

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

NASA CR.

147803

Technical Memorandum

DIGITIZING ZONE MAPS, USING MODIFIED LARSYS PROGRAM

(NASA-CR-147803) DIGITIZING ZONE MAPS,
USING MODIFIED LARSYS PROGRAM (Lockheed
Electronics Co.) 175 p HC \$6.75 CSCL 08B

N76-27649

Unclas

G3/43 45744

Prepared By

Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Contract NAS 9-12200

For

EARTH OBSERVATIONS DIVISION



National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

May 1976

LEC-7498

TECHNICAL REPORT INDEX/ABSTRACT
(See instructions on reverse side.)

1. TITLE AND SUBTITLE OF DOCUMENT TECHNICAL MEMORANDUM DIGITIZING ZONE MAPS, USING MODIFIED LARSYS PROGRAM	2. JSC NO. JSC- 10757
--	---------------------------------

3. CONTRACTOR/ORGANIZATION NAME Lockheed Electronics Company, Inc.	4. CONTRACT OR GRANT NO. NAS 9-12200
--	--

5. CONTRACTOR/ORIGINATOR DOCUMENT NO. LEC-7498	6. PUBLICATION DATE (THIS ISSUE) April 1976
--	---

7. SECURITY CLASSIFICATION Unclassified	8. OPR (OFFICE OF PRIMARY RESPONSIBILITY)
---	---

9. LIMITATIONS GOVERNMENT HAS UNLIMITED RIGHTS <input checked="" type="checkbox"/> YES <input type="checkbox"/> NO IF NO, STATE LIMITATIONS AND AUTHORITY <p align="center">N/A</p>	10. AUTHOR(S) L. Giddings S. Boston
--	---

11. DOCUMENT CONTRACT REFERENCES WORK BREAKDOWN STRUCTURE NO. <p align="center">N/A</p>	12. HARDWARE CONFIGURATION SYSTEM <p align="center">N/A</p>
CONTRACT EXHIBIT NO. <p align="center">N/A</p>	SUBSYSTEM <p align="center">N/A</p>
DRL NO. AND REVISION <p align="center">N/A</p>	MAJOR EQUIPMENT GROUP <p align="center">N/A</p>
DRL LINE ITEM NO. <p align="center">N/A</p>	

13. ABSTRACT

A method for digitizing zone maps is presented, starting with colored images and producing a final one-channel digitized tape. This method automates the work previously done interactively on the Image-100 and Data Analysis System (DAS) computers of the Johnson Space Center (JSC) Earth Observations Division (EOD). A color-coded map is digitized through color filters on a scanner to form a digital tape in LARSYS-2 or JSC Universal format. The taped image is classified by the EOD LARSYS program on the basis of training fields included in the image. Numerical values are assigned to all pixels in a given class, and the resulting coded zone map is written on a LARSYS or Universal tape. A unique spatial filter option permits zones to be made homogeneous and edges of zones to be abrupt transitions from one zone to the next. A zoom option allows the output image to have arbitrary dimensions in terms of number of lines and number of samples on a line.

14. SUBJECT TERMS

_____	_____	_____
_____	_____	_____
_____	_____	_____

JSC-10757

TECHNICAL MEMORANDUM

DIGITIZING ZONE MAPS, USING MODIFIED LARSYS PROGRAM

PREPARED BY

Spencer Boston
S. Boston

Loran Gidding
L. Gidding

APPROVED BY

Milton L. Bertrand Jr.
M. L. Bertrand, Jr., Manager
Earth Observations Data Products Department

Prepared By

Lockheed Electronics Company, Inc.

For

EARTH OBSERVATIONS DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

May 1976

LEC-7498

SUMMARY

A method for digitizing zone maps is presented, starting with colored images and producing a final one-channel digitized tape. This method automates the work previously done interactively on the Image-100 and Data Analysis System (DAS) computers of the Johnson Space Center (JSC) Earth Observations Division (EOD). A color-coded map is digitized through color filters on a scanner to form a digital tape, which is then converted to a multichannel LARSYS-2 or JSC Universal formatted tape for input to the EOD-LARSYS system.

The processors STAT, CLASSIFY and a modified DISPLAY processor classify each of the input multichannel pixel values into one of the training zone values or a threshold value. An output tape is then created by the modified DISPLAY processor which is input to the Spatial Filter and Zoom Scanner Data Program (FILZOM).

FILZOM has several options including: (1) to process a specified area of the input image; (2) to filter the specified area up to eight times using an eight-point spatial filtering scheme; (3) to zoom the specified area; (4) to output a multifile LARSYS-2 or Universal formatted tape; and (5) to output a line printer map with overprint capabilities.

Detailed examples are presented for digitizing Crop Moisture Index (CMI) maps and other zonal maps. A list of suitable tempera paint colors used in the preparation of images for a multichannel tape is also included.

CONTENTS

Section	Page
SUMMARY	iv
1. INTRODUCTION.	1-1
2. THE EOD-LARSYS SYSTEM AND FILZOM.	2-1
2.1 <u>INTRODUCTION</u>	2-1
2.2 <u>THE EOD-LARSYS SYSTEM.</u>	2-2
2.3 <u>SPATIAL FILTER AND ZOOM SCANNER DATA (FILZOM).</u>	2-2
2.3.1 THE NEED FOR THE FILTER.	2-2
2.3.2 THE FILZOM FILTER.	2-3
2.3.2.1 <u>Filter</u>	2-4
2.3.2.2 <u>Island and Hole Lead Cards</u>	2-5
2.3.3 THE OUTPUT IMAGE	2-7
2.3.3.1 <u>SKIP Lead Card</u>	2-7
2.3.3.2 <u>ZOOM Lead Card</u>	2-7
2.3.3.3 <u>LIMIT Lead Card.</u>	2-8
2.3.4 OTHER FEATURES	2-9
3. PREPARING DIGITIZED ZONE MAPS	3-1
3.1 <u>INTRODUCTION</u>	3-1
3.2 <u>PREPARATION OF INPUT TAPE TO EOD-LARSYS.</u>	3-1
3.3 <u>TAPE ASSIGNMENTS AND DECK SETUPS FOR UNIVAC 1108, EXEC II.</u>	3-1
4. SELECTION OF COLORS	4-1
4.1 <u>INTRODUCTION</u>	4-1
4.2 <u>AVAILABLE SYSTEMS FOR PREPARING COLORED IMAGES</u>	4-1
4.3 <u>AVAILABLE TEMPERA PAINTS</u>	4-2

Section	Page
4.4 <u>SELECTION OF COLORS</u>	4-2
4.5 <u>USE OF THE COLOR TABLES</u>	4-4
5. REFERENCES.	5-1
 APPENDIX	
A. COMPLETE EXAMPLE: PREPARATION OF A SIMPLE CROP MOISTURE INDEX TAPE	A-1
B. COMPLETE EXAMPLE: PREPARATION OF A COMPLEX CROP MOISTURE INDEX TAPE	B-1
C. ALPHABETICAL LISTING OF LEAD CARDS FOR FILZOM	C-1
D. PROGRAM LISTING AND PROGRAM PRINTER OUTPUT.	D-1
E. EXCERPT FROM DISPLAY PROCESSOR, DOCUMENTATION EOD-LARSYS WITH MODIFICATIONS.	E-1

TABLES

Table		Page
I	FILE ASSIGNMENTS FOR PROCESSORS AND FILZOM	3-2
II	COLORS CONSIDERED IN THIS STUDY	4-3
III	TABLE OF BEST COLORS FOR THREE PRIMARY COLOR CHANNELS.	4-5
IV	TABLE OF BEST COLORS FOR FOUR COLOR CHANNELS.	4-6

FIGURES

Figure		Page
A-1	Hand-painted color-coded CMI image composed of nine colors.	A-3
A-2	Input cards	A-4
A-3	Results of the discriminator.	A-5
B-1	Color-coded Crop Moisture Index Map (original in colors). . .	B-3
B-2	Cards used to process the image in figure B-1	B-4
B-3	Results of the discriminator.	B-6

1. INTRODUCTION

The need for a general method of digitizing zonal maps is detailed in the previous document of this series (An Interactive Method for Digitizing Crop Moisture Index and Other Maps) (ref. 1). In a few words, the need appeared for digitizing Crop Moisture Index (CMI) maps and other background images for use in the Screwworm Eradication Data System (SEDS). When it appeared that no method was available at JSC, the methods described in these documents were developed.

In essence, the work reported here is an automatization of the interactive procedure described earlier (ref. 1). This procedure bypasses the classification steps performed on the Image-100 computer system and the tape construction steps performed on the DAS. It still requires that an image be colored and digitized to form a multichannel tape. A method now in development promises to bypass these steps by producing a final one-channel tape based on the raw CMI data (ref. 2).

In brief, this procedure consists of the following three steps:

- Preparation of image with zones color coded (as described in ref. 1; optimum set of colors described in section 4 of this document)
- Digitization of colored image (as described in ref. 1; see appendix F for reformatting output tape from scanning densitometer of Photo Technology Laboratories)
- Processing of tape with modified EOD-LARSYS system and the FILZOM program to yield an appropriately coded final tape.

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

2. THE EOD-LARSYS SYSTEM AND FILZOM

2.1 INTRODUCTION

The fundamental program for classifying scanner data at JSC in June, 1975, was the operating version of the EOD-LARSYS program. This is an elaboration of classification algorithms developed at the Purdue University Laboratory for Applications of Remote Sensing at West Lafayette, Indiana. The original program has been progressively modified at JSC to meet local needs, and is continually being revised.

The program had most of the elements for classifying the type of data produced by the table scanner of the Image-100 computer of NASA's Earth Observations Division. As such, it had the potential of eliminating the extensive interactive processing that was being performed routinely to digitize maps. Although the classification program was adequate, there was no mechanism for preparing arbitrarily-coded tapes with the classification results. Nor were there special "filter" methods needed to ensure that zones were homogeneous and that discrete boundaries of zones were classified properly as abrupt transitions from one homogeneous zone to the next.

To develop these capabilities, modifications were made to the DISPLAY processor of the current EOD-LARSYS system and the program, Spatial Filter and Zoom Scanner Data (FILZOM), was written. The program FILZOM is extensively documented elsewhere (ref. 4), but the chief features of FILZOM are presented here as a guide to the user.

The modified EOD-LARSYS system and FILZOM are contained on one program tape. The programs may be executed consecutively or separately. Information on FILZOM's deck setup can be obtained from reference 4 and appendix C. Later modifications, as well as current tape numbers, may be obtained from the supervisor of the Data Processing Systems Department, Lockheed Electronics Company, Inc. Phone 713-483-3246 (FTS 525-3246).

2.2 THE EOD-LARSYS SYSTEM

The three EOD-LARSYS processors which are used in producing an input tape, INFIZ, for FILZOM are STAT, CLASSIFY, and DISPLAY. The statistical processor, STAT, computes statistical parameters for the training blocks which are defined as colored blocks on the image. The output tape from STAT along with the original multispectral scanner (MSS) image data tape are input to the classification processor, CLASSIFY. Each pixel on the MSS tape is classified into one of the defined zones and a statistical value of the likelihood that the pixel is classified correctly is assigned to each pixel. The output tape from CLASSIFY (MAPTAP) is input to the display processor, DISPLAY. DISPLAY can perform the following options:

- Produce a line printer map
- Produce a classification summary
- Perform thresholding
- Perform four point spatial filtering
- Output a tape in LARSYS-2 or Universal format.

2.3 SPATIAL FILTER AND ZOOM SCANNER DATA (FILZOM)

The program FILZOM performs many functions. The following topics concerning FILZOM are discussed:

- The need for the FILZOM filter
- The FILZOM filter
- The output image
- Other features.

2.3.1 THE NEED FOR THE FILZOM FILTER

The most unique feature of FILZOM is the filter option, which is described in greater detail in a separate memorandum (ref. 5).

At the time this work was undertaken, there was no filter available. By coincidence, a four-neighbor filter was designed and implemented in the DISPLAY processor parallel to our development of the eight-neighbor multi-filter option.

The version of the DISPLAY processor with the four-neighbor filter was, however, used to make the modifications to output an INFIZ tape. This means that the image filtered in DISPLAY may be further filtered in FILZOM. The FILZOM filter has greater capabilities than the four-point spatial filter because it can do fine filtering as well as gross filtering.

A need for the FILZOM filter was caused by two problems. The first problem comes from imperfections in painting and classifying the image. By their nature, zones must be homogeneous, but as a practical problem, misclassifications occur. This filter allows most of the poorly classified pixels within a zone to be replaced by values assigned to their neighbors.

The second need for the filter is caused by the edge problem between zones. An abrupt change of classification over the boundary is needed between two zones. In practice, boundaries are almost always classified as something other than two zones, because pixels between zones are composed of two or more zones, or paints mix, or shadows occur due to change in relief across a border, thus, causing indistinct boundaries.

2.3.2 THE FILZOM FILTER

FILZOM filters a specified area of the image output by DISPLAY. The extent of the filtering is determined by the following variables which have default values that can be overridden by input lead cards.

- The number of times the image is filtered — lead card FILTER
- The island test variable — lead card ISLAND
- The hole test variable — lead card HOLE.

A discussion of each of the lead cards and the filter algorithm follows:

2.3.2.1 Filter

FILZOM processes a specified area of the input image. (Refer to FILZOM's Lead Cards, appendix C.) The number of times the specified area is filtered is determined by the variable TIMFIL on the lead card FILTER. The maximum number of times an area can be filtered is arbitrarily set in the program to eight. The example below illustrates a multifilter variable of five. (TIMFIL equals 5.)

<u>INPUT IMAGE</u>	<u>WORK FILE Q</u>	<u>WORK FILE R</u>	<u>FINAL TAPE</u>
	<u>Output of Filter 1</u>	<u>Output of Filter 2</u>	
XXXXX } XXXXX } XXXXX } Filter XXXXX } XXXXX }	XXXXX } X111X } X111X } Filter X111X } XXXXX }	XXXXX } X222X } X222X } Filter X222X } XXXXX }	
	<u>Output of Filter 3</u>	<u>Output of Filter 4</u>	<u>Output of Filter 5</u>
	XXXXX } X333X } X333X } Filter X333X } XXXXX }	XXXXX } X444X } X444X } Filter & X444X } Zoom XXXXX }	XXXXX X555X X555X X555X XXXXX

where

X are pixels that are not filtered,

and

1 through 5 represents the number of times a pixel is filtered.

In the example, an input scan line is filtered and output on work file Q. Then, the next scan line is filtered and output on work file Q. This procedure continues until all lines in the specific area have been output on work file Q. Work file Q rewinds and becomes the input file and work file R is the output file. The work files R and Q are used simultaneously with one being the input file and the other being the output file, until the last filtering of the image. The output from the last filter is zoomed and written in Universal or LARSYS-2 format on the final output tape.

The border of the image is not filtered. Only the pixels which are surrounded by eight neighboring pixels can be filtered. Two examples are shown below, one is a classified pixel surrounded by eight neighbors, and the other is the order in which the neighboring pixels are examined.

AAB	517
CXB	3X4
CBB	826

The closest neighbors (1, 2, 3, 4) are examined first.

2.3.2.2 ISLAND and HOLE Lead Cards

The difference between a hole and an island in FILZOM is that an island is a classified pixel surrounded by eight pixels, and a hole is an unclassified (thresholded) pixel surrounded by eight neighbors.

The test variables for spatial filtering an island or a hole are input on lead cards ISLAND and HOLE. These test variables specify the number of neighboring pixels that have to be alike before the center pixel (island or hole) is changed. A test variable for a hole and a test variable for an island can vary for each time the image is filtered. Refer to lead cards ISLAND and HOLE in appendix C.

Using the previous image, which is duplicated on the following page, if the island test variable equals three, the following pixels in the image are examined.

ISLAND TEST VALUE = 3	
	ROWS
	<u>1 2 3</u>
Line 1	A A B
Line 2	C X B
Line 3	C B B

- | | |
|----------------------|----------------------|
| 1) Line 1, Row 2 = A | 4) Line 2, Row 3 = B |
| 2) Line 3, Row 2 = B | 5) Line 1, Row 1 = A |
| 3) Line 2, Row 1 = C | 6) Line 3, Row 3 = B |

There are three B pixels, two A pixels and one C pixel. The center pixel is changed to a B. The remaining two pixels (line 1, row 3; line 3, row 1) are not counted.

If the island test value equals five, all eight neighbors are examined but the center pixel remains X, since there are less than five like surrounding pixels.

Shown below is an image with threshold pixels indicated as T's.

```

A T B
A T T
A T B

```

If the hole test value equals three, the hole (center threshold pixel) would remain a T. The three closest like pixels are T's.

Since it is often desirable to replace as many of the threshold pixels with a classified value, a negative test value on either the ISLAND or HOLE lead card indicates that only the neighboring classified pixels are counted. In the above example using a hole test value of -3, all eight neighbors are examined. The center pixel becomes an A, since there are three pixels that have an A classification.

2.3.3 THE OUTPUT IMAGE

The specific samples and lines which are output on the final tape are determined by parameters which are default values or are input on lead cards SKIP, ZOOM, and LIMIT. (Refer to FILZOM's lead cards.) A discussion of these lead cards follows.

2.3.3.1 SKIP Lead Card

The SKIP lead card (1) designates the number of scan lines that are skipped on the image output by DISPLAY and (2) designates the number of samples that are skipped on each processed scan line.

Example:

	SKIP LINES	=	3			
	SKIP SAMPLES	=	2			
	SAMPLES					
	<u>1 2 3 4 5 6</u>					
Line 1	X	X	X	X	X	
Line 2	X	X	X	X	X	Skipped lines
Line 3	X	X	X	X	X	
Line 4	X	X	X	X	X	
Line 5	X	X	X	X	X	Processed Image
Line 6	X	X	X	X	X	
	Skipped samples					

2.3.3.2 ZOOM Lead Card

The ZOOM lead card designates which pixels within the specified area are multiply defined or skipped. In the previous example, the processing area has three lines and four samples. If the zoom sample factor equals 2, each pixel on a scan line is output two times. If the zoom line factor equals 3,

each scan line is output three times. The equation for calculating the number of times a pixel is output on a scan line follows:

$$Z \cdot (N - S) \leq Y + T$$

where

Z = zoom sample factor

N = sample number on a scan line

S = number of samples to skip on a scan line

T = number of times the N pixel value will be output. T is a whole number.

$$Y = \sum_{s+1}^{n-1} T \quad (\text{sum of previous output pixels})$$

The same equation is used to determine if a scan line is multiply defined or skipped.

2.3.3.3 LIMIT Lead Card

The LIMIT lead card contains the total number of samples and lines that are output on the final tape. The following formula determines if all the pixels on an image are processed or if fill data is needed.

$$T = L - (P - S) * Z$$

where

L = total number of output samples

P = number of pixels on an input scan line

Z = zoom sample factor

S = number of samples to skip on a scan line

T = the number of output pixels which are skipped or the number of threshold pixel values which are added to the end of an output scan line

2.3.4 OTHER FEATURES

Other features of FILZOM are the varied codes which are assigned to the output pixels, the overprint symbols on the line printer map, and the multiple output files.

On the FILZOM's output tape in LARSYS-2 or Universal format, any code between 1 and 255 may be assigned to a classified zone, a threshold field, an unidentifiable field or an unidentifiable 'other' field. The present logic in the DISPLAY processor assigns codes ranging between zero and the number of subclasses or class +5. (Refer to lead cards CODE, DEFAULT C, and DESIGNATE in appendix C.)

After each filter, the image can be zoomed and output in a LARSYS-2 or Universal tape format. In addition, the image can be represented on a line printer map (refer to lead card FILTER). This means that if the data is filtered three times, there can be three files of the zoomed output image and three line printer maps of the image.

The symbols, on the line printer map which represent the various zones, may be the CMI default overprint symbols, the symbols used in the DISPLAY processor, or the symbols input on lead cards.

3. PREPARING DIGITIZED ZONE MAPS

3.1 INTRODUCTION

The use of EOD-LARSYS and FILZOM greatly simplifies the digitization of zone maps. For brevity, this section will not discuss any matters mentioned in the previous document on the interactive process, since these are not changed here.

3.2 PREPARATION OF THE INPUT TAPE TO EOD-LARSYS

Images may be prepared on the Image-100 table scanner as before. In this case, instructions of the previous document would be followed on scanning devices. Then the memory of the Image-100 would be released to a 9-track tape in LARSYS-2 or Universal format.

The densitometer of the Photo Techniques Laboratory might be used. In this case, the tape conversion program in appendix G would convert the tape format to one compatible with the EOD-LARSYS. This would be a 9-track tape. In general, the initial source of digitized data could be a wide variety of densitometers.

3.3 TAPE ASSIGNMENTS AND DECK SETUPS FOR UNIVAC 1108, EXEC II

In executing EOD-LARSYS and FILZOM several files are used. The logical unit for each file is pre-assigned in the programs. (Refer to table I.) The tapes produced on the building 17 Image-100 computer system are 9-track tapes. This presents a problem, in that the EXEC II 1108 computer system at JSC has only one 9-track tape drive. The input 9-track tape has to be copied onto a 7-track tape. Examples of deck setups along with their appropriate Form 588 are presented in the following pages.

TABLE I. — FILE ASSIGNMENTS FOR PROCESSORS AND FILZOM

Processor	Logical Unit and Internal File Name												
	A (SAVTAP)	B (MAPTAP)	C (DATAPE)	H (BMFILE)	K (HISFIL)	L (TRFORM)	M (ERIPTP)	N (DASTAP)	Unassigned (SCRATCH)	I (INFIZ)	P (SAVEFZ)	R (Work File)	Q (Work File)
\$HIST			Tape		FASTRAND								
\$GRAYMAP			Tape		FASTRAND								
\$STAT	Tape or FASTRAND		Tape										
\$ISOCLS	Tape or FASTRAND		Tape				7-track tape	9-track tape	FH432				
\$SELECT	Tape or FASTRAND		Tape	Tape or FASTRAND					FH432				
\$CLASSIFY	Tape or FASTRAND	Tape or FASTRAND	Tape	Tape or FASTRAND									
\$DISPLAY		Tape or FASTRAND						9-track tape	FH432	Tape			
\$DATA- TRANSFORMATION	Tape or FASTRAND		Tape	Tape or FASTRAND		Tape							
FILZOM										Tape	Tape 9-track	Tape or FASTRAND	Tape or FASTRAND

Note: It is suggested that the program tape always be assigned to logical unit Z.

Example I. — Run 1 of 2 copies the input 9-track tape onto a 7-track tape.
Run 2 of 2 executes LARSAA (processors, \$STAT, \$CLASSIFY and \$DISPLAY) and
executes FILZOM.

```

* RUN
* SARN ASG N=1039
* RS ASG C=V01689
* ASG L=$UTIL
* XQT CUR
  IN L
* XQT TAPACK
REWIND,19
REWIND,3
TAPECOPY,19,1,1,,3
REWIND,19
REWIND,3
$END
* FIN

```

```

* RUN
* S ASG Z=V03345      * PROGRAM TAPE FOR LARSAA AND FILZOM
* NSR ASG P=SAV      * OUTPUT FROM FILZOM-----9 TRACK TAPE
* SR ASG B=MAPTAP
* SR ASG i=INFIZ
* SR ASG C=V01689      * DATAPE
* RW ASG R=WORK1
* RW ASG Q=WORK2
* ASG H,A
* COM 03477776
* XQT CUR
  TRW Z
  IN Z
* XQT LARSAA
$STAT
COMMENT      AUTOMATIC PREPARATION OF CMI, 4 CHANNELS FROM IMAGE 100
CHANNELS     1,2,3,4
OPTION       MAXCLS = 22
OPTION       COVAR=C
*END*
CLASSNAME -6.5
SUBCLASS   -6.5
-6.5      ( 1, 1), ( 3, 356), ( 23, 356), ( 23, 375), ( 3, 375)
CLASSNAME +7.5
SUBCLASS   +7.5
+7.5      ( 1, 1), ( 7, 395), ( 20, 395), ( 20, 415), ( 7, 415)
CLASSNAME +6.5
SUBCLASS   +6.5
+6.5      ( 1, 1), ( 7, 440), ( 20, 440), ( 20, 459), ( 7, 459)
CLASSNAME +5.5
SUBCLASS   +5.5
+5.5      ( 1, 1), ( 37, 440), ( 51, 440), ( 51, 459), ( 37, 459)
CLASSNAME +4.5
SUBCLASS   +4.5
+4.5      ( 1, 1), ( 67, 440), ( 84, 440), ( 84, 459), ( 67, 459)
CLASSNAME +3.5
SUBCLASS   +3.5
+3.5      ( 1, 1), ( 98, 440), ( 112, 440), ( 112, 459), ( 98, 459)
CLASSNAME +2.5
SUBCLASS   +2.5
+2.5 C    ( 1, 1), ( 128, 440), ( 142, 440), ( 142, 459), ( 128, 459)
CLASSNAME +1.5

```

```

SUBCLASS +1.5
+1.5 C (1, 1), (155, 440), (172, 440), (172, 459), (155, 459)
CLASSNAME +0.75
SUBCLASS +0.75
+0.75 C (1, 1), (185, 440), (202, 440), (202, 459), (185, 459)
CLASSNAME +0.25
SUBCLASS +0.25
+0.25 C (1, 1), (216, 440), (232, 440), (232, 459), (216, 459)
CLASSNAME -0.25
SUBCLASS -0.25
-0.25 C (1, 1), (247, 440), (263, 440), (263, 459), (247, 459)
CLASSNAME -0.75
SUBCLASS -0.75
-0.75 C (1, 1), (277, 440), (293, 440), (293, 459), (277, 459)
CLASSNAME -1.5
SUBCLASS -1.5
-1.5 C (1, 1), (308, 440), (324, 440), (324, 459), (308, 459)
CLASSNAME -2.5
SUBCLASS -2.5
-2.5 C (1, 1), (339, 440), (355, 440), (355, 459), (339, 459)
CLASSNAME -3.5
SUBCLASS -3.5
-3.5 C (1, 1), (368, 440), (384, 440), (384, 459), (368, 459)
CLASSNAME -4.5
SUBCLASS -4.5
-4.5 C (1, 1), (400, 440), (416, 440), (416, 459), (400, 459)
CLASSNAME -5.5
SUBCLASS -5.5
-5.5 C (1, 1), (431, 440), (447, 440), (447, 459), (431, 459)
CLASSNAME WATER
SUBCLASS WATER
WATER (1, 1), (465, 440), (495, 440), (495, 459), (465, 459)
$END*
$CLASSIFY
CHANNELS 1,2,3,4
OPTION STATS
*END*
WHOLE (1, 1), (1, 1), (512, 1), (512, 512), (1, 512)
$END*
$DISPLAY
COMMENT CMI 27
SYMBOLS G,7,6,5,4,3,2,1,0,+,-,A,B,C,D,E,F,J
OPTION THRESHOLD VALUE
THRESHOLD 17*50.0,500.0
OPTION FIZKEY
*END*
$END*
$EXIT
@ XQT FILZOM
COMMENT CMI 27
FILTER 2 MAP FILES
ISLAND -3 -3
HOLE -3 -3
@E PMD
@ FIN

```


INSTRUCTIONS FOR CENTRAL COMPUTER COMPLEX COMPUTER RUNS

(DO NOT FILL IN SHADED AREAS)

PROGRAMMER'S COMMENTS:

TAPE 1039 IS 9 TRACK

PROGRAMMER			BADGE NO.	BOX NO.	PHONE NO.	DATE	PRIORITY & INITIALS
X			X	X	X	X	
DIVISION CODE	PRG. NO.	PROJ. NO.	EST. TIME	MAX. TIME	PAGES OUTPUT	SEG. NO.	
X	X	X	X	X	X	142	X
OPERATING SYSTEM			TYPE OF RUN		NO. TAPES	NO. FASTR FILES	NO. DRUM FILES
1108 EXEC II	<input checked="" type="checkbox"/>	3200 SCOPE	<input type="checkbox"/>	PROD	<input checked="" type="checkbox"/>	TEST	<input type="checkbox"/>
1108 EXEC VIII	<input type="checkbox"/>	3200 SMARTS	<input type="checkbox"/>	OTHER (EXPLAIN BELOW)			
1108 COBOL	<input type="checkbox"/>	3200 OTHER	<input type="checkbox"/>		2	1	0
INPUT TAPES		WORKING TAPES	OUTPUT TAPES			PERMANENT FASTRAND FILES	
UNIT	REEL NO.	FILE NAME	UNIT	REEL NO.	FILE NAME	SAVE	
N	1039		C	V01689	DATAP	S	
							\$ UTIL
4080	<input type="checkbox"/>	REEL NO.	FILE NO.	PUNCHED OUTPUT	<input type="checkbox"/>	REEL NO.	NO. CARDS
18 MM	<input type="checkbox"/>						
35 MM	<input type="checkbox"/>						
CAL COMP PLOT	<input type="checkbox"/>	REEL NO.	NO. PLOTS	ACTUAL TIME USAGE			
ABNORMAL STOPS		SYSTEM OPERATOR	SYSTEM NO.	STOP			
EXCESS OUTPUT	<input type="checkbox"/>			START			
EXCESS TIME	<input type="checkbox"/>						
OTHER (EXPLAIN BELOW)	<input type="checkbox"/>		PAGES OUTPUT				

OPERATORS COMMENTS:

Example I
Run 1 of 2

INSTRUCTIONS FOR CENTRAL COMPUTER COMPLEX COMPUTER RUNS

(DO NOT FILL IN SHADED AREAS)

PROGRAMMER'S COMMENTS

TAPE SAVE IS 9 TRACK

PROGRAMMER			BADGE NO.	BOX NO.	PHONE NO.	DATE	PRIORITY & INITIALS
X			X	X	X	X	
DIVISION CODE	PROG. NO.	PROJ. NO.	EST. TIME	MAX. TIME	PAGES OUTPUT	REG. NO.	
X	X	X	X	X	X	8431	X
OPERATING SYSTEM			TYPE OF RUN		NO TAPES	NO FASTR FILES	NO DRUM FILES
1108 EXEC II	<input checked="" type="checkbox"/>	3200 SCOPE	<input type="checkbox"/>	PROD	<input checked="" type="checkbox"/>	TEST	<input type="checkbox"/>
1108 EXEC VIII	<input type="checkbox"/>	3200 SMARTS	<input type="checkbox"/>	OTHER (EXPLAIN BELOW)			
1108 COBOL	<input type="checkbox"/>	3200 OTHER	<input type="checkbox"/>				
					7	2	1
INPUT TAPES			WORKING TAPES	OUTPUT TAPES			PERMANENT FASTRAND FILES
UNIT	REEL NO.	FILE NAME	UNIT	REEL NO.	FILE NAME	SAVE	
Z	V03345		P		SAVE		\$
C	V01689		B		MAPTAP		\$
			I		INFIZ		\$
			R		WORK1		\$
			Q		WORK2		\$
							\$
4080	<input type="checkbox"/>	REEL NO.	FILE NO.	PUNCHED OUTPUT	REEL NO.	NO. CARDS	
16 MM	<input type="checkbox"/>			<input type="checkbox"/>			
35 MM	<input type="checkbox"/>						
CAL COMP PLOT	<input type="checkbox"/>	REEL NO.	NO. PLOTS	ACTUAL TIME USAGE			
ABNORMAL STOPS			SYSTEM OPERATOR	SYSTEM NO.		STOP	
EXCESS OUTPUT	<input type="checkbox"/>						
EXCESS TIME	<input type="checkbox"/>						
OTHER (EXPLAIN BELOW)	<input type="checkbox"/>			PAGES OUTPUT		START	

OPERATORS COMMENTS:

Example I
Run 2 of 2

Example II. — In this deck setup, one run card is used. The input 9-track tape on unit P is copied onto unit C. LARSAA and FILZOM are executed. FILZOM writes over the input data on unit P.

```

0Z RUN
0S ASG Z=V03345      0 PROGRAM TAPE FOR LARSAA AND FILZCM
0SARN ASG P=1039     0 INPUT DATAPE OVERWRITTEN WITH FINAL OUTPUT FROM INFIZ
0SR ASG B=MAPTAP
0SR ASG I=INFIZ
0SR ASG C=DATAPE
0RW ASG R=WORK1
0RW ASG Q=WORK2
0 ASG H,A
0 ASG L=$UTIL
0 COM 03477776
0 XQT CUR
  IN L
0 XQT TAPACK
REWIND,19
REWIND,3
TAPECOPY,19,1,1,03
REWIND,19
REWIND,3
$END
0 XQT CUR
  TRW Z
  IN Z
0 XQT LARSAA
$STAT
COMMENT      AUTOMATIC PREPARATION OF CMI, 4 CHANNELS FROM IMAGE 10--CMI 27
CHANNELS     1,2,3,4
OPTION       MAXCLS = 22
OPTION       COVAR=C
*END*
CLASSNAME -6.5
SUBCLASS -6.5
-6.5 (1, 1), ( 3, 356), ( 23, 356), ( 23, 375), ( 3, 375)
CLASSNAME +7.5
SUBCLASS +7.5
+7.5 (1, 1), ( 7, 395), ( 20, 395), ( 20, 415), ( 7, 415)
CLASSNAME +6.5
SUBCLASS +6.5
+6.5 (1, 1), ( 7, 440), ( 20, 440), ( 20, 459), ( 7, 459)
CLASSNAME +5.5
SUBCLASS +5.5
+5.5 (1, 1), (37, 440), ( 51, 440), ( 51, 459), ( 37, 459)
CLASSNAME +4.5
SUBCLASS +4.5
+4.5 (1, 1), ( 67, 440), (84, 440), ( 84, 459), ( 67, 459)
CLASSNAME +3.5
SUBCLASS +3.5
+3.5 (1, 1), ( 98, 440), (112, 440), (112, 459), ( 98, 459)
CLASSNAME +2.5
SUBCLASS +2.5
+2.5 C (1, 1), (128, 440), (142, 440), (142, 459), (128, 459)
CLASSNAME +1.5
SUBCLASS +1.5
+1.5 C (1, 1), (155, 440), (172, 440), (172, 459), (155, 459)
CLASSNAME +0.75
SUBCLASS +0.75
+0.75C (1, 1), (185, 440), (202, 440), (202, 459), (185, 459)
CLASSNAME +0.25
SUBCLASS +0.25
+0.25 (1, 1), (216, 440), (232, 440), (232, 459), (216, 459)

```

```

CLASSNAME -0.25
SUBCLASS -0.25
-0.25 (1, 1), (247, 440), (263, 440), (263, 459), (247, 459)
CLASSNAME -0.75
SUBCLASS -0.75
-0.75 (1, 1), (277, 440), (293, 440), (293, 459), (277, 459)
CLASSNAME -1.5
SUBCLASS -1.5
-1.5 (1, 1), (308, 440), (324, 440), (324, 459), (308, 459)
CLASSNAME -2.5
SUBCLASS -2.5
-2.5 (1, 1), (339, 440), (355, 440), (355, 459), (339, 459)
CLASSNAME -3.5
SUBCLASS -3.5
-3.5 (1, 1), (368, 440), (384, 440), (384, 459), (368, 459)
CLASSNAME -4.5
SUBCLASS -4.5
-4.5 (1, 1), (400, 440), (416, 440), (416, 459), (400, 459)
CLASSNAME -5.5
SUBCLASS -5.5
-5.5 (1, 1), (431, 440), (447, 440), (447, 459), (431, 459)
CLASSNAME WATER
SUBCLASS WATER
WATER (1, 1), (465, 440), (495, 440), (495, 459), (465, 459)
$END*
$CLASSIFY
CHANNELS 1,2,3,4
OPTION STATS
*END*
WHOLE (1, 1), (1, 1), (512, 1), (512, 512), (1, 512)
$END*
$DISPLAY
COMMENT CMI 27
SYMBOLS G,7,6,5,4,3,2,1,0,+,-,A,B,C,D,E,F,J
OPTION THRESHOLD VALUE
THRESHOLD 17*50.0,500.0
OPTION FIZKEY
*END*
$END*
$EXIT
@ XQT FILZOM
COMMENT CMI 27
FILTER 2 MAP FILES
ISLAND -3 -3
HOLE -3 -3
@E PMD

```

INSTRUCTIONS FOR CENTRAL COMPUTER COMPLEX COMPUTER RUNS

(DO NOT FILL IN SHADED AREAS)

PROGRAMMER'S COMMENTS

Tape 1039 IS 9 TRACK.

PROGRAMMER			HADGE NO.	BOX NO.	PHONE NO.	DATE	PRIORITY & INITIALS	
<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
DIVISION CODE	PRIG. NO.	PROJ. NO.	EST. TIME	MAX. TIME	PAGES OUTPUT	SEG. NO.		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	
OPERATING SYSTEM			TYPE OF RUN			NO. TAPES	NO. FASTR FILES	NO. DRUM FILES
1108 EXEC II	<input checked="" type="checkbox"/>	3200 SCOPE	<input type="checkbox"/>	PROD	<input checked="" type="checkbox"/>	TEST	<input type="checkbox"/>	
1108 EXEC VIII	<input type="checkbox"/>	3200 SMARTS	<input type="checkbox"/>	OTHER (EXPLAIN BELOW)				
1108 COBOL	<input type="checkbox"/>	3200 OTHER	<input type="checkbox"/>					
						7	3	1
INPUT TAPES			WORKING TAPES	OUTPUT TAPES			PERMANENT FASTRAND FILES	
UNIT	REFL. NO.	FILE NAME	UNIT	REEL NO.	FILE NAME	%AVI		
Z	V03345		P	1039			\$	
			B		MAPTAP		\$ # UTIL	
			I		INFIZ		\$	
			C		DATAP		\$	
			R		WORK1		\$	
			Q		WORK2		\$	
4080	<input type="checkbox"/>	REEL NO.	FILE NO.	PUNCHED OUTPUT	<input type="checkbox"/>	REEL NO.	NO. CARDS	
18 MM	<input type="checkbox"/>							
35 MM	<input type="checkbox"/>							
CAL COMP PLOT	<input type="checkbox"/>	REEL NO.	NO. PLOTS	ACTUAL TIME USAGE				
ABNORMAL STOPS			SYSTEM OPERATOR	SYSTEM NO.	STOP			
EXCESS OUTPUT	<input type="checkbox"/>				START			
EXCESS TIME	<input type="checkbox"/>			PAGES OUTPUT				
OTHER (EXPLAIN BELOW)	<input type="checkbox"/>							
OPERATORS COMMENTS:								

Example II

Example III. — FILZOM is executed using the saved output tape, INFIZ, from a previous execution of processor \$DISPLAY.

```

● RUN
●RW ASG R=WORK1
●RW ASG Q=WORK2
●SR ASG I=VOI524
●NSR ASG P=SAV
●S ASG Z=V03345
● XQT CUR
  TRW Z
  IN Z
  TOC
● XQT FILZOM
SKIP          29
ZOOM          .5   .5
LIMIT        151  220
OUTPUT        L
FILTER        2    MAP FILES
COMMENT      CMI TEST ----- NO THRESHOLDING HAS BEEN USED
ISLAND       4    5
●E PND
● FIN

```

● INFIZ TAPE

● OUTPUT FROM FILZOM-----9 TRACK TAPE

● PROGRAM TAPE FOR LAR9AA AND FILZOM

INSTRUCTIONS FOR CENTRAL COMPUTER COMPLEX COMPUTER RUNS

(DO NOT FILL IN SHADED AREAS)

PROGRAMMER'S COMMENTS

TAPE SAVE IS 9 TRACK

PROGRAMMER <input checked="" type="checkbox"/>			BADGE NO. <input checked="" type="checkbox"/>	BOX NO. <input checked="" type="checkbox"/>	PHONE NO. <input checked="" type="checkbox"/>	DATE <input checked="" type="checkbox"/>	PRIORITY & INITIALS
DIVISION CODE <input checked="" type="checkbox"/>	PROG. NO. <input checked="" type="checkbox"/>	PROJ. NO. <input checked="" type="checkbox"/>	EST. TIME <input checked="" type="checkbox"/>	MAX. TIME <input checked="" type="checkbox"/>	PAGES OUTPUT <input checked="" type="checkbox"/>	SLG. NO.	<input checked="" type="checkbox"/>
OPERATING SYSTEM			TYPE OF RUN		NO TAPES	NO. FASTR FILES	NO DRUM FILES
1108 EXEC II	<input checked="" type="checkbox"/>	3200 SCOPE	<input type="checkbox"/>	PROD.	<input checked="" type="checkbox"/>	TEST	<input type="checkbox"/>
1108 EXEC VIII	<input type="checkbox"/>	3200 SMARTS	<input type="checkbox"/>	OTHER (EXPLAIN BELOW)			
1108 COBOL	<input type="checkbox"/>	3200 OTHER	<input type="checkbox"/>				
INPUT TAPES			WORKING TAPES	OUTPUT TAPES			PERMANENT FASTRAND FILES
UNIT	REEL NO.	FILE NAME	UNIT	REEL NO.	FILE NAME	SAVL	
Z	V03345		P		SAVE		\$
I	V01524INEI2		R		WORK1		\$
			Q		WORK2		\$
							\$
							\$
							\$
							\$
4060	<input type="checkbox"/>	REEL NO.	FILE NO.	PUNCHED OUTPUT	<input type="checkbox"/>	REEL NO.	NO. CARDS
16 MM	<input type="checkbox"/>						
36 MM	<input type="checkbox"/>						
CAL COMP PLOT	<input type="checkbox"/>	REEL NO.	NO. PLOTS	ACTUAL TIME USAGE			
ABNORMAL STOPS			SYSTEM OPERATOR	SYSTEM NO.	STOP		
EXCESS OUTPUT	<input type="checkbox"/>				[]		
EXCESS TIME	<input type="checkbox"/>				[]		
OTHER (EXPLAIN BELOW)	<input type="checkbox"/>				START		
				PAGES OUTPUT	[]		
OPERATORS COMMENTS:							

Example III

4. SELECTION OF COLORS

4.1 INTRODUCTION

Crop Moisture Index maps have been digitized weekly for a year at this writing. The interactive procedure described previously and the automatic procedure described herein were applied to hand-colored images prepared with tempera paints.

