1 S		(190, 30)
11 S	1 S		(30, 50)
12 S	1 S		(70, 50)
13 S	1 S		(130, 50)
14 S	1 S		(190, 50)
15 S	1 S		(30, 70)
16 S	1 S		(50, 70)
17 S	1 S		(130, 70)
18 S	1 S		(190, 70)
19 S	1 S		(30, 90)
20 S	1 S		(70, 90)
21 S	1 S		(30, 110)
22 S	1 S		(50, 110)
23 S	1 S		(70, 110)
24 S	1 S		(110, 110)
25 N	2 N		(50, 10)
26 N	2 N		(70, 10)
27 N	2 N		(130, 10)
28 N	2 N		(190, 10)
29 N	2 N		(10, 30)
30 N	2 N		(30, 30)
31 N	2 N		(90, 30)
32 N	2 N		(130, 30)
33 N	2 N		(150, 30)
34 N	2 N		(170, 30)
35 N	2 N		(10, 50)
36 N	2 N		(50, 50)
37 N	2 N		(90, 50)
38 N	2 N		(110, 50)
39 N	2 N		(150, 50)
40 N	2 N		(170, 50)
41 N	2 N		(10, 70)
42 N	2 N		(70, 70)
43 N	2 N		(90, 70)
44 N	2 N		(110, 70)
45 N	2 N		(150, 70)
46 N	2 N		(170, 70)
47 N	2 N		(10, 90)
48 N	2 N		(50, 90)
49 N	2 N		(90, 90)
50 N	2 N		(110, 90)
51 N	2 N		(130, 90)
52 N	2 N		(150, 90)

55	N		2	N				(170,	90)
56	N		2	N				(190,	90)
57	N		2	N				(10,	110)
58	N		2	N				(90,	110)
59	N		2	N				(130,	110)
60	N		2	N				(150,	110)
								(170,	110)
								(190,	110)

NO.	SAMPLE	LINE	TYPE	CATEGORY	DATA			
					CH(1)	CH(2)	CH(3)	CH(4)
1.	10	10	2	1	28	36	33	14
2.	30	10	2	1	24	31	28	11
3.	90	10	2	1	26	36	33	14
4.	110	10	2	1	28	38	39	15
5.	150	10	2	1	21	23	24	10
6.	170	10	2	1	31	38	39	15
7.	50	30	2	1	21	26	22	10
8.	70	30	2	1	24	35	37	14
9.	110	30	2	1	24	35	34	14
10.	190	30	2	1	21	27	25	10
11.	30	50	2	1	28	38	34	15
12.	70	50	2	1	28	38	44	16
13.	130	50	2	1	21	26	27	11
14.	190	50	2	1	25	32	34	14
15.	30	70	2	1	26	33	33	14
16.	50	70	2	1	24	27	28	12
17.	130	70	2	1	26	33	38	16
18.	190	70	2	1	24	25	42	20
19.	30	90	2	1	20	27	26	12
20.	70	90	2	1	29	40	48	19
21.	30	110	2	1	28	40	44	19
22.	50	110	2	1	25	35	34	15
23.	70	110	2	1	22	30	31	14
24.	110	110	2	1	28	38	42	18
25.	50	10	2	2	28	33	31	13
26.	70	10	2	2	31	38	36	14
27.	130	10	2	2	21	27	27	11
28.	190	10	2	2	28	33	33	13
29.	10	30	2	2	27	35	31	13
30.	30	30	2	2	24	32	31	12
31.	90	30	2	2	21	27	26	12
32.	130	30	2	2	27	35	31	13
33.	150	30	2	2	27	35	34	13
34.	170	30	2	2	21	27	29	11
35.	10	50	2	2	22	32	34	14
36.	50	50	2	2	22	26	23	10
37.	90	50	2	2	22	26	25	11

```
38.    110     50   2        2        21       26       25       11
39.    150     50   2        2        21       25       27       11
40.    170     50   2        2        25       30       31       13
41.     10     70   2        2        26       31       31       12
42.     70     70   2        2        24       31       33       12
43.     90     70   2        2        28       36       36       13
44.    110     70   2        2        28       33       33       14
45.    150     70   2        2        24       28       27       11
46.    170     70   2        2        24       25       28       11
47.     10     90   2        2        24       27       29       12
48.     50     90   2        2        32       43       43       17
49.     90     90   2        2        27       37       48       21
50.    110     90   2        2        29       40       46       19
51.    130     90   2        2        32       40       40       15
52.    150     90   2        2        24       30       34       14
53.    170     90   2        2        21       27       29       11
54.    190     90   2        2        29       43       40       16
55.     10    110   2        2        33       40       47       19
56.     90    110   2        2        22       30       30       12
57.    130    110   2        2        28       35       34       15
58.    150    110   2        2        18       22       20        9
59.    170    110   2        2        30       35       34       14
60.    190    110   2        2        21       28       25       12

TIME FOR DOTDATA     0.188 MINUTES

    PLEASE BE ADVISED OF THE CHANGE TO CLASSIFY        AND DISPLAY PROCESSOR CARD IMAGES
    MAPTAP FILE AND UNIT NUMBER NOW APPEAR AS           ORDINARY CONTROL CARDS  FIRST ONES AFTER PROCESSOR CARD
1                                                  LYNDON B. JOHNSON SPACE CENTER
                                                        HOUSTON, TEXAS
0       PROCEDURE 1 RUN

$TESTSP


INPUT SUMMARY

      DATA    UNIT=11,FILE=1
      DOTF    INPUT/UNIT=14,FILE=1
      STAT    OUTPUT/UNIT=20,FILE=1
      CHAN    DATA=1,2,3,4
      CHAN    STAT=1,2,3,4
      SEP     1.0
      STOM    15.0
      DLMI    0
      NMIN    20
      PMIN    20
      ISTO    0
      OPTI    STATS
      COMM    PROCEDURE 1 RUN
      DOTS       1,10,15,20,30,35,40
      *END


YOU HAVE SELECTED THE FOLLOWING PARAMETER VALUES AND   OPTIONS

STOP AFTER     0 ITERATION(S)
ALLOW A MINIMUM OF     20 PIXELS PER CLUSTER
```