For the interactive procedure, the principal requirement for selection of colors was that colors of adjacent zones be very distinct. It was convenient if each zone was classified unambiguously, but this was not necessary. However, in the automatic system described herein, it is vital that colors classify correctly. Major misclassifications will result in major errors on the final digital tapes. For this reason, a modest study of appropriate colors was made. The results are reported in the following sections.

4.2 AVAILABLE SYSTEMS FOR PREPARING COLORED IMAGES

There is a wide variety of coloring systems available for preparation of opaque images. For this purpose, we needed a system that met the following requirements.

1. It would need to be rapid and to be completed in a few hours.
2. It needed to be suitable for a relatively untrained person to do.
3. Color zones needed to be homogeneous with the colors the same over the entire image and not to be glossy.
4. There should be no relief in the surface.
5. Colors needed to be opaque if they were to be used to paint over a figure on paper.
6. Fairly small detail (for example, a millimeter) needed to be easy to color.

- | | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | |
|--|--|--|--|--|--|--|--|
7. Boundaries should be abrupt — without transition.
 8. Final image size should be about 6 or 8 inches on a side for digitization on the Image-100.

Probably, the best system in the hands of an artist involved opaque acrylic paints, but which are difficult for an untrained person to use. For a trained person, homogeneous colored paper could be the best system. This would be convenient if the image were large enough and were to be reduced photographically. No experiments were made with this system. Preparation of a transparency was also tried using colored film. The edges were not sufficiently precise in our trial cases, and preparation of the colored image was difficult. By far, the best system was one involving the printing of the master image, as illustrated in appendix C, but it is much too slow and expensive for routine use. For operational use in support of the Screw-worm Eradication Project, a system was developed for overpainting paper diagrams with opaque tempera paints. This represented a compromise in all the requirements, but it satisfied the needs of the project.

4.3 AVAILABLE TEMPERA PAINTS

For routine use, it was necessary to use a wide variety of tempera paints. It was quickly found that available paints vary widely in quality. Some would not cover a pencil or xerox image adequately, even though they were classified as opaque paints. Others did not leave uniform colors because of separation of pigments.

On the advice of professionals, only colors of a single quality brand were used. The one most accessible to us were the Rich Art colors shown in table I. These proved to be good colors for this purpose, although undoubtedly other brands of paints would also be useful.

4.4 SELECTION OF COLORS

A brief study was made of the separability of tempera colors. A color table was prepared to include all the colors in our possession. The ISOCLS

TABLE II. —COLORS CONSIDERED IN THIS STUDY

RICH ART POST (TEMPERA) COLOR		RICH GLO FLUORESCENT WATER COLOR
1 White (Spectrum)	*64 Prussian Blue	Daylight Fluorescent — Glows Under Black Light
14 Poster White	*65 Navy Blue	
18 Gray (Spectrum)	70 Yellow Green (Spectrum)	
20 Black (Spectrum)	71 Emerald Green	
*24 Poster Black	72 Green (Spectrum)	
30 Light Yellow	*73 Dark Green	D14 White
31 Yellow (Spectrum)	*74 Extra Dark Green	D30 Lemon Yellow
32 Dark Yellow	75 Blue Green (Spectrum)	D40 Yellow Orange
40 Yellow Orange (Spectrum)	76 Olive Green	D42 Orange
42 Orange (Spectrum)	*77 Holly Green	D43 Red Orange
43 Red Orange (Spectrum)	*78 Cyprus Green	D51 Red
*44 Flesh	80 Lavender	D52 Cerise Red
50 Blazing (Light) Red	*81 Light Magenta	D62 Blue
51 Vermillion	82 Red Violet (Spectrum)	D72 Green
52 Red (Spectrum)	83 Violet (Spectrum)	*D81 Lt. Magenta (Pink)
53 Crimson Red	84 Blue Violet (Spectrum)	D82 Deep Magenta
*54 Dark Red	90 Yellow Ochre	D83 Violet
56 Brilliant Red	91 Brown (Spectrum)	
*57 Pink	92 Burnt Sienna	
58 Maroon	93 Burnt Umber	
60 Light Blue	*94 Vandyke Brown	
61 Turquoise Blue	*95 Raw Umber	
62 Blue (Spectrum)	*97 Raw Sienna	
63 Ultramarine Blue		

All "Rich Art" Poster (Tempera) Colors and "Rich Glo" Fluorescent colors are listed, but starred colors were not used. Probably the "Rich Glo Acral" fluorescent acrylic poster colors would be identical in color to the fluorescent colors but more convenient to use.

These colors are available from art supply houses or direct from the Rich Art Color Company, 31 West 21st. Street, New York, N.Y., 10010.

No endorsement of Rich Art products is implied. Undoubtedly, other brands would be equally useful to this use. Rich Art colors happened to be the most accessible high quality tempera colors at the time of this investigation.

processor of EOD-LARSYS was used to determine which colors were most easily separable, and the results are shown in tables II and III.

This is only a quick way of determining separability of colors, and its results are only presented as a convenient starting point. This brief study suffered from the following defects:

- Only a single experiment was made.
- Results depend on the filters used, as specified in the table. Other filters might give a different set of colors.
- Lighting of the image was arbitrary. Photofloods were used with varying voltages produced by an autotransformer. Fluorescent lights would probably yield different color sets.
- The experiment was limited to the Image-100 system. The spectra characteristics of its television cameras are not known.

In spite of these defects, these color sets should provide a useful starting point for choosing an optimum set.

4.5 USE OF THE COLOR TABLES

Since a near-infrared channel is sometimes used in addition to three visible colors, two tables were needed. Both are used in the same way, but table II will be used for illustration.

If 23 zones are needed, use all the colors in the 23-zone column. If 22 are needed, remove No. 75 because it is listed as 1 in the elimination order column. If 21 are needed, eliminate color 84. Table II, used in this way, can specify optimum sets for 5 to 23 zones.

Colors in these tables are arranged such that similar colors are adjacent. As a result, this order should be preserved for zones in numerical order. For example, if ten zones are to be coded numerically, 10, 20, 30, etc.,

TABLE III. - TABLE OF BEST COLORS FOR THREE PRIMARY COLOR CHANNELS

23 ZONES		15 ZONES		10 ZONES		5 ZONES
Color	Elimi- nation Order	Color	Elimi- nation Order	Color	Elimi- nation Order	Color
1		1		1		1
D30		D30		D30	3	50
31		31	2	50		63
D43	3	50		blue + 1	4	90
50		42	1	63		20
D51	6	53	5	18	1	
42		blue + 1		70	5	
53		61	3	80	2	
blue + 1		63		90		
61		18		20		
63		70				
18		72	4			
72 + 1	4	80				
70		90				
72		20				
75	1					
82 + 1	8					
80						
84	2					
90						
91	7					
93	5					
20						

These lists are based on LARSYS clustering of hand-painted color tables. Data were read through blue, yellow, and red filters (Wratten filters 98, 50Y, and 25 into three channels of the Image-100. Data was read from the machine in LARSYS-2 format and clustered using the EOD LARSYS program.

The elimination order numbers specify the order of removal of colors. For example, if only thirteen zones are to be distinguished, colors 31 and 42 should be omitted from the 15 zone list of best colors.

TABLE IV. — TABLE OF BEST COLORS FOR FOUR COLOR CHANNELS
 (THREE PRIMARY COLORS PLUS NEAR INFRARED)

23 ZONES		15 ZONES		10 ZONES		5 ZONES
Color	Elimi- nation Order:	Color	Elimi- nation Order	Color	Elimi- nation Order	Color
1		1		1		1
D30		D30		D30	4	50
31		31	1	50		62 + 1
D43	5	50		62 + 1		90
50		53	2	61	3	20
D51	8	62 + 1		18	2	
42	1	61		72	5	
53		63	4	90		
62 + 1		18		93	1	
61		72		20		
63		75	5			
18		80	3			
72 + 1	7	90				
70	2	93				
72		20				
75						
82 + 1	3					
80						
84	4					
90						
91	6					
93						
20						

These lists were compiled like those on table 2, except that near-infrared radiation, using a Wratten 89B filter, was used as a fourth color channel.

they should be represented respectively, by colors 1, D30, 50, blue plus 1, etc., as shown in the table. Then, if a pixel in zone 30 is not properly classified, it will most likely be misclassified as 20 or 40, not by numbers that are more erroneous.

These color tables are also used for interactive classification schemes, as in reference 1. In such a case, colors should be attenuated in some way such that adjacent zones are unlike colors. In the above example the same ten zones might be coded 1, 18, D30, 70, 50, 80, blue + 1, 90, 63, and 20.

5. REFERENCES

1. Giddings, L. E.; and Thompson, E. J.: An Interactive Method for Digitizing Crop Moisture Index and Other Maps. JSC-09809, LEC-6498 (1975).
2. Giddings, L. E.; and Mejia, Raúl: Preparation of Digital Tapes by Interpolation. JSC-10758, LEC-7499, to be issued in 1976.
3. Minter, Ruth; Gardner, C. T.; Wills, B. E.; and Corbett, B. W.: User Documentation EOD-LARSYS, Earth Observation Divisions Version of the Laboratory for Applications of Remote Sensing System Program Q619, LEC-3984 (1975).
4. Boston, Sydney: User Documentation Spatial Filter and Zoom Scanner Data (FILZOM) Program Q945. JSC 11130. LEC-8097 to be published in 1976.
5. Giddings, L. E.; and Boston, Sydney: The Eight-Neighbor Filter Option of the LSD DISPLAY Processor. JSC-10795. LEC-7619.

APPENDIX A

DIGITIZATION OF A SIMPLE ARTIFICIAL IMAGE

The method described here was designed for use with images such as that shown in figure A-1. This hand-painted image is composed of nine colors, each of which identifies a zone which is to be assigned a given value of crop moisture index. Adjacent colors were chosen for contrast, not for similarity. Therefore, misclassifications will tend to be serious.

Figure A-2 shows the input cards for this case, starting from an output tape from LARSYS "CLASSIFY", and figure A-3 shows the results of the discriminator. Note, again that adjacent zones were chosen for contrast, whereas they would best have been chosen for similarity.

Figure A-3 shows the final result of the classification and assignment of zone values after four applications of a three-neighbor filter. Note that there are a few misclassifications, which are to be expected because of this low quality of illumination used with the Image-100 table scanner.

#37

A-3

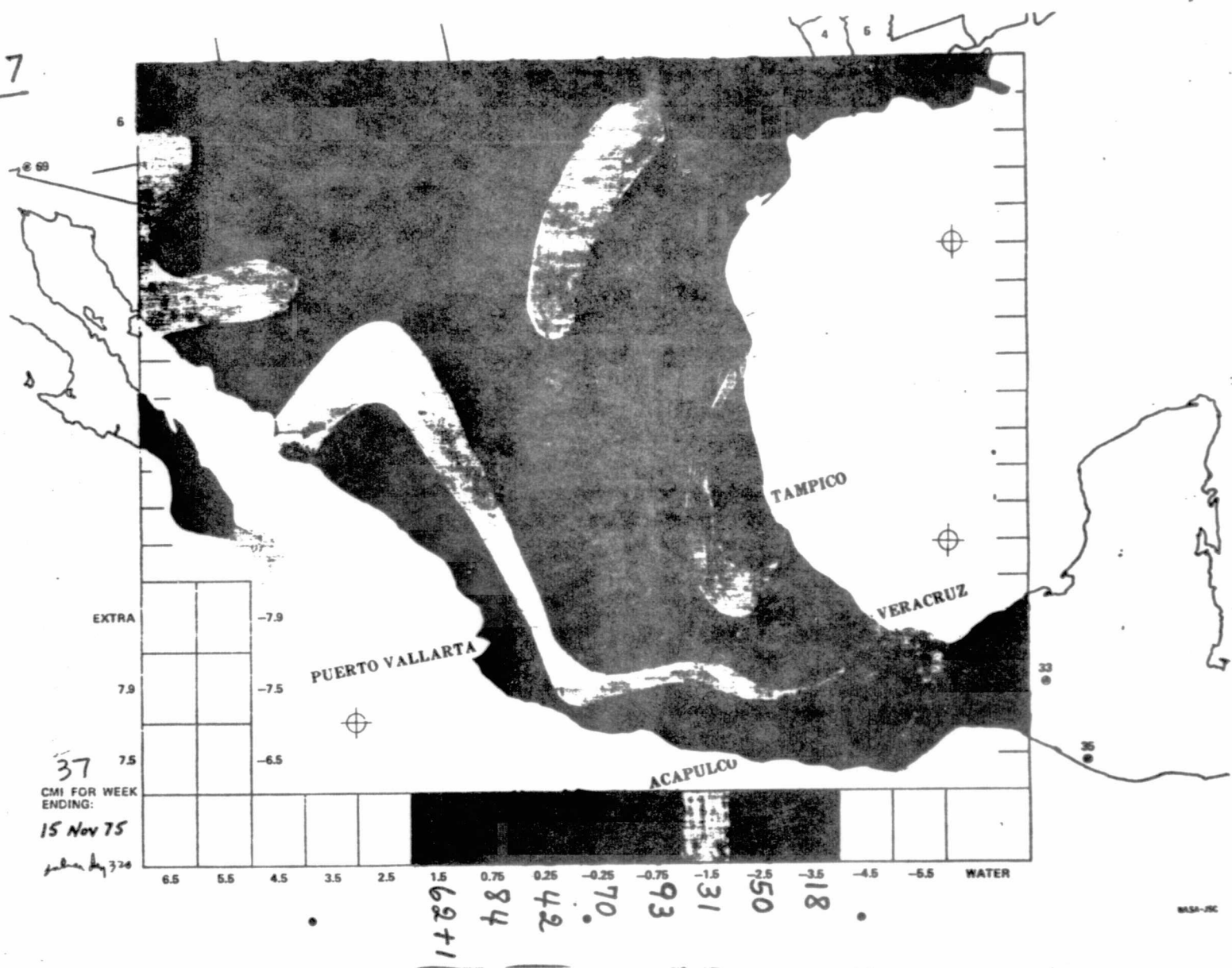


Figure A-1. - Hand-painted color-coded CMI image composed of nine colors.

BOSTON CMI 37

```

@Z RUN L7885,FD52,H3A,4803D,J999,C,30,99
@NSR ASG P=SAVE      @ OUTPUT FROM FILZOM-----9 TRACK TAPE
@SR ASG I=INFIZ      @ CMI INFIZ
@RW ASG R=WORK1
@RW ASG Q=WORK2
@S ASG Z=V00577      @ PROGRAM TAPE FOR LARSAA AND FILZOM
@SR ASG B=V02530     @ CMI MAPTAP
@ XQT CUR
  TRW Z
  IN Z
  TOC
@ XQT LARSAA
$DISPLAY
COMMENT  CMI NO. 37 FOR WEEK ENDING 15 NOV 75 THRESHOLDING
THRESHOLD @*50.0,500.0
OPTION   THRESHOLD VALUE
SYMBOLS  1,2,3,4,5,6,7,8,W.
OPTION   FIZKEY
*END*
$END*
$EXIT
@ XQT FILZOM
OUTPUT      L
FILTER      8   MAP FILES
COMMENT     CMI NO. 37 FOR WEEK ENDING 15 NOV 75 THRESHOLDING
ISLAND      0   -3   -3   -3   -3   -3   -3   -3
HOLE        0   -3   -3   -3   -3   -3   -3   -3
@E PMD
@ FIN

```

Figure A-2. - Input cards.



Figure A-3.— Results of the discriminator.

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B

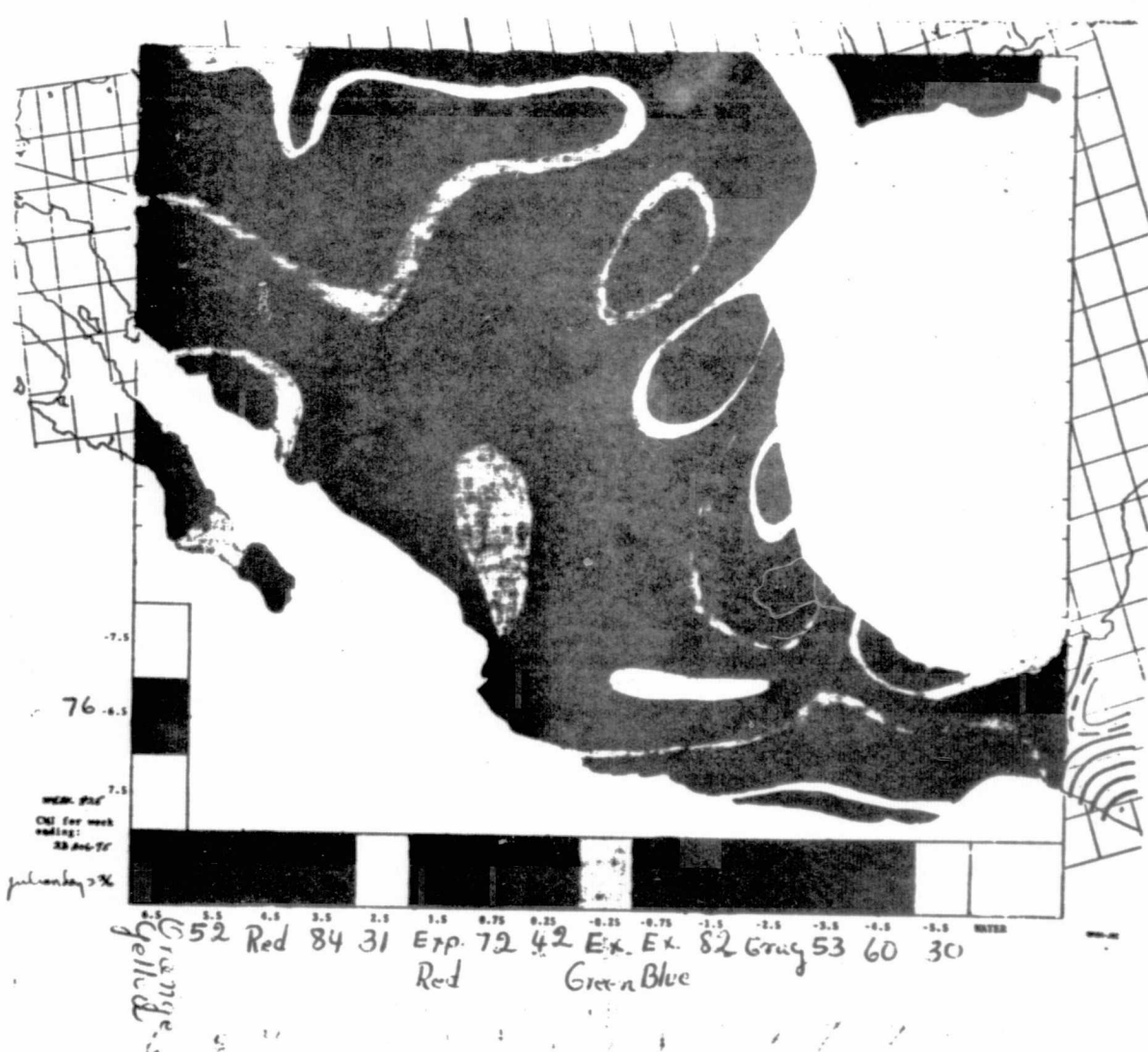
DIGITIZATION OF A COMPLEX ARTIFICIAL IMAGE

The quality of zone-discrimination of this method depends very strongly on the homogeneity of illumination of the scanning device. Since illumination for opaque images on the Image-100 is only makeshift, zone discrimination should be expected to be only fair.

Figure B-1 presents an image which is considerably more complex than A-1. There are 17 zones to be discriminated in this figure.

Figure B-2 reproduces the cards used to process this image, starting with a 7-track copy of the 9-track tape produced by the Image-100 computer.

The final zone image (fig. B-3) shows that the discriminator worked quite well. In all cases, misidentifications are serious only because similar colors are not adjacent; for example, three blues are coded as 3.5, -0.75, and -4.5. If similar colors were adjacent, no misidentifications would have been serious.



#95

B-3

Figure B-1. - Color-coded Crop Moisture Index Map (original in colors).

STARTED PAINTING AT 1 P.M.
FINISHED PAINTING AT 5 P.M.
AUGUST 26/75 *bi*

```

@Z RUN L78385,FD52,H3A,4803D,J999,C,30,99
@S ASG Z=V03345 @ PROGRAM TAPE FOR LARSAA AND FILZOM
@SR ASG C=V01689
@SR ASG B=X02162
@NSR ASG P=X01741 @ OUTPUT FROM FILZOM-----9 TRACK TAPE
@RW ASG R=WORK1
@RW ASG Q=WORK2
@SR ASG I=X00643
@SR ASG A
* ASG H
@ XQT CUR
  TRW Z
  IN Z
  TOC

```

BOSTON CMI 37

```

@ XQT LARSAA
$STAT
COMMENT      AUTOMATIC PREPARATION OF CMI, 4 CHANNELS FROM IMAGE 100
CHANNELS     1,2,3,4
OPTION       MAXCLS = 22
OPTION       COVAR=C
*END*
CLASSNAME   -6.5
SUBCLASS    -6.5
-6.5        (1,1),(13,363),(20,363),(20,377),(13,377)
CLASSNAME   +6.5
SUBCLASS    +6.5
+6.5        (1,1),(13,440),(20,440),(20,453),(13,453)
CLASSNAME   +5.5
SUBCLASS    +5.5
+5.5        (1,1),(43,440),(51,440),(51,453),(43,453)
CLASSNAME   +4.5
SUBCLASS    +4.5
+4.5        (1,1),(73,440),(84,440),(84,453),(73,453)
CLASSNAME   +3.5
SUBCLASS    +3.5
+3.5        (1,1),(104,440),(112,440),(112,453),(104,453)
CLASSNAME   +2.5
SUBCLASS    +2.5
+2.5        (1,1),(134,440),(142,440),(142,453),(134,453)
CLASSNAME   +1.5
SUBCLASS    +1.5
+1.5        (1,1),(161,440),(172,440),(172,453),(161,453)
CLASSNAME   +0.75
SUBCLASS    +0.75
+0.75       (1,1),(191,440),(202,440),(202,453),(191,453)
CLASSNAME   +0.25
SUBCLASS    +0.25
+0.25       (1,1),(222,440),(232,440),(232,453),(222,453)
CLASSNAME   -0.25
SUBCLASS    -0.25
-0.25       (1,1),(253,440),(263,440),(263,453),(253,453)
CLASSNAME   -0.75
SUBCLASS    -0.75
-0.75       (1,1),(283,440),(293,440),(293,453),(283,453)
CLASSNAME   -1.5
SUBCLASS    -1.5
-1.5        (1,1),(314,440),(324,440),(324,453),(314,453)

```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure B-2. — Cards used to process the
image in figure B-1.

```

CLASSNAME -2.5
SUBCLASS -2.5
-2.5 (1,1),(345,440),(355,440),(355,453),(345,453)
CLASSNAME -3.5
SUBCLASS -3.5
-3.5 (1,1),(374,440),(384,440),(384,453),(374,453)
CLASSNAME -4.5
SUBCLASS -4.5
-4.5 (1,1),(406,440),(416,440),(416,453),(406,453)
CLASSNAME -5.5
SUBCLASS -5.5
-5.5 (1,1),(437,440),(447,440),(447,453),(437,453)
CLASSNAME WATER
SUBCLASS WATER
BLANK (1,1),(13,323),(20,323),(20,337),(13,337)
PAINT (1,1),(471,440),(495,440),(495,453),(471,453)
$END*
$CLASSIFY
CHANNELS 1,2,3,4
OPTION STATS
*END*
WHOLE (1, 1), (1, 1), (512, 1), (512, 512), (1, 512)
$END*
$DISPLAY
COMMENT CMI WEEK ENDING 23 AUG 75 NUMBER 25
SYMBOLS G,6,5,4,3,2,1,0,+,-,A,B,C,D,E,F,J
OPTION THRESHOLD VALUE
THRESHOLD 16*50.0,500.0
OPTION FIZKEY
*END*
$END*
$EXIT
@ XQT FILZOM
COMMENT CMI WEEK ENDING 23 AUG 75 NUMBER 25
FILTER 5 MAP FILES
ISLAND 0 -3 -3 -3 -3
HOLE 0 -3 -3 -3 -3
OUTPUT L
@E PMD
@FIN

```

Figure B-2. — Cards used to process the image in figure B-1. (concluded)

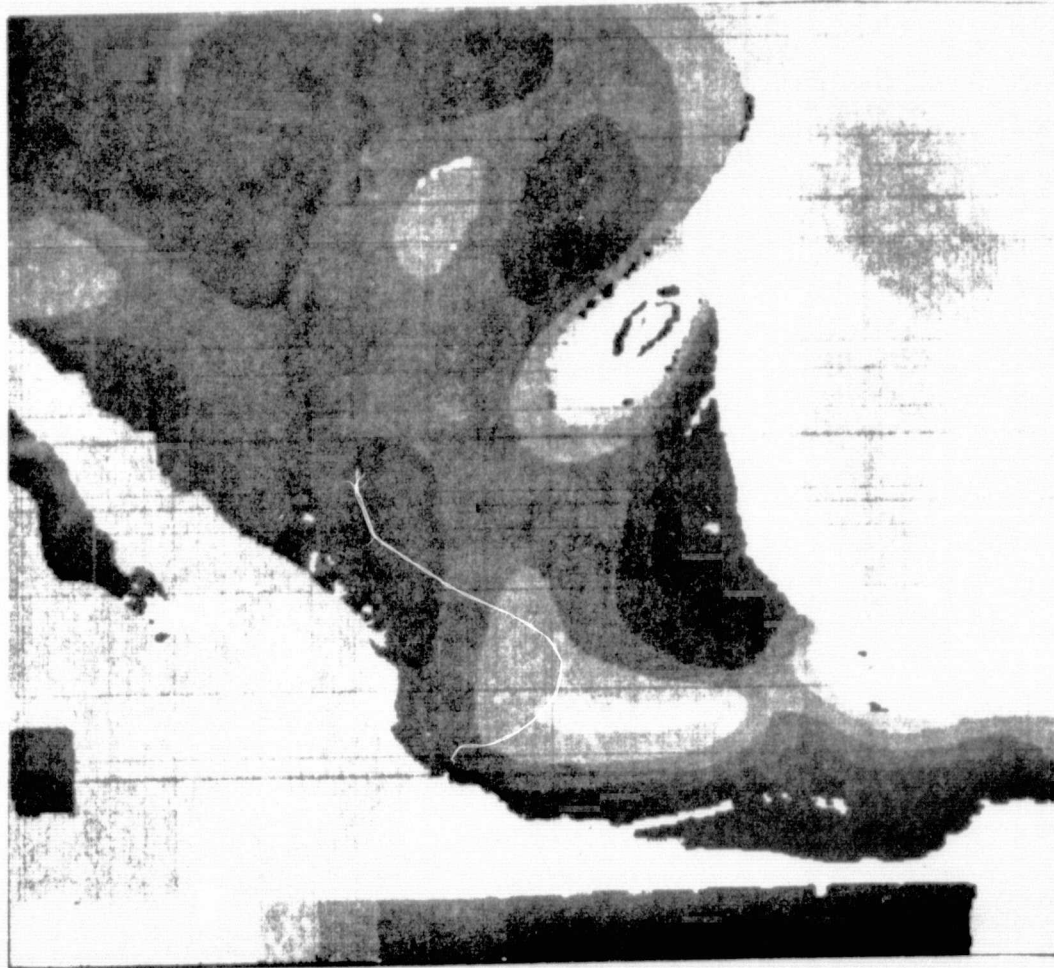


Figure B-3. — Results of the discriminator.

APPENDIX C

ALPHABETICAL LISTING OF LEAD CARDS FOR FILZOM

APPENDIX C

ALPHABETICAL LISTING OF LEAD CARDS FOR FILZOM

<u>IDENTIFICATION</u>	<u>PAGE</u>
CODE	1
COMMENT	2
DATE	3
DEBUG	4
DEFAULT C	5
DEFAULT S	6
DESIGNATE	7
FILE	8
HED1	9
HED2	10
HOLE	11
ISLAND	12
FILTER	13
LIMIT	14
*NOSEDS	15
OUTPUT	16
SKIP	17
SYMBOL	18
ZOOM	19

*If a NOSEDS lead card is read, check the default values for cards LIMIT, SKIP and ZOOM.

CARD NO. _____

JOB FILZOM
NAME DEBUGΔΔΔΔΔ

LEAD CARD SET UP
PROGRAMMER Boston DATE _____

PAGE NO. 1 OF 1

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1&2	1-10	A6,A4	CARD1 CARD2	Enter 'DEBUGΔΔΔΔΔ'.
3	11-15	I5	DEBUG F	Enter a "1" (right justified) for debug information. If DEBUGF equals one, the input training fields and test fields are printed.

COMMENTS Optional card. Default is no debug information is printed.

CARD

NO. _____
JOB FILZOM
NAME DEFAULTASA

LEAD CARD SET UP
PROGRAMMER Boston

PAGE NO. 1 OF 1
DATE _____

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1&2	1-10	A6,A4	CARD1 CARD2	Enter 'DEFAULTASA'.
3	15	A1	DFTSYM(1)	Enter one of the default symbols for a pixel that is unclassified.
4	20	A1	DFTSYM(2)	Enter the overprint symbol for a pixel that is unclassified.

COMMENTS Optional card. The default values are DFTSYM(1) = 0 and DFTSYM(2) = blank

CARD

NO. _____
 JOB FILZOM
 NAME DESIGNATEA

LEAD CARD SET UP PAGE NO. 1 OF 1
 PROGRAMMER BOSTON DATE _____

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1&2	1-10	A6,A4	CARD 1 CARD 2	Enter 'DESIGNATEA'.
3	11-15	I5	DESCOD(1)	Enter a code (right justified) for fields that are designated as 'unidentifiable'. The code must be unique. Refer to lead cards CODE and DEFAULTAC. Also a code cannot be smaller than 1 or larger than 255.
4	20	A1	DESSYM(1,1)	Enter one of the symbols for a field that is designated as 'unidentifiable'.
5	25	A1	DESSYM(2,1)	Enter the overprint symbol for a field that is designated as 'unidentifiable'.
6	31-35	I5	DESCOD(2)	Enter a code (right justified) for fields that are designated as 'other'.
7	45	A1	DESSYM(1,2)	Enter one of the symbols for a field that is designated as 'other'.
8	45	A1	DESSYM(2,2)	Enter the overprint symbol for a code that is designated as 'other'.

COMMENTS Optional card. The default values are DESCOD(1)=255, DESSTM(1,1)=?, DESSYM(2,1)=Δ, DESCOD(2)=2, DESSYM(1,2)=#, DESSYM(2,2)=Δ

CARD

NO.

JOB FILZOM

LEAD CARD SET UP

PAGE NO. 1 OF 1

NAME HED2ΔΔΔΔΔΔΔ

PROGRAMMER Boston

DATE _____

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1&2	1-10	A6,A4	CARD1 CARD2	Enter 'HED2ΔΔΔΔΔΔΔ'.
3-12	11-70	10A6	HED2	Any 60 characters which are desired for the 2nd line of the page heading.

COMMENTS Optional card. Default: HOUSTON, TEXAS

CARD

NO. _____

JOB FILZOM

NAME ISLAND

LEAD CARD SET UP

PAGE NO. _____ OF _____

PROGRAMMER BOSTON

DATE _____

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1&2	1-10	A6,A4	CARD1 CARD2	Enter 'ISLANDΔΔΔΔΔ'.
3	11-35	5I5	ISLTST	For each time the input data is filtered enter an ISLTST value (right justified). A positive ISLTST value is the number of classified or unclassified pixels that have to be equal before the center classified pixel is changed. A negative ISLTST value is the number of classified pixels that have to be equal before the center classified pixel is changed.
				Valid ISLTST values follow:
				3 -3
				4 -4
				5 -5
				6 -6
				7 -7
				8 -8
				0 for no filtering

COMMENTS Optional card. For each time the input data is filtered, the default ISLTST value equals 5.

CARD

NO. _____

JOB FILZOM

LEAD CARD SET UP

PAGE NO. 1 OF 1

NAME FILTER

PROGRAMMER Boston DATE _____

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1&2	1-10	A6,A4	CARD1 CARD2	Enter 'FILTERΔΔΔΔ'.
3	11-15	I5	TIMFIL	Enter the number of times (right justified) the input data will be filtered. The TIMFIL value cannot be larger than 8. The default is TIMFIL = 1.
4	20-22	A3	FILMAP	If a greymap is desired for each time the input data is filtered, enter 'MAP'. The default is a greymap of the last time the data is filtered.
5	25-29	A5	FILEFIL	If an output file is desired for each time the data is filtered, enter 'FILES'. The default is only an output file of the last time the data is filtered.

COMMENTS Optional card.

CARD

NO. _____
JOB FILZOM
NAME LIMIT

LEAD CARD SET UP
PROGRAMMER _____

PAGE NO. 1 OF 1
Boston DATE _____

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1&2	1-10	A6,A4	CARD1 CARD2	Enter 'LIMITΔΔΔΔΔ'.
3	11-15	I5	LIMLIN	Enter the number of lines (right justified) desired on the output tape. A blank or zero is not allowed. Also LIMLIN cannot be larger than 1,000.
4	16-20	I5	LIMSAM	Enter the number of samples (right justified) per line desired on the output tape. A blank or zero is not allowed. Also LIMSAM cannot be larger than 1,000.

COMMENTS Optional card. The default values are LIMLIN = 550 and LIMSAM = 625.

CARD

NO. _____

JOB FILZOM

LEAD CARD SET UP PAGE NO. 1 OF 1

NAME SYMBOLΔΔΔΔ

PROGRAMMER Boston

DATE _____

FIELD I. D.	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION																																												
1&2	1-10	A6,A4	CARD1 CARD2	Enter 'SYMBOLΔΔΔΔ'.																																												
3-62	11-70	60A1	SYM	Enter symbols S_1, S_2, \dots, S_n to subclasses 1, 2, .. n respectively. If overprinting is desired, then the overprint symbols are entered on a second SYMBOLΔΔΔΔ lead card. Overprinted or non-overprinted symbols may be used in the same greymap, however, each symbol or symbol set for a subclass must be unique. The default SEDS symbols are shown below.																																												
				<table border="0"> <thead> <tr> <th>SEDS CLASS NAMES</th> <th>SYMBOL SETS</th> </tr> </thead> <tbody> <tr><td>-7.9</td><td>8/</td></tr> <tr><td>-7.5</td><td>7/</td></tr> <tr><td>-6.5</td><td>6/</td></tr> <tr><td>-5.5</td><td>5/</td></tr> <tr><td>-4.5</td><td>4/</td></tr> <tr><td>-3.5</td><td>3/</td></tr> <tr><td>-2.5</td><td>2/</td></tr> <tr><td>-1.5</td><td>1/</td></tr> <tr><td>-0.75</td><td>1=</td></tr> <tr><td>-0.25</td><td>0/</td></tr> <tr><td>+0.25</td><td>0+</td></tr> <tr><td>+0.75</td><td>1+</td></tr> <tr><td>+1.5</td><td>1Δ</td></tr> <tr><td>+2.5</td><td>2Δ</td></tr> <tr><td>+3.5</td><td>3Δ</td></tr> <tr><td>+4.5</td><td>4Δ</td></tr> <tr><td>+5.5</td><td>5Δ</td></tr> <tr><td>+6.5</td><td>6Δ</td></tr> <tr><td>+7.5</td><td>7Δ</td></tr> <tr><td>+7.9</td><td>8Δ</td></tr> <tr><td>WATER</td><td>ΔΔ</td></tr> </tbody> </table>	SEDS CLASS NAMES	SYMBOL SETS	-7.9	8/	-7.5	7/	-6.5	6/	-5.5	5/	-4.5	4/	-3.5	3/	-2.5	2/	-1.5	1/	-0.75	1=	-0.25	0/	+0.25	0+	+0.75	1+	+1.5	1Δ	+2.5	2Δ	+3.5	3Δ	+4.5	4Δ	+5.5	5Δ	+6.5	6Δ	+7.5	7Δ	+7.9	8Δ	WATER	ΔΔ
SEDS CLASS NAMES	SYMBOL SETS																																															
-7.9	8/																																															
-7.5	7/																																															
-6.5	6/																																															
-5.5	5/																																															
-4.5	4/																																															
-3.5	3/																																															
-2.5	2/																																															
-1.5	1/																																															
-0.75	1=																																															
-0.25	0/																																															
+0.25	0+																																															
+0.75	1+																																															
+1.5	1Δ																																															
+2.5	2Δ																																															
+3.5	3Δ																																															
+4.5	4Δ																																															
+5.5	5Δ																																															
+6.5	6Δ																																															
+7.5	7Δ																																															
+7.9	8Δ																																															
WATER	ΔΔ																																															

COMMENTS Optional card. If only one SYMBOLΔΔΔΔ card is read the second set will be blanks. The default symbols are the SEDS symbols.