PRINT A CLUSTER SUMMARY EVERY 20 ITERATION(S)
PRINT A CLUSTER MAP EVERY 20 ITERATION(S)
ALLOW A MAXIMUM OF 60 CLUSTERS PER CLASS
THE STATISTICS FILE WILL BE WRITTEN AFTER 1 CLASS(ES) HAVE BEEN CLUSTERED
CHANNELS ARE--- 1 2 3 4
OLMIN = 0.0
STDMAX= 15.000
PERCENT = 10
PMIN = 20
NSPOTS = 7
NO. SUN ANGLES FROM CARDS = 0
SUN ANGLE TAPE SET = 0

SEP = 1.000

INPUT IMAGE DATA TAPE INFORMATION

FORMAT	UNIVERSAL
NO. OF CHANNELS	4
NO. OF PIXELS/LINE	196
FIRST SCAN LINE NO.	1
FIRST PIXEL REFERENCE PT	1

1 LYNDON B. JOHNSON SPACE CENTER
0 PROCEDURE 1 RUN HOUSTON, TEXAS

FIELDS TO BE CLUSTERED FOR CLASS WHEA

	FIELD NAME	SAMPLE INC.	LINE INC.	VERTICES (SAMPLE,LINE)
1	SEGM	10	10	(10, 10) (196, 10) (196, 117) (10, 117)

DO/DU CLUSTER POP FOR THIS CLASS 0 0

1 LYNDON B. JOHNSON SPACE CENTER
0 PROCEDURE 1 RUN HOUSTON, TEXAS

FINAL CLUSTER SUMMARY FOR CLASS WHEA

TOTAL NUMBER OF CLUSTERS = 7
TOTAL NUMBER OF POINTS = 209

CLUSTER	SYMBOL	POINTS IN CLUSTER
1	1	23
2	2	16
3	3	38
4	4	41
5	5	22
6	6	46
7	7	23

MEANS

CLUSTER	CH(1)	CH(2)	CH(3)	CH(4)
1	22.17	27.78	29.61	12.00
2	25.31	29.44	29.50	12.13
3	26.42	35.03	36.58	14.84
4	30.20	40.10	44.46	18.05
5	24.68	32.45	30.91	12.91
6	21.57	25.98	25.70	11.17
7	26.04	32.00	33.04	13.39

STANDARD DEVIATIONS

CLUSTER	CH(1)	CH(2)	CH(3)	CH(4)
1	1.34	1.02	1.24	1.53

2	1.04	1.32	1.32	0.86
3	2.00	2.59	3.71	1.93
4	3.44	6.26	7.64	3.12
5	2.14	1.67	0.95	0.51
6	1.44	1.41	2.04	0.87
7	1.16	1.22	1.27	0.87

DISTANCES BETWEEN CLUSTERS

CLUSTER	1	2	3	4	5	6	7
1	0.0	3.02	6.32	8.28	4.18	3.00	5.74
2	3.02	0.0	4.94	7.46	2.70	4.73	3.76
3	6.32	4.94	0.0	2.75	3.89	7.37	2.62
4	8.28	7.46	2.75	0.0	7.18	8.82	5.86
5	4.18	2.70	3.89	7.18	0.0	6.46	2.27
6	3.00	4.73	7.37	8.82	6.46	0.0	7.78
7	5.74	3.76	2.62	5.86	2.27	7.78	0.0

1 LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0 PROCEDURE 1 RUN

SEGM

TOTAL NUMBER OF POINTS IN THIS FIELD 209

```
          00000000011111111111
          12345678901234567890
          00000000000000000000

     10   37255134374266624 77
     20   43661735266411565 3
     30   55526733663654311 6
     40   44351666152626612 6
     50   66336344656165632 7
     60   15166366152752634 5
     70   77733377374444126 3
     80   66633217516734442 3
     90   11644444344546711 4
    100   71536446443444737 6
    110   43473454134333663 6
```

POINTS PER CLUSTER IN THIS FIELD

CLUSTER	SYMBOL	POINTS
1	1	23
2	2	16
3	3	38
4	4	41
5	5	22
6	6	46
7	7	23

1

CLASS : WHEA
SUBCLASS: WH01
0 MEAN: 22.17 27.78 29.61 12.00

COVARIANCE MATRIX:

```
0          1.80
0         -0.14     1.04
0         -0.37    -0.30     1.54
0         -0.48     0.0      1.22     2.35
0
```

CLASS : WHEA
SUBCLASS: WH02
0 MEAN: 25.31 29.44 29.50 12.13

COVARIANCE MATRIX:

```
0          1.04
0         -0.34     1.75
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                -0.53      0.03      1.75
                -0.16      0.01      0.13      0.73


  CLASS     :   #MEA
  SUBCLASS  :   #H03
  MEAN      :   26.42     35.03     36.58     14.84

  COVARIANCE MATRIX:
                3.95
                2.54      6.71
               -0.85      0.75     13.77
               -0.20      0.32      6.12      3.71


  CLASS     :   #MEA
  SUBCLASS  :   #H04
  MEAN      :   30.20     40.10     44.46     18.05

  COVARIANCE MATRIX:
               11.42
               19.44     39.21
               23.11     43.93     58.44
                9.24     16.17     22.51      9.71


  CLASS     :   #MEA
  SUBCLASS  :   #H05
  MEAN      :   24.68     32.45     30.91     12.91

  COVARIANCE MATRIX:
                4.56
                2.33      2.79
               -0.67     -0.09      0.98
               -0.12     -0.23      0.13      0.26


  CLASS     :   #MEA
  SUBCLASS  :   #H06
  MEAN      :   21.57     25.98     25.70     11.17

  COVARIANCE MATRIX:
                2.07
                0.82      1.98
                0.72      0.97      4.17
                0.36      0.50      1.31      0.75


  CLASS     :   #MEA
  SUBCLASS  :   #H07
  MEAN      :   26.04     32.00     33.04     13.39

  COVARIANCE MATRIX:
                1.35
                0.70      1.48
               -0.22      0.22      1.61
               -0.23      0.04      0.33      0.76

THE STATISTICS FILE FOR    1 CLASSES AND    7 SUBCLASSES HAS BEEN WRITTEN

THE STATS FOR A PARTICULAR CLASS OR SUBCLASS SHOULD BE REFERRED TO IN LATER RUNS BY
THE FOLLOWING NAMES AND NUMBERS (WHICHEVER APPLICABLE)
```

292.65

```
     CLASS  1     WHEA      SUBCLASSES (TOTAL=  7)

                          1     WH01
                          2     WH02
                          3     WH03
                          4     WH04
                          5     WH05
                          6     WH06
                          7     WH07

 TIME FOR TESTSP      0.084 MINUTES

     PLEASE BE ADVISED OF THE CHANGE TO CLASSIFY        AND DISPLAY PROCESSOR CARD IMAGES
     MAPTAP FILE AND UNIT NUMBER NOW APPEAR AS            ORDINARY CONTROL CARDS  FIRST ONES AFTER PROCESSOR CARD
1                                                    LYNDON B. JOHNSON SPACE CENTER
                                                          HOUSTON, TEXAS
0         PROCEDURE 1 RUN

 $LABEL

            INPUT SUMMARY

 NOTE       INPUT/UNIT=19,FILE=1
 STAT       INPUT/UNIT=20,FILE=1
 STAT       OUTPUT/UNIT=20,FILE=2
 PROC       K-NEAREST
 THRE       25.0
 COMM       PROCEDURE 1 RUN
 *END

 USER HAS REQUESTED THE FOLLOWING OPTIONS :

 K-NEAREST PROCEDURE WILL BE USED
 L1 DISTANCE WILL BE USED
 THRESHOLD DISTANCE =      25.000
   1-NEAREST DOIS WILL BE USED
 NO SUN ANGLE CORRECTION WILL BE APPLIED
 NOTFIL FILE IS BEING INPUT
 SAVTAP FILE IS BEING INPUT
1                                                    LYNDON B. JOHNSON SPACE CENTER
                                                          HOUSTON, TEXAS
0         PROCEDURE 1 RUN


                                  INPUT FIELDS

      FIELD          CLASS        SUBCLASS                    VERTICES  (SAMPLE,LINE)

     1  S          1  S                          (  50,  20)
     2  S          1  S                          ( 110,  20)
     3  S          1  S                          ( 190,  20)
     4  S          1  S                          ( 110,  40)
     5  S          1  S                          (  30,  60)
     6  S          1  S                          (  50,  60)
     7  S          1  S                          ( 110,  60)
     8  S          1  S                          ( 130,  60)
     9  S          1  S                          ( 150,  60)
    10  S          1  S                          ( 170,  80)
    11  S          1  S                          (  10, 100)
    12  S          1  S                          (  90, 100)
    13  N          2  N                          (  10,  20)
    14  N          2  N                          (  30,  20)
    15  N          2  N                          (  70,  20)
    16  N          2  N                          (  90,  20)
    17  N          2  N                          ( 130,  20)
    18  N          2  N                          ( 150,  20)
    19  N          2  N                          ( 170,  20)
    20  N          2  N                          (  10,  40)
    21  N          2  N                          (  30,  40)
    22  N          2  N                          (  50,  40)
    23  N          2  N                          (  70,  40)
    24  N          2  N                          ( 130,  40)
    25  N          2  N                          ( 150,  40)
    26  N          2  N                          ( 190,  40)
    27  N          2  N                          (  10,  60)
    28  N          2  N                          (  90,  60)
    29  N          2  N                          ( 170,  60)
    30  N          2  N                          ( 190,  60)
```

31	N	2	N	(10, 80)
32	N	2	N	(50, 80)
33	N	2	N	(70, 80)
34	N	2	N	(90, 80)
35	N	2	N	(110, 80)
36	N	2	N	(190, 80)
37	N	2	N	(30, 100)
38	N	2	N	(50, 100)
39	N	2	N	(70, 100)
40	N	2	N	(150, 100)
41	N	2	N	(170, 100)
42	N	2	N	(190, 100)