APPENDIX D
PROGRAM LISTING
AND
PROGRAM PRINTER OUTPUT

PROGRAM LISTING

OR. FILZOR/M0945A,FILZOR,FILZOR
 UBIVAC 1108 FORTRAN V EXEC II LEVEL 25A -(EXECR LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:23:34

08 APR 76

11:23:34

MAIN PROGRAM

STORAGE USED: CODE(1) 007573; DATA(0) 016627; BLANK COMMON(2) 000000

COMMON BLOCKS:

0003 FIZ 000434
 0004 DRUM 000046
 0005 INFIZ1 000015
 0006 CONSAM 015560
 0007 UNIFZ2 003216
 0010 GLOBAL 000076
 0011 DISPL 001370

EXTERNAL REFERENCES (BLOCK, NAME)

0012 RESET
 0013 RREAD
 0014 FSDSFL
 0015 TDATE
 0016 NDCODS
 0017 FIZNRF
 0020 WRTFLO
 0021 FIZSON
 0022 RINIT
 0023 NDCODS
 0024 FIZTMO
 0025 FIZFLT
 0026 FIZURT
 0027 RWRITE
 0030 FIZRNP
 0031 CLOCK
 0032 CMERR
 0033 NREMS
 0034 NROUS
 0035 BIOZE
 0036 NROUS
 0037 BIOIS
 0040 NMEFS
 0041 NSTOP

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0000	014175	10F	0000	014326	100F	0001	000474	1000G	0001	000530	1020G	0001	000020	1040L
0000	014330	105F	0001	000144	1050L	0001	000631	1056G	0000	014334	106F	0001	000247	1060L
0001	000637	1063G	0001	000645	1070G	0001	000272	1070L	0001	000651	1074G	0001	000307	1075L
0001	000327	1080L	0000	014347	110F	0001	000674	1106G	0001	000703	1114G	0001	000742	1137G
0001	000747	11430	0001	000775	1156G	0001	001004	1164G	0000	014361	120F	0001	001043	1207G
0001	001050	1213G	0001	001107	1234G	0001	001132	1247G	0001	001155	1260G	0001	001171	1270G
0001	001224	1307G	0001	001263	1330G	0001	001271	1335G	0001	001275	1341G	0001	001337	1363G
0001	001345	1370G	0001	001351	1374G	0000	014363	140F	0001	001304	1444G	0001	001310	1450G
0001	001516	1455G	0001	001530	1464G	0001	001534	1472G	0001	001561	1505G	0001	001604	1517G
0001	001627	1531G	0001	001652	1543G	0001	001730	1574G	0000	014365	160F	0001	002142	1663G
0000	014413	170F	0000	014466	180F	0000	014476	181F	0000	014507	182F	0000	014520	183F
0000	014543	190F	0000	014551	191F	0000	014556	200F	0001	000402	2000L	0001	000422	2006L
0001	000451	2020L	0001	000501	2040L	0001	000535	2060L	0001	002653	2067G	0001	000621	2070L
0001	000656	2080L	0001	000721	2085L	0001	000726	2090L	0000	014572	210F	0001	000757	2100L

ORIGINAL PAGE IS
 OF POOR QUALITY

0001	001022	2105L	0000	014601	211F	0001	001027	2110L	0001	002721	2112G	0001	001060	2120L					
0001	002735	2123G	0001	002760	2135G	0001	001114	2140L	0001	003040	2154G	0001	003043	2160G					
0001	001134	2140L	0001	001176	2180L	0000	014606	220F	0001	001231	2200L	0001	003106	2202G					
0001	003113	2210G	0001	003126	2217G	0001	001305	2220L	0001	003134	2225G	0001	001361	2225L					
0001	001471	2230L	0001	003150	2236G	0001	001543	2240L	0001	003161	2245G	0001	001566	2260L					
0001	001671	2280L	0000	014616	230F	0001	001634	2300L	0001	001657	2310L	0001	003305	2315G					
0001	001671	2315L	0001	003331	2331G	0001	003342	2336G	0001	001735	2350L	0001	001745	2360L					
0001	003423	2365G	0001	001764	2365L	0001	002001	2370L	0001	002027	2375L	0001	003443	2377G					
0000	014657	240F	0001	003571	2435G	0001	002045	2440L	0001	003600	2441G	0001	002065	2445L					
0001	003642	2442G	0001	003645	2445G	0001	002116	2445L	0000	014704	250F	0001	002160	2500L					
0001	003730	2514G	0001	003753	2526G	0001	002353	2540L	0001	004002	2543G	0001	004017	2554G					
0001	002403	2560L	0000	014730	260F	0001	004317	2647G	0000	014772	265F	0001	004331	2655G					
0000	015000	270F	0001	004530	2722G	0001	004645	2736G	0001	005062	2764G	0000	015043	280F					
0000	014177	30F	0000	015077	300F	0001	005177	3000G	0000	015117	302F	0001	002453	3020L					
0001	005336	3021G	0000	015123	303F	0001	002515	3040L	0001	002566	3060L	0001	005474	3065G					
0001	002637	3080L	0000	015133	310F	0001	005377	3117G	0001	005614	3127G	0000	015216	315F					
0000	015237	317F	0000	015240	320F	0001	006011	3204G	0000	015303	322F	000T	006101	3224G					
0001	006131	3242G	0000	015325	350F	0001	006311	3301G	0000	015332	331F	0000	015343	332F					
0000	015371	333F	0001	006415	3332G	0000	015431	335F	0000	015464	340F	0000	015563	342F					
0000	015570	345F	0001	007047	3450G	0001	007114	3446G	0000	015573	350F	0001	007135	3500G					
0001	007221	3526G	0001	007226	3534G	0000	015574	360F	0001	007421	3640G	0000	015642	363F					
0001	007470	3672G	0001	007513	3705G	0000	015632	400F	0001	002660	4000L	0001	002727	4040L					
0000	015645	405F	0001	002744	4050L	0001	003064	4080L	0000	015674	410F	0001	003066	4100L					
0000	015713	415F	0000	015757	416F	0001	003121	4160L	0000	016022	420F	0001	003171	4300L					
0001	003356	4360L	0001	003363	4380L	0000	016035	440F	0001	003435	4400L	0001	003460	4450L					
0001	003464	4460L	0001	003555	4480L	0001	003610	4500L	0001	003660	4530L	0001	003662	4535L					
0001	003664	4536L	0001	003764	4540L	0001	004006	4540L	0000	016057	460F	0001	004024	4600L					
0000	016063	465F	0000	014236	50F	0000	016066	500F	0000	016077	510F	0000	016101	515F					
0000	016102	520F	0001	000031	546G	0000	016121	580F	0000	016130	585F	0000	016252	60F					
0001	004444	6040L	0001	000237	632G	0001	000262	647E	0001	000302	663G	0001	000322	676G					
0000	014266	70F	0001	004771	7080L	0001	000360	716G	0001	003325	7200L	0001	003355	7205L					
0001	005401	7215L	0001	005503	7220L	0001	005064	7250L	0001	000372	727G	0001	005677	7280L					
0001	005764	7300L	0001	006012	7320L	0001	006144	7348L	0001	006166	7370L	0001	006242	7380L					
0001	006267	7420L	0001	000417	743G	0001	006316	7460L	0001	006335	7480L	0001	006366	7485L					
0001	006403	7490L	0000	014301	75F	0001	006426	7500L	0001	000431	751G	0001	006641	7511L					
0001	006466	7515L	0001	006454	7520L	0001	000444	762G	0000	014313	80F	0001	006653	8000L					
/0001	006636	8040L	0001	006722	8050L	0001	007153	8070L	0001	007171	8075L	0001	007231	8120L					
0001	007275	8300L	0001	007310	8980L	0000	014324	90F	0000	016134	900F	0001	007350	9000L					
0001	007405	9010L	0001	007426	9030L	0001	007437	9040L	0001	007452	9045L	0000	016156	905F					
0001	007453	9050L	0001	007476	9060L	0001	007524	9070L	0001	007534	9075L	0001	007542	9080L					
0001	007551	9090L	0000	016202	910F	0000	016221	915F	0000	016243	920F	0000	016277	925F					
0000	016355	930F	0000	016410	940F	0000	016442	950F	0000	016471	990F	0011	001342	ACR0P					
0004	I	000042	ALADD	0011	001351	ANALYS	0011	001343	ADTHER	0005	I	000000	ARF121	0000	I	014140	ARNAWE		
0000	I	000242	ARRAY	0003	I	000375	ASAN	0000	I	014136	ASAMF	0003	I	000374	ASAM1	0007	I	003215	ATFILE
0011	001344	ATOTAL	0000	I	014173	BADREC	0000	I	014146	BFILE	0000	I	014044	BLANK	0011	000570	BRCONB		
0011	000571	BNFEAT	0010	000035	BNFILE	0011	000567	BNFLG	0010	000056	BRKEY	0010	000056	BRKEY	0006	I	015554	BUFFT	
0011	001354	CAMS	0000	I	014077	CARD1	0000	I	014100	CARDZ	0011	000000	CATFLG	0011	000001	CATRAM	0011	000001	CATRAM
0011	I	000572	CDATE	0000	I	014115	CHECK	0000	I	014123	CR	0011	I	000076	CLSRAM	0011	000640	CLSSUB	
0003	I	000206	CODE	0004	I	000001	COLADD	0004	I	000013	COLINC	0003	I	000356	COMMENT	0011	R	001075	CON
0000	I	014072	CRDCT	0011	001341	CROP	0011	001366	CRPKEY	0007	I	000021	CTFILE	0000	I	014107	CTRECD		
0000	I	014032	CTSED	0003	000400	CTSEDS	0000	I	014045	DASH	0010	000053	DATAPE	0003	I	000354	DATE		
0010	000071	DATFIL	0000	I	014147	DDRMYY	0003	I	000370	DEBUGF	0000	I	014026	DESCOD	0011	I	001356	DESKEY	
0003	I	008431	DESMO1	0003	I	000432	DESMO2	0011	I	001340	DESOTM	0000	I	014022	DESSYM	0011	I	001337	DESUM1
0003	I	000315	DFTCOD	0003	I	000320	DFTPT	0003	I	000316	DFTSYM	0000	I	014144	DRMED	0000	I	014141	DRMST
0010	000067	DRMWB	0004	I	000031	DRMURT	0010	000066	DRURAD	0005	I	000001	DSPYDT	0005	I	000003	DSPYTR		
0011	I	000000	DURDIS	0010	I	000000	DURGLD	0000	I	014033	DURRY	0003	I	000433	DUPWD	0000	I	014040	DUPSVM
0000	I	014172	ELATIN	0000	I	014163	ELINE	0011	000564	ERPTRS	0000	I	014126	ENDFLD	0003	I	000401	EOFHIT	
0010	000062	ERIPTP	0010	000063	ERPKEY	0000	I	014075	ERRCOD	0000	I	014073	ERROR	0000	I	014117	ERRSED		
0000	I	014163	ERRTST	0003	I	000377	ESAM	0000	I	014137	ESAMF	0003	I	000376	ESAR1	0011	001175	FETVC2	
0011	I	000575	FIELDZ	0011	I	000600	FIELDS	0000	I	014034	FILE	0000	I	014106	FILFIL	0000	I	014105	FILHNP
0011	I	001334	FILTER	0010	I	000075	FIZFIL	0010	000074	FIZKEY	0005	I	000000	FIZNAM	0010	000073	FIZHNP		
0005	000004	FLDESC	0011	I	001171	FLOKEY	0011	I	000574	FLOSVZ	0011	I	000577	FLOSV3	0004	I	000032	FOR	
0000	I	014145	FYF	0010	I	000000	HEAD	0003	I	000327	MED1	0003	I	000341	MED2	0011	001001	HIGH	

ORIGINAL PAGE IS
OF POOR QUALITY

0010	000057	MISFIL	0010	000060	MISKEY	0006	011634	MOLARR	0007	003203	MOLECH	0004	000027	MOLLIN
0000	014142	MOLNAP	0007	000032	MOLTST	0004	000030	MOLWRD	0000	014101	I	0004	000000	IBADDR
0000	000000	ICARD	0000	014062	IDBUG	0000	014060	IDCOD	0000	014067	IDCOM	0000	014066	IDDAT
0000	014061	IDDC	0000	014074	IDDES	0000	014052	IDDS	0000	014063	IDFIL	0000	014071	IDFIT
0000	014121	IDFTPT	0000	014064	IDHED1	0000	014065	IDHED2	0000	014056	IDHOL	0000	014055	IDISL
0000	014054	IDLIN	0000	014050	IDOP	0000	014070	IDSEDS	0000	014057	IDSXP	0000	014051	IDSYN
0000	014053	IDZOM	0000	014112	IMR	0005	000010	ILINED	0005	000011	ILINIC	0005	000007	ILINST
0000	014113	IMIN	0005	000006	INLINE	0005	000005	INPTS	0006	001750	IR	0000	000106	IRSTAT
0000	000112	INWRDS	0005	000013	ISAMED	0005	000014	ISAMIC	0005	000012	ISAMST	0000	014114	ISEC
0000	014161	ISKWR	0007	003173	ISLDCN	0007	000022	ISLSTT	0000	014132	I1	0000	014133	I2
0000	014111	I3	0000	014102	J	0000	014122	JDFPT	0006	015556	JSTAT	0000	014123	J1
0000	014130	J2	0000	014131	J3	0000	014132	J4	0000	014104	K	0011	001237	KATNO
0000	014157	KEYFIL	0003	000371	LINMIT	0003	000325	LINLIN	0003	000326	LINSAN	0000	014154	LINE
0000	014160	LINEXT	0000	014042	LININC	0003	000373	LWPSAN	0007	000000	RAP	0011	001335	RAPFMT
0010	000052	RAPTAP	0010	000064	RAPUNT	0000	014047	RAXCRD	0000	014135	RAXSAR	0011	001367	RFILE
0011	001333	ROCAT	0011	001236	ROCLS2	0004	000025	ROCOL	0004	000026	ROCOMR	0011	001174	ROFET2
0010	000045	ROFILE	0011	001172	ROFLD2	0011	001173	ROFLD3	0011	000554	ROMAP	0011	001233	ROSUB2
0011	000556	ROSUB3	0011	001234	ROTRFD	0000	014174	ROWRMS	0000	014143	RQDS	0007	000010	OTFILE
0003	000353	OTLINE	0003	000312	OTUNT	0000	014150	OUTFAS	0000	014076	OUTLWP	0010	000070	PAGSIZ
0011	000557	PCFDRY	0011	000602	PCTD3	0007	000043	PERCNT	0011	000566	PLTKEY	0003	000372	PROSAN
0000	014120	PT	0006	001750	PTSYN	0004	000043	REALIN	0000	014134	SANDFT	0000	014041	SARINC
0000	014156	SAMPLE	0006	000000	SANSON	0000	014043	SANSTR	0010	000054	SAVTAP	0000	014033	SEKKEY
0003	000402	SEDNAM	0011	001345	SITE	0000	014110	SKFILE	0003	000313	SKPLIN	0003	000314	SKPSAR
0000	014151	SKP3	0000	014162	SLINE	0010	000072	STAFIL	0011	000563	STATKY	0011	000364	SUBCAT
0011	000173	SUBNAR	0011	000270	SUBNO	0000	014155	SV	0003	000000	SYN	0006	011634	SYNCHK
0011	000677	SYMNTX	0004	000044	SYRPAG	0003	000311	TAPOUT	0007	003214	TMOLECH	0011	000603	THRES
0011	000562	THRSKY	0011	000565	THRSVA	0007	000020	TINFIL	0004	000045	TIMREA	0007	003213	TIBLCH
0000	014116	TOPPAG	0007	000042	TOTPCT	0000	014124	TOTTRN	0000	014123	TOTYST	0011	001235	TOTYST2
0011	000555	TOTYST3	0010	000061	TRFORM	0011	000561	TRNKEY	0003	000427	TRNRD	0000	014036	TRNSYN
0011	000560	TSTKEY	0003	000430	TSTNO	0000	014037	TSTSYN	0000	014133	TSTNRD	0004	015555	UNTFAS
0000	014030	UNTFAT	0000	014031	UNTFAZ	0000	014046	UNTFIZ	0011	000576	VRTX2	0011	000601	VRTX3
0007	D 001073	ZCNI	0000	D 000170	ZDUNY	0000	D 014170	ZHOL	0003	D 000321	ZOHLIN	0003	D 000323	ZORSAN
0007	D 001113	ZPC	0000	D 000116	ZSED	0000	D 014166	ZTCHM	0000	D 014164	ZTOTPC			

```

00100 1* C PROGRAM FILZOR
00100 2* C
00100 3* C
00100 4* C
00100 5* C
00101 6* C
00103 7* IMPLICIT INTEGER (A-Y)
00104 7* INCLUDE FIZCOM.LIST
00105 7* IMPLICIT DOUBLE PRECISION (Z)
00105 7* REAL COM,THRES *NEW
00105 7* C
00105 7* C PARAMETERS
00105 7* C
00106 7* PARAMETER PAGCOL = 10, PAGSYN = 110, PARSAR = 1000, PARLIN = 1000
00107 7* PARAMETER PAGWRD = 19, PSCLAD = 61, ISUBCL = PSCLAD - 1
00110 7* PARAMETER PSEDS = 21, POUTLN = PSCLAD*6, PSYN = 2*POUTLN,
00110 7* PSAMPS = PARSAR * 5, PARFIL = 8 *NEW
00110 7* C
00110 7* C COMMON BLOCKS
00110 7* C
00111 7* COMMON /FIZ/ SYR(POUTLN,2), CODE(POUTLN), TAPOUT, OTUNT, SKPLIN,
00111 7* SKPSAN, DFTCOD, DFTSYN(2), DFTPT, ZORLIN, ZORSAN,
00111 7* LINLIN, LINSAN, MED1(10), MED2(10), OTLINE, DATE(2),
00111 7* CORENT(10),DEBUGF,LINMIT,PROSAN,LRPSAN,
00111 7* ASAM1, ASAM, ESAM1, ESAN, CTSEDS, EOFMIT,
00111 7* SEDNAM(PSEDS),TRNRD,TSTNO,DESNO1,DESNO2,DUPNO
00112 7* COMMON /DRUN/ IBADDR, COLADD(PAGCOL), COLINC(PAGCOL), BOCOL,

```

ORIGINAL PAGE IS
OF POOR QUALITY



```

00112 7*      *      NOCOMR, MOLLIN, HOLWRD, DRHWRT,
00112 7*      *      FOR(8), ALADD, REALIN, SYMPAG, TIRREA
00113 7*      *      COMMON /INFIZ1/ FIZNAM, DSPYOT(2), DSPYTH, FLOESC, INPTS, IRLINE,
00113 7*      *      ILINST, IRLINED, IRLINIC, ISANST, ISAMED, ISAMIC
00114 7*      *      COMMON /CONSAM/ SAMSON(PARSAN), IR(PARSAN,4), HOLARR(PARSAN,2),
00114 7*      *      SUFFY, UNTFAS, JSTAT(2)
00115 7*      *      COMMON/UNTFIZ/ NAP(PARFIL), OTFILE(PARFIL), TINFIL,
00115 7*      *      CTFIL, ISLTST(PARFIL), HOLTST(PARFIL), TOTPCT,
00115 7*      *      PERCPT(POUTLB,PARFIL), ZCMI(PARFIL),
00115 7*      *      ZPC(POUTLB,PARFIL), ISLDCM(PARFIL),
00115 7*      *      NOLECH(PARFIL), TISLCH, THOLCH
00115 7*      *      ,ATFILE
00116 7*      *
00116 7*      *      EBD
00117 8*      *      INCLUDE COMBK6.LIST
00120 8*      *      COMMON/GLOBAL/HEAD(42),NAPTAP,DATAPE,SAVTAP,BNFIE,BNKEY,
00120 8*      *      MISFIL,MISKEY,TRFORM,ERIPPT,ERPKEY,NAPUNT,NOFILE,
00120 8*      *      DRUNAD,DRRMD5,PAGSIZ,DATAFIL,STAFIL
00120 8*      *      ,FIZUNT,FIZKEY,FIZFIL
00121 8*      *      EBD
00122 9*      *      INCLUDE COMBK10.LIST
00123 9*      *      COMMON/DISPL/CATFLG,CATNAR(61),CLSNAR(61),SUBNAR(61),SUBNO(60),
00123 9*      *      SUBCAT(60),CLSSUB(60),BORAP,TOTVT3,NOSUB3,
00123 9*      *      PCFDKY,TSTKEY,TRKEY,TMRKY,STATKY,ERPTAS,TMRSA,
00123 9*      *      PLTKY,BNFLG,BNCOMB,BNFEAT,CDATE(2),
00123 9*      *      FLDV2,FLD2,VERTX2,FLOSV3,FLD3,VERTX3,PCTIDS,
00123 9*      *      THRS(60),SYNMX(66),HIGH(60),CON(60)
00123 9*      *      ,FLDKY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)
00123 9*      *      ,NOUB2,NOTRFD,TOTVT2,NOCLB2
00123 9*      *      ,KATNO(60),NOCAT,FILTER,NAPPRT
00123 9*      *      ,DESKY,DESURI,DESOTH,CROP ,ACROP,OTHER,ATOTAL
00123 9*      *      ,SITE(4),ANALYS(3),CARB(10),CRPKY
00123 9*      *      ,WFILE
00124 9*      *      EBD
00125 10*     *      DIMENSION ICARD(79),IRSTAT(4),IRWRDS(4)
00126 11*     *      PARAMETER PARDIS=760, PARGLO=62, PARARF=13
00127 12*     *      DIMENSION DUNDIS(PARDIS), DUNGLO(PARGLO), ARFIZ1(PARARF)
00130 13*     *      DIMENSION PTSYN(POUTLN), SYNCHK(POUTLN,2)
00131 14*     *      DIMENSION ZSED(PSEDS), ZDUNY(PSEDS)
00132 15*     *      PARAMETER PARARR = 6000
00133 16*     *      DIMENSION ARRAY (PARARR)
00134 17*     *      DIMENSION DESSYN(2,2),DESCOD(2)
00134 18*     *      C
00134 19*     *      C
00134 20*     *      C
00135 21*     *      EQUIVALENCE STATEMENTS
00136 22*     *      EQUIVALENCE (PTSYN(1),IR(1,1)), (HOLARR(1),SYNCHK(1,1))
00136 22*     *      EQUIVALENCE (DUNDIS(1),CATFLG), (DUNGLO(1),HEAD(1))
00137 23*     *      EQUIVALENCE (ARFIZ1(1),FIZNAM)
00137 24*     *      C
00137 25*     *      C
00137 26*     *      C
00140 27*     *      DATA SEDNAR( 1)/6H-7.9 /, SEDNAR( 2)/6H-7.5 /,
00140 28*     *      SEDNAR( 3)/6H-6.5 /, SEDNAR( 4)/6H-5.5 /,
00140 29*     *      SEDNAR( 5)/6H-4.5 /, SEDNAR( 6)/6H-3.5 /,
00140 30*     *      SEDNAR( 7)/6H-2.5 /, SEDNAR( 8)/6H-1.5 /,
00140 31*     *      SEDNAR( 9)/6H-0.75 /, SEDNAR(10)/6H-0.25 /,
00140 32*     *      SEDNAR(11)/6H-0.25 /, SEDNAR(12)/6H-0.75 /,
00140 33*     *      SEDNAR(13)/6H-1.5 /, SEDNAR(14)/6H-2.5 /,
00140 34*     *      SEDNAR(15)/6H-3.5 /, SEDNAR(16)/6H-4.5 /,
00140 35*     *      SEDNAR(17)/6H-5.5 /, SEDNAR(18)/6H-6.5 /,
00140 36*     *      SEDNAR(19)/6H-7.5 /, SEDNAR(20)/6H-7.9 /,
00140 37*     *      SEDNAR(21)/6HWATER /
00166 38*     *      DATA TAPOUT/1/, OTUNT/19/, SKPLIN/20/, SKPSAM/0/, UNTF1/20/,
00166 39*     *      UNTF2/21/, HOLTST/PARFIL*5/, ISLTST/PARFIL*5/, DFTCOD/128/,

```

ORIGINAL PAGE IS
OF POOR QUALITY



00166 40*
 00166 41*
 00166 42*
 00166 43*
 00166 44*
 00217 45*
 00221 46*
 00225 47*
 00231 48*
 00234 49*
 00236 50*
 00236 51*
 00236 52*
 00236 53*
 00236 54*
 00236 55*
 00263 56*
 00263 57*
 00263 58*
 00263 59*
 00263 60*
 00263 61*
 00263 62*
 00263 63*
 00263 64*
 00263 65*
 00263 66*
 00312 67*
 00312 68*
 00312 69*
 00325 70*
 00325 71*
 00325 72*
 00341 73*
 00343 74*
 00343 75*
 00352 76*
 00352 77*
 00352 78*
 00352 79*
 00352 80*
 00402 81*
 00402 82*
 00404 83*
 00404 84*
 00406 85*
 00412 86*
 00414 87*
 00414 88*
 00414 89*
 00414 90*
 00417 91*
 00420 92*
 00420 93*
 00420 94*
 00420 95*
 00421 96*
 00421 97*
 00422 98*
 00422 99*
 00423 100*
 00423 101*
 00424 102*

```

* DFTSYN(1)/1M0/, DFTSYN(2)/1M /, MAP/PARFIL=1/,
* DTFILE/PARFIL=1/,TINFIL/1/,CTFILE/0/,
* ZOMLEB/1.2500/,ZOMSAN/1.2500/,LIRLIN/550/,LIRSAN/625/,
* DEBUG/0/, LIMHIT/0/, CTSED/21/, SEDKEY/1/
* FILE/1/
DATA DURNV/0/
DATA TRNSYR/'*', TSYSYR/'*', DUPSYR/'S'/
DATA SARINC/1/, LININC/1/, SARSTR/1/
DATA BLANK/6M /, DASH/6M-----/
DATA SYN/PSYR=6M /
DATA SYN(1,1)/1M0/, SYN(1,2)/1M1/, SYN(2,1)/1M7/, SYN(2,2)/1M1/,
* SYN(3,1)/1M6/, SYN(3,2)/1M1/, SYN(4,1)/1M5/, SYN(4,2)/1M1/,
* SYN(5,1)/1M4/, SYN(5,2)/1M1/, SYN(6,1)/1M3/, SYN(6,2)/1M1/,
* SYN(7,1)/1M2/, SYN(7,2)/1M1/, SYN(8,1)/1M1/, SYN(8,2)/1M1/,
* SYN(9,1)/1M1/, SYN(9,2)/1M=/,
* SYN(10,1)/1M0/, SYN(10,2)/1M1/
DATA SYN(11,1)/1M0/, SYN(11,2)/1M=/,
* SYN(12,1)/1M1/, SYN(12,2)/1M=/,
* SYN(13,1)/1M1/, SYN(13,2)/1M /,
* SYN(14,1)/1M2/, SYN(14,2)/1M /,
* SYN(15,1)/1M3/, SYN(15,2)/1M /,
* SYN(16,1)/1M4/, SYN(16,2)/1M /,
* SYN(17,1)/1M5/, SYN(17,2)/1M /,
* SYN(18,1)/1M6/, SYN(18,2)/1M /,
* SYN(19,1)/1M7/, SYN(19,2)/1M /,
* SYN(20,1)/1M8/, SYN(20,2)/1M /,
* SYN(21,1)/1M /, SYN(21,2)/1M /,
DATA CODE(1)/ 1/, CODE(2)/ 8/, CODE(3)/ 24/, CODE(4)/ 40/,
* CODE(5)/ 56/, CODE(6)/ 72/, CODE(7)/ 88/, CODE(8)/104/,
* CODE(9)/116/, CODE(10)/124/
DATA CODE(11)/132/, CODE(12)/140/, CODE(13)/152/, CODE(14)/168/,
* CODE(15)/184/, CODE(16)/200/, CODE(17)/216/, CODE(18)/232/,
* CODE(19)/248/, CODE(20)/254/, CODE(21)/176/
DATA UNTF12/11/
DATA DESSYR(1,1)/1M7/, DESSYR(2,1)/1M /, DESCOD(1)/255/,
* DESSYR(1,2)/1M0/, DESSYR(2,2)/1M /, DESCOD(2)/2 /
DATA MAXCRD/27/, IDOP/0/, IDSYN/0/, IDDS/0/, IDZOR/0/, IDLIN/0/,
* IDISL/0/, IDHOL/0/, IDSKP/0/, IDCOD/0/, IDDC/0/, IDBUG/0/,
* IDFIL/0/, IDHED1/0/, IDHED2/0/, IDDAT/0/, IDCOM/0/,
* IDSEDS/0/, IDFIT/0/, CRDCT/0/, ERROR/0/, IDDES/0/
* ,CTFILE/0/
DATA MED1/6M . 6M . 6M LVB, 6H00 B., 6M JOMB.,
* 6H0B SPA, 6HCE CER, 6HTER . 6M . 6M /
DATA MED2/6M . 6M . 6M . 6M H, 6H0USTOR,
* 6M, TEXA, 6MS . 6M . 6M /
DATA CORENT/10*6M /, ERRCOD/99999/, EOFHIT/0/
DATA OUTLBP /POUTLW/
DATA DATE(1)/6M /, DATE(2)/6M /

```

FORMAT STATEMENTS

```

10 FORMAT (A6,A4,70A1)
30 FORMAT (1M1,38X,'CHECK LEAD CARDS FOR PROGRAM FILZDM',/,
* 39X,35(' '),/,/,/,T50,9('0'),10('1'),10('2'),10('3'),
* 10('4'),10('5'),10('6'),10('7'),'8',/,T50,8('1234567890')
* ,/)
50 FORMAT (2X,'...ERROR... CARD IDENTIFICATION',T50,A6,A4,70A1,
* /,T50,A6,A4,/)
60 FORMAT (2X,'...ERROR... DUPLICATE CARD ID.',T50,A6,A4,70A1,/,
* T50,A6,A4,/)
70 FORMAT (2X,'...ERROR... ERROR IN CODE(S)',T50,A6,A4,70A1,/,
* T60,70A1,/)
75 FORMAT (2X,'...ERROR... BAD REREAD FORMAT',T50,A6,A4,70A1,/,/)

```




00425	103.	88	FORMAT (2X, '***GOOD *** INPUT CARD', T50, A6, A4, 70A1, /, /)
00426	104.	90	FORMAT (10X, 2F5, 0)
00427	105.	100	FORMAT (10X, 10I5)
00430	106.	105	FORMAT (10X, I5, 4X, A3, 2X, A5)
00431	107.	106	FORMAT (2X, '***ERROR*** ERROR IN CODE(S)', T50, A6, A4, 70A1, /, /
00431	108.	.	T60, 3A5, /)
00432	109.	110	FORMAT (2X, '***VARIABLES CHECKED LATER***', T50, A6, A4, 70A1, /, /)
00433	110.	120	FORMAT (10X, 10I5)
00434	111.	140	FORMAT (10X, 10A6)
00435	112.	160	FORMAT (/, T8, '***ERROR*** FILZON', /, T20, 'TOO MANY LEAD CARDS '
00435	113.	.	'HAVE BEEN READ.', /, T20, 'NO MORE LEAD CARDS WILL BE '
00435	114.	.	'READ.', /)
00436	115.	170	FORMAT (/, /, /, T40, 'SUMMARY OF INPUT TAPE', /, T40, 21('-'), /, /, /, /
00436	116.	.	T25, 'INFIZ INPUT ON LOGICAL UNIT ', /, T2, ' ', /, /, /
00436	117.	.	T25, 'INFIZ FILE ', /, T2, ' GENERATED ON ', /, T2A6, ' AT '
00436	118.	.	A2, ' ', /, T2, ' ', /, T2, ' '
00436	119.	.	' WILL BE PROCESSED.', /, /, /, /
00436	120.	.	T25, 'SUMMARY OF INPUT SCANNER DATA', /)
00437	121.	180	FORMAT (T29, 'NO THRESHOLDING HAS BEEN APPLIED.', /)
00440	122.	181	FORMAT (T29, 'CMI SQUARE THRESHOLDING HAS BEEN APPLIED.', /)
00441	123.	182	FORMAT (T29, 'EMPIRICAL THRESHOLDING HAS BEEN APPLIED.', /)
00442	124.	183	FORMAT (T29, 'USER-INPUT THRESHOLDING HAS BEEN APPLIED.', /)
00443	125.	191	FORMAT (T29, 'PROCESSOR DISPLAY APPLIED SPATIAL FILTERING.', /)
00444	126.	190	FORMAT (T29, 'PROCESSOR DISPLAY DID NOT APPLY SPATIAL FILTERING.', /)
00444	127.	.	' '
00445	128.	200	FORMAT (T29, 'DATA CLASSIFIED FOR NAPTAP FILE ', /, T11, ' ON ', /, T6, A6, /
00445	129.	.	' ', /)
00446	130.	210	FORMAT (T29, 'NUMBER OF INPUT LINES = ', /, T15, ' ')
00447	131.	211	FORMAT (1H, T15, '***ERROR***')
00450	132.	220	FORMAT (/, T29, 'NUMBER OF INPUT SAMPLES = ', /, T15, ' ')
00451	133.	230	FORMAT (/, T29, 'NUMBER OF CLASSES = ', /, T12, ' ', /, /, /)
00451	134.	.	T29, 'NUMBER OF SUBCLASSES = ', /, T12, ' ', /, /, /)
00451	135.	.	T29, 'NUMBER OF TRAINING FIELDS = ', /, T12, ' ', /, /, /)
00451	136.	.	/, T29, 'NUMBER OF TEST OR DESIGNATE FIELDS = ', /, T13, ' ')
00452	137.	240	FORMAT (/, T8, '***ERROR*** FILZON', /, T20, 'NUMBER OF SAMPLES '
00452	138.	.	' TO SKIP = ', /, T15, /, T20, 'NUMBER OF SAMPLES ON INPUT TAPE = '
00452	139.	.	' /5)
00453	140.	250	FORMAT (/, T8, '***ERROR*** FILZON', /, T20, 'NUMBER OF LINES TO '
00453	141.	.	' SKIP = ', /, T15, /, T20, 'NUMBER OF LINES ON INPUT TAPE = '
00453	142.	.	' /5)
00454	143.	260	FORMAT (/, T8, '***ERROR*** FILZON', /, T20, 'ERROR CODES ARE '
00454	144.	.	' LESS THAN 1 OR GREATER THAN 255 OR ', /, T20, 'THE CODES '
00454	145.	.	' ARE NOT UNIQUE.', /, /, T25, 'SUBCLASS', /, T35, 'CODE', /
00454	146.	.	/, T25, 8('-'), /, T35, '----', /)
00455	147.	265	FORMAT (T20, 12, ' ', /, T26, A6, T35, 14, T40, A6)
00456	148.	270	FORMAT (/, T8, '***ERROR*** FILZON', /, T20, 'ZOOM LINE FACTOR = '
00456	149.	.	' F10.3, /, T20, 'THE SKIP LINE VARIABLE IS IN ERROR OR '
00456	150.	.	' /, T20, 'THE ZOOM LEAD CARD IS IN ERROR OR '
00456	151.	.	' /, T20, 'THE LIMITS LINE VARIABLE IS IN ERROR.')
00457	152.	280	FORMAT (/, T8, '***ERROR*** FILZON', /, T20, 'ZOOM SAMPLE FACTOR = '
00457	153.	.	' F10.3, /, T20, 'THE SKIP SAMPLE VARIABLE IS IN ERROR OR '
00457	154.	.	' /, T20, 'THE LIMITS SAMPLE VARIABLE IS IN ERROR.')
00460	155.	300	FORMAT (/, /, /, T40, 'PROCESSING FOR FILZON', /, T40, 21('-'), /, /, /
00460	156.	.	T25, 'TAPE OUTPUT ON LOGICAL UNIT ', /, T13)
00461	157.	302	FORMAT (1H, T56, '--- UNIVERSAL FORMAT.', /)
00462	158.	303	FORMAT (1H, T56, '--- LARSYS II FORMAT.', /)
00463	159.	310	FORMAT (T25, 'ON INPUT TAPE BEFORE PROCESSING BEGINS ', /, T14,
00463	160.	.	' LINES WILL BE SKIPPED.', /, /, T25, 'ON EACH LINE '
00463	161.	.	' OF THE INPUT TAPE ', /, T14, ' SAMPLES WILL BE SKIPPED.', /
00463	162.	.	' /, /, T25, 'ZOOM LINE FACTOR = ', /, T10, 3, /, /, T25,
00463	163.	.	' ZOOM SAMPLE FACTOR = ', /, T10, 3, /, /, T25,
00463	164.	.	' LINE LIMIT = ', /, T10, /, /, T25, 'SAMPLE LIMIT = ', /, T10, /)
00464	165.	315	FORMAT (T25, 'INPUT LINES ', /, T15, ' TO ', /, T15, ' WILL NOT BE PROCESSED '

ORIGINAL PAGE IS
OF POOR QUALITY



```

00464 166* * "DUE TO THE ZOOM LINE FACTOR.",/)
00465 167* 317 FORMAT (T25,"OUTPUT LINES ",I5," TO ",I5," WILL BE DEFAULT ",
00466 168* * "VALUES DUE TO THE ZOOM LINE FACTOR.",/)
00466 169* 320 FORMAT (T25,"ON EACH INPUT LINE, SAMPLES",I5," TO ",I5,
00466 170* * " WILL NOT BE PROCESSED DUE TO THE ZOOM SAMPLE FACTOR.",
00466 171* * "/)
00467 172* 322 FORMAT (T25,"OUTPUT SAMPLE ",I5," TO ",I5," WILL BE DEFAULT ",
00467 173* * "VALUES DUE TO THE ZOOM SAMPLE FACTOR.",/)
00470 174* 330 FORMAT (T25,"ISLAND TEST = ",I10)
00471 175* 331 FORMAT (/,T25,"UNCLASSIFIED PIXEL TEST (HOLE) = ",I10)
00472 176* 332 FORMAT (/,T25,"DEFAULT CODE FOR UNCLASSIFIED PIXELS = ",T66,I3,/,/
00472 177* * ,T25,"DEFAULT SYMBOL FOR UNCLASSIFIED PIXELS = ",T66,A1,
00472 178* * /,1M*,T66,A1)
00473 179* 333 FORMAT (/,T25,"DESIGNATED CODES:",T50,"UNIDENTIFIED = ",T65,I3,
00473 180* * T75,"OTHER = ",T85,I4,/,/,T25,"DESIGNATED SYMBOLS:",
00473 181* * T50,"UNIDENTIFIED = ",T65,A1,T75,"OTHER = ",T85,A1,/,1M*,
00473 182* * T65,A1,T85,A1)
00474 183* 335 FORMAT (/,T8,"***ERROR*** FILZOM",/,T20,"A GROUP CARD SHOULD ",
00474 184* * "HAVE BEEN USED IN SCLASSIFY TO GROUP THE TWO OR MORE",
00474 185* * "DIFFERENT SUBCLASSES IN A CLASS.",/)
00475 186* 340 FORMAT (/,T8,"***ERROR*** FILZOM",/,T20,"SOME OF THE INPUT SED
00475 187* * "CLASS NAMES ARE NOT VALID.",/,T20,"RERUN THE JOB USING",
00475 188* * "A NOSEDB LEAD CARD, A SYMBOL LEAD CARD, A CODE LEAD
00475 189* * "CARD.",/,T20,"A DEFAULT S LEAD CARD, A DEFAULT C
00475 190* * "LEAD CARD AND A DESIGNATE ",
00475 191* * "LEAD CARD.",/,T20,"THE CLASS NAMES IN ERROR ARE MARKED.",
00475 192* * /,/,T50,"CLASS NAMES",/,T50,"-----",/,)
00476 193* 342 FORMAT (T50,I2," ",2X,A6,5X,A6)
00477 194* 345 FORMAT (T49,A6,1X,A6)
00500 195* 350 FORMAT (I6)
00501 196* 360 FORMAT (/,T8,"***ERROR*** FILZOM",/,T20,"SOME OF THE SYMBOLS ",
00501 197* * "ARE NOT UNIQUE.",/,T20,"CORRECT THE SYMBOL LEAD CARDS ",
00501 198* * "AND/OR DEFAULT S LEAD CARD.",
00501 199* * /,/,T50,"SUBCLASS NAME",T70,"SYMBOL",/,T50,I6(" "),
00501 200* * T70,6(" "),/)
00502 201* 365 FORMAT (T49,I2," ",2X,A6,T73,A1,/,1M*,T73,A1,T76,A6)
00503 202* 400 FORMAT (/,/,/,T40,"SUMMARY OF OUTPUT FOR FILZOM",/,
00503 203* * T40,28(" "),/,/)
00504 204* 405 FORMAT (T25,"OUTPUT FILE",I2," ON UNIT",I3)
00505 205* 410 FORMAT (T25,"NUMBER OF RECORDS THAT HAVE BEEN READ FROM ",
00505 206* * "LOGICAL UNIT",I3," = ",I5,".")
00506 207* 415 FORMAT (T25,"NUMBER OF UNCLASSIFIED PIXELS THAT HAVE BEEN ",
00506 208* * "RECLASSIFIED DUE TO THE HOLE TEST (",I2," ) FOR FILTER",
00506 209* * "I2." = ",I6,".",/,T25,"TOTAL NUMBER OF HOLE PIXELS THAT",
00506 210* * "HAVE BEEN RECLASSIFIED = ",I6,".")
00507 211* 416 FORMAT (T25,"NUMBER OF ISLAND PIXELS THAT HAVE BEEN ",
00507 212* * "RECLASSIFIED DUE TO THE ISLAND(",I2," ) FOR FILTER",
00507 213* * "I2." = ",I6,".",/,T25,"TOTAL NUMBER OF ISLAND PIXELS THAT",
00507 214* * "HAVE BEEN RECLASSIFIED = ",I6,".")
00510 215* 420 FORMAT (T25,"TOTAL NUMBER OF ZOOMED OUTPUT POINTS = ",I6,".",/,/)
00511 216* 440 FORMAT (T55,"PERCENT TABLE",/,T55,I3(" "),/,/,T57,
00511 217* * "SUBCLASS NAME",T57,"TOTAL POINTS",T74,"PERCENTAGES",/)
00512 218* 460 FORMAT (T41,A6,T59,I6,T78,F5.2)
00513 219* 465 FORMAT (10X,I5,15X,I5)
00514 220* 500 FORMAT (/,/,T20,"TIME FOR FILZOM = ",F6.3," MINUTES.")
00515 221* 510 FORMAT (1X,F3.0)
00516 222* 515 FORMAT (F6.0)
00517 223* 520 FORMAT (/,/,T25,"FOR FILTER ",I1," THE AVERAGE CRT FOR ALL ",
00517 224* * "LAND PIXELS IS ",F7.3,".")
00520 225* 580 FORMAT (1M1,10X,"DATA IN ARRAY ",A6,/,/,/)
00521 226* 585 FORMAT (10(4X,I3," ",1X,I3))
00522 227* 900 FORMAT (/,T8,"***ERROR*** FILZOM",/,T20,"RECORD ",I4,
00522 228* * "ON FILE",I3," ON LOGICAL UNIT",I3," HAS AN EOF.")