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

PROCEDURE 1 RUN

CLUSTER-DOT INTER-DISTANCE TABLE

DOTS	CLST (1)	CLST (2)	CLST (3)	CLST (4)	CLST (5)	CLST (6)	CLST (7)
1.	2.78	7.38	22.87	42.80	10.95	7.93	14.48
2.	7.57	12.38	28.87	48.80	16.95	3.02	20.48
3.	18.43	13.63	6.03	22.80	9.05	25.59	6.30
4.	5.00	3.75	17.87	37.80	8.59	10.59	9.48
5.	5.00	6.50	21.87	41.80	9.95	9.72	13.48
6.	6.57	11.38	27.87	47.80	15.95	2.07	19.48
7.	6.65	4.25	15.87	35.80	6.14	12.59	8.70
8.	14.43	9.63	8.03	26.80	5.05	21.59	6.39
9.	5.57	10.38	26.87	46.80	14.95	2.72	18.48
10.	6.43	2.50	16.87	36.80	5.59	11.93	8.48
11.	14.43	8.63	7.87	27.80	4.05	20.59	1.48
12.	89.43	84.63	68.13	48.20	80.05	96.59	76.52
13.	30.43	25.63	9.13	10.80	21.05	37.59	17.52
14.	16.22	9.38	25.87	45.80	14.59	4.33	17.48
15.	18.43	15.25	2.71	21.80	10.05	26.59	8.51
16.	6.43	2.25	14.87	34.80	3.77	13.59	6.48
17.	0.78	5.38	20.87	40.80	8.95	7.59	12.48
18.	12.43	7.63	12.03	28.80	3.95	18.59	4.39
19.	9.43	5.25	11.87	31.80	0.95	16.59	3.48
20.	21.43	16.63	5.87	19.80	12.05	28.59	8.52
21.	20.43	15.63	2.82	20.80	11.05	27.59	7.61
22.	2.43	3.38	18.87	38.80	6.95	9.59	10.48
23.	8.57	13.38	29.87	49.80	17.95	3.46	21.48
24.	7.22	5.75	20.87	40.80	11.59	7.93	12.48
25.	4.22	5.38	21.87	41.80	9.45	6.59	13.48
26.	5.65	6.38	22.87	42.80	10.95	5.59	14.48
27.	5.35	10.13	19.87	39.80	10.32	9.72	12.70
28.	3.57	8.38	24.87	44.80	12.95	5.07	16.48
29.	34.43	29.63	13.13	6.80	25.05	41.59	21.52
30.	8.43	6.25	12.87	32.80	1.32	15.59	4.48
31.	12.57	17.38	33.87	53.80	21.95	5.46	25.48
32.	15.43	14.63	5.03	21.80	10.05	26.59	6.52
33.	5.78	10.13	18.18	35.80	10.32	12.59	12.70
34.	5.78	7.50	15.87	35.80	4.14	12.59	7.48
35.	5.57	10.38	26.87	46.80	14.95	2.72	18.48
36.	7.43	3.25	13.87	33.80	2.95	14.59	5.48
37.	11.65	6.63	12.87	32.80	4.86	15.59	6.48
38.	7.22	8.38	24.87	44.80	12.95	5.54	16.48
39.	37.43	32.63	16.13	8.78	28.05	44.59	24.52
40.	11.43	6.63	9.87	29.80	4.95	18.59	1.48
41.	14.43	9.63	6.87	26.80	5.05	21.59	1.70
42.	7.57	12.38	23.87	48.80	16.95	3.02	20.48

1
1 0 PROCEDURE 1 RUN

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

LABELING BY 1-NEAREST NEIGHBOR PROCEDURE

LABEL : N

MEANS : 22.17 27.78 29.61 12.00

DOTS	ANALYST LABEL	DISTANCE
17.	N	0.78

TOTALS

LABEL NO.

S 0
N 1

1 LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0 PROCEDURE 1 RUN

LABELING BY 1-NEAREST NEIGHBOR PROCEDURE

LABEL : N

MEANS : 25.31 29.44 29.50 12.13

DOTS	ANALYST LABEL	DISTANCE
16.	N	2.25

TOTALS

LABEL NO.

S 0
N 1

1 LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0 PROCEDURE 1 RUN

LABELING BY 1-NEAREST NEIGHBOR PROCEDURE

LABEL : N

MEANS : 26.42 35.03 36.58 14.84

DOTS	ANALYST LABEL	DISTANCE
15.	N	2.71

TOTALS

LABEL NO.

S 0
N 1

1 LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0 PROCEDURE 1 RUN

LABELING BY 1-NEAREST NEIGHBOR PROCEDURE

LABEL : N

MEANS : 30.20 40.10 44.46 18.05

DOTS	ANALYST LABEL	DISTANCE
29.	N	6.80

TOTALS

LABEL NO.

S 0
N 1

1 LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

0 PROCEDURE 1 RUN

LABELING BY 1-NEAREST NEIGHBOR PROCEDURE

LABEL : N
MEANS : 24.08 32.45 30.91 12.91

DOTS	ANALYST LABEL	DISTANCE
19.	N	0.95

TOTALS
LABEL NO.
S 0
N 1

1 -
0 - PROCEDURE 1 RUN

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

LABELING BY 1-NEAREST NEIGHBOR PROCEDURE

LABEL : S
MEANS : 21.57 25.98 25.70 11.17

DOTS	ANALYST LABEL	DISTANCE
6.	S	2.07

TOTALS
LABEL NO.
S 1
N 0

1
0 PROCEDURE 1 RUN

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

LABELING BY 1-NEAREST NEIGHBOR PROCEDURE

LABEL : N
MEANS : 26.04 32.00 33.04 13.39

DOTS	ANALYST LABEL	DISTANCE
40.	N	1.48

TOTALS
LABEL NO.
S 0
N 1

1

THE STATISTICS FILE FOR 2 CLASSES AND 7 SUBCLASSES HAS BEEN WRITTEN
THE STATS FOR A PARTICULAR CLASS OR SUBCLASS SHOULD BE REFERRED TO IN LATER RUNS BY THE FOLLOWING NAMES AND NUMBERS (WHICHEVER APPLICABLE)

CLASS 1 S SUBCLASSES (TOTAL= 1)

1 S 01

CLASS 2 N SUBCLASSES (TOTAL= 6)

2 N 01
3 N 02

92

```
                    4    N 03
                    5    N 04
                    6    N 05
                    7    N 06

 TIME FOR LABEL     0.053 MINUTES

     PLEASE BE ADVISED OF THE CHANGE TO CLASSIFY        AND DISPLAY PROCESSOR CARD IMAGES
     MAPTAP FILE AND UNIT NUMBER NOW APPEAR AS            ORDINARY CONTROL CARDS  FIRST ONES AFTER PROCESSOR CARD
1                                                    LYNDON B. JOHNSON SPACE CENTER
                                                           HOUSTON, TEXAS
0         PROCEDURE 1 RUN

 $CLASSIF

     MAPT       OUTPUT/UNIT=2,FILE=1
     DATA       UNIT=11,FILE=1
     STAT       UNIT=20,FILE=2
     CATE       FILE
     APRI       FILE
     COMM       PROCEDURE 1 RUN
     *END
1                                                    LYNDON B. JOHNSON SPACE CENTER
                                                           HOUSTON, TEXAS
0         PROCEDURE 1 RUN

 THE FOLLOWING OPTIONS HAVE BEEN SELECTED

    CATEGORY CLASSIFIER OPTION HAS BEEN SELECTED.
      ALSO CLASSES FROM STATFILE WILL BE CONSIDERED THE CATEGORIES FOR CLASSIFICATION
    APRIORI VALUES FROM STATFILE  2   APRIORI= NO. PI   XELS IN SUBCLASS/TOTAL NO. PIXELS IN ALL SUBCLASSES ***

 SUPERVISOR INFORMATION :

    FILE NUMBER ....          1
    NO. OF FIELDS.....        1
    NO. OF CLASSES.....       2
    NO. OF SUBCLASSES.....    7
    NO. OF CHANNELS.....      4


1                                                    LYNDON B. JOHNSON SPACE CENTER
                                                           HOUSTON, TEXAS
0         PROCEDURE 1 RUN


                AREA USED TO COMPUTE TRAINING STATISTICS
              FIELD          CLASS        SUBCLASS                    VERTICES  (SAMPLE,LINE)
1             1  SEGM        1  S                    (  10,  10) ( 196,  10) ( 196, 117) (  10, 117)
                                                   LYNDON B. JOHNSON SPACE CENTER
                                                          HOUSTON, TEXAS
0         PROCEDURE 1 RUN

   ***  CLASSIFICATION STUDY  ***   MAPTAP FILE       1


                    SUBCLASSES CONSIDERED
                 SYMBOL      SUBCLASS      A PRIOR
*                  1          S 01          0.2201
*
*                  2          N 01          0.1100
*
*                  3          N 02          0.0766
*
*                  4          N 03          0.1818

                                          CHANNELS CONSIDERED
                                          TRAINING   RECOGNITION
                                             1           1
                                             2           2
                                             3           3
```

```
+
+                    5          N 04          0.1962
+                    6          N 05          0.1053
+                    7          N 06          0.1100

 INPUT IMAGE DATA TAPE INFORMATION

      FORMAT                    UNIVERSAL
      NO. OF CHANNELS                4
      NO. OF PIXELS/LINE           196
      FIRST SCAN LINE NO.            1
      FIRST PIXEL REFERENCE PT       1
1                                          LYNDON B. JOHNSON SPACE CENTER
                                                 HOUSTON, TEXAS
0         PROCEDURE 1 RUN


                                  AREA OF CLASSIFICATION

      FIELD NAME    NO. OF VERTICES   SAMPLE INC.   LINE INC.   VERTICES
        SEGM                                                    ( 10, 10) ( 196, 10) ( 196, 117) ( 10, 117) ( 10, 10)
1                                          LYNDON B. JOHNSON SPACE CENTER
                                                 HOUSTON, TEXAS
0         PROCEDURE 1 RUN


                          MAP OF CATEGORY CLASSIFIER CLASSIFICATION RESULTS

                          CATEGORY          CLASS                SUBCLASS
                          NO.  NAME         NO.  NAME       NO.   NAME    SYMBOL
+                          1   S            1   S
+                                                           1    S 01      1
+                          2   N            2   N
+                                                           2    N 01      2
                                                            3    N 02      3
                                                            4    N 03      4
                                                            5    N 04      5
                                                            6    N 05      6
                                                            7    N 06      7


                 00000000001111111111
                 12345678901234567890
                 00000000000000000000

      10    433662444743111 3577
      20    541127463115227 1634
      30    666317441141654 2251
      40    444621112631311 2371
      50    414414561612161 4327
      60    262114112737631 4546
      70    777414774474451 3174
      80    111443276217455 5343
      90    221455555555617 2215
     100    725414515544547 4741
     110    545745652454441 2441


 *** %CLASSIFY - COMPLETED ***
 !

 TIME FOR CLASSIFY      0.074 ,
```

```
PLEASE BE ADVISED OF THE CHANGE TO CLASSIFY          AND DISPLAY PROCESSOR CARD IMAGES
MAPTAP FILE AND UNIT NUMBER NOW APPEAR AS                ORDINARY CONTROL CARDS  FIRST ONES AFTER PROCESSOR CARD
1                                                LYNDON B. JOHNSON SPACE CENTER
                                                        HOUSTON, TEXAS
0        PROCEDURE 1 RUN

 $DISPLAY

      GTUN             UNIT=23,FILE=1
      PPUN             UNIT=27,FILE=1
      AIUN             UNIT=28,FILE=1
      NAME              S
      ALPH              .5,.5
      DOTF        INPUT/UNIT=19,FILE=1
      *END



    YOU HAVE SELECTED THE FOLLOWING OPTIONS:

        PROCESS THE CLASSIFICATION RESULTS FROM MAPTAP (      B ), UNIT    2  ,  FILE    1
        APPLY NO THRESHOLDING
0    LIST PROCESSING OPTION SELECTED


     NUMBERS OF GROUND TRUTH ,  AI, AND            DISCRIMINATOR UNITS AND FILES ARE AS FOLLOWS


         23       1      28       1      27       1


      SELECTED CATEGORY NAME FOR LIST IS  S


1     BIAS CORRECTION  ALPHAS ARE    0.500000  0.500000
                                                LYNDON B. JOHNSON SPACE CENTER
0        PROCEDURE 1 RUN                                HOUSTON, TEXAS



                  AREA USED TO COMPUTE TRAINING STATISTICS
               FIELD           CLASS        SUBCLASS                          VERTICES  (SAMPLE,LINE)
1               1  SEGM         1  S                            (  10,  10) ( 196,  10) ( 196, 117) (  10, 117)
                                                LYNDON B. JOHNSON SPACE CENTER
0        PROCEDURE 1 RUN                                HOUSTON, TEXAS

DISPLAY OF CLASSIFIED FIELD............SEGM
        CLASSIFICATION  DATE........
        CLASSIFICATION CHANNELS...  1  2  3  4

                         MAP OF CATEGORY CLASSIFIER CLASSIFICATION RESULTS

                       CATEGORY               CLASS                     SUBCLASS
                      NO.  NAME             NO.  NAME              NO.    NAME    SYMBOL
+                      1    S
+                                            1    S
                                                                    1    S 01      1
+                      2    N
+                                            2    N

                                                                    2    N 01      2
                                                                    3    N 02      3
                                                                    4    N 03      4
                                                                    5    N 04      5
                                                                    6    N 05      6
                                                                    7    N 06      7


       000000000111111111
```