```

```

00523 229* 905 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,'LOGICAL UNIT',I3,
00523 230*   * HAS A '
00523 231*   * 'PARITY ERROR ON FILE ',I2, ' ON RECORD',I4,'.')
00524 232* 910 FORMAT (/,T8,'THE PREVIOUSLY MARKED ERROR(S) CAUSED THE PROGRAM',
00524 233*   * TO ERROR EXIT.',/,I1,/,I1,/,I1)
00523 234* 915 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,'ERROR IN SUBROUTINE ',
00523 235*   * 'FILZON',/,T20,'CALL A PROGRAMMER.',/)
00526 236* 920 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,'INFIZ ON LOGICAL UNIT',
00526 237*   * I3, ' IS NOT IN THE CORRECT FORMAT.',/,T20,'A LISTING',
00526 238*   * ' OF THE PHYSICAL RECORD FOLLOWS:',/,I1, 5(5X,013))
00527 239* 925 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,'FLDKEY IN COMMON ',
00527 240*   * 'BLOCK DISPL = ',I3, ' ',/,T20,'FLDKEY MUST EQUAL ',
00527 241*   * 'WHICH INDICATES GROUND TRUTH FIELDS ARE ASSOCIATED',
00527 242*   * ' TO SUBCLASSES.',/,T20,'GIVE ALL GROUND TRUTH FIELDS ',
00527 243*   * 'SUBCLASS NAMES AND RERUN THOSE JOBS.(')
00530 244* 930 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,'FILE NUMBER ON ',
00530 245*   * 'INFIZ TAPE = ',I2, ' ',/,T20,'THE DESIRED FILE ',
00530 246*   * 'NUMBER = ',I2, ' ',/,T20,'CALL A PROGRAMMER.(')
00531 247* 940 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,'ON RECORD',I5, ' OF ',
00531 248*   * 'FILE ',I2, ' ',I5, ' WORDS WERE READ.',/,T20,'THERE ARE ',
00531 249*   * I5, ' PIXELS ON A LINE.(')
00532 250* 950 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,
00532 251*   * 'ERROR CONDITION ON ATTEMPT TO POSITION INFIZ OVER',
00532 252*   * I3, ' FILES.',/,T20,'PSUSFL STATUS CODE = ',I4)
00533 253* 999 FORMAT (/,T8,'***ERROR*** FILZON',/,T20,'TAPE UNIT ',I2,
00533 254*   * ' HAS A WRITE ERROR FOR RECORD ',I4, ' FILE ',I1, ' ')
00533 255*
00533 256* C
00533 257* C.....
00533 258* C READ AND PRINT LEAD CARDS
00533 259* C.....
00533 260* C
00534 260* CALL RESET
00535 261* REMIND UNTFIZ
00536 262* REMIND OTURT
00537 263* WRITE (6,30)
00541 264*
00542 265* 1040 CONTINUE
00552 266* READ (5,10,END=2360) CARD1, CARD2, ICARD
00553 267* CRDCT = CRDCT + 1
00555 268* IF (CRDCT.GT. MAXCRD) GO TO 2350
00557 269* IF (CARD2.NE. BLANK) GO TO 1050
00561 270* IF (CARD1.EQ. 6HOUTPUT) GO TO 1080
00563 271* IF (CARD1.EQ. 6HZOOM) GO TO 2040
00565 272* IF (CARD1.EQ. 6HLIMIT) GO TO 2060
00567 273* IF (CARD1.EQ. 6HISLAND) GO TO 2080
00571 274* IF (CARD1.EQ. 6HOLE) GO TO 2100
00573 275* IF (CARD1.EQ. 6HSKIP) GO TO 2120
00575 276* IF (CARD1.EQ. 6HCODE) GO TO 2140
00577 277* IF (CARD1.EQ. 6HDEBUG) GO TO 2200
00601 278* IF (CARD1.EQ. 6HFILE) GO TO 2220
00603 279* IF (CARD1.EQ. 6HFILTER) GO TO 2225
00605 280* IF (CARD1.EQ. 6HHED1) GO TO 2240
00607 281* IF (CARD1.EQ. 6HHED2) GO TO 2260
00611 282* IF (CARD1.EQ. 6HDATE) GO TO 2280
00613 283* IF (CARD1.EQ. 6HROSEDS) GO TO 2310
00615 284* 1050 IF (CARD1.EQ. 6HCONHEW.AND. CARD2.EQ. 4MT) GO TO 2300
00617 285* IF (CARD1.EQ. 6HDEFAUL.AND. CARD2.EQ. 4MT S) GO TO 2020
00621 286* IF (CARD1.EQ. 6HDEFAUL.AND. CARD2.EQ. 4MT C) GO TO 2180
00623 287* IF (CARD1.EQ. 6HDESIGN.AND. CARD2.EQ. 4MATE) GO TO 2315
00625 288* ERROR = ERROR + 1
00626 289* WRITE (6,50) CARD1, CARD2, ICARD, DASH, DASH      0 BAD IDENTIFICATION
00640 290* GO TO 1040
00641 291* 1060 CONTINUE

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

00642 292*      ERROR = ERROR + 1
00643 293*      WRITE (6,60) CARD1, CARD2, ICARD, DASH, DASH      @ DUPLICATE ID.
00655 294*      GO TO 1040
00656 295*      1070 CONTINUE
00657 296*      WRITE (6,80) CARD1,CARD2,ICARD      @ GOOD LEAD CARD
00667 297*      GO TO 1040
00670 298*      1075 CONTINUE
00671 299*      ERROR = ERROR + 1
00672 300*      WRITE (6,75) CARD1, CARD2, ICARD      @ FORMAT REREAD
00702 301*      GO TO 1040
C*****
C LEAD CARD OUTPUT
C*****
00703 305*      1080 IDOP = IDOP + 1
00704 306*      IF (IDOP .GT. 1) GO TO 1060
00706 307*      IF (ICARD(5) .EQ. 1ML) TAPOUT = 2
00710 308*      IF (ICARD(5) .EQ. 1ML) GO TO 1070
00712 309*      WRITE (6,70) CARD1, CARD2, ICARD, BLANK, BLANK, BLANK, BLANK,
00712 310*      *DASH, (BLANK, 1 = 6, 70)
00733 311*      ERROR = ERROR + 1
00734 312*      GO TO 1040
C*****
C LEAD CARD SYMBOL
C*****
00735 316*      2000 IDSYN = IDSYN + 1
00736 317*      IF (IDSYN .GT. 2) GO TO 1060
00740 318*      IF (IDSYN .EQ. 2) GO TO 2006
00742 319*      DO 2005 I = 1, ISUBCL
00745 320*      SYN(I,2) = BLANK
00746 321*      2005 CONTINUE
00750 322*      2006 DO 2010 I = 1, ISUBCL
00753 323*      SYN(I,IDSYN) = ICARD(I)
00754 324*      2010 CONTINUE
00756 325*      WRITE (6,110) CARD1,CARD2, ICARD
00766 326*      GO TO 1040
C*****
C LEAD CARD DEFAULT B
C*****
00767 330*      2020 IDDS = IDDS + 1
00770 331*      IF (IDDS .GT. 1) GO TO 1060
00772 332*      DFTSYN(1) = ICARD(5)
00773 333*      DFTSYN(2) = ICARD(10)
00774 334*      WRITE (6,110) CARD1,CARD2, ICARD
01004 335*      GO TO 1040
C*****
C LEAD CARD ZOOM
C*****
01004 338*      2040 IDZON = IDZON + 1
01006 340*      IF (IDZON .GT. 1) GO TO 1060
01010 341*      READ (30,90, ERR = 1075) ZONLIN, ZONPAR
01014 342*      WRITE (6,110) CARD1, CARD2, (ICARD(I), I=1,70)
01024 343*      GO TO 1040
C*****
C LEAD CARD LIMIT
C*****
01024 346*      2060 IDLIM = IDLIM + 1
01026 348*      IF (IDLIM .GT. 1) GO TO 1060
01030 349*      READ (30, 100, ERR=1075) LIMLIN, LINSAN
01034 350*      ARRAY(1) = BLANK
01035 351*      ARRAY(2) = BLANK
01036 352*      IF (LINSAN .LE. 0) ARRAY(1) = DASH
01040 353*      IF (LIMLIN .LE. 0) ARRAY(2) = DASH
01042 354*      IF (LINSAN .GT. PARSAN) ARRAY(1) = DASH

```

```

01044 355*      IF (LIRLIN .GT. PARLIN) ARRAY(2) = DASH
01046 356*      IF (ARRAY(1) .EQ. DASH) GO TO 2070
01050 357*      IF (ARRAY(2) .EQ. BLANK) GO TO 1070
01052 358*      2070 WRITE (6,70) CARD1,CARD2,ICARD,ARRAY(1),J=1,6,
01052 359*           *BLANK,(ARRAY(2),J=1,4),(BLANK,J=1,60)
01100 360*      GO TO 1040
01100 361*      C*****
01100 362*      C LEAD CARD ISLAND
01100 363*      C*****
01101 364*      2080 IDISL = IDISL + 1
01102 365*      IF (IDISL .GT. 1) GO TO 1060
01104 366*      READ (30, 120, ERR = 1075) ISLTST
01112 367*      ERRST = 0
01113 368*      DO 2090 I = 1,PARFIL
01116 369*           ARRAY(I) = BLANK
01117 370*           IF (ISLTST(I) .EQ. 0) GO TO 2090
01121 371*           IF (IABS(ISLTST(I)) .LT. 3) GO TO 2085
01123 372*           IF (IABS(ISLTST(I)) .LE. 8) GO TO 2090
01125 373*      2085 ERRST = ERRST + 1
01126 374*           ARRAY(I) = DASH
01127 375*      2090 CONTINUE
01131 376*      IF (ERRST .EQ. 0) GO TO 1070
01133 377*      WRITE (6,106) CARD1,CARD2, ICARD, (ARRAY(I), I = 1,PARFIL)
01147 378*      ERROR = ERROR + 1
01150 379*      GO TO 1040
01150 380*      C*****
01150 381*      C LEAD CARD MOLD
01150 382*      C*****
01151 383*      2100 IDMOL = IDMOL + 1
01152 384*      IF (IDMOL .GT. 1) GO TO 1060
01154 385*      READ (30, 120, ERR = 1075) MOLTST
01162 386*      ERRST = 0
01163 387*      DO 2110 I = 1,PARFIL
01166 388*           ARRAY(I) = BLANK
01167 389*           IF (MOLTST(I) .EQ. 0) GO TO 2110
01171 390*           IF (IABS(MOLTST(I)) .LT. 3) GO TO 2105
01173 391*           IF (IABS(MOLTST(I)) .LE. 8) GO TO 2110
01175 392*      2105 ERRST = ERRST + 1
01176 393*           ARRAY(I) = DASH
01177 394*      2110 CONTINUE
01201 395*      IF (ERRST .EQ. 0) GO TO 1070
01203 396*      WRITE (6,106) CARD1, CARD2, ICARD,(ARRAY(I),I=1,PARFIL)
01217 397*      ERROR = ERROR +1
01220 398*      GO TO 1040
01220 399*      C*****
01220 400*      C LEAD CARD SKIP
01220 401*      C*****
01221 402*      2120 IDSKP = IDSKP + 1
01222 403*      IF (IDSKP .GT. 1) GO TO 1060
01224 404*      READ (30,100, ERR = 1075) SKPLIN, SKPSAN
01230 405*      WRITE (6,110) CARD1, CARD2, ICARD
01240 406*      GO TO 1040
01240 407*      C*****
01240 408*      C LEAD CARD CODE
01240 409*      C*****
01241 410*      2140 IDCOD = IDCOD + 1
01242 411*      IF (IDCOD .GT. 6) GO TO 1060
01244 412*      IF (IDCOD .GT. 1) GO TO 2160
01246 413*      DO 2150 I = 1, ISUBCL
01251 414*           CODE(I) = 0
01252 415*      2150 CONTINUE
01254 416*      2160 J = (IDCOD - 1) * 10 + 1
01255 417*           K = J * 10 - 1

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

01266 418. READ (30,120,ERR=1075) (CODE(I), I = J, K)
01264 419. WRITE (6,110) CARD1, CARD2, ICARD
01274 420. GO TO 1040
01274 421. C.....
01274 422. C LEAD CARD DEFAULT C
01274 423. C.....
01275 424. 2180 IDDC = IDDC + 1
01276 425. IF (IDDC .GT. 1) GO TO 1060
01300 426. READ (30,100, ERR=1075) DFTCOD
01303 427. WRITE (6,110) CARD1, CARD2, ICARD
01313 428. GO TO 1040
01313 429. C.....
01313 430. C LEAD CARD DEBUG
01313 431. C.....
01314 432. 2200 IDBUG = IDBUG + 1
01315 433. IF (IDBUG .GT. 1) GO TO 1060
01317 434. READ (30,100, ERR=1075) DEBUGF
01322 435. IF (DEBUGF .EQ. 1) GO TO 1070
01324 436. WRITE (6,70) CARD1, CARD2, ICARD, BLANK, (DASH, I = 1, 4),
01324 437. *(BLANK, I = 6,70)
01345 438. ERROR = ERROR + 1
01346 439. GO TO 1040
01346 440. C.....
01346 441. C LEAD CARD FILE
01346 442. C.....
01347 443. 2220 IDFIL = IDFIL + 1
01350 444. IF (IDFIL .GT. 1) GO TO 1060
01352 445. READ (30,100, ERR=1075) FILE
01355 446. IF (FILE .GT. 0) GO TO 1070
01357 447. WRITE (6,70) CARD1, CARD2, ICARD, BLANK, (DASH, I = 1, 4),
01357 448. *(BLANK, I=6,70)
01400 449. ERROR = ERROR + 1
01401 450. GO TO 1040
01401 451. C.....
01401 452. C LEAD CARD FILTER
01401 453. C.....
01402 454. 2225 IDFIT = IDFIT + 1
01403 455. IF (IDFIT .GT. 1) GO TO 1060
01405 456. READ (30,105, ERR=1075) TINFIL, FILMAP, FILFIL
01412 457. ARRAY (1) = BLANK
01413 458. IF (TINFIL .GT. PARFIL) ARRAY(1) = DASH
01415 459. IF (TINFIL .LT. 1) ARRAY(1) = DASH
01417 460. ARRAY (2) = DASH
01420 461. IF (FILMAP .EQ. BLANK) ARRAY(2) = BLANK
01422 462. IF (FILMAP .EQ. 6HMAP ) ARRAY(2) = BLANK
01424 463. ARRAY(3) = DASH
01425 464. IF (FILFIL .EQ. BLANK) ARRAY(3) = BLANK
01427 465. IF (FILFIL .EQ. 6HFILES ) ARRAY(3) = BLANK
01431 466. IF (ARRAY(1) .EQ. DASH) GO TO 2230
01433 467. IF (ARRAY(2) .EQ. DASH) GO TO 2230
01435 468. IF (ARRAY(3) .EQ. BLANK) GO TO 1070
01437 469. 2230 ERROR = ERROR + 1
01440 470. WRITE (6,70) CARD1, CARD2, ICARD, (BLANK, I=1,4), ARRAY(1),
01440 471. *(BLANK, I = 1,4), ARRAY(2), ARRAY(2), ARRAY(2), BLANK, BLANK,
01440 472. *(ARRAY(3), I = 1,5), (BLANK, I = 20,70)
01476 473. IDFIL = 999
01477 474. GO TO 1040
01477 475. C.....
01477 476. C LEAD CARD HED1
01477 477. C.....
01500 478. 2240 IDMED1 = IDMED1 + 1
01501 479. IF (IDMED1 .GT. 1) GO TO 1060
01503 480. READ (30,140, ERR=1075) HED1

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

01511 481* GO TO 1070
01511 482* C.....
01511 483* C LEAD CARD WEDZ
01511 484* C.....
01512 485* 2260 IDMEDZ = IDMEDZ + 1
01513 486* IF (IDMEDZ .GT. 1) GO TO 1070
01515 487* READ (30,140,ERR=1075) HED.
01523 488* GO TO 1070
01523 489* C.....
01523 490* C LEAD CARD DATE
01523 491* C.....
01524 492* 2280 IDDAT = IDDAT + 1
01525 493* IF (IDDAT .GT. 1) GO TO 1060
01527 494* READ (30,140,ERR=1075) DATE
01535 495* GO TO 1070
01535 496* C.....
01535 497* C LEAD CARD CORMENT
01535 498* C.....
01536 499* 2300 IDCOR = IDCOR + 1
01537 500* IF (IDCOR .GT. 1) GO TO 1060
01541 501* READ (30,140,ERR=1075) CONENT
01547 502* GO TO 1070
01547 503* C.....
01547 504* C LEAD CARD SEDS
01547 505* C.....
01550 506* 2310 IDSEDS = IDSEDS + 1
01551 507* IF (IDSEDS .GT. 1) GO TO 1060
01553 508* SEDKEY = 2
01554 509* GO TO 1070
01554 510* C.....
01554 511* C LEAD CARD DESIGNATE
01554 512* C.....
01555 513* 2315 IDDES = IDDES + 1
01556 514* IF (IDDES .GT. 1) GO TO 1060
01560 515* READ (30,463,ERR=1075) DESCOD(1),DESCOD(2)
01564 516* DESSYN(1,1) = ICARD(10)
01565 517* DESSYN(2,1) = ICARD(15)
01566 518* DESSYN(1,2) = ICARD(30)
01567 519* DESSYN(2,2) = ICARD(35)
01570 520* WRITE (6,110) CARD1, CARD2, ICARD
01600 521* GO TO 1040
01600 522* C.....
01600 523* C TOO MANY LEAD CARDS HAVE BEEN READ
01600 524* C.....
01600 525* C
01601 526* 2350 CONTINUE
01602 527* ERROR = ERROR + 1
01603 528* WRITE (6,160)
01603 529* C
01603 530* C.....
01603 531* C 1. VALIDATE FIRST RECORD ON INFIZ TAPE.
01603 532* C 2. SKIP THE NECESSARY FILES IF FILE IS GREATER THAN 1. THEN
01603 533* C READ RECORD 1 OF THE FILE THAT IS GOING TO BE PROCESSED
01603 534* C
01603 535* C.....
01603 536* C
01605 537* 2360 CONTINUE
01606 538* IF (FILE .LT. 1) GO TO 0090
01610 539* CTRECD = 1
01611 540* SKFILE = 0
01612 541* I3 = 1
01613 542* IRSTAT(I3) = 0
01614 543* IF (FILE .GT. 1) GO TO 2375

```

```

01616 344* 2365 CALL MREAD (UNTFIZ, 1, ARFIZ1, PARARF, IRSTAT(13), IRNRDS(13))
01617 345* 2370 IF (IRSTAT(13) .EQ. 1) GO TO 2370
01621 346* IF (IRSTAT(13) .GT. 1) GO TO 9000
01623 347* IF (FIZNAM .NE. 4MINFIZ ) GO TO 9010
01625 348* IF (FILE .EQ. 1) GO TO 2440
01627 349* IF (SKFILE .GT. 0) GO TO 2440
01631 350* 2375 CONTINUE
01632 351* SKFILE = FILE - 1
01633 352* CALL F80SFL (UNTFIZ,SKFILE,IRSTAT(13))
01636 353* IF (IRSTAT(13) .NE. 0) GO TO 9030
01636 354* GO TO 2365
01636 355* C
01636 356* C.....
01636 357* C READ RECORD FROM INFIZ AND STORE DATA IN COMMON BLOCK GLOBAL
01636 358* C.....
01636 359* C
01637 360* 2440 CONTINUE
01640 361* CTRECD = CTRECD + 1
01641 362* CALL MREAD (UNTFIZ, 1, DUNGLO, PARGLO, IRSTAT(13), IRNRDS(13))
01642 363* 2445 IF (IRSTAT(13) .EQ. 1) GO TO 2445
01644 364* IF (IRSTAT(13) .GT. 1) GO TO 9000
01644 365* C
01644 366* C.....
01644 367* C READ RECORD FROM INFIZ AND STORE DATA IN COMMON BLOCK DISPL
01644 368* C.....
01644 369* C
01646 370* 2460 CONTINUE
01647 371* CTRECD = CTRECD + 1
01650 372* CALL MREAD (UNTFIZ, 1, DUNDIS, PARDIS, IRSTAT(13), IRNRDS(13))
01651 373* 2465 IF (IRSTAT(13) .EQ. 1) GO TO 2465
01653 374* IF (IRSTAT(13) .GT. 1) GO TO 9000
01655 375* IF (FIZFIL .NE. FILE) GO TO 9070
01657 376* IF (FLDKEY .NE. 1) GO TO 9080
01657 377* C
01657 378* C.....
01657 379* C PREPARE HEADING FOR PRINTER (OUTPUT AND PRINT SUMMARY OF INFIZ
01657 380* C.....
01657 381* C
01661 382* 2470 CONTINUE
01662 383* DO 2480 I = 1,10
01665 384* HEAD(I+2) = HED1(I)
01666 385* HEAD(I+19) = HED2(I)
01667 386* HEAD(I+31) = CORENT(I)
01670 387* 2480 CONTINUE
01672 388* IF (ICDDAT .GT. 0) GO TO 2500
01674 389* CALL TDATE (DATE)
01675 390* 2500 HEAD(15) = DATE(1)
01676 391* HEAD(16) = DATE(2)
01677 392* WRITE (6,HEAD)
01701 393* FLD(0,12,INR) = FLD(0,12,DSPTYM)
01702 394* FLD(0,12,ININ) = FLD(12,12,DSPTYM)
01703 395* FLD(0,12,ISEC) = FLD(24,12,DSPTYM)
01704 396* WRITE (6,170) UNTFIZ, FILE, DSPYOT(1), DSPYOT(2), INR, ININ, ISEC
01715 397* IF (THRSKY .EQ. 0) WRITE (6,180)
01720 398* IF (THRSKY .EQ. 1) WRITE (6,181)
01723 399* IF (THRSKY .EQ. 2) WRITE (6,182)
01726 400* IF (THRSKY .EQ. 3) WRITE (6,183)
01731 401* IF (FILTER .EQ. 0) WRITE (6,190)
01734 402* IF (FILTER .EQ. 1) WRITE (6,191)
01737 403* WRITE (6,200) BFILE,CDATE(1),CDATE(2)
01740 404* CHECK = ((ILINED-ILINST)*ILINIC)/ILINIC
01745 405* WRITE (6,210) INLINE
01750 406* IF (CHECK .EQ. INLINE) GO TO 2540

```



```

01752 607*      ERROR = ERROR + 1
01753 608*      WRITE (6,211)
01755 609*      2540 CHECK = ((ISAMED-ISANST)*ISARIC)/ISARIC
01756 610*      WRITE (6,220) INPTS
01761 611*      IF (CHECK .EQ. INPTS) GO TO 2560
01763 612*      ERROR = ERROR + 1
01764 613*      WRITE (6,211)
01766 614*      2560 WRITE (6,230) NOCLS2, NOSUB2, NOFLD2, NOFLD3
01766 615*      C
01766 616*      C.....
01766 617*      C          VALIDATE LEAD CARDS THAT WERE NOT VALIDATED WHEN READ.
01766 618*      C          PERFORM THE NECESSARY CALCULATIONS PERTAINING TO THE
01766 619*      C          LEAD CARDS.
01766 620*      C.....
01766 621*      C
01774 622*      3060 CONTINUE
01774 623*      C.....
01774 624*      C          LEAD CARD SKIP
01774 625*      C.....
01775 626*      TOPPAG = 0
01776 627*      IF (SKPSAM .LT. INPTS .AND. SKPSAM .GE. 0) GO TO 3020
02000 628*      ERROR = ERROR + 1
02001 629*      TOPPAG = 1
02002 630*      WRITE (6,HEAD)
02004 631*      WRITE (6,240) SKPSAM, INPTS
02010 632*      3020 IF (SKPLIN .LT. INLINE .AND. SKPSAM .GE. 0) GO TO 3040
02012 633*      IF (TOPPAG .LT. 1) WRITE (6,HEAD)
02013 634*      TOPPAG = 1
02016 635*      ERROR = ERROR + 1
02017 636*      WRITE (2,250) SKPLIN, INLINE
02017 637*      C.....
02017 638*      C          CHECK ZOOM FACTORS
02017 639*      C.....
02023 640*      3040 CONTINUE
02023 641*      C
02023 642*      C          CALCULATE ZOOM LINE FACTORS
02023 643*      C
02023 644*      C
02024 645*      IF (ZONLIN .GT. 0) GO TO 3060
02026 646*      ZONLIN = FLOAT(LINLIN) / FLOAT(INLINE-SKPLIN)
02027 647*      ZONLIN = ZONLIN + .00001
02030 648*      IF (ZONLIN .GT. 0) GO TO 3060
02032 649*      IF (TOPPAG .LT. 1) WRITE (6,HEAD)
02035 650*      TOPPAG = 1
02036 651*      ERROR = ERROR + 1
02037 652*      WRITE (6,270) ZONLIN
02037 653*      C.....
02037 654*      C          ZOOM SAMPLE FACTOR
02037 655*      C.....
02042 656*      3060 CONTINUE
02043 657*      IF (ZORSAM .GT. 0) GO TO 3080
02045 658*      ZORSAM = FLOAT(LINSAM) / FLOAT(INPTS-SKPSAM)
02046 659*      ZORSAM = ZORSAM + .00001
02047 660*      IF (ZORSAM .GT. 0) GO TO 3080
02051 661*      IF (TOPPAG .LT. 1) WRITE (6,HEAD)
02054 662*      TOPPAG = 1
02055 663*      ERROR = ERROR + 1
02056 664*      WRITE (6,280) ZORSAM
02056 665*      C.....
02056 666*      C          LEAD CARD SECS
02056 667*      C.....
02061 668*      3080 CONTINUE
02062 669*      IF (SECKEY .EQ. 1) GO TO 4040

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

02064 670*      IF (IDSYM .GT. 0) GO TO 4000
02066 671*      DO 3090 I = 1, NOSUB2
02071 672*          SYM(I,1) = SYMNTX(I)
02072 673*          SYM(I,2) = BLANK
02073 674*      3090 CONTINUE
02075 675*      4000 IF (IDDS .EQ. 0) DFTSYN(1) = SYMNTX(NOSUB2 + 1)
02077 676*          IF (IDDS .EQ. 0) DFTSYN(2) = BLANK
02101 677*          IF (IDDC .EQ. 0) DFTCOD = NOSUB2 + 1
02103 678*          IF (IDDES .EQ. 0) DESCOD(1) = DESUNI
02105 679*          IF (IDDES .EQ. 0) DESCOD(2) = DESOTM
02107 680*          IF (IDCOD .GT. 0) GO TO 4300
02111 681*          DO 4010 I = 1, NOSUB2
02114 682*              CODE(I) = I
02115 683*      4010 CONTINUE
02117 684*          GO TO 4300
02117 685*      C *****
02117 686*      C VALIDATE SEDS CLASS NAMES WITH THE CLASS NAMES ON THE INFIZ TAPE.
02117 687*      C IF THE INPUT CLASS NAMES AGREE WITH THE SEDS NAMES, THE DEFAULT SYM
02117 688*      C ARRAY AND THE DEFALUT CODE ARRAY WILL BE PUT IN THE SAME ORDER AS THE
02117 689*      C INPUT CLASS NAMES.
02117 690*      C *****
02120 691*      4040 CONTINUE
02121 692*          CHECK = 1
02122 693*          DO 4050 I = 2, NOSUB2
02125 694*              IF (SUBNAM(I) .EQ. SUBNAM(I-1)) GO TO 4050
02127 695*              CHECK = CHECK + 1
02130 696*      4050 CONTINUE
02132 697*          IF (CHECK .NE. NOCLS2) GO TO 9050
02134 698*          DO 4070 I = 1, PSEDS
02137 699*              IF (SEDNAM(I) .GT. 6M* ) DECODE (6,510,SEDNAM(I)) ZSED(I)
02143 700*              IF (SEDNAM(I) .LT. 6M* .AND. SEDNAM(I) .NE. 6HWATER )
02143 701*              * DECODE (6,515,SEDNAM(I)) ZSED(I)
02147 702*              IF (SEDNAM(I) .EQ. 6HWATER ) ZSED(I) = 0.0
02151 703*      4070 CONTINUE
02153 704*          DO 4100 I = 1, NOCLS2
02156 705*              ARRAY(I) = BLANK
02157 706*              DO 4060 J=1,CTSED
02162 707*                  IF (SEDNAM(J) .EQ. CLSNAM(I)) GO TO 4060
02164 708*      4060 CONTINUE
02166 709*              ARRAY(I) = 6HERROR
02167 710*              ERROR = ERROR + 1
02170 711*              ERRSED = ERRCOD
02171 712*              GO TO 4100
02172 713*      4080 PTSYM(I) = J
02173 714*      4100 CONTINUE
02175 715*          IF (ERRSED .EQ. ERRCOD) GO TO 9060
02175 716*      C *****
02175 717*      C ARRANGE SEDS CODES IN THE ORDER OF THE SED INPUT CLASS NAMES
02175 718*      C ON INFIZ TAPE
02175 719*      C *****
02177 720*          IF (IDCOD .GT. 0) GO TO 4160
02201 721*          DO 4120 I = 1,CTSED
02204 722*              MOLARR(I,1) = CODE(I)
02205 723*      4120 CONTINUE
02207 724*          DO 4140 I = 1, NOCLS2
02212 725*              PT = PTSYM(I)
02213 726*              CODE(I) = MOLARR(PT,1)
02214 727*      4140 CONTINUE
02216 728*          4160 DO 4165 I = 1, NOCLS2
02221 729*              ZOUHY(I) = ZSED(I)
02222 730*      4165 CONTINUE
02224 731*          DO 4170 I = 1,NOCLS2
02227 732*              PT = PTSYM(I)

```

```

02230 733*      ZSED(I) = ZDUMY(PT)
02231 734*
02233 735*      4170 CONTINUE
02233 736*      IF (IDSYM .GT. 0) GO TO 4300
02233 737*      C
02233 738*      *****
02233 739*      C ARRANGE SEDS SYMBOLS IN THE ORDER OF THE SED INPUT CLASSES.
02233 740*      C
02233 741*      *****
02233 742*      DO 4180 I = 1, CTSED
02240 743*          SYNCHK(I,1) = SYN(I,1)
02241 744*          SYNCHK(I,2) = SYN(I,2)
02242 745*
02244 746*      4180 CONTINUE
02244 747*      DO 4200 I = 1, NOCLS2
02247 748*          PT = PTSVR(I)
02250 749*          SYN(I,1) = SYNCHK(PT,1)
02251 750*          SYN(I,2) = SYNCHK(PT,2)
02252 751*
02252 752*      4200 CONTINUE
02252 753*      C*****
02252 754*      C PLACE DEFAULT SYMBOLS AND BORDER SYMBOLS IN ARRAY SYN
02252 755*      C PLACE DEFAULT CODE IN ARRAY CODE
02252 756*      C*****
02254 757*      4300 CONTINUE
02255 758*          DFTPT = NOSUB2-1
02256 759*          TRNO = DFTPT + 1
02257 760*          TSTNO = DFTPT + 2
02260 761*          DESNO1 = DFTPT + 4
02261 762*          SUBNAM(TRNO) = 6HTRNFLD
02262 763*          SUBNAM(TSTNO) = 6HTSTFLD
02263 764*          DUPNO = TSTNO + 1
02264 765*          SUBNAM(TSTNO+1) = 6HDUPFLD
02265 766*          DESNO2 = DFTPT + 5
02266 767*          CODE(DFTPT) = DFTCOD
02267 768*          SYN(DFTPT,1) = DFTSYN(1)
02270 769*          SYN(DFTPT,2) = DFTSYN(2)
02271 770*          SYN(DFTPT+1,1) = TRNSYM
02272 771*          SYN(DFTPT+1,2) = BLANK
02273 772*          SYN(DFTPT+2,1) = TSTSYM
02274 773*          SYN(DFTPT+2,2) = BLANK
02275 774*          SYN(DFTPT+3,1) = DUPSYM
02276 775*          SYN(DFTPT+4,1) = DESSYN(1,1)
02277 776*          SYN(DFTPT+4,2) = DESSYN(2,1)
02300 777*          SYN(DFTPT+5,1) = DESSYN(1,2)
02301 778*          SYN(DFTPT+5,2) = DESSYN(2,2)
02302 779*          CODE(DFTPT+4) = DESCOD(1)
02303 780*          CODE(DFTPT+5) = DESCOD(2)
02304 781*          SUBNAM(DESNO1) = 6MUNIDEN
02305 782*          SUBNAM(DESNO2) = 6MOTHER
02306 783*          CLSNAM(NOCLS2+1) = 6MHOLE
02307 784*          SUBNAM(DFTPT) = 6MHOLE
02307 785*      C*****
02307 786*      C LEAD CARDS SYMBOL AND DEFAULT S
02307 787*      C*****
02310 788*      4310 CONTINUE
02310 789*      C
02310 790*      *****
02310 791*      C REARRANGE THE TWO SYMBOLS FOR EASY VALIDATION
02310 792*      C
02310 793*      *****
02311 794*          IDFTPT = DFTPT + 5
02312 795*          IF (DESKEY .EQ. 0) IDFTPT = IDFTPT -2
02314 796*          DO 4320 I = 1, IDFTPT
02317 797*              SYNCHK(I,1) = BLANK
02320 798*              SYNCHK(I,2) = BLANK
02321 799*              FLD (0, 0, SYNCHK(I,1)) = FLD (0, 0, SYN(I,1))
02322 800*              FLD (0, 0, SYNCHK(I,2)) = FLD (0, 0, SYN(I,1))
02323 801*              FLD (0, 0, SYNCHK(I,1)) = FLD (0, 0, SYN(I,2))
02324 802*              FLD (0, 0, SYNCHK(I,2)) = FLD (0, 0, SYN(I,2))

```

```

02323 796* 4320 CONTINUE
02325 797* C *****
02325 798* C CHECK THE UNIQUENESS OF EACH PAIR OF SYMBOLS FOR A CODE.
02325 799* C *****
02327 800* JDFPT = IDFTPT - 1
02330 801* DO 4380 I = 1, JDFPT
02333 802* K = I + 1
02334 803* ARRAY(I) = BLANK
02335 804* DO 4340 J = K, IDFTPT
02340 805* IF(SYNCHK(I,1) .EQ. SYNCHK(J,1)) GO TO 4360
02342 806* IF(SYNCHK(I,1) .EQ. SYNCHK(J,2)) GO TO 4360
02344 807* 4340 CONTINUE
02346 808* GO TO 4380
02347 809* 4360 ARRAY(J) = 6HERROR
02350 810* IDSYN = ERRCOD
02351 811* 4380 CONTINUE
02353 812* IF (IDSYN .LT. ERRCOD) GO TO 4400
02355 813* IF (TOPPAG .LT. 1) WRITE (6,HEAD)
02360 814* ERROR = ERROR + 1
02361 815* WRITE (6,360)
02363 816* WRITE (6,365) (1,SUBNAM(I),SYN(I,1),SYN(I,2),ARRAY(I),
02363 817* *I=1,IDFTPT)
02363 818* C*****
02363 819* C CODE LEAD CARDS-----CODE, DEFAULT C, DESIG. C
02363 820* C*****
02375 821* 4400 CONTINUE
02375 822* C *****
02375 823* C ARE THE CODES VALID
02375 824* C *****
02376 825* DO 4460 I = 1, DFTPT
02401 826* ARRAY(I) = BLANK
02402 827* IF (CODE(I) .LT. 1) GO TO 4450
02404 828* IF (CODE(I) .GT. 255) GO TO 4450
02406 829* GO TO 4660
02407 830* 4450 ARRAY(I) = 6HERROR
02410 831* IDCOD = ERRCOD
02411 832* 4460 CONTINUE
02413 833* IF (DESKEY .EQ. 0) GO TO 4480
02415 834* ARRAY(DESNO1) = BLANK
02416 835* ARRAY(DESNO2) = BLANK
02417 836* IF (CODE(DESNO1) .LT. 1) ARRAY(DESNO1) = 6HERROR
02421 837* IF (CODE(DESNO1) .GT. 255) ARRAY(DESNO1) = 6HERROR
02423 838* IF (CODE(DESNO2) .LT. 1) ARRAY(DESNO2) = 6HERROR
02425 839* IF (CODE(DESNO2) .GT. 255) ARRAY(DESNO2) = 6HERROR
02427 840* IF (ARRAY(DESNO1) .EQ. 6HERROR) IDCOD = IDCOD + 1
02431 841* IF (ARRAY(DESNO2) .EQ. 6HERROR) IDCOD = IDCOD + 1
02431 842* C *****
02431 843* C CHECK TO INSURE THAT EACH CODE IS UNIQUE
02431 844* C *****
02433 845* 4480 CONTINUE
02434 846* DO 4520 I = 1, NOSUBZ
02437 847* K = I + 1
02440 848* DO 4520 J = K, DFTPT
02443 849* IF (CODE(I) .NE. CODE(J)) GO TO 4500
02445 850* ARRAY(J) = 6HERROR
02446 851* IDCOD = ERRCOD
02447 852* 4500 CONTINUE
02450 853* 4520 CONTINUE
02453 854* IF (DESKEY .EQ. 0) GO TO 4536
02455 855* IF (CODE(DESNO1) .EQ. CODE(DESNO2)) ARRAY(DESNO2) = 6HERROR
02457 856* IF (ARRAY(DESNO2) .EQ. 6HERROR) IDCOD = IDCOD + 1
02461 857* DO 4535 I = DESNO1,DESNO2
02464 858* DO 4530 J = 1,DFTPT

```

ORIGINAL PAGE IS
OF POOR QUALITY



```

02467 859.          IF (CODE(I) .NE. CODE(J)) GO TO 4530
02471 860.          ARRAY(I) = 0HERROR
02472 861.          IDCOD = IDCOD + 1
02473 862.          GO TO 4535
02474 863. 4530 CONTINUE
02476 864. 4535 CONTINUE
02500 865. 4550 CONTINUE
02501 866.          IF (IDCOD .LT. ENRCOD) GO TO 4540
02503 867.          IF (TOPPAG .LT. 1) WRITE (6,HEAD)
02506 868.          TOPPAG = 1
02507 869.          ERROR = ERROR + 1
02510 870.          WRITE (6,240)
02512 871.          WRITE (6,265) (I, SUBNAM(I), CODE(I), ARRAY(I), I = 1,DFTPT)
02523 872.          IF (DESKEY .EQ. 1)
02523 873.          *WRITE (6,265) (I,SUBNAM(I),CODE(I),ARRAY(I),I=DESN01,DESN02)
02523 874.          C
02523 875.          C CHECK FLAGS FOR PRINTING GRAYMAP FOR EACH TIME DATA IS FILTERED.
02523 876.          C CHECK FLAGS FOR OUTPUTTING A FILE FOR EACH TIME THE DATA IS FILTERED.
02523 877.          C *****
02535 878. 4540 CONTINUE
02536 879.          IF (IDFIL .GT. 1) GO TO 4600
02540 880.          IF (FILMAP .EQ. 0HMAP ) GO TO 4560
02542 881.          DO 4550 I = 1,TINFIL
02545 882.             MAP(I) = 0
02546 883. 4550 CONTINUE
02550 884.             MAP(TINFIL) = 1
02551 885. 4560 IF (FILFIL .EQ. 0HFILES ) GO TO 4600
02553 886.             DO 4580 I = 1,TINFIL
02556 887.                OTFILE(I) = 0
02557 888. 4580 CONTINUE
02561 889.                OTFILE(TINFIL) = 1
02562 890.                GO TO 4600
02562 891.          C
02562 892.          C *****
02562 893.          C IF NO ERRORS HAVE OCCURRED THE SUMMARY FOR LEAD CARDS AND THE
02562 894.          C CALCULATIONS MADE FROM THE LEADS CARDS ARE PRINTED.
02562 895.          C *****
02562 896.          C
02563 897. 4600 CONTINUE
02564 898.          IF (ERROR .GT. 0) GO TO 9090
02566 899.          WRITE (6,HEAD)
02570 900.          WRITE (6,300) OTUNT
02573 901.          IF (TAPOUT .EQ. 1) WRITE (6,302)
02576 902.          IF (TAPOUT .EQ. 2) WRITE (6,303)
02601 903.          WRITE (6,310) SKPLIN, SKPSAM, ZONLIN, ZONSAM, LIMLIN, LINSAM
02611 904.          CHECK = (FLOAT(LIMLIN)/ZONLIN) * SKPLIN
02612 905.          CK = CHECK * 1
02613 906.          IF (CHECK .LT. INLINE) WRITE (6,315) CK,INLINE
02620 907.          CHECK = (FLOAT(LINSAM)/ZONSAM) * SKPSAM
02621 908.          CK = CHECK * 1
02622 909.          IF (CHECK .LT. INPTS) WRITE (6,320) CK,INPTS
02627 910.          CHECK = FLOAT (INLINE - SKPLIN) * ZONLIN
02630 911.          CK = CHECK * 1
02631 912.          IF (CHECK .LT. LIMLIN) WRITE (6,317) CK,LIMLIN
02636 913.          CHECK = FLOAT (INPTS - SKPSAM) * ZONSAM
02637 914.          CK = CHECK * 1
02640 915.          IF (CHECK .LT. LINSAM) WRITE (6,322) CK, LINSAM
02645 916.          WRITE (6,330) (ISLTST(I), I = 1,TINFIL)
02653 917.          WRITE (6,331) (HOLTST(I), I = 1, TINFIL)
02661 918.          WRITE (6,332) CODE(DFTPT), SYM(DFTPT,1), SYM(DFTPT,2)
02666 919.          IF (DESKEY .EQ. 1) WRITE (6,333) DESCOD(1),DESCOD(2),
02666 920.          *DESSYN(1,1),DESSYN(1,2),DESSYN(2,1),DESSYN(2,2)
02666 921.          C

```

```

02666 922. C.....
02666 923. C
02666 924. C   THE RECTANGULAR VERTICES WHICH OUTLINE A FIELD ARE STORED AS FOLLOWS:
02666 925. C   LINE MIN., LINE MAX., SAMPLE MIN., SAMPLE MAX., AND THEN A POINTER
02666 926. C   WHICH POINTS TO THE ACTUAL VERTICES.
02666 927. C
02666 928. C   THE ACTUAL FIELD OUTLINE IS STORED AS FOLLOWS:
02666 929. C   SAMPLE, LINE ETC.
02666 930. C
02666 931. C   TRAINING FIELD DATA, TEST FIELDS, OR DESIGNATE FIELDS ARE
02666 932. C   ALL STORED THE SAME WAY.
02666 933. C
02666 934. C.....
02666 935. C
02666 936. C
02666 937. C
02666 938. C   CALCULATE ADDRESSES FOR TRAINING FIELDS AND TEST OR DESIGNATE FIELDS.
02666 939. C.....
02677 940.   FLDV2 = 1
02700 941.   FIELD2 = (4 * NOFLD2) + 1
02701 942.   VERTX2 = (4 * NOFLD2) + (5 * NOFLD2) + 1
02702 943.   TOTTRN = (4 * NOFLD2) + (5 * NOFLD2) + (2 * TOTVT2)
02703 944.   FLDV3 = TOTTRN + 1
02704 945.   FIELD3 = (4 * NOFLD3) + TOTTRN + 1
02705 946.   VERTX3 = (4 * NOFLD3) + (5 * NOFLD3) + TOTTRN + 1
02706 947.   TOTV3 = (4 * NOFLD3) + (5 * NOFLD3) + (2 * TOTVT3) + TOTTRN
02706 948. C.....
02706 949. C   HEAD TRAINING FIELDS
02706 950. C.....
02707 951.   6020 CONTINUE
02710 952.   CTRECD = CTRECD + 1
02711 953.   CALL XREAD (UNTFIZ, 1, ARRAY(FLDSV2), TOTTRN, IRSTAT(I3),
02711 954.   *IRVROS(I3))
02712 955.   6040 IF (IRSTAT(I3) .EQ. 1) GO TO 6040
02714 956.   IF (IRSTAT(I3) .GT. 1) GO TO 9000
02716 957.   IF (DEBUG .EQ. 1) CALL FIZRF (ARRAY(FLDSV2), ARRAY(VERTX2),
02716 958.   *NOFLD2, 1, CLSNAM, SUBNAM, CODE, SYN, DFTPT, OUTLBP, SYNRTX)
02716 959. C.....
02716 960. C   200H RECTANGULAR VERTICES FOR THE TRAINING FIELDS
02716 961. C.....
02720 962.   ENDFLD = VERTX2 - 1
02721 963.   DO 6080 J1 = FIELD2, ENDFLD, 5
02724 964.     J2 = J1 + 1
02725 965.     J3 = J1 + 2
02726 966.     J4 = J1 + 3
02727 967.     ARRAY(J1) = FLOAT ((ARRAY(J1) / ILINIC) - SKPLIN) * ZONLIN
02730 968.     ARRAY(J2) = FLOAT ((ARRAY(J2) / ILINIC) - SKPLIN) * ZONLIN
02731 969.     ARRAY(J3) = FLOAT ((ARRAY(J3) / ISANIC) - SKPSAM) * ZONSAM
02732 970.     ARRAY(J4) = FLOAT ((ARRAY(J4) / ISANIC) - SKPSAM) * ZONSAM
02733 971.   6080 CONTINUE
02733 972. C.....
02733 973. C   200H ACTUAL TRAINING FIELDS
02733 974. C.....
02735 975.   DO 7000 I = VERTX2, TOTTRN, 2
02740 976.     J = I + 1
02741 977.     ARRAY(I) = FLOAT ((ARRAY(I) / ISANIC) - SKPSAM) * ZONSAM
02742 978.     ARRAY(J) = FLOAT ((ARRAY(J) / ILINIC) - SKPLIN) * ZONLIN
02743 979.   7000 CONTINUE
02745 980.   CALL FIZRF (ARRAY(FLDSV2), ARRAY(VERTX2), NOFLD2, 3, CLSNAM,
02745 981.   *SUBNAM, CODE, SYN, DFTPT, OUTLBP, SYNRTX)
02745 982. C.....
02745 983. C   READ TEST FIELDS OR DESIGNATE FIELDS
02745 984. C.....