272

```
          1234567890123456789
          000000000000000000

     10   43366244474311113577
     20   54112746311522716J4
     30   66631744114165422251
     40   44442111263131123?1
     50   41441455161215143?7
     60   2621141127376314546
     70   777414774474451?174
     80   11144327421744553?3
     90   2214555555565172215
    100   72?.1451554454747?1
    110   5457456524544412441

                                        LYNDON B. JOHNSON SPACE CENTER
                                              HOUSTON, TEXAS
          PROCEDURE 1 RUN

CLASSIFICATION SUMMARY FOR FIELD   SEGM
TOTAL NUMBER OF SAMPLED POINTS        209

                PTS. BEFORE   PCT. OF          PTS. AFTER   PCT.OF       PCT.OF          PTS.      PCT.OF       PCT. OF
+   SUBCLASS      THRES.      TOTAL            THRES.       TOTAL        SUBCLASS        THRES.    TOTAL        SUBCLASS
                              CLSF.FLD.                     CLSF.FLD.                              CLSF.FLD.    THRES.

    S 01            46        22.01              46        22.01        100.00           0         0.0          0.0
    N 01            23        11.00              23        11.00        100.00           0         0.0          0.0
    N 02            17         8.13              17         8.13        100.00           0         0.0          0.0
    N 03            50        23.92              50        23.92        100.00           0         0.0          0.0
    N 04            30        14.35              30        14.35        100.00           0         0.0          0.0
    N 05            19         9.09              19         9.09        100.00           0         0.0          0.0
    N 06            24        11.48              24        11.48        100.00           0         0.0          0.0

          PTS. THRESHOLDED IN DISPLAY          0
          PTS. THRESHOLDED IN CLASSIFY         0
                          TOTAL                0        PCT. =    0.0
1                                                 LYNDON B. JOHNSON SPACE CENTER
                                                        HOUSTON, TEXAS
0         PROCEDURE 1 RUN

CLASSIFICATION SUMMARY FOR FIELD   SEGM
TOTAL NUMBER OF SAMPLED POINTS        209

                PTS. BEFORE   PCT. OF          PTS. AFTER   PCT.OF       PCT.OF          PTS.      PCT.OF       PCT. OF
+   CLASS         THRES.      TOTAL            THRES.       TOTAL        CLASS           THRES.    TOTAL        CLASS
                              CLSF.FLD.                     CLSF.FLD.                              CLSF.FLD.    THRES.

    S               46        22.01              46        22.01        100.00           0         0.0          0.0
    N              163        77.99             163        77.99        100.00           0         0.0          0.0

          PTS. THRESHOLDED IN DISPLAY          0
          PTS. THRESHOLDED IN CLASSIFY         0
                          TOTAL                0        PCT. =    0.0
1                                                 LYNDON B. JOHNSON SPACE CENTER
                                                        HOUSTON, TEXAS
0         PROCEDURE 1 RUN

CLASSIFICATION SUMMARY FOR FIELD   SEGM
TOTAL NUMBER OF SAMPLED POINTS        209

                PTS. BEFORE   PCT. OF          PTS. AFTER   PCT.OF       PCT.OF          PTS.      PCT.OF       PCT. OF
+   CATEGORY      THRES.      TOTAL            THRES.       TOTAL        CATEGORY        THRES.    TOTAL        CATEGORY
                              CLSF.FLD.                     CLSF.FLD.                              CLSF.FLD.    THRES.

    S               46        22.01              46        22.01        100.00           0         0.0          0.0
    N              163        77.99             163        77.99        100.00           0         0.0          0.0

          PTS. THRESHOLDED IN DISPLAY          0
          PTS. THRESHOLDED IN CLASSIFY         0
```

TOTAL PCT. = 0.0

NSITE = 1021 TYPE = PPT
NSITE = 1361 TYPE = GT
NSITE = 1021 TYPE = AI

PPTC VS CLASSIFIED LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		N/N		N/N		N/N		N/N		S/N		N/N		N/S		N/N		S/N	
20	S/N		N/S		N/N		N/N		N/N		N/S		N/N		N/N		N/N		N/N
30		N/N		N/N		S/N		S/N		N/S		S/S		N/N		N/N		N/N	
40	N/N		S/N		N/N		S/S		N/N		N/N		N/N		S/S		S/N		N/S
50		N/S		N/N		N/N		N/N		S/N		N/N		N/N		S/N		N/N	
60	S/N		S/N		N/S		S/S		S/N		N/N		N/N		N/S		S/N		S/N
70				N/N		N/N		N/N		N/N		S/N		S/N		N/N		N/N	
80	N/S		N/S		N/N		S/N		N/N		N/S		N/N		N/N		N/N		S/N
90		N/N		N/N		S/N		N/N		N/N		S/N		N/S		N/N		S/S	
100	S/N		N/N		N/S		S/N		N/N		N/N		N/N		N/N		S/N		S/S
110		N/N		N/N		S/N		S/N		N/N		S/N		N/N		N/N		N/N	

PPTC VS CLASSIFIED LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/N		N/N		N/N		N/N		S/N		N/N		N/S		N/S		S/N		S/N
20		N/N		N/S		N/N		N/N		N/S		N/N		N/N		S/S		N/N	
30	N/N		S/N		N/S		S/N		S/S		N/N		N/N		N/N		N/N		N/S
40		S/N		N/N		N/S		N/S		N/N		N/S		N/S		S/N		N/N	
50	S/N		N/N		S/S		S/N		S/S		N/S		S/S		S/S		N/N		N/N
60		S/N		S/S		S/N		N/S		N/N		N/N		S/N		S/N		S/N	
70	N/N	N/N	N/N		N/S		N/N		N/N		N/N		S/N		S/S		N/S		N/N
80		N/S		N/N		S/N		N/N		N/N		S/N		N/N		N/N		N/N	
90	N/N		S/S		N/N		N/N		N/N		N/N		N/N		N/N		S/N		S/N

100		N/N		N/N		S/N		N/S		N/N		N/N		N/N		N/N		S/N	
110	N/N		S/N		N/N		S/N		S/N		N/N		N/N		N/S		N/N		N/S

FOR LIST PROCESSING PPC VS MACHINE CLASS
PROPORTION SUMMARY
P(1) = 0.2201 P(2) = 0.7799

N11	N12	N21	N22	N81	N82
9	24	18	55	0	0

P11	P12	PH1	P32	P	ALP		
0.3333	0.3038	0.0	0.0	0.3103	0.5000	0.5000	

GROUND TRUTH VS CLASSIFIED LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		N/N		N/N		N/N		N/N		N/N		S/N		N/S		N/N		N/N	
20	N/N		N/S		N/N		N/N		N/N		N/S				N/N		N/N		
30		S/N		N/N		N/N		N/N		N/S		N/S		N/N		N/N		S/N	
40	N/N		N/N		N/N		N/S		N/N		S/N		S/N		N/S		N/N		N/S
50		N/S		N/N		N/N		S/N		S/N		N/N		N/N		S/N		N/N	
60	N/N		N/N		N/S		S/S		N/N		N/N				N/S		N/N		S/N
70		S/N		N/N		S/N		S/N		N/N		N/N				N/N		N/N	
80	S/S		S/S		N/N		N/N		S/N		N/S		S/N		N/N		N/N		N/N
90		N/N		S/N		S/N		S/N		S/N		S/N		N/S		N/N		N/S	
100	S/N		N/N		N/S		N/N		N/N		N/N		S/N		N/N		N/N		N/S
110		N/N		N/N		N/N		N/N						N/N		N/N			