```

```

02746 985. 7060 CONTINUE
02747 986. IF (TSTKEY .EQ. 0 .AND. DESKEY .EQ. 0) GO TO 7200
02751 987. TSTWRD = TOTST - TOTTRN
02752 988. CTRECD = CTRECD * 2
02753 989. CALL XREAD (UNTFIZ, 1, ARRAY(FLOSV3), TSTWRD, ISTAT(13),
02753 990. *IRWRDS(13))
02754 991. 7080 IF (IRSTAT(13) .EQ. 1) GO TO 7080
02756 992. IF (IRSTAT(13) .GT. 3) GO TO 9000
02760 993. IF (DEBUGF .EQ. 1 .AND. TSTKEY .EQ. 1)
02760 994. *CALL FIZMRF (ARRAY(FLOSV3), ARRAY(VERTX3), NOFLD3, 2,
02760 995. *CLSNR, SUBNR, CODE, SYN, DFTPT, OUTLNP, SYNRTX)
02760 996. C.....
02760 997. C ZOOM RECTANGULAR VERTICES FOR TEST FIELDS OR DESIGNATE FIELDS
02760 998. C.....
02762 999. ENDFLD = VERTX3 - 1
02763 1000. DO 7100 J1 = FIELDS, ENDFLD, 3
02766 1001. J2 = J1 + 1
02767 1002. J3 = J1 + 2
02770 1003. J4 = J1 + 3
02771 1004. ARRAY(J1) = FLOAT ((ARRAY(J1) / ILINIC) - SKPLIN) * ZONLIN
02772 1005. ARRAY(J2) = FLOAT ((ARRAY(J2) / ILINIC) - SKPLIN) * ZONLIN
02773 1006. ARRAY(J3) = FLOAT ((ARRAY(J3) / ISANIC) - SKPSAN) * ZONSAN
02774 1007. ARRAY(J4) = FLOAT ((ARRAY(J4) / ISANIC) - SKPSAN) * ZONSAN
02775 1008. 7100 CONTINUE
02775 1009. C.....
02775 1010. C ZOOM ACTUAL TEST FIELDS
02775 1011. C.....
02777 1012. DO 7120 I = VERTX3, TOTST, 2
03002 1013. J = I + 1
03003 1014. ARRAY(I) = FLOAT ((ARRAY(I) / ISANIC) - SKPSAN) * ZONSAN
03004 1015. ARRAY(J) = FLOAT ((ARRAY(J) / ILINIC) - SKPLIN) * ZORLIN
03005 1016. 7120 CONTINUE
03007 1017. IF (DESKEY .EQ. 1) CALL WRTFLD (ARRAY(FLOSV3), ARRAY(VERTX3),
03007 1018. *NOFLD3, 3, CLSNR, SUBNR)
03011 1019. IF (TSTKEY .EQ. 1) CALL FIZMRF (ARRAY(FLOSV3), ARRAY(VERTX3),
03011 1020. *NOFLD3, 4, CLSNR, SUBNR, CODE, SYN, DFTPT, OUTLNP, SYNRTX)
03011 1021. C
03011 1022. C.....
03011 1023. C CALCULATE HOW MANY TIMES A PIXEL WILL BE WRITTEN ON TAPE.
03011 1024. C.....
03011 1025. C
03013 1026. 7200 CONTINUE
03014 1027. ASAN = SKPSAN + 1
03015 1028. SANDFT = 0
03016 1029. J = 0
03017 1030. MAXSAN = 0
03020 1031. DO 7210 I = ASAN, INPTS
03023 1032. J = J + 1
03024 1033. CALL FIZSOM (J, MAXSAN, ZONSAN, SANSON(J), LINSAN, 87205)
03025 1034. 7205 MAXSAN = MAXSAN + SANSON(J)
03026 1035. IF (MAXSAN .EQ. LINSAN) GO TO 7215
03030 1036. 7210 CONTINUE
03032 1037. I = INPTS
03033 1038. J = INPTS + 1
03034 1039. SANDFT = LINSAN - MAXSAN
03035 1040. SANSON(J) = SANDFT
03036 1041. MAXSAN = SANSON(J) + MAXSAN
03037 1042. 7215 PROSAN = J
03040 1043. ESAN = I
03041 1044. ASAN1 = ASAN
03042 1045. IF (ASAN .EQ. 1) ASAN1 = 2
03044 1046. ASANF = ASAN
03045 1047. IF (ASAN .GT. 1) ASANF = ASAN-1

```

ORIGINAL PAGE IS
OF POOR QUALITY



03047 1048.
 03050 1049.
 03052 1050.
 03053 1051.
 03055 1052.
 03057 1053.
 03060 1054.
 03063 1055.
 03063 1056.
 03063 1057.
 03063 1058.
 03063 1059.
 03063 1060.
 03072 1061.
 03073 1062.
 03074 1063.
 03075 1064.
 03077 1065.
 03100 1066.
 03102 1067.
 03103 1068.
 03104 1069.
 03106 1070.
 03107 1071.
 03110 1072.
 03111 1073.
 03112 1074.
 03113 1075.
 03114 1076.
 03116 1077.
 03121 1078.
 03122 1079.
 03123 1080.
 03125 1081.
 03126 1082.
 03131 1083.
 03132 1084.
 03134 1085.
 03135 1086.
 03136 1087.
 03137 1088.
 03141 1089.
 03142 1090.
 03143 1091.
 03144 1092.
 03145 1093.
 03150 1094.
 03151 1095.
 03152 1096.
 03153 1097.
 03154 1098.
 03154 1099.
 03154 1100.
 03154 1101.
 03154 1102.
 03154 1103.
 03154 1104.
 03154 1105.
 03154 1106.
 03154 1107.
 03154 1108.
 03155 1109.
 03156 1110.

```

    ESAN1 = 1
    IF (ESAN .EQ. INPTS) ESAN1 = INPTS - 1
    ESANF = ESAN + 1
    IF (ESAN .EQ. INPTS) ESANF = INPTS
    IF (DEBUG .EQ. 0) GO TO 7220
    ARBARE = 6NSANSON
    WRITE (6,580) ARBARE
    WRITE (6,585) (I, SANSON(I), I = 1,PROSAN)
  C
  C-----
  C          CALCULATE VARIABLES FOR WRITING ON THE DRUM
  C-----
  C
  7220 CONTINUE
    DRNST = TOTYST + 1
    NOCOL = LINSAM / PAGSYN
    IF (MOD(LINSAM,PAGSYN) .NE. 0) NOCOL = NOCOL + 1
    NOCOMR = PAGSYN/4
    IF (MOD(PAGSYN,4) .NE. 0) NOCOMR = NOCOMR + 1
    MOLHAP = PARARR - TOTYST
    MOLLIN = (MOLHAP / (NOCOMR + NOCOL * 2))
    IF (MOLLIN .LT. 1) GO TO 9045
    MOLMRD = MOLLIN * NOCOMR + 2
    CALL RIBIT (IBADDR, NWDS)
    COLINC(1) = IBADDR
    COLADD(1) = IBADDR
    ALADD = NOCOMR * LIRLIN + 2
    DRHRT = 0
    IF (NOCOL .EQ. 1) GO TO 7230
    DO 7230 I = 2, NOCOL
      COLADD(I) = COLADD(I-1) + ALADD
      COLINC(I) = COLADD(I)
  7230 CONTINUE
  7250 DRHED = DRNST + (MOLMRD + NOCOL)
    DO 7260 I = DRHST, DRHED
      ARRAY(I) = BLANK
  7260 CONTINUE
    SYNPA6 = PAGSYN
    REALIN = MOLHAP/(NOCOMR+4)
    TIRREA = LIRLIN / REALIN
    IF (MOD(LIRLIN,REALIN) .NE. 0) TIRREA = TIRREA + 1
    LNPSAM = MOD(LINSAM,6) + LINSAM
    FOR(1) = 6N(TX,15)
    FOR(2) = 6N(2X,T9)
    FOR(3) = 1H,
    ENCODE (6,350,FTT) NOCOMR
    FOR(4) = FTT
    FOR(5) = 6NA6,/,1
    FOR(6) = 6NM, T9,
    FOR(7) = FOR(4)
    FOR(8) = 3NA6)
  C
  C-----
  C          COUNTER FOR THE LOOP TO FILTER THE INPUT FIELD
  C          FROM INFIZ AND THE FASTRAN OR TAPE UNITS.
  C-----
  C
  C
  7280 CONTINUE
    BUFT = 1
  
```



```

03157 1111. CTFILE = CTFILE + 1
03160 1112. IF (MOLTST(CTFILE) .NE. 0 .AND. ISLYST(CTFILE) .NE. 0)
03160 1113. *ATFILE = ATFILE + 1
03162 1114. IF (CTFILE(CTFILE) .NE. 0) BFILE = BFILE + 1
03164 1115. IF (OTFILE(CTFILE) .EQ. 1)
03164 1116. *CALL FIZTMD (LINSAN, TAPOUT, OTUNT, DDHVV, SEDKEY)
03166 1117. IF (CTFILE .EQ. TIMFIL) GO TO 7300
03170 1118. OUTFAS = UNTFA1
03171 1119. IF (MOD(CTFILE,2) .EQ. 0) OUTFAS = UNTFA2
03171 1120. C
03171 1121. C
03171 1122. C.....
03171 1123. C SKIP THE NUMBER OF SKIP LINE + 3 LINES
03171 1124. C.....
03171 1125. C
03171 1126. C
03173 1127. 7300 CONTINUE
03174 1128. SKP3 = SKPLIN + 3
03175 1129. I1 = 2
03176 1130. I2 = -3
03177 1131. I3 = 4
03200 1132. J = 1
03201 1133. INRDS(I3) = INPTS
03202 1134. IRSTAT(I3) = 0
03203 1135. DO 7340 LINE = 1, SKP3
03206 1136. 7320 IF (IRSTAT(I3) .EQ. 1) GO TO 7320
03210 1137. IF (IRSTAT(I3) .GT. 1) GO TO 9000
03212 1138. IF (INRDS(I3) .NE. INPTS) GO TO 9000
03214 1139. CALL MREAD (UNTFIZ, 1, IR(I,J), INPTS, IRSTAT(J), INRDS(J))
03215 1140. CTRECD = CTRECD + 1
03216 1141. SV = I1
03217 1142. I1 = I2
03220 1143. I2 = I3
03221 1144. I3 = J
03222 1145. J = SV
03223 1146. 7340 CONTINUE
03225 1147. DO 7360 SAMPLE = ASAMP,ESAMP
03230 1148. IF ( IR(SAMPLE,I1) .LT. 1) IR(SAMPLE,I1) = DFTPT
03232 1149. IF (IR(SAMPLE,I2) .LT. 1) IR(SAMPLE,I2) = DFTPT
03234 1150. 7360 CONTINUE
03236 1151. KEYFIL = 0
03237 1152. IF (SKP3 .EQ. 3) GO TO 7368
03241 1153. DO 7365 SAMPLE = ASAMP,ESAMP
03244 1154. IF (IR(SAMPLE,J) .LT. 1) IR(SAMPLE,J) = DFTPT
03246 1155. 7365 CONTINUE
03250 1156. KEYFIL = 1
03251 1157. 7368 CALL FIZFLT (I1,KEYFIL,J,I2)
03252 1158. OTLINE = 0
03253 1159. LINECT = 1
03254 1160. CALL FIZSON (LINECT, OTLINE, ZONLIN, ISKMR, LTNLIN, S7370)
03255 1161. GO TO 9075
03256 1162. 7370 CONTINUE
03257 1163. CALL FIZMRT (I1,ISKMR,S9090,ARRAY(FLOSV2), ARRAY(FIELD2),
03257 1164. *ARRAY(VRTX2), ARRAY(FLOSV3), ARRAY(FIELD3), ARRAY(VRTX3),
03257 1165. *ARRAY(DRMST),J)
03260 1166. IF (CTFILE .GE. TIMFIL) GO TO 7420
03262 1167. REMIND OUTFAS
03263 1168. JSTAT(1) = 0
03264 1169. JSTAT(2) = 0
03265 1170. 7380 IF (JSTAT(BUFPT) .EQ. 1) GO TO 7380
03267 1171. IF (JSTAT(BUFPT)) 7400,7400,7380
03272 1172. 7400 CALL MWRITE (OUTFAS,1,MOLARR(1,BUFPT),PROSAN,JSTAT(BUFPT))
03273 1173. 7420 BUFPT = 3 - BUFPT

```

```

03273 1174 C
03273 1175 C.....
03273 1176 C          MAIN PROCESSING DO-LOOP
03273 1177 C.....
03273 1178 C
03274 1179     KEYFIL = 1
03275 1180     SLINE = SKPS + 1
03276 1181     LIMHIT = 0
03277 1182     ELINE = ILINE + 2
03300 1183     DO 7400 LINE = SLINE, ELINE
03301 1184         IF (LIMHIT + EOFHIT) 7480, 7460, 8000
03306 1185     7400     IF (IRSTAT(I3) .EQ. 1) GO TO 7460
03310 1186         IF (IRSTAT(I3) .EQ. 0) GO TO 7480
03312 1187         IF (IRSTAT(I3) .EQ. 3) GO TO 9000
03314 1188         EOFHIT = 1
03315 1189         KEYFIL = 0
03316 1190         GO TO 7485
03317 1191     7480     IF (IRNRDS(I3) NRE. INPTS) GO TO 9040
03321 1192         CALL MREAD (UNTFIZ, 1, IR(I,J), INPTS, IRSTAT(J), IRNRDS(J))
03322 1193         CTRECD = CTRECD + 1
03322 1194     C          *****
03322 1195     C          DETERMINES THE NUMBER OF TIMES THE LINE WILL BE WRITTEN ON THE OUTPUT
03322 1196     C          *****
03323 1197     7485     CONTINUE
03324 1198         LINECT = LINECT + 1
03325 1199         CALL FIZSOM (LINECT, OTLINE, ZONLIN, ISKMR, LIRLIN, S7490)
03326 1200         LIMHIT = 1
03327 1201     7490     IF (KEYFIL .LT. 1) GO TO 7500
03331 1202         DO 7495 SAMPLE = ASAMF,ESAMF
03334 1203             IF (IR(SAMPLE,I3) .LT. 1) IR(SAMPLE,I3) = DFTPT
03336 1204     7495     CONTINUE
03336 1205     C          *****
03336 1206     C          FILTERS THE SCAN LINE BY USING THE HOLE AND ISLAND VALUES.
03336 1207     C          *****
03340 1208     7500     CALL FIZFLT (I2,KEYFIL,I1,I3)
03340 1209     C          *****
03340 1210     C          STORES FILTERED INPUT DATA ON FASTRAM OR TAPE FILE
03340 1211     C          *****
03341 1212         IF (CTFILE .GE. TIMFIL) GO TO 7515
03343 1213     7511     IF (JSTAT(BUFPT) .EQ. 1) GO TO 7511
03345 1214         IF (JSTAT(BUFPT)) 7512,7512,8980
03350 1215     7512     CALL MWRITE (OUTFAS,1,MOLARR(1,BUFPT),PROSAM,JSTAT(BUFPT))
03351 1216     7515     IF (ISKMR .LT. 1) GO TO 7520
03351 1217     C          *****
03351 1218     C          ZOOMS THE SCAN LINE. COUNTES THE NUMBER OF PIXELS IN EACH CLASS
03351 1219     C          WRITES THE LINE ON THE OUTPUT TAPE ISKMR TIMES AND
03351 1220     C          WRITES THE DATA FOR THE GRAYMAP ON THE DRUM.
03351 1221     C          *****
03353 1222         CALL FIZMRT (I2,ISKMR,S9090,ARRAY(FLDV2), ARRAY(FIELD2),
03353 1223         * ARRAY(VRTX2), ARRAY(FLDV3), ARRAY(FIELD3), ARRAY(VRTX3),
03353 1224         * ARRAY(DRNST),I1)
03354 1225     7520     SV = I1
03355 1226         BUFPT = 3 - BUFPT
03356 1227         I1 = I2
03357 1228         I2 = I3
03360 1229         I3 = J
03361 1230         J = SV
03362 1231     7400 CONTINUE
03364 1232     8000 CONTINUE
03365 1233         IF (LIMHIT .GT. 0) GO TO 8040
03367 1234         ISKMR = LIRLIN - OTLINE
03370 1235         KEYFIL = -1
03371 1236         LIMHIT = 1

```

ORIGINAL PAGE IS
OF POOR QUALITY

03372 1237*
 03373 1238*
 03373 1239*
 03373 1240*
 03373 1241*
 03373 1242*
 03373 1243*
 03374 1244*
 03375 1245*
 03376 1246*
 03377 1247*
 03400 1248*
 03401 1249*
 03403 1250*
 03405 1251*
 03407 1252*
 03413 1253*
 03416 1254*
 03421 1255*
 03422 1256*
 03426 1257*
 03434 1258*
 03442 1259*
 03445 1260*
 03447 1261*
 03452 1262*
 03453 1263*
 03460 1264*
 03462 1265*
 03463 1266*
 03464 1267*
 03464 1268*
 03474 1269*
 03476 1270*
 03477 1271*
 03502 1272*
 03504 1273*
 03505 1274*
 03506 1275*
 03507 1276*
 03511 1277*
 03512 1278*
 03516 1279*
 03517 1280*
 03521 1281*
 03522 1282*
 03524 1283*
 03525 1284*
 03530 1285*
 03531 1286*
 03533 1287*
 03536 1288*
 03537 1289*
 03541 1290*
 03542 1291*
 03544 1292*
 03546 1293*
 03547 1294*
 03550 1295*
 03551 1296*
 03552 1297*
 03553 1298*
 03554 1299*

```

CALL FIZFLT (12, KEYFIL, DUMMY, DUMMY)
CALL FIZMRT (12, ISKMR, 5909, ARRAY(FLOSV2), ARRAY(FIELD2),
+ARRAY(VRTX2), ARRAY(FLOSV3), ARRAY(FIELD3), ARRAY(VRTX3),
+ARRAY(DRMST), 1)
C.....
C          OUTPUT SUMMARY FOR FILZON
C.....
8040 CONTINUE
  THOLCM = THOLCM + MOLECM(CTFILE)
  TISLCH = TISLCH + ISLDCM(CTFILE)
  TOTPCY = LINSAM + LIMLIN
  ZTOTPC = FLOAT(TOTPCY) * .01
  WRITE (6, HEAD)
  WRITE (6, 400)
  IF (OTFILE(CTFILE) .EQ. 0) GO TO 8050
  WRITE (6, 405) BFILE, OTUNT
  IF (TAPOUT .EQ. 1) WRITE (6, 302)
  IF (TAPOUT .EQ. 2) WRITE (6, 303)
8050 CONTINUE
  WRITE (6, 410) URTFIZ, CTRECD
  WRITE (6, 415) HOLTST(CTFILE), ATFILE, MOLECM(CTFILE), THOLCM
  WRITE (6, 416) ISLTST(CTFILE), ATFILE, ISLDCM(CTFILE), TISLCH
  WRITE (6, 420) TOTPCY
  WRITE (6, 440)
  DO 8060 I = 1, DFTPT
    ZPC(I, CTFILE) = FLOAT(PERCNT(I, CTFILE)) / ZTOTPC
    WRITE (6, 460) SUBNAM(I), PERCNT(I, CTFILE), ZPC(I, CTFILE)
8060 CONTINUE
  ZPC(DESNO1, CTFILE) = FLOAT(PERCNT(DESNO1, CTFILE)) / ZTOTPC
  ZPC(DESNO2, CTFILE) = FLOAT(PERCNT(DESNO2, CTFILE)) / ZTOTPC
  WRITE (6, 460) (SUBNAM(I), PERCNT(I, CTFILE), ZPC(I, CTFILE),
+ I = DESNO1, DESNO2)
  IF (SEKKEY .NE. 1) GO TO 8075
  ZTOCNI = 0.0
  DO 8070 I = 1, NOSUB2
    IF (SUBNAM(I) .EQ. 6HWATER) GO TO 8070
    ZHOL = FLOAT(PERCNT(I, CTFILE))
    ZTOCNI = ZHOL * ZTOCNI
    ZCHI(CTFILE) = ZCHI(CTFILE) + (ZHOL * ZSED(I))
8070 CONTINUE
  ZCHI(CTFILE) = ZCHI(CTFILE) / ZTOCNI
  WRITE (6, 520) ATFILE, ZCHI(CTFILE)
8075 CONTINUE
  IF (MAP(CTFILE) .EQ. 0) GO TO 8120
  CALL FIZMAP (ARRAY(DRMST), 5909)
  IF (CTFILE .GE. TINFIL) GO TO 8300
  DRHMRT = 0
  DO 8090 I = DRNST, DRHED
    ARRAY(I) = BLANK
8090 CONTINUE
  DO 8100 I = 1, NCOL
    COLINC(I) = COLADD(I)
8100 CONTINUE
8120 CONTINUE
  IF (JSTAT(1) .NE. 0) GO TO 8980
  IF (JSTAT(2) .NE. 0) GO TO 8980
  END FILE OUTFAS
  INPTS = PROSAM
  CTRECD = 0
  FILE = 1
  INLINE = INLINE - SKPLIN
  SKPLIN = 0
  SKPSAM = 0
  
```

ORIGINAL PAGE IS
 OF POOR QUALITY

C.2

```

03553 1300•      EOFMT = 0
03556 1301•      LIMMT = 0
03557 1302•      ASAMF = 1
03560 1303•      ESAMF = 1
03561 1304•      ASAM = 1
03562 1305•      ASAM1 = 2
03563 1306•      ESAM = PROSAM
03564 1307•      ESAM1 = PROSAM - 1
03565 1308•      UNTFIZ = OUTFAS
03566 1309•      REMIND UNTFIZ
03567 1310•      GO TO 7280
03570 1311•      8300 CONTINUE
03571 1312•      CALL CLOCK (ELATIN)
03572 1313•      WRITE (6,500) ELATIN
03575 1314•      STOP
C.....
C          FATAL ERROR WRITE STATEMENTS AND ERROR EXIT
C.....
03576 1318•      8900 CONTINUE
03577 1319•      BADREC = LINECT -2
03600 1320•      IF (JSTAT(BUFPT) .EQ. 2) WRITE (6,900) BADREC, CTFILE, OUTFAS
03600 1321•      IF (JSTAT(BUFPT) .EQ. 3) WRITE (6,900) OUTFAS, BADREC, CTFILE
03614 1322•      GO TO 9090
03615 1323•      9000 CONTINUE
03616 1324•      IF (IRSTAT(I3) .EQ. 2) WRITE (6,900) CTRECD, FILE, UNTFIZ
03624 1325•      IF (IRSTAT(I3) .EQ. 3) WRITE (6,905) UNTFIZ, FILE, CTRECD
03632 1326•      GO TO 9090
03633 1327•      9010 CONTINUE
03634 1328•      NUMHRS = IRHRDS(I3)
03635 1329•      WRITE (6,920) UNTFIZ, (ARFIZ(I), I = 1, NUMHRS)
03644 1330•      GO TO 9090
03645 1331•      9030 CONTINUE
03646 1332•      WRITE (6,950) SKFILE, IRSTAT(I3)
03652 1333•      GO TO 9090
03653 1334•      9040 CONTINUE
03654 1335•      WRITE (6,940) CTRECD, FILE, IRHRDS(I3), IMPTS
03662 1336•      GO TO 9090
03663 1337•      9045 CONTINUE
03664 1338•      GO TO 9090
03665 1339•      9050 CONTINUE
03666 1340•      WRITE (6,335)
03670 1341•      WRITE (6,345) (ARRAY(I), CLSNAN(I), I = 1, NOCLS2)
03677 1342•      GO TO 9090
03700 1343•      9060 CONTINUE
03701 1344•      WRITE (6,340)
03703 1345•      WRITE (6,342) (1, CLSNAN(I), ARRAY(I), I = 1, NOCLS2)
03713 1346•      GO TO 9090
03714 1347•      9070 CONTINUE
03715 1348•      WRITE (6,930) FIZFIL, FILE
03721 1349•      GO TO 9090
03722 1350•      9075 CONTINUE
03723 1351•      WRITE (6,915)
03725 1352•      GO TO 9090
03726 1353•      9080 CONTINUE
03727 1354•      WRITE (6,925) FLDKEY
03732 1355•      GO TO 9090
03733 1356•      9090 CONTINUE
03734 1357•      CALL CLOCK (ELATIN)
03735 1358•      WRITE (6,500) ELATIN
03740 1359•      WRITE (6,910)
03742 1360•      CALL CHRR
03743 1361•      END

```



END OF COMPILATION:
FILZON SYMBOLIC

NO DIAGNOSTICS.

08 APR 76 11:23:49 0 03012014 14 1361 (DELETED)

ORIGINAL PAGE IS
OF POOR QUALITY

BL PDP-6 COMB6, COMB6
PDP BL1 2403 0010
THIS PROC ELEMENT PROCESSED ON 08 APR 76 AT 11:23:49

08 APR 76

11:23:49

```
000001  PROC ORIGIN 1  ENTRY POINT 1  COMB6*  FCOPY
000002                                     COMMON/GLOBAL/HEAD(42),NAPTAP,DATAPE,SAVTAP,BNFILE,BRKEY,
000003                                     *           MISFIL,MISKEY,TRFORM,ERIPTP,ERPKEY,NAPUNT,NOFILE,
000004                                     *           BRUNAD,BRNHDS,PAGSIZ,DATFIL,STAFIL
000005                                     *           ,FIZUNT,FIZKEY,FIZFIL                                     FILZON
000006                                     END
000007  PROC ORIGIN 2  ENTRY POINT 2  COMB6*  FCOPY
000008  C*
000009  C*  GLOBAL COMMON IS USED IN EVERY PROCESSOR. IT IS ALWAYS IN CORE.
000010  C*  ALL PARAMETERS ARE INITIALIZED IN THE MONITOR ROUTINE OR BLKCON
000011  C*  EXCEPT AS NOTED BELOW
000012  C*  DEFINITIONS
000013  C*  HEAD - STANDARD HEADING PRINTED ON MOST OUTPUT PAGES.
000014  C*  NAPTAP - FORTRAN UNIT NUMBER ON WHICH THE NAPTAP FILE IS
000015  C*           WRITTEN (=2)
000016  C*  DATAPE - UNIT NO. FOR THE IMAGE DATA TAPE (=3)
000017  C*  SAVTAP - UNIT NO. ON WHICH THE STATISTICS FILE IS WRITTEN (=1)
000018  C*  BNFILE - UNIT NO. ON WHICH THE B-MATRIX FILE IS WRITTEN (=10)
000019  C*  BRKEY - TRIGGER INDICATING THAT THE B-MATRIX FILE HAS BEEN
000020  C*           WRITTEN. CAN BE SET IN SELECT CLASSIFY OR DATA-TR.
000021  C*  MISFIL - UNIT NO. ON WHICH THE HISTOGRAM FILE IS WRITTEN (=13)
000022  C*  MISKEY - TRIGGER INDICATING THE HISTOGRAM FILE HAS BEEN
000023  C*           WRITTEN. SET IN HIST PROCESSOR.
000024  C*  TRFORM - UNIT NO. ON WHICH THE TRANSFORMED IMAGE IS WRITTEN BY
000025  C*           THE DATA-TRANSFORMATION PROCESSOR. (=14)
000026  C*  ERIPTP - UNIT NO. ON WHICH THE ISOCLS PROCESSOR WRITES
000027  C*           CLUSTER STATISTICS FOR THE ERIPS SYSTEM. (=15)
000028  C*  ERPKEY - TRIGGER INDICATING THAT THE ERIPS INTERFACE TAPE
000029  C*           HAS BEEN WRITTEN.
```



000030
 000031
 000032
 000033
 000034
 000035
 000036
 000037
 000038
 000039
 000040
 000041
 000042
 000043
 000044
 000045
 000046
 000047
 000048
 000049
 000050
 000051
 000052

CORBK6

PROCEDURE

C* WAPUNT - 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
 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* DRUMDS - NO. OF WORDS AVAILABLE ON THE RANDOM ACCESS FILE.
 C* PAGESZ - 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* FIZUNT - LOGICAL UNIT ON WHICH THE FILES FOR PROGRAM FILZON FILZON
 C* ARE WRITTEN FILZON
 C* FIZKEY - TRIGGER TO WRITE A FILE FOR INPUT TO FILZON. FILZON
 C* 0 = DO NOT WRITE A FILE FILZON
 C* 1 = WRITE A FILE FILZON
 C* FIZFIL - TOTAL NUMBER OF FILES THAT ARE WRITTEN ON FIZUNT. FILZON
 C*

END

14 JAN 76 16:24:02 0 01447740 14 52 (DELETED)
 1 01451270 12 1



BL PDP, * CRBK10, CRBK10
 PDP BL1 2403 0010
 THIS PROC ELEMENT PROCESSED ON 08 APR 76 AT 11:24:16

08 APR 76

11:24:15

```

000001  PROC ORIGIN 1  ENTRY POINT 1  CRBK10  FCOPY
000002  COMMON/DISPL/CATFLG, CATNAM(61), CLSNAM(61), SUBNAM(61), SUBNO(60),
000003  * SUBCAT(60), CLSSUB(60), NONAP, TOTVT3, NOSUB3,
000004  * PCFBKY, TSTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
000005  * PLYKEY, BNFLG, BRCORB, BNFEAT, CDATE(2),
000006  * FLOBV2, FIELD2, VERTX2, FLOBV3, FIELD3, VERTX3, PCTIDS,
000007  * THRES(60), SYNTAX(66), HIGH(60), CON(60)
000008  * ,FLDKY, NOFLD2, NOFLD3, NOFETZ, FETVC2(30)
000009  * ,NOSUB2, NOTRFD, TOTVT2, NOCLS2
000010  * ,RAYNO(60), NOCAT, FILTER, MAPFRT
000011  * ,DEBKY, DESUNI, DESOTH, CROP ,ACROP, ADTHER, ATOTAL
000012  * ,SITE(4), ANALYS(3), CARB(10), CAPKEY
000013  * ,BFILE  FILZOR
000014  END
000015  PROC ORIGIN 2  ENTRY POINT 2  CONT10  FCOPY
000016  C*
000017  C* COMMON BLOCK DISPL IS USED ONLY IN THE DISPLAY PROCESSOR
000018  C*
000019  C*
000020  C* DEFINITIONS
000021  C*
000022  C* CATFLG - FLAG INDICATING WHETHER OR NOT CATEGORY PERFORMANCE
000023  C* REPORTS MUST BE GENERATED.
000024  C* CATNAM - NAMES OF CATEGORIES. READ FROM NAPTAP.
000025  C* CLSNAM - NAMES OF CLASSES. READ FROM NAPTAP.
000026  C* SUBNAM - NAMES OF SUBCLASSES. READ FROM NAPTAP.
000027  C* SUBCAT - SUBCLASS-CATEGORY CORRESPONDENCE VECTOR
000028  C* (SUBCAT(I)=N MEANS SUBCLASS I BELONGS TO CATEGORY N)
000029  C* CLSSUB - SUBCLASS-CLASS CORRESPONDENCE VECTOR.

```




000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061

C* (CLASSUB(I)=N MEANS SUBCLASS I BELONGS TO CLASS N)
C* NONAP - TRIGGER INDICATING WHETHER OR NOT A MAP IS TO BE PRINTED
C* TOTVTS - TOTAL NO. OF VERTICES IN INPUT TEST FIELDS.
C* NOSUB3 - NO. OF SUBCLASSES USED IN CLASSIFY PLUS ONE, FOR THE
C* THRESHOLD CLASS.
C* DCFDRY - KEY INDICATING WHETHER OR NOT GROUND TRUTH PERFORMANCE
C* REPORTS ARE TO BE PRINTED ON A PER FIELD BASIS.
C* YSTKEY - KEY INDICATING WHETHER OR NOT TEST FIELDS WERE INPUT.
C* TRBKEY - KEY INDICATING WHETHER OR NOT TRAINING FIELDS ARE TO
C* BE OUTLINED.
C* THRSKY - THRESHOLD KEY
C* #1 APPLY CHI-SQUARE THRESHOLDS
C* #2 APPLY EMPIRICAL THRESHOLDS
C* #3 APPLY USER-INPUT THRESHOLDS
C* #0 NO THRESHOLDING
C* STATKY - KEY FOR PRINTING STATS FROM MAPTAP
C* ENPYRS - EMPIRICAL THRESHOLDING FLAG
C* THRSVA - USER-INPUT THRESHOLD VALUE FLAG
C* PLYKEY - FLAG FOR PRINTING CUMULATIVE HISTOGRAMS OF QUADRATIC
C* FORM.
C* BRFLG - FLAG INDICATING WHETHER OR NOT A B-MATRIX WAS
C* APPLIED IN CLASSIFY.
C* BRCONB - NO. OF LINEAR COMBINATIONS IN B-MATRIX
C* BRFEAT - NO. OF CHANNELS USED IN COMPUTING B-MATRIX
C* CDATE - DATE OF CLASSIFICATION
C* FLD5VZ - ADDRESS IN 'ARRAY' FOR TRAINING FIELD INFORMATION.
C* FOR EACH TRAINING FIED 4 PIECES OF INFORMATION ARE
C* STORED - 1=FIELD NAME
C* 2=CLASS NO.
C* 3=SUBCLASS NO.
C* 4=NO. OF VERTICES
C* FIELDZ - ADDRESS IN 'ARRAY' FOR RECTANGULAR AREA SURROUNDING

ORIGINAL PAGE IS
OF POOR QUALITY

000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093

C* EACH TRAINING FIELD. FOR EACH TRAINING FIELD 5 PIECES
C* OF INFORMATION ARE STORED.
C* 1=LINE START
C* 2=LINE END
C* 3=SAMPLE START
C* 4=SAMPLE END
C* 5=POINTER INTO VERTEX ARRAY FOR VERTICES
C* OF THIS FIELD.
C* VERTX2 - ADRESS IN 'ARRAY' FOR TRAINING FIELD VERTICES.
C* FLD8V3 - SAME AS FLD8V2 FOR TEST FIELDS
C* FIELD3 - SAME AS FIELD2 FOR TEST FIELDS
C* VERTX3 - SAME AS VERTX2 FOR TEST FIELDS
C* PCTID3 - ADDRESS IN 'ARRAY' FOR PERFORMANCE TABLE.
C* THRES - THRESHOLD VALUES
C* SYMNTX - SYMBOLS FOR EACH SUBCLASS, PLUS THRESHOLD SYMBOL
C* AND OUTLINE SYMBOLS.
C* HIGH - THRESHOLD REJECTION PERCENTAGE - EMPIRICAL OPTION
C* CON - CONSTANT FACTOR FROM PROBABILITY DENSITY FUNCTION
C* FROM CLASSIFY. ONE FOR EACH SUBCLASS.
C* FLDKEY - KEY INDICATING WHETHER GROUND TRUTH FIELDS ARE
C* ASSOCIATED WITH CLASSES OR SUBCLASSES.
C* NOFLD2 - NO. OF TRAINING FIELDS
C* NOFLD3 - NO. OF TEST FIELDS
C* NOFET2 - NO. OF CHANNELS USED IN CLASSIFICATION.
C* PETVC2 - CHANNELS USED IN CLASSIFICATION.
C* NOSUB2 - NO. OF SUBCLASSES USED IN CLASSIFICATION.
C* NOTRFD - NO. OF GROUND TRUTH FIELDS FOR WHICH PERFORMANCE
C* TABLES WILL BE MADE. EQUALS NOFLD3 OR NOFLD2.
C* TOTVT2 - TOTAL NO. OF VERTICES FOR TRAINING FIELDS.
C* NOCLS2 - NO. OF CLASSES USED IN CLASSIFICATION.
C* KAT80 - CLASS - CATEGORY CORRESPONDENCE VECTOR
C* (KAT80(I)=N MEANS CLASS I IS IN CATEGORY N)



000094

000095

000096

000097

000098

000099

000100

000101

000102

000103

000104

000105

000106

000107

000108

000109

000110

000111

000112

CHK10

PROCEDURE

- C* NOCAT - NO. OF CATEGORIES.
- C* FILTER - FLAG FOR SPATIAL FILTERING OPTION.
- C* WAPPRT - FORMAT FOR OUTPUT MAP TAPE
- C* DESKEY - KEY INDICATING WHETHER OR NOT DESIGNATED FIELDS WERE INPUT
- C* DESUNI - NO. FOR DESIGNATED UNIDENTIFIABLE (NOSUB2-5)
- C* DESOTH - NO. FOR DESIGNATED OTHER (NOSUB2-6)
- C* CROP - NAME OF CROP FOR WHICH INTENSIVE TEST SITE SUMMARY REPORT IS TO BE PRINTED, CROP IS TO BE COMPARED WITH OTHER
- C* ACROP - ACRES OF 'CROP' - USER INPUT
- C* AOTHR - ACRES OF 'OTHER' - USER INPUT
- C* ATOTAL - TOTAL ACRES IN CLASSIFIED SEGMENT
- C* SITE - NAME OF SITE (CLASSIFIED SEGMENT)
- C* ANALYS - NAME OF ANALYST PERFORMING STUDY
- C* CARB - NAME OF PROCEDURE CONFIGURATION USED IN STUDY
- C* CRPREY - KEY FOR GENERATING INTENSIVE TEST SITE SUMMARY REPORT
- C* BFILE - WAPTAP FILE NUMBER FOR SDISPLAY RUN FILZOR
- C*
- C*

END

14 JAN 76 16:24:04 0 01453694 14 112 (DELETED)
 1 01456714 12 1

BL PDP. FIZPDP.FIZPDP
PDP BL1 2403 0010
THIS PROC ELEMENT PROCESSED ON 08 APR 76 AT 11:24:42

08 APR 76

11:24:42

```
000001  PROC ORIGIN 1  ENTRY POINT 1  FIZCOM FCOPY
000002                                     IMPLICIT DOUBLE PRECISION (2)
000003                                     REAL COM.THRES
000004                                     C
000005                                     C          PARAMETERS
000006                                     C
000007                                     PARAMETER PAGCOL = 10, PAGSYN = 110, PARSAN = 1000, PARLIN = 1000
000008                                     PARAMETER PAGWRD = 10, PSCLAD = 61, ISUBCL = PSCLAD - 1
000009                                     PARAMETER PSEDS = 21, POUTLN = PSCLAD*6, PSYN = 2*POUTLN,
000010                                     * PSANPS = PARSAN * 5, PARFIL = 8
000011                                     C
000012                                     C          COMMON BLOCKS
000013                                     C
000014                                     COMMON /FIZ/ SYN(POUTLN,2), CODE(POUTLN), TAPOUT, DTUNT, SKPLIN,
000015                                     * SRPSAN, DFTCOD, DFTSYN(2), DFTPT, ZORLIN, ZORSAN,
000016                                     * LINLIN , LINSAN, MED1(10), MED2(10), OTLIN, DATE(2),
000017                                     * COMERT(10),DEBUGF,LINMIT,PROSAN,LNPSAN,
000018                                     * ASAN1, ASAN, ESAN1, ESAN, CTSBDS, EOFHIT,
000019                                     * SEDNAM(PSEDS),TRNO,TSTNO,DESN01,DESN02,DUPNO
000020                                     COMMON /DRUM/ IBADDR, COLADD(PAGCOL), COLINC(PAGCOL), NOCOL,
000021                                     * BGCORR, MOLLIN, MOLWRD, DRHWRT,
000022                                     * FOR(S), ALADD, REALIN, SYRPAG, TIMREA
000023                                     COMMON /INFIZ1/ FIZNAM, DSPYDT(2), DSPYTN, FLDESC, INPTS, INLINE,
000024                                     * ILINST, ILINED, ILINIC, ISANST, ISANED, ISANIC
000025                                     COMMON /CONSAH/ SANSON(PARSAN),IR(PSANPS,4),MOLARR(PARSAN,2),
000026                                     * BUFPY, UNTFAS, JSTAT(2)
000027                                     COMMON/UNTIF2/ NAP(PARFIL), OTFILE(PARFIL), TIMFIL,
000028                                     * CTFILE, ISLTST(PARFIL), MOLTST(PARFIL), TOTPCT,
000029                                     * PERCNT(POUTLN,PARFIL),ZCHI(PARFIL),
```



000030

000031

000032

000033

000034

000035

000036

000037

FIZPOP

PROCEDURE

PROC ORIGIN 2 ENTRY POINT 2 DEFCON= FCOPY

END

END

• ZPC(POUTLN,PARFIL), ISLCH(PARFIL),
 • NOLECH(PARFIL), TISLCH, THOLCH
 • ,ATFILE

28 FEB 76	13:58:26	0	02655732	14	37	(DELETED)
		1	02656740	12	1	

@ FOR, FIZBOR/SKAAA,FIZBOR,FIZBOR
 UNIVAC 1100 FORTRAN V EXEC II LEVEL 25A -(EXEC8 LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:24:45

08 APR 76

11:24:45

SUBROUTINE FIZBOR ENTRY POINT 000402

STORAGE USED: CODE(1) 000461; DATA(0) 000074; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 FDLINT
 0004 HERR35

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000017	107G	0001	000104	131G	0001	000264	166G	0001	000322	175G	0001	000217	20L	
0001	000222	30L	0001	000241	40L	0001	000257	43L	0001	000351	46L	0001	000356	50L	
0000	1	000036	BORNUM	0000	1	000000	FL	0000	1	000026	I	0000	1	000034	IB
0000	1	000037	IJ	0000	1	000040	IRJPS	0000	1	000030	IPT	0000	1	000033	J
0000	1	000027	NV	0000	1	000031	SAMPS					0000	1	000032	NI

```

00101 1* SUBROUTINE FIZBOR(ISH,LINUM,IR,NOFLD,FIELD,FLOSAV,VERTEX,
00101 2* * NOSUB3,SANSTR,SANEND,SANINC,LININC)
00101 3* C*
00101 4* C* THIS SUBROUTINE SETS THE SYMBOL INDEX IN THE CLASSIFIED LINE (IR)
00101 5* C* ARRAY TO OUTLINE TRAINING OR TEST FIELDS IN THE MAP.
00101 6* C*
00103 7* IMPLICIT INTEGER(A-Z)
00104 8* DIMENSION IR(1),FIELD(5,NOFLD),FLOSAV(4,NOFLD),VERTEX(1)
00105 9* DIMENSION FL(22)
00106 10* DO 50 I=1,NOFLD
00111 11* IF(LINUM-LININC .LT. FIELD(1,I))GO TO 50
00113 12* IF(LINUM-LININC .GT. FIELD(2,I))GO TO 50
00115 13* IF(FIELD(3,I) .GT. SANEND)GO TO 50
00117 14* IF(FIELD(4,I) .LT. SANSTR)GO TO 50
00117 15* C*
00117 16* C* FOUND A FIELD THAT NEEDS BORDER ON THIS LINE. NOW FIND FIELD
00117 17* C* INTERSECTIONS ON THIS LINE.
00117 18* C*
00121 19* NV=FLOSAV(4,I)
00122 20* IPT=FIELD(5,I)
00122 21* C*
00122 22* C* TOP OR BOTTOM
00122 23* C*
00123 24* IF(LINUM.GT.FIELD(2,I))GO TO 40
00125 25* IF(LINUM.LT.FIELD(1,I))GO TO 30
00127 26* CALL FDLINT(VERTEX(IPT),NV,FL,LINUM,SAMPS,NI)
00130 27* DO 20 J=1,NI,2
00133 28* IB=(FL(J)-SANSTR)/SANINC
00134 29* IE=(FL(J+1)-SANSTR)/SANINC + 2
00135 30* IF(MOD(SANSTR,SANINC).NE.MOD(FL(J),SANINC))IB=IB+1
00137 31* IF (IB .LT. SANSTR) GO TO 20
00141 32* IF (IE .GT. SANEND) GO TO 20
00143 33* IF (IB .GT. SANEND) GO TO 20
00145 34* IF (IE .LT. SANSTR) GO TO 20
  
```

FILZON
 FILZON
 FILZON
 FILZON

```

00147 35• BORNUN=ISYM
00150 36• IF(IR(IB).GT.NOSUB3)BORNUN = NOSUB3 + 3
00152 37• IR(IB)=BORNUN
00153 38• IF(IR(IE).GT.NOSUB3)BORNUN = NOSUB3 + 3
00155 39• IR(IE) = BORNUN
00156 40• 20 CONTINUE
00160 41• GO TO 50
00161 42• 30 CONTINUE
00161 43• C•
00161 44• C• GET INTERCEPTS FOR TOP LINE IN FIELD
00161 45• C•
00162 46• CALL FDLINT(VERTEX(IPT),BV,FL,FIELD(1,I),SARPS,NI)
00163 47• GO TO 45
00163 48• C•
00163 49• C• GET INTERCEPTS FOR BOTTOM LINE IN FIELD
00163 50• C•
00164 51• 40 CALL FDLINT(VERTEX(IPT),BV,FL,FIELD(2,I),SARPS,NI)
00165 52• 45 DO 47 J=1,NI,2
00170 53• IB = (FL(J)-SARSTR)/SARINC
00171 54• IE = (FL(J+1)-SARSTR)/SARINC + 2
00172 55• IF(MOD(SARSTR,SARINC).NE.MOD(FL(J),SARINC))IB=IB+1
00174 56• DO 46 IJ=IB,IE
00177 57• IF (IJ .LT. SARSTR) GO TO 46
00201 58• IF (IJ .GT. SAREND) GO TO 46
00203 59• BORNUN=ISYM
00204 60• IF(IR(IJ).GT.NOSUB3)BORNUN=NOSUB3+3
00206 61• IR(IJ) = BORNUN
00207 62• 46 CONTINUE
00211 63• 47 CONTINUE
00213 64• 50 CONTINUE
00215 65• END

```

FILZON
FILZON

FILZON
FILZON

FILZON
FILZON

FLO00720

END OF COMPILATION: NO DIAGNOSTICS.
FILZON SYMBOLIC

08 APR 76 11:24:47 0 03031330 14 65 (DELETED)



FOR FIZFLT/SKAAA, FIZFLT, FIZFLT
 UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A (EXEC8 LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:24:47

08 APR 76

11:24:47

SUBROUTINE FIZFLT ENTRY POINT 000314

STORAGE USED: CODE(1) 000335; DATA(0) 000036; BLANK COMMON(2) 000000

COMMON BLOCKS:

0003 FIZ 000434
 0004 DRUR 000046
 0005 INFIZ1 000015
 0006 CONSAM 015560
 0007 UNTIFZ 003216
 0010 DISPL 001370

EXTERNAL REFERENCES (BLOCK, NAME)

0011 FIZTST
 0012 BERRSS

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001 000024 13ZG	0001 000045 143G	0001 000122 164G	0001 000030 2000L	0001 000065 3000L
0001 000102 30Z0L	0001 000270 4000L	0010 001342 ACROP	0004 000042 ALADD	0010 001351 ANALYS
0010 001343 ADTHER	0003 I 000375 ASAN	0003 I 000374 ASAN1	0007 003215 ATFILE	0010 001344 ATOTAL
0010 000570 BRCONB	0010 000571 BRFEAT	0010 000567 BRFLG	0006 I 015554 BUFFT	0010 001354 CANS
0010 000000 CATFLG	0010 000001 CATNAM	0010 000572 CDATE	0007 I 003173 CHANGE	0010 000076 CLSNAR
0010 000460 CLSSUB	0003 000204 CODE	0004 000001 COLADD	0004 000013 COLINC	0003 000356 CORENT
0010 001075 COM	0010 001341 CROP	0010 001346 CRPKEY	0007 I 000021 CTFILE	0003 000400 CTSEDS
0003 000354 DATE	0003 000370 DEBUG	0010 001336 DESKEY	0003 000431 DESNO1	0003 000432 DESNO2
0010 001340 DESOYM	0010 001337 DESUNI	0003 000315 DFTCOD	0003 I 000320 DFTPT	0003 000316 DFTSYM
0004 000031 DRWRT	0005 000001 DSPYOT	0005 000003 DSPYTN	0003 000433 DUPRO	0010 000564 ENPTRS
0003 000401 EOPMIT	0003 I 000377 ESAN	0003 I 000376 ESAN1	0010 001175 FETVC2	0010 000575 FIELD2
0010 000600 FIELDS	0010 001334 FILTER	0005 000000 FIZNAM	0005 000004 FDESC	0010 001171 FLDREY
0010 000574 FLDVZ	0010 000577 FLDV3	0004 000032 FOR	0003 000327 HED1	0003 000341 HED2
0010 001001 HIGH	0006 I 011634 HOLARR	0007 003203 MOLECM	0004 000027 HOLLIN	0000 I 000001 HOLSAM
0007 000032 HOLTST	0004 000030 HOLWRD	0000 I 000002 I	0004 000000 IBADDR	0005 000010 ILINED
0005 000011 ILINIC	0005 000007 ILINST	0000 I 000003 IRINUS	0000 000006 IBJPS	0005 000006 INLINE
0003 000005 INPTS	0000 I 000004 IPLUS	0006 I 001750 IR	0005 000013 ISAMED	0005 000014 ISANIC
0005 000012 ISARST	0007 003173 ISLDCH	0007 000022 ISLTST	0006 015556 JSTAT	0010 001237 KATRO
0003 000371 LIMHIT	0003 000325 LIMLIN	0003 000326 LIRSAN	0003 000373 LIRPSAN	0007 000000 MAP
0010 001335 MAPFRT	0010 001367 MFILE	0010 001333 NOCAT	0010 001236 NOCLS2	0004 000025 NOCOL
0004 000026 NOCONR	0010 001174 NOFET2	0010 001172 NOFLD2	0010 001173 NOFLD3	0010 000554 NOMAP
0010 001235 NOSUB2	0010 000556 NOSUB3	0010 001234 NOTRFD	0007 000010 OTFILE	0003 000353 OTLINE
0003 000312 OTURT	0010 000557 PCFDKY	0010 000602 PCTID3	0007 000043 PERCNT	0010 000566 PLTRKY
0003 I 000372 PROSAM	0004 000043 REALIN	0000 I 000000 SAMPLE	0006 000000 SANSOW	0003 000402 SEDNAM
0010 001345 SITE	0003 000313 SEPLIN	0003 000314 SKPSAM	0010 000563 STATKY	0010 000364 SUBCAT
0010 000173 SUBNAM	0010 000270 SUBNO	0003 000000 SYR	0010 000677 SYMRTX	0004 000044 SVRPAG
0003 000311 TAPOUT	0007 I 000022 TEST	0007 003214 THOLCM	0010 R 000603 THRES	0010 000562 THRSKY
0010 000565 THRSVA	0007 000020 TIRFIL	0004 000045 TIRREA	0007 003213 TISLCH	0007 000042 TOTPCY
0010 001235 TOTVTZ	0010 000555 TOTVTS	0010 000561 TRNKEY	0003 000427 TRNNO	0010 000560 TSTREY
0003 000430 TSTRD	0006 015555 URTFAS	0010 000576 VERTX2	0010 000601 VERTX3	0007 D 001073 ZCHI
0003 D 000321 ZORLIN	0003 D 000323 ZONSAM	0007 D 001113 ZPC		

ORIGINAL PAGE IS
 OF POOR QUALITY



```

00101 1* SUBROUTINE FIZFLT (LINE,KEYFIL,11,13)
00103 2* IMPLICIT INTEGER (A-Y)
00104 3* INCLUDE FIZCOM, LIST
00105 3* IMPLICIT DOUBLE PRECISION (Z)
00106 3* REAL CON,THRES
00106 3* C
00106 3* C
00106 3* C
00107 3* PARAMETERS
00110 3* PARAMETER PAGCOL = 10, PAGSYN = 110, PARSAN = 1000, PARLIN = 1000
00110 3* PARAMETER PAGWRD = 19, PSCLAD = 61, ISUBCL = PSCLAD - 1
00111 3* PARAMETER PSEDS = 21, POUTLN = PSCLAD-6, PSYN = 2*POUTLN,
00111 3* PSANPS= PARSAN * 5, PARFIL = 8
00111 3* C
00111 3* C
00111 3* C
00112 3* COMMON BLOCKS
00112 3* COMMON /FIZ/ SYN(POUTLN,2), CODE(POUTLN), TAPOUT, OTUNT, SKPLIN,
00112 3* SKPSAN, DFTCOD, DFTSYN(2), DFTPT, ZONLIN, ZONSAN,
00112 3* LTNLIN, LINSAN, MEDY(10), MEDZ(10), OTLINE, DATE(2),
00112 3* CONENT(10),DEBUGF,LIMIT,PROSAN,LWPSAN,
00112 3* ASANT, ASAN, ESANT, EGAN, CTSEDS, EOFMIT,
00112 3* SEDNA(PSEDS),TRNO,TSTNO,DESNO1,DESNO2,DUPNO
00113 3* COMMON /DRUM/ IBADDR, COLADD(PAGCOL), COLINC(PAGCOL), NOCOL,
00113 3* NOCONR, HOLLIN, HOLWRD, DRUMAT,
00113 3* FOR(8), ALADD, REALIN, SYNPAQ, TIMREA
00114 3* COMMON /INFIZ/ FIZNAM, DSPYDT(2), DSPYTH, FLDESC, INPTS, INLINE,
00114 3* ILINST, ILINED, ILINIC, ISANST, ISANED, ISANIC
00115 3* COMMON /CONAN/ SANSON(PARSAN),IR(PSANPS,4),HOLARR(PARSAN,2),
00115 3* BUPT, URTFAS, JSTAT(2)
00116 3* COMMON/URTIFZ/ RAP(PARFIL), OTFILE(PARFIL), TIRFIL,
00116 3* CTFILE, ISLTST(PARFIL), HOLTST(PARFIL), TOTPCY,
00116 3* PERCNT(POUTLN,PARFIL),ZCR(PARFIL),
00116 3* ZPC(POUTLN,PARFIL), ISLDCM(PARFIL),
00116 3* NOLECH(PARFIL), TISLCH, THLCH
00116 3* ,ATFILE
00117 3* END
00120 4* INCLUDE CMR10.LIST
00121 4* COMMON/DISPL/CATFLG, CATNAM(61), CLSN/M(61), SUBNAM(61), SUBNO(60),
00121 4* SUBCAT(60), CLSSUB(60), IOMAP, TOTVTS, NOSUB3,
00121 4* PCFONY, TSTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
00121 4* PLTKEY, BRFLG, BRCONB, BRFEAT, CDAT(2),
00121 4* FLDSVZ, FIELDZ, VERTXZ, FL/SV3, FIELD3, VERTX3, PCTID3,
00121 4* THRES(60), SYMHTX(66), H/GH(60), COR(60)
00121 4* ,FLDKEY, NOFLDZ, NOFLD3, NOFETZ, FETVCZ(30)
00121 4* ,NOSUB2, NOTRFD, TOTVTZ, NOCLSZ
00121 4* ,KATNO(60), NOCAT, FILTER, MAPERT
00121 4* ,DESKEY, DESUNI, DESOTH, CROP , ACROP, AOTHER, ATOTAL
00121 4* ,SITE(4), ANALYS(3), CAHS(10), CRPKEY
00121 4* ,NFILE
00122 4* END
00123 5* DIMENSION TEST(PARFIL,2), CHANGE(PARFIL,2)
00123 6* C
00123 7* C
00123 8* C
00124 9* EQUIVALENCE (TEST(1,1),ISLTST(1)),(TEST(1,2),HOLTST(1)),
00124 10* (CHANGE(1,1),ISLDCM(1)),(CHANGE(1,2),NOLECH(1))
00124 11* C
00124 12* C
00124 13* C
00124 14* C
00124 15* C
00124 16* C
00124 17* C
00124 18* C

```

FILZON

```

00124 19. C
00124 20. C.....
00124 21. C
00125 22. IF (KEYFIL) 1000, 2000, 3000
00125 23. C
00125 24. C.....
00125 25. C CREATE A SCAN LINE OF POINTER DEFAULT VALUES
00125 26. C.....
00125 27. C
00130 28. 1000 CONTINUE
00131 29. DO 1050 SAMPLE = 1, PROSAM
00134 30. MOLARR(SAMPLE,BUFPT) = DFTPT
00133 31. 1050 CONTINUE
00137 32. GO TO 4000
00137 33. C
00137 34. C.....
00137 35. C SET VALUES IN MOLARR EQUAL TO VALUES IN IR ARRAY
00137 36. C.....
00137 37. C
00140 38. 2000 CONTINUE
00141 39. MOLSAM = 0
00142 40. DO 2050 SAMPLE = ASAM, ESAM
00145 41. MOLSAM = MOLSAM + 1
00146 42. MOLARR(SAMPLE,BUFPT) = IR(SAMPLE,LINE)
00147 43. 2050 CONTINUE
00151 44. IF (MOLSAM .NE. PROSAM) MOLARR(MOLSAM+1,BUFPT) = DFTPT
00153 45. GO TO 4000
00153 46. C
00153 47. C.....
00153 48. C SPATIAL FILTER SCANNER LINE OF INPUT DATA
00153 49. C.....
00153 50. C
00154 51. 3000 CONTINUE
00155 52. MOLSAM = 0
00156 53. IF (ASAM1 .EQ. ASAM) GO TO 3020
00160 54. MOLARR(1,BUFPT) = IR(ASAM,LINE)
00161 55. MOLSAM = 1
00162 56. 3020 CONTINUE
00163 57. DO 3080 SAMPLE = ASAM1, ESAM1
00166 58. MOLSAM = MOLSAM + 1
00167 59. I = 1
00170 60. IF (IR(SAMPLE,LINE) .EQ. DFTPT) I = 2
00172 61. IRINUS = SAMPLE - 1
00173 62. IPLUS = SAMPLE + 1
00174 63. CALL FIZTST (TEST(CTFILE,I), CHANGE(CTFILE,I),
00174 64. * IR(SAMPLE,LINE), MOLARR(MOLSAM,BUFPT),
00174 65. * IR(SAMPLE,I), IR(SAMPLE,I3), IR(IRINUS,LINE),
00174 66. * IR(IPLUS,LINE), IR(IRINUS,I), IR(IPLUS,I3),
00174 67. * IR(IPLUS,I), IR(IRINUS,I3), DFTPT)
00175 68. 3080 CONTINUE
00177 69. IF (MOLSAM .EQ. PROSAM) GO TO 4000
00201 70. MOLSAM = MOLSAM + 1
00202 71. MOLARR(MOLSAM,BUFPT) = IR(ESAM,LINE)
00203 72. IF (MOLSAM .NE. PROSAM) MOLARR(MOLSAM+1,BUFPT) = DFTPT
00205 73. 4000 RETURN
00206 74. END

```

END OF COMPILATION: NO DIAGNOSTICS.
FIZFLT SYMBOLIC

08 APR 76 11:24:49 0 03032231 14 74 (DELETED)

B FOR, FIZHAP/SKAAA, FIZHAP, FIZHAP
 UNIVAC 1108 FORTRAN V EXEC 11 LEVEL 25A - (EXECB LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:24:50

08 APR 76

11:24:49

SUBROUTINE FIZHAP ENTRY POINT 000670

STORAGE USED: CODE(1) 000706; DATA(0) 000731; BLANK COMMON(2) 000000

COMMON BLOCKS:

0003 FIZ 000434
 0004 DRUM 000046
 0005 INFIZ1 000015
 0006 CONSAM 015560
 0007 UNTIFZ 003216
 0010 DISPL 001370
 0011 GLOBAL 000076

EXTERNAL REFERENCES (BLOCK, NAME)

0012 SETHRC
 0013 RREAD
 0014 RNDUS
 0015 R10ZS
 0016 R101S
 0017 RERR4S
 0020 RERR3S

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0000	000537	100F	0001	000012	1000L	0001	000121	1020L	0001	000132	1100L	0001	000205	1110L				
0001	000374	1180L	0001	000446	1200L	0001	000683	1300L	0001	000073	162G	0000	000540	200F				
0001	000147	214G	0001	000164	226G	0001	000236	253G	0001	000277	265G	0000	000613	270F				
0001	000304	271G	0000	000621	280F	0000	000632	285F	0000	000633	290F	0000	000635	300F				
0001	000636	3000L	0001	000550	341G	0001	000354	344G	0000	000635	350F	0001	000560	350G				
0001	000620	364G	0001	000625	370G	0000	000665	400F	0010	001342	ACROP	0004	I	000042	ALADD			
0010	001351	ANALYS	0010	001343	AOTMER	0003	000375	ASAM	0003	000374	ASAM1	0007	I	003215	ATFILE			
0010	001344	ATOTAL	0010	000570	BRCORB	0010	000571	BNFEAT	0011	000055	BNFILE	0010	000567	BNFLG				
0011	000056	BNKEY	0006	015534	BUFPT	0010	001354	CARS	0010	000000	CATFLG	0010	000001	CATNAM				
0010	000572	COATE	0010	I	000076	CLSNAM	0000	I	000516	CLSBUR	0010	I	000460	CLSSUB				
0000	I	000002	COL	0004	I	000001	COLADD	0004	000013	COLINC	0003	000356	COMERT	0010	R	001075	CON	
0010	001341	CROP	0010	001366	CRPKEY	0000	I	000514	ETCOL	0007	I	000021	CTFILE	0003	000400	CTSEDS		
0000	I	000523	CTMR	0011	000033	DATAPE	0003	000354	DATE	0011	000071	DATFIL	0003	000370	DEBUGF			
0010	I	001334	DESKEY	0003	I	000431	DESNO1	0003	I	000432	DESNO2	0010	001340	DESOTH	0010	001337	DESURI	
0003	000315	DFTCOD	0003	000320	DFTPT	0003	000316	DFTSYR	0011	000067	DRMDS	0004	000031	DRMRT				
0011	000066	DRUMAD	0005	000001	DSPYDT	0005	000003	DSPYTH	0003	I	000433	DUPNO	0000	I	000526	EDDMLI		
0010	000564	EMPTRS	0003	000401	EOFMIT	0011	000062	ERIPTP	0011	000063	ERPKEY	0003	000377	ESAM				
0003	000376	ESAM1	0010	001175	FETVCZ	0010	000575	FIELD2	0010	000600	FIELD3	0010	001334	FILTER				
0011	000075	FIZFIL	0011	000074	FIZKEY	0005	000000	FIZNAM	0011	000073	FIZURT	0003	I	000004	FDESC			
0010	001171	FLDKEY	0010	000574	FLDSVZ	0010	000577	FLDSV3	0004	I	000032	FOR	0011	I	000000	HEAD		
0003	000327	HED1	0003	000341	HED2	0010	001001	HIGH	0011	000057	HISFIL	0011	000060	HISKEY				
0006	011634	HOLARR	0007	003203	HOLECH	0004	000027	HOLLIN	0007	I	000032	HOLTST	0004	000030	HOLWRD			
0000	I	000517	I	0000	I	000532	IADDR	0004	000000	IBADDR	0000	I	000521	IC	0000	I	000527	IDR
0000	I	000515	IEND	0005	000010	ILINED	0003	000011	ILINIC	0005	000007	ILINST	0000	000701	INJPS			
0005	000006	INLINE	0005	000005	INPTS	0006	001750	IR	0000	I	000322	IS	0003	000013	ISANED			
0005	000014	ISARIC	0005	000012	ISARST	0007	003173	ISLDCN	0007	I	000022	ISLTST	0000	I	000520	ISRTY		
0000	I	000334	ISUB	0006	015556	JSTAT	0010	001237	KATRO	0003	000371	LIMNIT	0003	I	000325	LINLBI		
0003	I	000326	LIMBAN	0000	I	000535	LINE	0003	000373	LNPBAR	0007	000000	NAP	0010	001335	NAPFRT		

0011	000052	MAPTAP	0011	000064	MAPUNT	0010	001367	WFILE	0010	001333	WOCAT	0010	001236	WOCLSZ				
0004	I	000025	WOCOL	0004	I	000026	WOCOWR	0010	001174	WOFETZ	0011	000065	WOFILE	0010	001172	WOFLO2		
0010	001173	WOFLO3	0010	000534	WONAP	0010	I	001233	WOSUB2	0010	I	000536	WOSUB3	0010	001234	WOTRFD		
0007	000010	OTFILE	0003	000353	OTLINE	0003	000312	OTUNT	0011	000070	PAGC12	0010	000537	PCFDKY				
0010	000402	PCTID3	0007	000043	PERCNT	0010	000366	PLTKEY	0000	I	000531	PPT	0003	000372	PROSAM			
0000	I	000530	PT	0000	I	000523	PTS	0004	I	000043	REALIN	0000	I	000524	REAWR	0004	000000	SARSON
0011	000054	SAVTAP	0003	000402	SEDMAN	0010	001343	SITE	0003	000313	SKPLIN	0003	000314	SKPSAM				
0011	000072	STAFIL	0000	I	000000	STAT	0010	000563	STATKY	0000	I	000533	STDRLI	0010	000364	SUBCAT		
0010	I	000173	SUBMAN	0010	000270	SUBNO	0003	I	000000	SYN	0010	000677	SYRHTX	0004	I	000044	SYRPA6	
0003	000311	TAPOUT	0007	003214	THOLEM	0010	R	000603	THRES	0010	000562	THRKEY	0010	000565	THRSVA			
0007	000020	TINFIL	0004	I	000045	TINREA	0007	003213	TISLDM	0007	000042	TOTPCT	0010	001235	TOTVT2			
0010	000535	TOTVT3	0011	000061	TRFORM	0010	000561	TRNKEY	0003	000427	TRNNO	0010	000560	TSTKEY				
0003	000430	TSTNO	0004	013555	UNTFAS	0010	000576	VERTX2	0010	000601	VERTX3	0000	I	000536	WR			
0007	D	001073	ZCNI	0003	D	000321	ZONLIN	0003	D	000323	ZONSAM	0007	D	001113	ZPC			

```

00101 1* SUBROUTINE FIZMAP (DRMARR, S )
00103 2* IMPLICIT INTEGER (A-Z)
00104 3* INCLUDE FIZCOM.LIST
00105 3* IMPLICIT DOUBLE PRECISION (Z)
00106 3* REAL CON,THRES
00106 3*
00106 3* C
00106 3* C PARAMETERS
00106 3* C
00107 3* PARAMETER PAGCOL = 10, PAGSYN = 110, PARSAN = 1000, PARLIN = 1000
00110 3* PARAMETER PAGWRD = 19, PSCLAD = 61, ISUBCL = PSCLAD - 1
00111 3* PARAMETER PSEDS = 21, POUTLN = PSCLAD+6, PSYN = 2*POUTLN,
00111 3* PSANP5 = PARSAN + 5, PARFIL = 8
00111 3*
00111 3* C
00111 3* C COMMON BLOCKS
00111 3* C
00112 3* COMMON /FIZ/ SYN(POUTLN,2), CODE(POUTLN), TAPOUT, OTUNT, SKPLIN,
00112 3* SKPSAM, DFTCOD, DFTSYN(2), DFTPT, ZONLIN, ZONSAM,
00112 3* L1NLIN , L1NSAM, MED1(10), MED2(10), OTLIN, DATE(2),
00112 3* CORENT(10),DEBUGF,L1RHIT,PROSAM,LHPSAM,
00112 3* ASAR1, ASAM, ESAR1, ESAM, CTSEDS, EOFHIT,
00112 3* SEDNAM(PSEDS),TRNNO,TSTNO,DESNO1,DESNO2,DUPTNO
00113 3* COMMON /DRUN/ IBADDR, COLADD(PAGCOL), COLINC(PAGCOL), WCOL,
00113 3* WOCOWR, WOLLIN, WOLWRD, DRMRT,
00113 3* FOR(8), ALADO, REALIN, SYRPA6, TINREA
00114 3* COMMON /INFIZ/ FIZNAM, DSPYDT(2), DSPYTH, FLDESC, INPTS, INLINE,
00114 3* ILINST, ILINED, ILINIC, ISARST, ISAREO, ISANIC
00115 3* COMMON /CONSAR/ SARSON(PARSAN),IR(PSANP5,4),MOLARR(PARSAN,2),
00115 3* BUFPY, UNTFAS, JSTAT(2)
00116 3* COMMON/UNTFIZ/ MAP(PARFIL), OTFILE(PARFIL), TINFIL,
00116 3* CTFILE, ISLTST(PARFIL), MOLTST(PARFIL), TOTPCT,
00116 3* PERCNT(POUTLN,PARFIL),ZCNI(PARFIL),
00116 3* ZPC(POUTLN,PARFIL), ISLDM(PARFIL),
00116 3* NOLECH(PARFIL), TISLDM, THOLEM
00116 3* ,ATFILE
00117 3* END
00120 4* INCLUDE CMK10.LIST
00121 4* COMMON/DISPL/CATFLG,CATHAN(61),CLSNAM(61),SUBNAM(61),SUBNO(60),
00121 4* SUBCAT(60),CLSSUB(60),NONAP,TOTVT3,NOSUB3,
00121 4* PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,ENPTRS,THRSVA,
00121 4* PLTKEY,BNFLG,BNCOMB,BNFEAT,CDATE(2),
00121 4* FLDBV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,
00121 4* THRES(60),SYRHTX(66),HIGH(60),CON(60)
00121 4* ,FLDKEY,WOFLO2,WOFLO3,WOFETZ,FETVC2(30)
00121 4* ,WOSUB2,WOTRFD,TOTVT2,WOCLSZ
00121 4* ,KATHO(60),WOCAT,FILTER,WAPFHT

```

```

00121 4*      *      .DESKEY,DESUNI,DESOTM,CROP ,ACROP,AOTNER,ATOTAL
00121 4*      *      .SITE(4),ANALYS(3),CANS(10),CRPKEY
00121 4*      *      .BFILE
00122 4*      *      END
00123 5*      *      INCLUDE COMB6,LIST
00124 5*      *      COMMON/GLOBAL/HEAD(42),NAPTAP,DATAPE,SAVTAP,BNFILE,BNKEY,
00124 5*      *      MISFIL,MISKEY,TRFORM,ERIPTP,ERPKEY,NAPURT,NOFILE,
00124 5*      *      DRUMAD,DRUMOS,PAGSIZ,DATFIL,STAFIL
00124 5*      *      ,FIZUNT,FIZKEY,FIZFIL
00124 5*      *      END
00125 5*      *      DIMENSION DRUMR(NOCOMR,2,REALIN,2),STAT(2),COL(PAGSYN,3)
00126 6*
00126 7*      C
00126 8*      C.....
00126 9*      C      FORMAT STATEMENTS
00126 10*     C.....
00126 11*     C
00127 12*     100 FORMAT (/)
00130 13*     200 FORMAT (/,T42,'DISPLAY OF FIELD ',A6,' PRODUCED BY FILZON.',
00130 14*     *      /,T42,44(' '),/,/,T40,'FILTER = ',I1,T60,' ISLAND TEST = ',
00130 15*     *      /,T80,'HOLE TEST = ',I2,/,
00130 16*     *      /,T45,'CLASS',T77,'SUBCLASS',/,T42,'NO.',T50,
00130 17*     *      'NAME',T72,'NO.',T78,'NAME',T85,'SYMBOL')
00131 18*     270 FORMAT (T72,I2,T78,A6,T87,A1,/,1H*,T87,A1)
00132 19*     280 FORMAT (/,T42,I2,T50,A6,T72,I2,T78,A6,T87,A1,/,1H*,T87,A1)
00133 20*     285 FORMAT (/,/)
00134 21*     290 FORMAT (T9,110(1))
00135 22*     300 FORMAT (/,T8,'***ERROR*** FILZON. SUBROUTINE FIZMAP',/,T20,
00135 23*     *      /,T8,' IS AN ILLEGAL DRUM ADDRESS.',/)
00136 24*     350 FORMAT (/,T73,'SPECIAL SYMBOLS',/,T73,15(' '))
00137 25*     400 FORMAT (/,T72,I2,T78,A6,T87,A1,/,1H*,T87,A1,/)
00137 26*     C
00137 27*     C.....
00137 28*     C      INITIALIZATION
00137 29*     C.....
00140 30*     CALL SETNRG (68,0,68)
00141 31*     CTCOL = 0
00142 32*     IEND = 0
00143 33*     1000 CTCOL = CTCOL + 1
00144 34*     IF (CTCOL .GT. NOCOL) RETURN
00144 35*     IF (IEND .GE. LINSAN) RETURN
00146 36*     C
00146 37*     C.....
00146 38*     C      PRINT SYMBOLS USED FOR GREYMAP
00146 39*     C.....
00146 40*     C
00150 41*     WRITE (6,HEAD)
00152 42*     WRITE (6,200) FLDESC,ATFILE,ISLTST(CTFILE),MOLTST(CTFILE)
00160 43*     CLSNUN = 0
00161 44*     DO 1100 I = 1,NOSUBZ
00164 45*     IF (CLSNUN .EQ. CLSSUB(I)) GO TO 1020
00166 46*     CLSNUN = CLSNUN + 1
00167 47*     WRITE (6,280) CLSNUN, CLSNUN(CLSNUN), I, SUBNAR(I),
00167 48*     *      SYN(I,1), SYN(I,2)
00177 49*     GO TO 1100
00200 50*     1020 WRITE (6,270) I, SUBNAR(I), SYN(I,1), SYN(I,2)
00206 51*     1100 CONTINUE
00210 52*     WRITE (6,350)
00212 53*     WRITE (6,400) (I,SUBNAR(I),SYN(I,1),SYN(I,2),I=NOSUBZ,DUPNO)
00223 54*     IF (DESKEY .EQ. 0) GO TO 1110
00225 55*     DO 1105 I = DESR01,DESNO2
00230 56*     WRITE (6,100)
00232 57*     WRITE (6,270) I,SUBNAR(I),SYN(I,1),SYN(I,2)
00240 58*     1105 CONTINUE

```

```

00242 59. 1110 CONTINUE
00242 60. C
00242 61. C.....
00242 62. C PRINT COLUMN HEADING
00242 63. C.....
00242 64. C
00243 65. WRITE (6,285)
00245 66. IEND = SYRPAG * CTCOL
00246 67. ISTRY = (IEND - SYRPAG) + 1
00247 68. IF (IEND .GT. LINSAR) IEND = LINSAR
00251 69. IC = 0
00252 70. DO 1150 IS = ISTRY, IEND
00255 71. IC = IC + 1
00256 72. COL(IC,1) = IS/100
00257 73. COL(IC,2) = MOD(IS,100)/10
00260 74. COL(IC,3) = MOD(IS,10)
00261 75. 1150 CONTINUE
00263 76. PTS = (IEND - ISTRY) + 1
00264 77. DO 1160 IC = 1,3
00267 78. WRITE (6,290) (COL(IS,IC), IS = 1,PTS)
00275 79. 1160 CONTINUE
00277 80. WRITE (6,285)
00277 81. C
00277 82. C.....
00277 83. C READ DATA FROM DRUM AND PRINT
00277 84. C.....
00277 85. C
00301 86. REAWR = REALIN * NOCOMR + 2
00302 87. CTMR = 0
00303 88. EDDNLI = 0
00304 89. IDR = 1
00305 90. PT = 1
00306 91. PPT = 1
00307 92. STAT(1) = 0
00310 93. STAT(2) = 0
00311 94. IADDR = COLADD(CTCOL)
00312 95. IF ((CTMR-REAWR) .GT. ALADD) REAWR = ALADD-CTMR
00314 96. CALL RREAD (IADDR,DRHARR(1,1,1,PT),REAWR,STAT(PT))
00315 97. 1180 CONTINUE
00316 98. CTMR = CTMR + REAWR
00317 99. IF (IDR - TIMREA) 1190,1200,1300
00322 100. 1190 PPT = 3 - PPT
00323 101. IF ((CTMR-REAWR) .GT. ALADD) REAWR = ALADD -CTMR
00325 102. IADDR = COLADD(CTCOL) + CTMR
00326 103. CALL RREAD (IADDR,DRHARR(1,1,1,PPT),REAWR,STAT(PPT))
00327 104. 1200 IF (STAT(PT)-1) 1220,1200,3000
00332 105. 1220 STORLI = EDDNLI + 1
00333 106. EDDNLI = EDDNLI * REALIN
00334 107. IF (EDDNLI .GT. LIMLIN) EDDNLI = LIMLIN
00336 108. ISUB = STORLI - 1
00337 109. WRITE (6,FOR) (LINE,(DRHARR(WR,1,LINE-ISUB,PT),WR=1,NOCOMR),
00337 110. (DRHARR(WR,2,LINE-ISUB,PT),WR=1,NOCOMR),LINE = STORLI,EDDNLI)
00355 111. PT = 3 - PT
00356 112. IDR = IDR + 1
00357 113. GO TO 1180
00360 114. 1300 CONTINUE
00361 115. WRITE (6,285)
00363 116. DO 1310 IC = 1,3
00366 117. WRITE (6,290) (COL(IS,IC), IS = 1,PTS)
00374 118. 1310 CONTINUE
00376 119. GO TO 1000
00376 120. C.....
00376 121. C ERROR PRINT STATEMENT

```



00376 122• E.....
00377 123• 3000 CONTINUE
00400 124• WRITE (4,300)IADDR
00403 125• RETURN 2
00406 126• END

END OF COMPILATION: NO DIAGNOSTICS.
FIZNAP SYMBOLIC

08 APR 76 11:24:52 0 03033020 14 126 (DELETED)

ORIGINAL PAGE IS
OF POOR QUALITY

FOR, FIZSOW/SKAAA, FIZSOW, FIZSOW
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:24:53

08 APR 76

11:24:53

SUBROUTINE FIZSOW ENTRY POINT 000054

STORAGE USED: CODE(1) 000071; DATA(0) 000011; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 HERR4E
0004 HERR3E

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001 000007 10L 0000 000002 INJPS 0000 I 000001 K 0000 R 000000 RTEST

00101 1* SUBROUTINE FIZSOW (INNUM, OTRUM, RZOOM, ISKMR, LIMIT, S)
00102 2* IMPLICIT INTEGER (A-Z)
00104 3* REAL RTEST
00105 4* DOUBLE PRECISION RZOOM
00106 5* ISKMR = 0
00107 6* RTEST = RZOOM * FLOAT(INNUM)
00110 7* 10 K = OTRUM * ISKMR
00111 8* IF (K .GE. LIMIT) RETURN
00113 9* IF (K .GE. RTEST) RETURN 6
00115 10* ISKMR = ISKMR * 1
00116 11* GO TO 10
00117 12* END