GROUND TRUTH VS CLASSIFIED LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/N		N/N		N/N		N/N		N/N		N/N		N/S		N/S		N/N		N/N
20		N/N		N/S		N/N		N/N		N/S		M/N	M/N	N/N		N/S		N/N	M/N
30	N/N		N/N		N/S		N/N		N/S		N/N		N/N		N/N		N/N		N/S
40		R/N		N/N		S/S		N/S		N/N		S/S		M/S		N/N		N/N	
50	N/N		N/N		N/S		S/N		N/S		N/S		M/S		N/S		S/N		N/N
60		N/N		N/S		S/N		N/S		N/N		N/N	M/N	S/N		N/N		N/N	
70	N/N		S/N		N/S		S/N		N/N		N/N		N/N	M/N	S/S		N/S		M/N
80		N/S		N/N		N/N		N/N		N/N		S/N		N/N		N/N		N/N	

90	N/N		N/S		N/N		N/N		S/N		N/N		N/N		N/N		N/N		N/N
100		N/N		S/N		N/N		N/S		N/N		S/N		N/N		N/N		S/N	
110	N/N		N/N		N/N		N/N		S/N	R/N	N/N	M/N	N/N		N/S		M/N	M/N	N/S

FOR LIST PROCESSING GT VS MACHINE CLASS
PROPORTION SUMMARY
P(1) = 0.2201 P(2) = 0.7799

N11	N12	N21	N22	NB1	N82				
3	12	22	62	P2	11				
P11	P12	PR1	PR2	P	ALP				
0.1111	0.1412	0.0741	0.1294	0.1932	0.5000	0.5000			

A. I. LABELS VS CLASSIFIED LABELS

TYPE 1 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10		N/N		R/N		D/N		N/N		S/N		N/N		N/S		N/N		S/N	
20	S/N		N/S		N/N		N/N		S/N		N/S		N/N		N/N		N/N		N/N
30		R/N		N/N		S/N		S/N		N/S		N/S		N/N		N/N		N/N	
40	N/N		S/N		N/N		N/S		N/N		N/N		N/N		S/S		S/N		N/S
50		N/S		N/N		N/N		D/N		S/N		N/N		N/N		S/N		N/N	
60	S/N		S/N		N/S		S/S		N/N		N/N		N/N		N/S		D/N		S/N
70				N/N		S/N		N/N		D/N		S/N		S/N		N/N		D/N	
80	N/S		D/S		N/N		N/N		N/N		N/S		N/N		N/N		N/N		N/N
90		N/N		N/N		S/N		N/N		N/N		N/N		N/S		N/N		S/S	
100	S/N		N/N		N/S		S/N		N/N		N/N		N/N		N/N		S/N		N/S
110		N/N		S/N		N/N		S/N		D/N		R/N		M/N		D/N		D/N	

A. I. LABELS VS CLASSIFIED LABELS

TYPE 2 DOT CLASSIFICATION

	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190
10	N/N		N/N		N/N		N/N		N/N		M/N		N/S		N/S		S/N		S/N
20		N/N		N/S		N/N		S/N		N/S		N/N		N/N		N/S		N/N	
30	S/N		S/N		D/S		S/N		S/S		N/N		N/N		N/N		N/N		N/S
40		S/N		N/N		N/S		N/S		D/N		N/S		N/S		S/N		N/N	
50	S/N		N/N		S/S		N/N		D/S		N/S		N/S		S/S		N/N		R/N
60		D/N		M/S		N/N		N/S		N/N		M/N		R/N		N/N		N/N	

```
          70  N/N  N/N  N/N       N/S       N/N       N/N       N/N       S/N       S/S       S/S       D/N

          80       N/S       N/N       N/N       N/N       M/N       S/N       N/N       N/N       N/N

          90  N/N       N/S       S/N       N/N       S/N       N/N       N/N       N/N       S/N       S/N

         100       N/N       N/N       S/N       N/S       N/N       N/N       N/N       N/N       S/N

         110  N/N       S/N       N/N       M/N       S/N       N/N       S/N       N/S       D/N       N/S
1 FOR LIST PROCESSING AI    V5 MACHINE CLASS
0 PROPORTION SUMMARY
0  P(1) =  0.2201     P(2) =  0.7799
0  N11      N12       N21       N22       NR1       NR2
0        5        20        19        47         1         8
0  P11       P12       PR1       PR2       P    ALP
0  0.2000    0.2667    0.0400    0.1067    0.2980    0.5000    0.5000
1       ***** DISPLAY COMPLETED *****


 TIME FOR DISPLAY     0.149

     PLEASE BE ADVISED OF THE CHANGE TO CLASSIFY -        AND DISPLAY PROCESSOR CARD IMAGES
     MAPTAP FILE AND UNIT NUMBER NOW APPEAR AS              ORDINARY CONTROL CARDS  FIRST ONES AFTER PROCESSOR CARD
1                                                    LYNDON B. JOHNSON SPACE CENTER
                                                           HOUSTON, TEXAS
0        PROCEDURE 1 RUN

 $EXIT
```


SEP 1980
RECEIVED
NASA STI FACILITY
ACCESS DEPT.