END OF COMPILATION: NO DIAGNOSTICS.
FIZSOW SYMBOLIC

08 APR 76 11:24:56 0 03034536 14 12 (DELETED)

@ FOR, FIZTP/SKAAA,FIZTP,FIZTP
 UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A -(EXEC8 LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:24:56

08 APR 76

11:24:56

SUBROUTINE FIZTP ENTRY POINT 001735
 FIZTHD ENTRY POINT 001740
 FIZTLB ENTRY POINT 002000

STORAGE USED: CODE(1) 002012; DATA(0) 002144; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 CDATE
 0004 NDCODE
 0005 BYTRAN
 0006 NTRAN
 0007 CHERR
 0010 NIO1S
 0011 NIO2S
 0012 NWDUS
 0013 NERR2S
 0014 NERR3S

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	001136	110L	0001	001230	120L	0001	001330	130L	0001	001364	135L	0001	001367	140L
0001	001416	155L	0001	001437	160L	0001	001431	170L	0001	001457	180L	0001	001526	190L
0001	001533	200L	0001	000033	201G	0001	001536	201L	0001	001563	205L	0001	001566	210L
0001	000055	214G	0001	000067	222G	0001	000322	30L	0001	000753	374G	0001	000356	40L
0001	001005	406G	0001	001053	426G	0001	000445	45L	0000	001736	500F	0000	001756	510F
0000	002001	520F	0001	001371	524G	0000	002026	530F	0001	001502	565G	0001	000777	70L
0001	001031	76L	0001	001046	80L	0001	001104	90L	0000	001735	900F	0000	001715	ANC
0000	002071	ANCLNG	0000	001711	BEGBIT	0000	001710	BEGSKP	0000	001726	BGBIT	0000	001727	BGSKP
0000	000030	BITBEG	0000	002076	CONMRD	0000	001652	DECARR	0000	002072	DOI	0000	001660	EGC01C
0000	001657	FEAT	0000	001700	I	0000	001672	IA	0000	001705	IADR	0000	001663	IB
0000	001655	IBNONE	0000	000000	IBYTES	0000	001660	IC	0000	002055	ICMAN	0000	001675	ICOUNT
0000	001703	IDEM	0000	001730	IDERT	0000	001666	IE	0000	001664	IF	0000	001673	IFILE
0000	001713	INHOLD	0000	001712	IMOLE	0000	001662	II	0000	001665	IL	0000	001716	ILER
0000	001717	ILENGTH	0000	001661	IM	0000	001670	IN	0000	002115	INJPS	0000	001714	IPOS
0000	001330	IRAY	0000	001720	IREHD	0000	001676	ISTAT	0000	001667	IU	0000	001671	IV
0000	001677	IWD	0000	001701	IWD1	0000	001702	J	0000	001723	K	0000	001734	KA
0000	001674	KAUNIT	0000	002060	NBITS	0000	001725	NBYTES	0000	001656	NC	0000	002102	NCAR
0000	001733	NCLR	0000	002066	NCPR	0000	002057	NCS	0000	002100	NDSPR	0000	001706	NOBYTE
0000	001721	NOCHAN	0000	002074	NOSAN	0000	002067	NPRC	0000	001732	NR	0000	002070	NRPDS
0000	002103	NSAM	0000	002063	NVE	0000	001707	PACBIT	0000	000060	PACRAY	0000	001704	PACSKP
0000	001722	PREDIT	0000	002056	PROFLG	0000	002065	PRS2	0000	001731	ROM	0000	002073	SAMSTR
0000	002061	SVD	0000	002054	VARBL	0000	001724	WORD						

00100 1- C|.....CI
 00100 2- C FIZTP IS A MODIFICATION OF THE SUBROUTINE TAPMRT IN THE LARSYS
 00100 3- C SYSTEM.
 00100 4- C
 00100 5- C| CI
 00100 6- CI THE PURPOSE OF TAPMRT IS TO WRITE A DATA TAPE IN EITHER UNIVER- CI
 00100 7- CI SAL FORMAT OR LARSYS II FORMAT. THERE ARE TWO ENTRY POINTS TO CI

```

00100 8. CI THE SUBROUTINE -- WRTHDR AND WRTLIN. CI
00100 9. CI WRTHDR WRITES THE HEADER RECORD IN 32 BIT BYTES FOR LARSYS II CI
00100 10. CI AND 8 BIT BYTES FOR UNIVERSAL. ONE CALL TO WRTHDR MUST BE MADE CI
00100 11. CI FOR EACH REEL OF TAPE. THIS INFORMATION IS PACKED. CI
00100 12. CI
00100 13. C CALL WRTHDR(BC,FEAT,NSAMP,FORMAT,TRFORM) CI
00100 14. CI BC -- NO. OF CHANNELS TO BE WRITTEN FOR EACH DATA SET CI
00100 15. CI FEAT -- ARRAY CONTAINING CHANNELS TO BE WRITTEN CI
00100 16. CI NSAMP -- NO. OF SAMPLES PER CHANNEL CI
00100 17. CI FORMAT -- #1 FOR UNIVERSAL CI
00100 18. C TRFORM -- NO. OF TAPE OUTPUT UNIT CI
00100 19. CI #2 FOR LARSYS II CI
00100 20. CI WRTLIN WRITES THE DATA IN 8 BIT BYTES AND IS ALSO PACKED. A CI
00100 21. CI CALL TO THIS ROUTINE MUST BE MADE FOR EACH DATA SET TO BE WRIT- CI
00100 22. CI TEN CI
00100 23. CI
00100 24. CI CALL WRTLIN(DATA,LSTLIN) CI
00100 25. CI DATA -- ARRAY CONTAINING DATA TO BE WRITTEN CI
00100 26. CI LSTLIN -- = 0 FOR N-1 DATA SETS CI
00100 27. CI =-1 FOR LAST DATA SET CI
00100 28. CI
00100 29. CI ICHAN -- ACTIVE CHANNELS HAVE CORRESPONDING BIT POSITION CI
00100 30. CI TURNED ON CI
00100 31. CI
00100 32. CI PROFLG -- CI
00100 33. CI NCS -- NO. OF CHANNELS CI
00100 34. CI NBITS -- NO. OF BITS PER BYTE CI
00100 35. CI SVD -- START OF VIDEO DATA CI
00100 36. CI NVE -- SAME AS NSAMP CI
00100 37. CI PRSZ -- PHYSICAL RECORD SIZE IN BYTES CI
00100 38. CI NCPN -- NO. CHANNELS PER RECORD CI
00100 39. CI NRPC -- NO. PHYSICAL RECORDS PER CHANNEL CI
00100 40. CI NRPDS -- NO. OF RECORDS PER DATA SET CI
00100 41. CI ANCLNG -- LENGTH OF ANCILLARY BLOCK IN BYTES CI
00100 42. CI DOI -- DATA ORDER INDICATOR CI
00100 43. CI SAMSTR -- SAMPLE START CI
00100 44. CI COMWRD -- SIZE OF COMPUTER WORD IN BITS CI
00100 45. CI NDSPR -- NO. OF DATA SETS PER RECORD CI
00100 46. CI NCAR -- NO. OF CHANNELS ON ANCILLARY RECORD CI
00100 47. CI PACRAY -- DATA IS PACKED INTO THIS ARRAY AND THEN WRITTEN ON CI
00100 48. CI TAPE BY CALLING NTRAN CI
00100 49. CI ICOUNT -- RUNNING TOTAL OF NO. OF DATA SETS WRITTEN CI
00100 50. CI.....CI
00101 51. SUBROUTINE FIZTP
00103 52. IMPLICIT INTEGER (A-Z)
00103 53. C
00103 54. C ENTRY POINT FOR HEADER RECORD
00103 55. C
00104 56. C ENTRY FIZTHD (NSAMP,FORMAT,TRFORM,DORNYV,SEDKEY)
00104 57. C
00106 58. C DIMENSION VARBL(24), IBYTES(24), BITBEG(24), PACRAY(680),
00106 59. C ICHAN(2), IRAY(200)
00107 60. C DIMENSION EBCDIC(10)
00110 61. C DIMENSION DECARR(3)
00111 62. C DATA IBYTES/80,8, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 1, 2, 2,641,
00111 63. C 1,1024, 1,6, 2, 2/
00113 64. C DATA BITBEG/ 1,1,29,29,21,21,21,21,21,29,29,21,29,21,21, 1,
00113 65. C 29, 1,29,1,21,21/
00115 66. C DATA IBRNE/0010104000000/
00117 67. C DATA NC/1/, FEAT/1/
00122 68. C DATA EBCDIC(1)/0361/, EBCDIC(2)/0362/, EBCDIC(3)/0362/,
00122 69. C EBCDIC(4)/0364/, EBCDIC(5)/0365/, EBCDIC(6)/0366/,
00122 70. C EBCDIC(7)/0367/, EBCDIC(8)/0370/, EBCDIC(9)/0371/,

```

ORIGINAL PAGE IS
OF POOR QUALITY



```

00142 71* * EBCDIC(10)/0360/
00143 72* DATA IC/0303/, IM/0324/, II/0311/, IO/0100/
00144 73* DATA IF/0306/, IL/0323/, IE/0305/, IU/0344/, IV/0325/,
00145 74* * IV/0345/,IA/0301/
00146 75* DATA IFILE/0/
00147 76* EQUIVALENCE (VARBL(17),NOSAR),
00148 77* 1 (VARBL(2),ICHA), (VARBL(10),PRSZ),
00149 78* 2 (VARBL(3),PROFLG), (VARBL(11),NCP), (VARBL(19),COMWRD)
00150 79* 3, (VARBL(4),NCS), (VARBL(12),NPRC),
00151 80* 4 (VARBL(5),NBITS), (VARBL(13),NRPS), (VARBL(21),NDSPR),
00152 81* 5 (VARBL(6),SVD), (VARBL(14),ANCLNG),
00153 82* 6 (VARBL(15),DOT), (VARBL(23),NCR),
00154 83* 7 (VARBL(8),NVE), (VARBL(16),SARSTR), (VARBL(24),NSAR)
00155 84* DATA VARBL(1),VARBL(7),VARBL(9),VARBL(18),VARBL(20),VARBL(22)/6*-9
00156 85* +9/
00164 86* 900 FORMAT (3I2)
00165 87* KAUNIT = TRFORM
00166 88* ICOUNT=0
00167 89* SARSTR=1
00170 90* ISTAT = 0
00171 91* NCS = NC
00172 92* NVE = NSAMP
00173 93* NSAR = NSAMP
00174 94* NOSAR = NSAMP
00175 95* IFILE = IFILE + 1
00176 96* CALL CDATE (DDHYY)
00177 97* DECODE (6,900,DDHYY) DECARR
00205 98* IF (FORMAT .EQ. 1) GO TO 40
00205 99* C
00205 100* C PACKING HEADER RECORD IN LARSYS II FORMAT
00205 101* C
00207 102* IRAY(4) = 0
00210 103* IRAY(5) = NC
00211 104* IRAY(6) = NSAMP
00211 105* C
00211 106* C IRAY(51) THRU IRAY(NC+5) WILL BE FILLED WITH ONES
00211 107* C
00212 108* IND = 5*NC
00213 109* DO 10 I=1,IND
00216 110* IRAY(50+I) = I#ONE
00216 111* C
00216 112* C FILL REST OF 200 WORDS OF IRAY WITH ZEROS
00216 113* C
00220 114* IND1 = IND + 51
00221 115* DO 20 J=IND1,199
00224 116* IRAY(J+1) = 0
00226 117* IDER = 178
00227 118* PACSKP = IABS(0)
00230 119* IADR = 1
00231 120* NBITS = 32
00232 121* NOBYTE = 200
00233 122* PACBIT = 1
00234 123* BEGSKP = 4
00235 124* BEGBIT = 5
00236 125* CALL BYTRAN (BEGBIT,IRAY(IADR),BEGSKP,NOBYTE,NBITS,PACBIT,
00236 126* PACRAY(IADR),PACSKP)
00237 127* FLD(8,28,PACRAY(2)) = FLD(8,28,IFILE)
00240 128* INHLE = DECARR(1)/10
00241 129* IF (INHLD .EQ. 0) INHLD = 10
00243 130* FLD(28,8,PACRAY(14)) = FLD(28,8,EBCDIC(INHLD))
00244 131* INHLD = MOD(DECARR(1),10)
00245 132* IF (INHLD .EQ. 0) INHLD = 10
00247 133* FLD (0,8,PACRAY(15)) = FLD(28,8,EBCDIC(INHLE))

```



```

00250 134*      IMOLD = DECARR(2)/10
00251 135*      IF (IMOLD .EQ. 0) IMOLD = 10
00253 136*      FLD(8,8,PACRAY(15)) = FLD(28,8,EBCDIC(IMOLD))
00254 137*      IMOLD = MOD(DECARR(2),10)
00255 138*      IF (IMOLD .EQ. 0) IMOLD = 10
00257 139*      FLD(16,8,PACRAY(15)) = FLD(28,8,EBCDIC(IMOLD))
00260 140*      IMOLD = DECARR(3)/10
00261 141*      IF (IMOLD .EQ. 0) IMOLD = 10
00263 142*      FLD(24,8,PACRAY(15)) = FLD(28,8,EBCDIC(IMOLD))
00264 143*      IMOLD = MOD(DECARR(3),10)
00265 144*      IF (IMOLD .EQ. 0) IMOLD = 10
00267 145*      FLD(32,4,PACRAY(15)) = FLD(28,4,EBCDIC(IMOLD))
00270 146*      FLD(0,4,PACRAY(16)) = FLD(32,4,EBCDIC(IMOLD))
00271 147*      CALL STRANCAUNIT,1,IDEN, PACRAY,1STAT,22)
00272 148*      30 IF (1STAT .EQ. -1) GO TO 30
00274 149*      IF (1STAT .GE. 0) RETURN
00276 150*      IF (1STAT .LT. -1) WRITE(6,500)
00301 151*      IF (1STAT .LT. -1) CALL CHERR
00301 152*
C
C   PACKING HEADER RECORD IN UNIVERSAL FORMAT
C
00303 155*      40 FLD(0,1,ICHAN(1)) = 1
00304 156*      FLD(4,8,PACRAY(8)) = FLD(28,8,1B)
00305 157*      FLD(12,8,PACRAY(8)) = FLD(28,8,1B)
00306 158*      FLD(20,8,PACRAY(8)) = FLD(28,8,1B)
00307 159*      FLD(28,8,PACRAY(8)) = FLD(28,8,1B)
00310 160*      IF (SECKEY .EQ. 0) GO TO 45
00312 161*      FLD(4,8,PACRAY(8)) = FLD(28,8,1C)
00313 162*      FLD(12,8,PACRAY(8)) = FLD(28,8,1M)
00314 163*      FLD(20,8,PACRAY(8)) = FLD(28,8,1I)
00315 164*      45 CONTINUE
00316 165*      FLD(0,8,PACRAY(1)) = FLD(28,8,1U)
00317 166*      FLD(8,8,PACRAY(1)) = FLD(28,8,1B)
00320 167*      FLD(16,8,PACRAY(1)) = FLD(28,8,1I)
00321 168*      FLD(24,8,PACRAY(1)) = FLD(28,8,1V)
00322 169*      FLD(32,4,PACRAY(1)) = FLD(28,4,1A)
00323 170*      FLD(0,4,PACRAY(2)) = FLD(32,4,1A)
00324 171*      FLD(4,8,PACRAY(2)) = FLD(28,8,1C)
00325 172*      FLD(0,8,PACRAY(9)) = FLD(28,8,1F)
00326 173*      FLD(8,8,PACRAY(9)) = FLD(28,8,1I)
00327 174*      FLD(16,8,PACRAY(9)) = FLD(28,8,1L)
00330 175*      FLD(24,8,PACRAY(9)) = FLD(28,8,1E)
00331 176*      FLD(32,4,PACRAY(9)) = FLD(28,4,1B)
00332 177*      FLD(0,4,PACRAY(10)) = FLD(32,4,1B)
00333 178*      FLD(4,8,PACRAY(10)) = FLD(28,8,EBCDIC(IFILE))
00334 179*      FLD(12,8,PACRAY(14)) = DECARR(1)
00335 180*      FLD(20,8,PACRAY(14)) = DECARR(2)
00336 181*      FLD(28,8,PACRAY(14)) = DECARR(3)
00337 182*      IPOS = 1
00340 183*      50 FLD(IPOS,1,ICHAN(1)) = 1
00341 184*      IDEN = 680
00342 185*      NPROC = 0
00343 186*      PROF LG = 1
00344 187*      NBITS = 8
00345 188*      SVD = 1
00346 189*      PRS2 = 3060
00347 190*      ANCLNG = 70
00350 191*      ANC = ANCLNG * Z
00351 192*      OOI = 0
00352 193*      CONHRD = 36
00353 194*      BDSPR = 1
00354 195*      HCPR = 0
00355 196*      IF (NSAMP .GT. 2998) WRITE(6,510)

```

ORIGINAL PAGE IS OF POOR QUALITY.

```

00360 197* IF (NSAMP .GT. 2998) NSAMP = 2998
00362 198* ILEN = (NC*NSAMP) + ANC
00363 199* ILENTM = (NC*NSAMP + ANC )/ 3000
00364 200* IREND = MOD((NC*NSAMP + ANC ),3000)
00365 201* IF (ILENTM .EQ. 0) NCAR = NC
00367 202* IF (ILENTM .EQ. 0) NRPDS = 1
00371 203* IF (ILENTM .EQ. 0) GO TO 80
00373 204* DO 60 J=1,NC
00376 205* IF((NSAMP+J+ANC ) .GT. 3000) NCAR = J-1
00400 206* 60 IF((NSAMP+J+ANC ) .GT. 3000) GO TO 70
00403 207* 70 CONTINUE
00404 208* NCCHAN = NC - NCAR
00405 209* DO 75 J=1,NCCHAN
00410 210* IF(NSAMP+J .GT. 2998) NCPR = J-1
00412 211* 75 IF(NSAMP+J .GT. 2998) GO TO 76
00415 212* NCPR = NCCHAN
00416 213* 76 CONTINUE
00417 214* NRPDS = NCCHAN / NCPR + 1
00420 215* IF (MOD(NCCHAN,NCPR) .NE. 0) NRPDS = NRPDS + 1
00422 216* 80 BEGSKP = IABS(0)
00423 217* PACSKP = IABS(0)
00424 218* PREBIT = 0
00425 219* DO 100 K=1,24
00430 220* IF (VARBL(K) .EQ. -99) GO TO 90
00432 221* CALL BYTRAN(BITBEG(K),VARBL(K),BEGSKP,IBYTES(K),NBITS,PACBIT,PACRA
00432 222* V(WORD),PACSKP)
00433 223* 90 WORD = ((IBYTES(K) + PREBIT) + NBITS) / 36 + 1
00434 224* PACBIT = MOD(((IBYTES(K)+PREBIT)+NBITS),36) + 1
00435 225* 100 PREBIT = PREBIT + IBYTES(K)
00437 226* CALL NTRAN(KAUNIT,1,IDER, PACRAY,ISTAT,22)
00440 227* 110 IF (ISTAT .EQ. -1) GO TO 110
00442 228* IF (ISTAT .GE. 0) RETURN
00444 229* IF (ISTAT .LT. -1) WRITE(6,500)
00447 230* IF (ISTAT .LT. -1) CALL CHERR
00447 231* C
00451 232* ENTRY FIZTLN (IDATA, LSTLIN)
00451 233* C
00453 234* DIMENSION IDATA(NSAMP,NC)
00454 235* ICOUNT = ICOUNT + 1
00455 236* ANC = ANCLNG + 2
00456 237* IF (FORMAT .EQ. 1) GO TO 140
00456 238* C
00456 239* C WRITES PACKED DATA ON TAPE IN LARSYS II FORMAT
00456 240* C
00456 241* C PACKING ONE SET OF DATA INTO ONE RECORD
00456 242* C
00460 243* NBITS = 8
00461 244* ANCLNG = 4
00462 245* NBYTES = NSAMP*NC
00463 246* BGBIT = 36 - NBITS + 1
00464 247* BGSKP = 28
00465 248* FLD(IABS(0),16,PACRAY(1)) = ICOUNT
00466 249* FLD(17,16,PACRAY(1)) = 0
00467 250* 120 IF (ISTAT .EQ. -1) GO TO 120
00471 251* IF (ISTAT .LT. -1) WRITE(6,520)ICOUNT
00475 252* IF (ISTAT .LT. -1) CALL CHERR
00477 253* CALL BYTRAN(BGBIT,IDATA(1,1),BGSKP,NBYTES,NBITS,33,PACRAY(1),PACSK
00477 254* P)
00500 255* IDER = ((NSAMP*NC + ANCLNG)+8)/36
00501 256* IDER1 = MOD((NSAMP*NC + ANCLNG)+8,36)
00502 257* IF (IDER1 .NE. 0) IDER = IDER + 1
00504 258* CALL NTRAN(KAUNIT,1,IDER,PACRAY,ISTAT)
00505 259* IF (LSTLIN) 130,135,135

```



```

00510 260. 130 IF (ISTAT .EQ. -1) GO TO 130
00512 261. IF (ISTAT .LT. -1) WRITE (6,520) ICOUNT
00516 262. IF (ISTAT .LT. -1) CALL CMERR
00520 263. CALL MTRAN(KAUNIT,9,22)
00521 264. LSTLIN = 0
00522 265. 135 RETURN
00522 266. C
00522 267. C WRITE PACKED DATA ON TAPE IN UNIVERSAL FORMAT
00522 268. C
00523 269. 140 DO 150 I=1,16
00526 270. 150 PACRAY(I) = 0
00530 271. BCBIT = 36 - NBITS + 1
00531 272. IDEN = 640
00532 273. PACBIT = 1
00533 274. ROM = 1
00534 275. BCSKP = 28
00535 276. NR = NRPDS - 1
00536 277. IF(NCPR.EQ.0) GO TO 155
00540 278. NCLR = MOD((NC-NCAR),NCPR)
00541 279. 155 WORD=17
00541 280. C
00541 281. C PACKING ANCILLARY INFORMATION INTO PACRAY
00541 282. C
00542 283. FLD(IABS(0),16,PACRAY(1)) = 1
00543 284. FLD(20,16,PACRAY(16)) = ICOUNT
00543 285. C
00543 286. C DATA IS NOT PACKED WITH ANCILLARY RECORD
00543 287. C
00544 288. IF (NCAR .NE. 0) GO TO 160
00546 289. KA = 1
00547 290. GO TO 210
00547 291. C
00547 292. C ALL DATA IS PACKED ON ANCILLARY RECORD
00550 293. 160 IF (NCAR .NE. NC) GO TO 170
00552 294. NBYTES = IREMD - ANC
00553 295. KA = 2
00554 296. GO TO 210
00554 297. C
00554 298. C PART OF DATA IS PACKED ON ANCILLARY RECORD
00554 299. C
00555 300. 170 NBYTES = NCAR*NSAMP
00556 301. KA = 3
00557 302. GO TO 210
00557 303. C
00557 304. C DATA IS PACKED ON MORE THAN ONE RECORD
00557 305. C
00560 306. 180 ANC = 2
00561 307. WORD = 1
00562 308. PACBIT = 17
00563 309. KA = 4
00564 310. 185 DO 190 I=1,NR
00567 311. FLD(IABS(0),16,PACRAY(I)) = I-1
00570 312. NBYTES = NCPR * NSAMP
00571 313. IF (NCLR .NE. 0 .AND. I .EQ. NR) NBYTES = NCLR*NSAMP
00573 314. GO TO 210
00574 315. 190 CONTINUE
00576 316. 200 IF (LSTLIN) 201,205,205
00601 317. 201 IF (ISTAT .EQ. -1) GO TO 201
00603 318. IF (ISTAT .LT. -1) WRITE (6,530) ICOUNT
00607 319. CALL MTRAN (KAUNIT,9,22)
00610 320. LSTLIN = 0
00611 321. 205 RETURN
00612 322. 210 IF (ISTAT .EQ. -1) GO TO 210

```

```

00614 323. IF (ISTAT .LT. -1) WRITE(6,330)ICOUNT
00620 324. IF (ISTAT .LT. -1) CALL CMERR
00622 325. IF (NCRAR .NE. 0) CALL BYTRAR(8GBIT, IDATA(1,ROM), NCSKP, NBYTES, NBYTES
00622 326. , PACBIT, PACRAY(NORD), PACSKP)
00624 327. CALL BYTRAR(KAUNIT, 1, IDEF, PACRAY, ISTAT, 22)
00625 328. IF (KA .NE. 4) ROM = NCRAR * ROM
00627 329. IF (KA .EQ. 4) ROM = ROM * NCRP
00631 330. GO TO (180, 200, 180, 190), KA
00632 331. 500 FORMAT (/, T8, '***ERROR*** SUBROUTINE FIZTP', /, T20,
00632 332. , 'ABORTED BECAUSE OF BAD TAPE OR WRONG ASG CARD.')
00633 333. 510 FORMAT (/, T8, '***ERROR*** SUBROUTINE FIZTP', /, T20,
00633 334. , 'USER NSAMP IS TOO LARGE, THUS NSAMP WAS RESET ',
00633 335. , 'TO 2998.')
00634 336. 520 FORMAT (/, T8, '***ERROR*** SUBROUTINE FIZTP', /, T20,
00634 337. , 'RECORD ', I5, ' WAS BEING WRITTEN WHEN PROGRAM ABORTED ',
00634 338. , 'DUE TO BAD TAPE.')
00635 339. 530 FORMAT (/, T8, '***ERROR*** SUBROUTINE FIZTP', /, T20,
00635 340. , 'DATA SET ', I5, ' WAS BEING WRITTEN WHEN PROGRAM ABORTED ',
00635 341. , 'DUE TO BAD TAPE.')
00636 342. END

```

END OF COMPILATION:
FIZTP SYMBOLIC

NO DIAGNOSTICS.

08 APR 76 11:24:58 0 03034717 14 342 (DELETED)

0 FOR. FIZTST/SKAAA,FIZTST,FIZTST
 UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A -(EXEC8 LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:24:59

08 APR 76

11:24:59

SUBROUTINE FIZTST ENTRY POINT 000161

STORAGE USED: CODE(1) 000214; DATA(0) 000134; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 NERR35

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001	000035	131G	0001	000076	156G	0001	000112	171G	0001	000031	2000L	0001	000056	2100L	
0001	000060	2120L	0001	000063	2140L	0001	000074	3000L	0001	000106	4000L	0001	000137	4100L	
0000	I	000000	ADJSAN	0000	I	000010	CTSAN	0000	I	000107	I	0000	000112	INJPS	
0000	I	000110	PT	0000	I	000106	SAVPT					0000	I	000105	MAXCT

```

00101 1* SUBROUTINE FIZTST (TEST,CHANGE,CENTER,ARRHOL,ADJSA1,
00101 2* ADJSA2,ADJSA3,ADJSA4,ADJSA5,ADJSA6,
00101 3* ADJSA7,ADJSA8,DFTPT)
00103 4* IMPLICIT INTEGER (A-Y)
00104 5* DIMENSION ADJSAN(8), CTSAN(61)
00105 6* DATA CTSAN /&1 = 0/
00107 7* ADJSAN(1) = ADJSA1
00110 8* ADJSAN(2) = ADJSA2
00111 9* ADJSAN(3) = ADJSA3
00112 10* ADJSAN(4) = ADJSA4
00113 11* ADJSAN(5) = ADJSA5
00114 12* ADJSAN(6) = ADJSA6
00115 13* ADJSAN(7) = ADJSA7
00116 14* ADJSAN(8) = ADJSA8
00117 15* IF (TEST) 4000,1000,2000
00122 16* 1000 CONTINUE
00123 17* ARRHOL = CENTER
00124 18* RETURN
00125 19* 2000 CONTINUE
00126 20* MAXCT = 0
00127 21* SAVPT = 0
00130 22* DO 2100 I = 1, 8
00133 23* PT = ADJSAN(I)
00134 24* CTSAN(PT) = CTSAN(PT) + 1
00135 25* IF (CTSAN(PT) - MAXCT) 2100,2100,2020
00140 26* 2020 MAXCT = CTSAN(PT)
00141 27* SAVPT = PT
00142 28* IF (MAXCT .GE. TEST) GO TO 2140
00144 29* 2100 CONTINUE
00146 30* 2120 ARRHOL = CENTER
00147 31* GO TO 3000
00150 32* 2140 IF (SAVPT .EQ. CENTER) GO TO 2120
00152 33* ARRHOL = SAVPT
00153 34* CHANGE = CHANGE + 1
00154 35* 3000 CONTINUE
00155 36* DO 3020 I = 1, 8
  
```



```

00160 37*      PT = ADJSAR(I)
00161 38*      CTSAR(PT) = 0
00162 39*      3020 CONTINUE
00164 40*      RETURN
00165 41*      4000 CONTINUE
00166 42*      MAXCT = 0
00167 43*      SAVPT = 0
00170 44*      DO 4100 I = 1,8
00173 45*          PT = ADJSAR(I)
00174 46*          IF (PT-DFTPT) 4010,4100,4010
00177 47*      4010  CTSAR(PT) = CTSAR(PT) + 1
00200 48*          IF (CTSAR(PT)-MAXCT) 4100,4100,4020
00203 49*      4020  MAXCT = CTSAR(PT)
00204 50*          SAVPT = PT
00205 51*          IF (MAXCT .GE. IABS(TEST)) GO TO 2140
00207 52*      4100 CONTINUE
00211 53*      GO TO 2120
00212 54*      END

```

END OF COMPILATION;
FIZTST SYMBOLIC

NO DIAGNOSTICS.

08 APR 76 11:25:00 0 03040174 14 54 (DELETED)

FOR. FIZWRT/SKAAA, FIZWRT, FIZWRT
 UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A (EXECB LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:25:00

08 APR 76

11:25:00

SUBROUTINE FIZWRT ENTRY POINT 000532

STORAGE USED: CODE(1) 000576; DATA(0) 004100; BLANK COMMON(2) 000000

COMMON BLOCKS:

0003 DISPL 001370
 0004 FIZ 000434
 0005 DRUM 000046
 0006 INF121 000015
 0007 COMSAM 015560
 0010 UNTIFZ 003216

EXTERNAL REFERENCES (BLOCK, NAME)

0011 FIZTLN
 0012 FIZBOR
 0013 RWRITE
 0014 NWDUS
 0015 BIOZS
 0016 HERR48
 0017 HERR35

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0000	003762	100F	0001	000046	1200L	0001	000112	1260L	0001	000117	1300L	0001	000140	1330L
0001	000351	1364L	0001	000356	1365L	0001	000431	1385L	0001	000025	140G	0001	000053	150G
0001	000442	1500L	0001	000463	1600L	0001	000102	167G	0001	000465	1700L	0001	000135	207G
0001	000144	216G	0001	000226	230G	0001	000263	241G	0001	000306	245G	0001	000371	273G
0001	000470	3000L	0001	000430	302G	0001	000460	321G	0003	001362	ACROP	0005	000042	ALADD
0003	001351	ANALYS	0003	001363	ADTHER	0004	000375	ASAM	0004	000374	ASAM1	0010	003215	ATFILE
0003	001344	ATOTAL	0000	003760	BIT	0003	000570	BMC08B	0003	000571	BHFEAT	0003	000567	BNFLG
0007	015554	BUFFT	0003	001354	CAMS	0003	000000	CATFLG	0003	000001	CATRAM	0003	000572	COATE
0003	000076	CLSNAM	0003	000460	CLSSUB	0004	000206	CODE	0000	003752	COL	0005	000001	COLADD
0005	000013	COLINC	0000	003754	COLSAM	0004	000356	COMENT	0003	001075	CON	0003	001341	CROP
0003	001366	CRPKEY	0010	000021	CTFILE	0004	000400	CTSE09	0004	000354	DATE	0004	000370	DEBUGF
0003	001336	DESKEY	0004	000431	DESM01	0004	000432	DESM02	0003	001340	DES0TH	0003	001357	DESUN0
0004	000315	DFTCOD	0004	000320	DFTPT	0004	000316	DFTSYR	0000	003753	DRMLIN	0005	000031	DRHWRT
0000	000000	DRNST	0006	000001	DSPLYDT	0006	000003	DSPLYTH	0004	000433	DUPND	0003	000564	EMPTRS
0004	000401	EOFHIT	0004	000377	ESAM	0004	000376	ESAM1	0003	001175	FETV2	0003	000575	FIELDZ
0003	000600	FIELD3	0003	001334	FILTER	0006	000000	FIZNAM	0006	000004	FLDESC	0003	001371	FLDKEY
0003	000574	FLDSV2	0003	000577	FLDSV3	0005	000032	FOR	0004	000327	HED1	0004	000361	MED2
0003	001001	HIGH	0007	011634	HOLARR	0010	003203	HOLECH	0005	000027	HOLLIN	0010	000032	HOLTST
0005	000030	HOLWRD	0000	003744	I	0005	000000	IBADDR	0006	000010	ILINED	0006	000011	ILINIC
0006	000007	ILINST	0000	004046	INJPS	0006	000006	INLINE	0006	000005	INPTS	0007	001750	IN
0006	000013	ISAMED	0006	000014	ISAMIC	0006	000012	ISAM5T	0010	003173	ISLDCH	0010	000022	ISLTST
0000	003757	J	0007	015556	JSTAT	0000	003761	K	0003	001237	KATNO	0000	003746	L
0004	000371	LIMIT	0004	000325	LIMLIN	0004	000326	LIMSAR	0000	003741	LININC	0004	000376	LMPSAM
0000	003737	LSTLIN	0010	000000	NAP	0003	001335	NAPFMT	0003	001367	NFILE	0003	001333	NOCAT
0003	001236	NDCLS2	0005	000025	NDCOL	0005	000026	NDCOVR	0003	001174	NOFET2	0003	001172	NOFLD2
0003	001173	NOFLD3	0003	000554	NOFRAP	0003	001233	NOFSUB2	0003	000556	NOFSUB3	0003	001234	NOFRFD
0006	003750	OEOLIN	0006	003743	OEDSAM	0000	003747	OSTLIN	0000	003742	OSTSAM	0010	000010	OTFILE
0004	000353	OTLINE	0004	000312	OTUNT	0000	000012	OUTCOD	0000	001762	OUTSYR	0003	000557	PCFDRV
0003	000602	PCTID3	0010	000043	PERCNT	0003	000566	PLTKRY	0004	000372	PROSAM	0000	003745	PT



```

00124 7. DIMENSION OUTCOD(PARSAN), OUTSYN(PARSAN)
00125 8. DATA LSTLIN/0/, SAMINC/1/, LININC/1/
00131 9. DIMENSION TRNSAV(4,NOFLD2), TRNFLO(5,NOFLD2), TRNVER(2,TOTVT2),
00131 10. * TSTSAV(4,NOFLD3), TSTFLD(5,NOFLD3), TSTVER(2,TOTVT3),
00131 11. * DRUARR(NOCOMR,2,NOLIN,NOCOL)
00132 12. DATA OUTSYN/PARSAN*POUTLE/
00134 13. 100 FORMAT (/,T8,'***ERROR*** FIZJON, SUBROUTINE FIZHAT',T20,
00134 14. * I10,' IS AN ILLEGAL ',
00134 15. * 'ADDRESS FOR COLUMN PRINTOUT',I4,'.',/,T20,
00134 16. * 'THE STARTING DRUM ADDRESS FOR THE COLUMN EQUALS',I10,
00134 17. * '.',/,T20,'ERROR OCCURRED WHILE TRYING TO WRITE THE ',
00134 18. * I5,' RECORD ON THE DRUM FOR THIS COLUMN PRINTOUT.')
00135 19. OSTSAN = 0
00136 20. OEDSAN = 0
00137 21. DO 1200 I = 1,PROSAM
00142 22. IF (SAMSON(I) .LT. 1) GO TO 1200
00144 23. OSTSAN = OEDSAN + 1
00145 24. OEDSAN = OEDSAN + SAMSON(I)
00146 25. PT = MOLARR(I,BUFPT)
00147 26. DO 1180 L = OSTSAN, OEDSAN
00152 27. OUTCOD(L) = CODE(PT)
00153 28. OUTSYN(L) = PT
00154 29. PERCNT(PT,CTFILE) = PERCNT(PT,CTFILE) + LINSOM
00155 30. TOTPCT = TOTPCT + LINSOM
00156 31. 1180 CONTINUE
00160 32. 1200 CONTINUE
00162 33. OSTLIN = OYLINE + 1
00163 34. OEDLIN = OYLINE + LINSOM
00163 35. C
00163 36. C
00163 37. C.....
00163 38. C DO-LOOP TO WRITE DATA OUT ON TAPE
00163 39. C.....
00163 40. C
00164 41. IF (OTFILE(CTFILE) .EQ. 0) GO TO 1300
00166 42. DO 1240 I = 1, LINSOM
00171 43. OYLINE = OYLINE + 1
00172 44. IF (LIRLIN - OYLINE) 1240,1240,1240
00175 45. 1240 LSTLIN = -1
00176 46. 1240 CALL FIZTLN (OUTCOD,LSTLIN)
00177 47. 1280 CONTINUE
00201 48. 1300 CONTINUE
00202 49. IF (MAP(CTFILE) .EQ. 0) GO TO 1700
00204 50. IF (LINSOM .LT. 2) GO TO 1330
00206 51. DO 1320 I = 1,LINPSAN
00211 52. IR(I,1) = OUTSYN(I)
00212 53. 1320 CONTINUE
00212 54. C
00212 55. C.....
00212 56. C DO-LOOP TO WRITE DATA ON DRUM FOR GREYRAP
00212 57. C.....
00212 58. C
00214 59. 1330 CONTINUE
00215 60. DO 1400 I = OSTLIN, OEDLIN
00220 61. CALL FIZJON (TRNNO, I, OUTSYN, NOFLD2, TRNFLO, TRNSAV,
00220 62. * TRNVER, DFTPT, SAMSTR, LINSAN, SAMINC, LININC)
00221 63. IF (TSTKEY .EQ. 1) CALL FIZBOR (TSTNO, I, OUTSYN, NOFLD3,
00221 64. * TSTFLD, TSTSAV, TSTVER, DFTPT, SAMSTR, LINSAN, SAMINC, LININC)
00223 65. OSTSAN = 0
00224 66. OEDSAN = 0
00225 67. COL = 0
00226 68. DRNLIN = DRNLIN + 1
00227 69. DO 1360 COLSAN = 1, LINSAN, PAGESR

```



```

00232 70•      WORD = 0
00233 71•      OSTSAN = COLSAN
00234 72•      OEDSAN = OSTSAN + PAGSYN - 1
00235 73•      COL = COL + 1
00236 74•      IF (OEDSAN .GT. LINSAN) OEDSAN = LINSAN
00240 75•      DO 1340 MRSAN = OSTSAN, OEDSAN, 6
00243 76•      WORD = WORD + 1
00244 77•      DO 1335 J = 0, 3
00247 78•      PT = OUTSYN(MRSAN + J)
00250 79•      BIT = J + 6
00251 80•      FLD(BIT,6,DRUARR(WORD,1,DRMLIN,COL)) =
00251 81•      FLD(0,6,SYN(PT,1))
00252 82•      FLD(BIT,6,DRUARR(WORD,2,DRMLIN,COL)) =
00252 83•      FLD(0,6,SYN(PT,2))
00253 84•      1335 CONTINUE
00255 85•      1340 CONTINUE
00257 86•      1360 CONTINUE
00261 87•      IF (LIMIT) 1364,1364,1363
00264 88•      1363 IF (I .EQ. OEDLIN) GO TO 1365
00266 89•      1364 IF (DRMLIN .LT. MOLLIN) GO TO 1500
00270 90•      1365 DRMRRT = DRMRRT + 1
00271 91•      MOLMRD = DRMLIN + NOCOMR + 2
00272 92•      DO 1380 K = 1, NOCOL
00275 93•      CALL RWRITE (COLINC(K), DRUARR(1,1,K), MOLMRD, DRSTAT(I))
00276 94•      COLINC(K) = COLINC(K) + MOLMRD
00277 95•      1380 CONTINUE
00301 96•      DO 1390 K = 1, NOCOL
00304 97•      1385 IF (DRSTAT(K) .EQ. 1) GO TO 1385
00306 98•      IF (DRSTAT(K) .NE. 0) GO TO 3000
00310 99•      1390 CONTINUE
00312 100•      DRMLIN = 0
00313 101•      1500 CONTINUE
00314 102•      IF (OEDLIN .EQ. 1) GO TO 1600
00316 103•      IF (I .EQ. OEDLIN) GO TO 1600
00320 104•      DO 1520 COLSAN = 1,LINSAN
00323 105•      OUTSYN(COLSAN) = IR(COLSAN,11)
00324 106•      1520 CONTINUE
00326 107•      1600 CONTINUE
00330 108•      1700 CONTINUE
00331 109•      RETURN
00331 110•      C
00331 111•      C.....
00331 112•      C          FATAL ERROR
00331 113•      C.....
00331 114•      C
00332 115•      3000 CONTINUE
00333 116•      WRITE (6,100) COLINC(K), K, COLADD(K), DRMRRT
00341 117•      RETURN 3
00342 118•      END

```

END OF COMPILATION: NO DIAGNOSTICS. 08 APR 76 11:25:03 0 03040660 14 118 (DELETED)

ORIGINAL PAGE IS
OF POOR QUALITY



8 ASB, BYTRAN, BYTRAN
 ASSEMBLED BY UNIVAC 1108 EXEC II ASSEMBLER 2404 0008A
 THIS ASSEMBLY WAS DONE ON 08 APR 76 AT 11:23:04

08 APR 76

11:25: 3

000001
 000002
 000003
 000004
 000005
 000006
 000007
 000008
 000009
 000010
 000011
 000012
 000013
 000014
 000015
 000016
 000017
 000018
 000019
 000020
 000021
 000022
 000023
 000024
 000025
 000026
 000027
 000028
 000029
 000030
 000031
 000032
 000033
 000034
 000035
 000036
 000037
 000038
 000039
 000040
 000041
 000042
 000043
 000044
 000045
 000046
 000047
 000048
 000049
 000050
 000051
 000052
 000053
 000054
 000055
 000056
 000057
 000058
 000059

```

CALL BYTRAN(SBIT1, SWORD1, SBSKP, NBYTES, BYTESZ, DBIT1, DWORD1, DBSKP)
.
.
.
SBIT1 = BIT NO. OF FIRST BIT, FIRST BYTE, FIRST
SOURCE WORD( = 1, 2, 3, ..., 36 LEFT-TO-RIGHT )
SWORD1 = FIRST SOURCE WORD ADDRESS
SBSKP = NO. OF BITS TO SKIP BETWEEN BYTES, IN SOURCE
BYTE STRING
IF = 0, BYTES ARE A CONTIGUOUS BYTE STRING
IF .GT. 0, STARTING WITH BYTE 1 LEFT JUSTIFIED TO SBIT1,
NBYTES BYTES ARE UNPACKED EVERY SBSKP BITS,
WORD BOUNDARIES IGNORED IN UNPACKING SOURCE BYTE
STRING
NBYTES = TOTAL NO. OF BYTES TO BE UNPACKED FROM SOURCE STRING
AND PACKED INTO DESTINATION STRING
BYTESZ = BYTE SIZE ( NO. OF BITS/BYTE )
DBIT1 = BIT NO. FOR FIRST BIT OF FIRST BYTE, FIRST WORD OF
DESTINATION STRING ( = 1, 2, 3, ..., 36 LEFT-TO-RIGHT )
DWORD1 = FIRST DESTINATION WORD ADDRESS
DBSKP = NO. BITS TO SKIP BETWEEN BYTES PACKED INTO DESTINATION
BYTE STRING
IF = 0, BYTES ARE PACKED IN A CONTIGUOUS BYTE STRING
WITH FIRST BYTE LEFT-JUSTIFIED TO DBIT1 OF DWORD1
IF .GT. 0, DBSKP BITS ARE SKIPPED OVER BETWEEN BYTES
WHEN PACKING THE BYTES IN THE DESTINATION
STRING
NBYTES AND BYTESZ ARE COMMON TO SOURCE AND DESTINATION BYTE
STRINGS. OTHER VARIABLES INDEPENDENT OF EACH OTHER.

NOTE: THIS VERSION OF BYTRAN TESTS FOR
THE PACKING MODE TO BE RIGHT-JUSTIFICATION OF
THE BYTE IN THE DESTINATION WORDS. IF
RIGHT-JUSTIFY IS THE PACKING MODE, BYTRAN WILL
ZERO THE DESTINATION WORD PRIOR TO PACKING .

ALSO, ONLY ASCENDING WORD ADDRESSES ARE
ALLOWED IN THE SOURCE STRING AND
DESTINATION STRING .

EXECUTION TIME = .075 MILLISEC PER BYTE

```

8(1)

BBPR



```

000060 01 000000 74 04 00 00 0 000000
000061 000001 06 01 13 00 0 000001
000062 000002 71 12 04 00 0 000002
000063 000003 01 00 06 00 0 000004
000064 000004 06 00 01 00 0 000005
000065 000005 06 00 02 00 0 000006
000066 000006 14 00 01 00 0 000007
000067 000007 04 00 02 00 0 000010
000068 000010 12 00 04 13 1 000003
000069 000011 11 00 04 00 0 000012
000070 000012 27 00 01 00 0 000020
000071 000013 12 00 04 13 1 000004
000072 000014 03 01 04 00 0 000000
000073 000015 12 00 03 13 1 000012
000074 000016 16 00 03 00 0 000020
000075 000017 03 00 03 00 0 000022
000076
000077
000078 000020 12 00 05 13 1 000007
000079 000021 16 00 05 00 0 000020
000080 000022 03 00 05 00 0 000023
000081 000023 17 00 04 00 0 000013
000082 000024 01 00 04 00 0 000021
000083 000025 10 00 04 13 0 000001
000084 000026 17 00 04 00 0 000012
000085 000027 31 00 04 00 0 000013
000086 000030 12 00 04 13 1 000000
000087 000031 17 00 06 00 0 000012
000088 000032 16 00 06 00 0 000020
000089 000033 03 00 06 00 0 000014
000090 000034 10 00 04 13 0 000006
000091 000035 17 00 04 00 0 000012
000092 000036 31 00 04 00 0 000013
000093 000037 12 00 11 13 1 000005
000094 000040 17 00 11 00 0 000012
000095 000041 12 00 05 00 0 000025
000096 000042 16 00 11 00 0 000020
000097 000043 03 00 11 00 0 000015
000098 000044 05 00 00 00 0 000020
000099 000045 16 00 03 13 1 000004
000100 000046 52 00 05 00 0 000013
000101 000047 74 04 00 00 0 000057
000102 000050 05 00 00 00 0 000020
000103 000051 12 00 05 00 0 000023
000104 000052 34 00 04 00 0 000013
000105 000053 52 00 05 00 0 000011
000106 000054 74 04 00 00 0 000057
000107 000055 10 00 05 00 0 000012
000108 000056 03 00 05 00 0 000020
000109 000057 10 00 04 00 0 000017
000110 000060 73 12 04 00 0 000000
000111 000061 23 00 02 00 0 000020
000112 000062 05 00 00 00 0 000021
000113 000063 34 00 05 00 0 000013
000114 000064 14 00 05 00 0 000012
000115 000065 03 00 05 00 0 000024
000116 000066 05 00 00 00 0 000024
000117 000067 34 00 10 00 0 000013
000118 000070 14 00 10 00 0 000012
000119 000071 03 00 10 00 0 000025
000120 000072 50 00 00 00 0 000020
000121 000073 05 00 00 00 1 000025
000122

```

```

BOP 0,BAHE
BYTRAN SX,MZ X11,RETADD
DS A4,RSVAE
SA A6,RSVAE+2
SX X1,RSVAE+3
SX X2,RSVAE+4
SR R1,RSVAE+5
SR R2,RSVAE+6
LWA A4,*3,X11
AWA A4,ONE
LX X1,A4
LWA A4,*4,X11
SM,MZ A4,MSK
LWA A5,*2,X11
AWA A5,A4
SW A5,1SKP

CONT1 LWA A5,*7,X11
AWA A5,A4
SW A5,OSKP

CONT2 AWB A4,THRTY6
SA A4,SIZM36
L A4,1,X11
AWB A4,ONE
MSI A4,THRTY6
LWA A6,*0,X11
AWB A6,ONE
AWA A6,A4
SW A6,CURIN
LA A4,*6,X11
AWB A4,ONE
MSI A4,THRTY6
LWA A9,*5,X11
AWB A9,ONE
LA A5,A9
AWA A9,A4
SW A9,CUMOUT
S2 RTJSFY
AR A5,*4,X11
YE A5,THRTY6
J S=8
S2 A4
LA A5,OSKP
DI A4,THRTY6
TE A5,ZERO
J S=3
LA A5,ONE
SW A5,RTJSFY
LA A4,MSK
MSK LSSL A4,0
LR R2,A4
NOWMD S2 A5
DI A5,THRTY6
A A5,ONE
SW A5,SMD
S2 A8
DI A8,THRTY6
A A8,ONE
SW A8,DMD
TZ RTJSFY
S2 *DMD

```

```

. X1 = LOOP COUNTER = NO. BYTES - 1
. A4 = BYTE SIZE
. ....
. *** = INITIALIZATION FOR FORWARD ADDRESSING
. ....
. ....
. SIZM36 = BYTES2 - 36
. A4 = INITIAL SOURCE WORD ADDRESS
. SOURCE WORD ADDRESS - 1
. CUMULATIVE BIT COUNT (SOURCE)
. A6 = BIT NO. OF FIRST BIT, FIRST BYTE (SOURCE)
. A6 = BIT 1 ADDRESS (SOURCE) (0-35)
. A6 = CURIN = INITIAL CUMULATIVE BIT COUNT (SOURCE)
. A4 = INITIAL DEST. WORD ADDRESS
. ....
. A4 = CUMULATIVE BIT COUNT (DEST.)
. A9 = BIT NO. OF FIRST BIT, FIRST BYTE (DEST.)
. A9 = BIT 1 ADDRESS (DEST.) (0-35)
. ....
. A9 = CUMOUT = INITIAL CUMULATIVE BIT COUNT (DEST)
. ....
. A5 = DEST. BIT 1 ADDRESS * BYTE SIZE
. ....
. RIGHT-JUSTIFY FLAG SET
. ....
. INITIALIZE MASK REGISTER
. ....
. A5 = SMD = CURRENT SOURCE WORD ADDRESS
. ....
. A8 = DMD = CURRENT DEST. WORD ADDRESS
. ....
. ZERO-FILL THE DESTINATION WORD
  IF THE RIGHT-JUSTIFY FLAG IS SET

```

ORIGINAL PAGE IS
OF POOR QUALITY



000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185

000074 31 00 03 00 0 000013
000075 17 00 03 00 0 000014
000076 18 00 04 00 0 000013
000077 17 00 04 00 0 000021
000100 14 00 04 00 0 000021
000101 03 00 04 00 0 000026
000102 71 13 12 00 1 000024
000103 51 00 00 00 0 000020
000104 74 04 00 00 0 000112
000105 40 00 00 00 0 000020
000106 74 04 00 00 0 000111
000107 73 13 12 00 1 000026
000110 74 04 00 00 0 000112
000111 73 02 12 00 1 000026
000112 71 13 16 00 1 000023
000113 31 00 10 00 0 000013
000114 17 00 10 00 0 000015
000115 10 00 05 00 0 000013
000116 17 00 05 00 0 000024
000117 14 00 05 00 0 000021
000120 03 00 05 00 0 000026
000121 51 00 00 00 0 000021
000122 74 04 00 00 0 000130
000123 60 00 00 00 0 000021
000124 74 04 00 00 0 000127
000125 73 11 16 00 1 000026
000126 74 04 00 00 0 000130
000127 73 01 16 00 1 000026
000130 43 00 12 00 0 000032
000131 10 00 16 00 0 000027
000132 51 00 00 00 0 000021
000133 74 04 00 00 0 000141
000134 60 00 00 00 0 000021
000135 74 04 00 00 0 000140
000136 73 01 16 00 1 000026
000137 74 04 00 00 0 000141
000140 73 13 16 00 1 000026
000141 71 12 16 00 1 000023
000142 12 00 06 00 0 000014
000143 14 00 06 00 0 000022
000144 03 00 06 00 0 000014
000145 12 00 11 00 0 000015
000146 14 00 11 00 0 000023
000147 03 00 11 00 0 000015
000150 70 00 01 00 0 000062
000151 71 13 04 00 0 000002
000152 10 00 06 00 0 000004
000153 27 00 01 00 0 000005
000154 27 00 02 00 0 000006

RSI A5,THRTY6
ARM A5,CURIB
LA A4,THRTY6
ARM A4,A5
AA A4,SIZM36
SR A4,SMFCNT
DL A10,+SMD
TRZ A4
J S+6
TP A4
J S+3
SMFT1 LDSL A10,+SMFCNT
J S+2
JL A10,+SMFCNT
DL A14,+DMD
RSI A8,THRTY6
ARM A8,CUROUT
LA A5,THRTY6
ARM A5,A8
AA A5,SIZM36
SR A5,SMFCNT
TRZ A5
J S+6
TP A5
J S+3
SMFT2 LDSL A14,+SMFCNT
J S+2
DSC A14,+SMFCNT
RLU A10,A14
LA A14,A11
TRZ A5
J S+6
TP A5
J S+3
DSC A14,+SMFCNT
J S+2
LDSL A14,+SMFCNT
DB A14,+DMD
LRA A6,CURIB
A A6,ISKP
SR A6,CURIB
LRA A9,CUROUT
A A9,OSKP
SR A9,CUROUT
JGD X1,ROWMD
DL A4,RSAVE
LA A6,RSAVE+2
LX X1,RSAVE+3
LH X2,RSAVE+4

. A5 = SOURCE WD1 X 36 - CURIB
. A4 = 36 - (WD1 X 36 - CURIB)
= BIT1 ADDRESS,CURRENT SOURCE WORD
. A4 = IBIT1 + BYTES2 - 36 = SHIFT DIRECTION
AND MAGNITUDE
. A4 = SOURCE WORD SHIFT COUNT (MAGNITUDE)
. A10,A11 = SOURCE WD1,WD1+1
. SHIFT COUNT = 0
. SHIFT COUNT POSITIVE-RIGHT JUSTIFY THE
SOURCE BYTE IN A10
. SHIFT COUNT NEGATIVE- RIGHT JUSTIFY THE
SOURCE BYTE IN A10
. A14,A15 = DESTINATION WD1,WD1+1
. A8 = DESTINATION WD1 X 36 - CUROUT
. A5 = 36 - (WD1X36 - CUROUT)
= BIT 1 ADDRESS IN DEST. WORD
. A5 = OBIT1 + BYTES2 - 36
= SHIFT DIRECTION AND MAGNITUDE
. SMFCNT = DEST. WORD SHIFT (MAGNITUDE)
. SHIFT COUNT NON-ZERO - TEST FOR POSITIVE COUNT
. SHIFT COUNT POSITIVE- RIGHT JUSTIFY THE
DESTINATION BYTE POSITION IN A14
. SHIFT COUNT NEGATIVE- RIGHT JUSTIFY THE
DESTINATION BYTE POSITION IN A14
. SOURCE WORD,WITH BYTE RIGHT-ADJUSTED,MASKED
WITH DESTINATION WORD,BYTE POSITION
RIGHT-ADJUSTED, RESULT IN A11
. A14 RE-INITIALIZED WITH ADJUSTED DEST. WORD,
INCLUDING PACKED BYTE
. SHIFT WAS POSITIVE- RESTORE DEST. WD1,WD1+1
TO ORIGINAL BIT POSITIONS
. SHIFT WAS NEGATIVE- RESTORE DEST. WD1,WD1+1
TO ORIGINAL BIT POSITIONS
. DESTINATION WD1,WD1+1 RETURNED,
WITH PACKED BYTE
. CURIB UPDATED FOR BYTE SIZE AND OFFSET
. CUROUT UPDATED FOR BYTE SIZE AND OFFSET
. DECREMENT BYTE COUNT, GO GET NEXT SOURCE BYTE



```

000186 000155 23 00 01 00 0 000007
000187 000156 23 00 02 00 0 000010
000188 000157 27 00 13 00 0 000001
000189 000160 74 04 00 13 0 000011
000190
000191 00 000000 073631270623
000192 000001 000000000000
000193 000002
000194 000011 000000000000
000195 000012 000000000001
000196 000013 000000000044
000197 000014 000000000000
000198 000015 000000000000
000199 000016 000000000000
000200 000017 777777777777
000201 000020 000000000000
000202 000021 000000000000
000203 000022 000000000000
000204 000023 000000000000
000205 000024 000000000000
000206 000025 000000000000
000207 000026 000000000000
000208

```

```

BYTRAN SYMBOLIC
BYTRAN CODE RELOCATABLE

```

```

LR R1,RSAVE-5
LR R2,RSAVE-6
LX X11,RETADD
J 0,X11 . RETURN TO CALLING PROGRAM

S(0)
BARE * 'BYTRAN'
RETADD * 0
RSAVE RES 7
ZERO * 0
ONE * 1
THRTY6 * 36
CURIN * 0
CUROUT * 0
BYTESZ * 0
MASK * 077777777777
RTJ5FY * 0
SIZR36 * 0
ISKP * 0
OSKP * 0
SHD * 0
DMD * 0
SHFCBT * 0
END

```

```

14 JAN 76 16:24:19 0 01477702 14 208 (DELETED)
14 JAN 76 16:24:19 1 01505442 12 1 (DELETED)
0 01505456 14 16

```

@ FOR: FDLINT,FDLINT,FDLINT
 UNIVAC 1108 FORTRAN V EXEC 11 LEVEL 25A -(EXEC8 LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:25:32

08 APR 76

11:25:32

SUBROUTINE FDLINT ENTRY POINT 001001

STORAGE USED: CODE(1) 00107Z; DATA(0) 000077; BLANK COMMON(2) 000000

EXTERNAL REFERENCES (BLOCK, NAME)

0003 HERR3S

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001 000034 100L	0001 000474 1000L	0001 000024 113C	0001 000073 200L	0001 000320 2000L
0001 000706 29L	0001 000102 300L	0001 000424 3000L	0001 000430 3001L	0001 000664 317C
0001 000675 323C	0001 000720 340C	0001 000740 35L	0001 000133 400L	0001 000465 4000L
0001 000651 5L	0001 000137 500L	0001 000563 5000L	0001 000264 510L	0001 000330 600L
0001 000546 6000L	0001 000365 700L	0001 000327 7000L	0001 000641 8000L	0001 000615 9000L
0000 I 000014 I	0000 I 000016 IM1	0000 000044 INJPS	0000 I 000017 IP1	0000 I 000022 IP2
0000 I 000015 J	0000 I 000011 L	0000 I 000031 NN	0000 I 000012 N	0000 I 000033 NI
0000 I 000035 NJ	0000 I 000037 NN	0000 I 000013 NPTSE	0000 I 000032 NPTS1	0000 I 000034 NP1
0000 I 000036 NTERP	0000 R 000023 RL	0000 R 000030 RXX	0000 R 000024 RX1	0000 R 000025 RX2
0000 R 000026 RY1	0000 R 000027 RYZ	0000 I 000005 XNM1	0000 R 000020 XNP1	0000 I 000007 XNP2
0000 I 000004 XX	0000 I 000000 X1	0000 I 000001 X2	0000 I 000006 VNM1	0000 R 000021 VNP1
0000 I 000010 YNP2	0000 I 000002 Y1	0000 I 000003 Y2		

```

00101 1* SUBROUTINE FDLINT(FIELD,NPTS,FL,VLINE,NSAMP,JJ)
00101 2* CI
00101 3* CI THIS SUBROUTINE WILL RETURN THE PIXEL NUMBERS OF THOSE
00101 4* CI PIXELS ON A GIVEN LINE THAT ARE CONTAINED WITHIN THE
00101 5* CI BOUNDARIES OF A NONE-RECTANGULAR FIELD
00101 6* CI
00101 7* CI
00101 8* CI INPUT FIELD - NONE-RECTANGULAR FIELD TABLE
00101 9* CI ALL THE VERTICES MUST BE IN CLOCKWISE
00101 10* CI ORDER AND THE LAST VERTEX HAS TO BE EQUAL
00101 11* CI TO THE FIRST VERTEX FOR FIELD CLOSURE
00101 12* CI THE FIRST VERTEX MUST HAVE MINIMUM
00101 13* CI PIXEL VALUE
00101 14* CI NPTS - NO OF POINTS OF THE N-R FIELD
00101 15* CI VLINE - SCAN LINE NUMBER
00101 16* CI
00101 17* CI OUTPUT FL - ARRAY CONTAINING THE ORDERED PIXEL INTERCEPTS
00101 18* CI NSAMP - NO OF SAMPLES CONTAINED IN THE FIELD OF
00101 19* CI A GIVEN SCAN LINE
00101 20* CI JJ - THE LENGTH OF THE ARRAY FL
00101 21* CI
00103 22* CI PARAMETER MAXI=8
00104 23* CI DIMENSION FIELD(2,NPTS),FL(MAXI)
00105 24* CI INTEGER X1,X2,Y1,Y2,XX,FL,FIELD,VLINE
00106 25* CI INTEGER XNM1,YNM1,XNP2,YNP2
00107 26* CI IF(NPTS.EQ.2)GO TO 35
00107 27* CI ONE VERTEX FIELD
00107 28* CI L= VLINE
  
```



```
00112 29* DO 7 B = 1,MAXI
00115 30* 7 FL(B) = 0
00117 31* NPTSE = NPTS-1
00120 32* I = 1
00121 33* JJ = 0
00122 34* 100 X1=FIELD(1,I)
00123 35* Y1=FIELD(2,I)
00124 36* J = I+1
00125 37* X2=FIELD(1,J)
00126 38* Y2=FIELD(2,J)
00127 39* IF ( I .EQ. 1 ) GO TO 200
00131 40* IM1 = I-1
00132 41* XIM1=FIELD(1,IM1)
00133 42* YIM1=FIELD(2,IM1)
00134 43* GO TO 300
00135 44* 200 XIM1=FIELD(1,NPTSE)
00136 45* YIM1=FIELD(2,NPTSE)
00137 46* 300 IP1 = I+1
00140 47* XIP1=FIELD(1,IP1)
00141 48* YIP1=FIELD(2,IP1)
00142 49* IF ( I .EQ. NPTSE ) GO TO 400
00144 50* IP2 = I+2
00145 51* XIP2=FIELD(1,IP2)
00146 52* YIP2=FIELD(2,IP2)
00147 53* GO TO 500
00150 54* 400 XIP2=FIELD(1,2)
00151 55* YIP2=FIELD(2,2)
00152 56* 500 IF ( Y1 .EQ. Y2 ) GO TO 1000
00154 57* IF((L.EQ.Y2).AND.(Y2.EQ.YIP2)) GO TO 2000
00156 58* IF((L.EQ.Y1).AND.(Y1.EQ.YIM1)) GO TO 2000
00160 59* RL = L
00161 60* RX1 = X1
00162 61* RX2 = X2
00163 62* RY1 = Y1
00164 63* RY2 = Y2
00165 64* RXX = ((RL-RY1)*(RX2-RX1))/(RY2-RY1)+RX1
00166 65* XX = RXX*.5
00167 66* IF(Y1.LT.Y2) GO TO 510
00171 67* XX=RX
00172 68* IF((RX-XX).GT..5) XX=XX+1
00174 69* 510 CONTINUE
00175 70* IF (XX.GE. X1) .AND. (XX .LE. X2) ) GO TO 600
00177 71* IF (XX.LE. X1) .AND. (XX .GE. X2) ) GO TO 600
00201 72* 2000 I = I+1
00202 73* IF ( I .GT. NPTSE ) GO TO 5
00204 74* GO TO 100
00205 75* 600 IF(L.LE.Y1.AND.L.GE.Y2) GO TO 700
00207 76* IF(L.LE.Y2.AND.L.GE.Y1) GO TO 700
00211 77* GO TO 2000
00212 78* 700 JJ = JJ+1
00213 79* FL(JJ) = XX
00214 80* IF ( JJ .EQ. 1 ) GO TO 2000
00216 81* IF ( I .NE. NPTSE ) GO TO 3000
00220 82* IF(L.NE.Y2) GO TO 3000
00222 83* XIM1=X1
00223 84* YIM1=Y1
00224 85* X1=X2
00225 86* Y1=Y2
00226 87* X2=FIELD(1,2)
00227 88* Y2=FIELD(2,2)
00230 89* GO TO 3000
00231 90* 3000 IF ( L .NE. Y1 ) GO TO 2000
00233 91* 3001 IF (Y1.LT. YIM1) .AND. (Y1 .GT. Y2 ) GO TO 4000
```

```

00235  92*      IF (Y1 .GT. YHR1) .AND. (V1 .LT. VZ) GO TO 4000
00237  93*      GO TO 2000
00240  94*      4000 FL(JJ) = 0
00241  95*      JJ = JJ-1
00242  96*      GO TO 2000
00243  97*      1000 IF(L.NE.V1) GO TO 2000
00245  98*      IF(X1.GT.X2) GO TO 5000
00247  99*      IF(YHR1.LT.V1) GO TO 6000
00251  100*     IF ( YHP2 .GT. VZ ) GO TO 7000
00253  101*     JJ = JJ-1
00254  102*     FL(JJ) = X1
00255  103*     GO TO 2000
00256  104*     7000 JJ = JJ+1
00257  105*     FL(JJ) = X1
00260  106*     NN = JJ+1
00261  107*     FL(NN) = X2
00262  108*     JJ = NN
00263  109*     GO TO 2000
00264  110*     6000 IF ( YHP2 .LT. VZ ) GO TO 2000
00266  111*     JJ = JJ-1
00267  112*     FL(JJ) = X2
00270  113*     GO TO 2000
00271  114*     5000 IF ( YHR1 .LT. V1 ) GO TO 9000
00273  115*     IF ( YHP2 .GT. VZ ) GO TO 2000
00275  116*     JJ = JJ-1
00276  117*     FL(JJ) = X2
00277  118*     IF(NPTS.EQ.2)FL(JJ)=X1
00301  119*     GO TO 2000
00302  120*     9000 IF ( YHP2 .GT. VZ ) GO TO 8000
00304  121*     JJ = JJ-1
00305  122*     FL(JJ) = X1
00306  123*     NN = JJ+1
00307  124*     FL(NN) = X2
00310  125*     JJ = NN
00311  126*     GO TO 2000
00312  127*     8000 JJ = JJ+1
00313  128*     FL(JJ) = X1
00314  129*     GO TO 2000
00315  130*     5 NPTS1 = JJ-1
00316  131*     DO 29 NI = 1,NPTS1
00321  132*     NI = NI-1
00322  133*     DO 29 NJ = NPT, JJ
00325  134*     IF ( FL(NI) - FL(NJ) ) 29,29,28
00330  135*     28 NTEMP = FL(NI)
00331  136*     FL(NI) = FL(NJ)
00332  137*     FL(NJ) = NTEMP
00333  138*     29 CONTINUE
00336  139*     NSAMP = 0
00337  140*     DO 30 N = 1, JJ,2
00342  141*     NN = N+1
00343  142*     NSAMP = NSAMP+(FL(NN) -FL(N))-1
00344  143*     30 CONTINUE
00346  144*     RETURN
00347  145*     35 IF(VLINE.NE.FIELD(2,1))RETURN
00351  146*     FL(1)=FIELD(1,1)
00352  147*     FL(2)=FIELD(1,1)
00353  148*     NSAMP=1
00354  149*     JJ=2
00355  150*     RETURN
00356  151*     END

```

END OF COMPILATION; NO DIAGNOSTICS.



FOLINT
FOLINT

SYMBOLIC
SYMBOLIC

14 JAN 76	16:25:40	0	01720516	14	151	(DELETED)
08 APR 76	11:25:34	0	03044511	14	151	(DELETED)

ORIGINAL PAGE IS
OF POOR QUALITY



FOR, MRTFLD, MRTFLD, MRTFLD
 UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A (EXECB LEVEL E12010010A)
 THIS COMPILATION WAS DONE ON 08 APR 76 AT 11:25:55

08 APR 76

11:25:55

SUBROUTINE MRTFLD ENTRY POINT 000350

STORAGE USED: CODE(1) 000372; DATA(0) 000200; BLANK COMMON(2) 000000

COMMON BLOCKS:

0003 GLOBAL 000076

EXTERNAL REFERENCES (BLOCK, NAME)

0004 HNDUS
 0005 HI02S
 0006 HI01S
 0007 HERR3S

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0000	000013	100F	0001	000072	133G	0001	000145	154G	0000	000021	200F	0001	000227	201G
0001	000260	220G	0001	000315	235G	0000	000027	230F	0000	000050	300F	0000	000066	400F
0001	000176	5L	0000	000100	500F	0001	000266	6L	0000	000115	650F	0001	000323	7L
0000	000123	700F	0000	000133	705F	0000	000140	710F	0003	000055	0MFILE	0003	000056	0MKEY
0003	000053	DATAPE	0003	000071	DATFIL	0003	000067	DRHMDS	0003	000066	DRUMAD	0003	000062	ERIPTP
0003	000063	ERPKEY	0003	000075	FIZFIL	0003	000074	FIZKEY	0003	000073	FIZUNT	0000	I 000011	FLDNAM
0003	I 000000	HEAD	0003	000057	HISFIL	0003	000060	HISKEY	0000	I 000002	I	0000	I 000001	IB
0000	I 000007	IC	0000	I 000006	IE	0000	000146	INJPS	0000	I 000010	IS	0000	I 000012	J
0000	I 000000	LPRN	0003	000052	MAPTAP	0003	000064	MAPUNT	0003	000065	NOFILE	0000	I 000004	NO
0000	I 000005	NR	0000	I 000003	NV	0003	000070	PAGSIZ	0003	000054	SAVTAP	0003	000072	STAFIL
0003	000061	TRFORM												

```

00101 1* SUBROUTINE MRTFLD(FLDSAV, VERTEX, NOFLD, KEY, CLSNAM, SUBNAM)
00103 2* DIMENSION CLSNAM(1), SUBNAM(1)
00104 3* IMPLICIT INTEGER(A-Z)
00104 4* C*
00104 5* C* THIS SUBROUTINE PRINTS SAVED TRAINING OR TEST FIELDS
00104 6* C*
00105 7* DIMENSION FLDSAV(4, NOFLD), VERTEX(2, 1)
00106 8* DATA LPRN/'/'
00110 9* INCLUDE COMBK6.LIST
00111 9* COMMON/GLOBAL/HEAD(62), MAPTAP, DATAPE, SAVTAP, 0MFILE, 0MKEY,
00111 9* HISFIL, HISKEY, TRFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE,
00111 9* DRUMAD, DRHMDS, PAGSIZ, DATFIL, STAFIL,
00111 9* FIZUNT, FIZKEY, FIZFIL FILZON
00112 9* END
00113 10* IS=1
00114 11* WRITE(6, HEAD)
00116 12* IF(KEY.EQ.1)WRITE(6,100)
00121 13* IF(KEY.EQ.2)WRITE(6,200)
00124 14* IF(KEY.EQ.3)WRITE(6,300)
00127 15* IF(KEY.EQ.3)WRITE(6,250)
00132 16* DO 10 I=1, NOFLD
00135 17* BV=FLDSAV(4, I)

```



```

00136 18*      NO=NV-1
00137 19*      NR=NO-5
00140 20*      IF(NO.GT.5)NO=5
00142 21*      IE=IB-NO-1
00143 22*      IC=FLDSAV(2,1)
00144 23*      IS=FLDSAV(3,1)
00145 24*      FLDNAM=FLDSAV(1,1)
00146 25*      IF(KEY.NE.3)GO TO 5
00150 26*      WRITE(6,700)I,FLDNAM,(LPRN,VERTEX(1,J),VERTEX(2,J),J=1B,1E)
00162 27*      IF(IS.EQ.1)WRITE(6,705)
00165 28*      IF(IS.EQ.2)WRITE(6,710)
00170 29*      GO TO 6
00171 30*      5 CONTINUE
00172 31*      IF(IS.EQ.0)WRITE(6,400)I,FLDNAM,IC,CLSNAM(IC),
00172 32*      *      (LPRN,VERTEX(1,J),VERTEX(2,J),J=1B,1E)
00207 33*      IF(IS.NE.0)WRITE(6,500)I,FLDNAM,IC,CLSNAM(IC),IS,SUBNAM(IS),
00207 34*      *      (LPRN,VERTEX(1,J),VERTEX(2,J),J=1B,1E)
00226 35*      6 CONTINUE
00227 36*      IF(NR.LE.0)GO TO 7
00231 37*      IB=IE+1
00232 38*      IE=IB-NR-1
00233 39*      WRITE(6,650)(LPRN,VERTEX(1,J),VERTEX(2,J),J=1B,1E)
00243 40*      7 CONTINUE
00244 41*      IB=IE+2
00245 42*      10 CONTINUE
00247 43*      RETURN
00250 44*      100 FORMAT(/// 45X,' SAVED TRAINING FIELDS'//)
00251 45*      200 FORMAT(/// 46X,' INPUT TEST FIELDS'//)
00252 46*      250 FORMAT(///45X,' DESIGNATED FIELDS'///T10,' FIELD',T40,' DESIGNATED',
00252 47*      *      T80,' VERTICES (SAMPLE,LINE)')
00253 48*      300 FORMAT(1X,T10,' FIELD',T36,' CLASS',T47,' SUBCLASS',T80,' VERTICES (S
00253 49*      *      AMPLE,LINE)')
00254 50*      400 FORMAT(T15,I3,T20,A6,T30,I3,T35,A6,T65,5(A1,I4,' ',I4,' '),1X)
00255 51*      500 FORMAT(T15,I3,T20,A6,T30,I3,T35,A6,T45,I3,T50,A6,T65,
00255 52*      *      5(A1,I4,' ',I4,' '),1X)
00256 53*      650 FORMAT(1X,T65,5(A1,I4,' ',I4,' '),1X)
00257 54*      700 FORMAT(T15,I3,T20,A6,T65,5(A1,I4,' ',I4,' '),1X)
00260 55*      705 FORMAT(1M,T40,' UNIDENTIFIABLE')
00261 56*      710 FORMAT(1M,T40,' OTHER')
00262 57*      END

```

```

END OF COMPILATION:      NO DIAGNOSTICS.
WRTFLD      SYMBOLIC
WRTFLD      SYMBOLIC

```

```

14 JAN 76 16:39:06 0 02531676 14 57 (DELETED)
08 APR 76 11:25:56 0 03042063 14 57 (DELETED)

```

ORIGINAL PAGE IS
OF POOR QUALITY

PROGRAM PRINTER OUTPUT



09 APR 76

9:51:15

STARTING ADDRESS 019000

CORE LIMITS 019000 042334 100000 154647 163772 163777

FILZOM/R0995A
0 100000-116626
1 019000-023572

RSTOP8/ALECS
1 023573-023604

NIEN8/ALECS
0 116627-116627
1 023605-024112
2 116630-116724

NFAT8/ALECS
1 024113-025050
2 116725-116741

NFTV8/AL22
1 025051-025073

NCNVT8/ALECS
1 025074-025320
2 116742-117030

NOTIN8/ALECS
1 025321-025770
2 117031-117074

FPACK8/CODE
1 025771-026034

DEPTH /*****
0 117075-117102

NERR8/ALECS
0 117103-117272
1 026035-026477

NIQIN8/ALECS
1 026500-026551
2 117273-117323

NRWMD8/ALECS
1 026552-026644

NTAB8 /CODE

ORIGINAL PAGE IS
OF POOR QUALITY

D-72



117329-117972

CHERR/ CODE

0 117473-117473

FIZMAP/ KAAA

0 117474-120424

1 026645-027552

SETMRG/ CODE

1 027553-027566

FIZWRT/ KAAA

0 120425-124524

1 027567-030364

FIZBOR/ KAAA

0 124525-124620

1 030365-031045

FOLINT/ CODE

0 124621-124717

1 031046-032137

FIZFLT/ KAAA

0 124720-124755

1 032140-032474

FIZTST/ KAAA

0 124756-125111

1 032475-032710

FIZTP / KAAA

0 125112-127255

1 032711-034722

NTRAM /ALECH

0 127256-127256

1 034723-036215

2 127257-127475

TLABL/ CODE

0 127476-127505

TSCRW/ CODE

0 127506-127560

THRU/ CODE

0 127561-127661

TSWAP/ CODE

0 127662-127767

TINTL/ CODE

0 127770-130071

BYTRAM/ CODE

0 130072-130120

1 036216-036376

MSC401/ CODE

0 130121-130142



1008 /CODE
1 036377-036433

WOUTS /RLECA
0 130143-130147
1 036434-037436
2 130150-130155

WDCVA /RLECA
0 130166-130352

RANDIO /CODE
0 130353-130420
1 037437-037626

FIZSOW /RKAAR
0 130421-130431
1 037627-037717

WATFLD /CODE
0 130432-130631
1 037720-040311

FIZWRF /RKAAR
0 130632-131062
1 040312-040742

WINTPS /RLECA
0 131063-131065
1 040743-042060
2 131066-131123

WINTVA /RLECA
1 042061-042216
2 131124-131153

YDATE /CODE
0 131154-131163

EDATEA /CODE
1 042217-042253

F505FL /CODE
0 131164-131236

WU /CODE
0 131237-131441

CLOCK /CODE
0 131442-131444
1 042254-042334

DISPL /*****
0 131445-133034

GLOBAL /*****
0 133035-133132

UNTYPE /*****
0 133133-134350

COMSON /*****
0 134351-154130

ORIGINAL PAGE IS
OF POOR QUALITY



FIZI/*****
0 159131-159195

BRUM /*****
0 159196-159213

FIZ /*****
0 159214-159297

END OF ALLOCATION 1103 0039A 09099

**ORIGINAL PAGE IS
OF POOR QUALITY**



CHECK LEAD CARDS FOR PROGRAM FILZOM

0000000011111111222222223333333344444444555555556666666677777777
 123456789012345678901234567890123456789012345678901234567890123456789

***GOOD *** INPUT CARD	COMMENT	CRI NO. 37 TEST AREA
***GOOD *** INPUT CARD	OUTPUT	L
VARIABLES CHECKED LATER	SKIP	200 160
VARIABLES CHECKED LATER	ZOOM	1 1
***GOOD *** INPUT CARD	LIMIT	250 220
***GOOD *** INPUT CARD	FILTER	7
***GOOD *** INPUT CARD	ISLAND	-3 -3 -3 -3 -3 -3 -3
***GOOD *** INPUT CARD	HOLE	-3 -3 -3 -3 -3 -3 -3

PRECEDING PAGE BLANK NOT FILMED

ORIGINAL PAGE IS
 OF POOR QUALITY



LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

09 APR 76

CMI NO. 37 TEST AREA

SUMMARY OF INPUT TAPE

INFIZ INPUT ON LOGICAL UNIT 11.

INFIZ FILE 1 GENERATED ON 06 APR 76 AT 16:07:33 WILL BE PROCESSED.

SUMMARY OF INPUT SCANNER DATA:

USER-INPUT THRESHOLDING HAS BEEN APPLIED.

PROCESSOR DISPLAY DID NOT APPLY SPATIAL FILTERING.

DATA CLASSIFIED FOR HAPTAP FILE 1 ON 10 FEB 76 .

NUMBER OF INPUT LINES = 512.

NUMBER OF INPUT SAMPLES = 512.

NUMBER OF CLASSES = 9.

NUMBER OF SUBCLASSES = 9.

NUMBER OF TRAINING FIELDS = 9.

NUMBER OF TEST OR DESIGNATE FIELDS = 0.

LYNNON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

09 APR 76

CME NO. 37 TEST AREA

PROCESSING FOR FILZOM

TAPE OUTPUT ON LOGICAL UNIT 19 -- LARSYS II FORMAT.

ON INPUT TAPE BEFORE PROCESSING BEGINS 200 LINES WILL BE SKIPPED.

ON EACH LINE OF THE INPUT TAPE 160 SAMPLES WILL BE SKIPPED.

ZOOM LINE FACTOR = 1.000

ZOOM SAMPLE FACTOR = 1.000

LINE LIMIT = 250

SAMPLE LIMIT = 220

INPUT LINES 481 TO 512 WILL NOT BE PROCESSED DUE TO THE ZOOM LINE FACTOR.

ON EACH INPUT LINE, SAMPLES 381 TO 512 WILL NOT BE PROCESSED DUE TO THE ZOOM SAMPLE FACTOR.

ISLAND TEST = -3 -3 -3 -3 -3 -3 -3

UNCLASSIFIED PIXEL TEST (HOLE) = -3 -3 -3 -3 -3 -3 -3

DEFAULT CODE FOR UNCLASSIFIED PIXELS = 120

DEFAULT SYMBOL FOR UNCLASSIFIED PIXELS = 0



LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

09 APR 76

CAL NO. 37 TEST AREA

TRAINING FIELDS FOR FILZON OUTPUT

FIELD	CLASS	SUBCLASS	CODE	SYMBOL	VERTICES (SAMPLE,LINE)
1 +1.5 C	1 +1.5	1 +1.5	152	1	(4, 240) (12, 240) (12, 259) (4, 259)
2 +0.75C	2 +0.75	2 +0.75	140	9	(39, 240) (92, 240) (92, 259) (39, 259)
3 +0.25	3 +0.25	3 +0.25	132	0	(65, 240) (72, 240) (72, 259) (65, 259)
4 -0.25	4 -0.25	4 -0.25	124	0	(96, 240) (103, 240) (103, 259) (96, 259)
5 -0.75	5 -0.75	5 -0.75	116	0	(126, 240) (133, 240) (133, 259) (126, 259)
6 -1.5	6 -1.5	6 -1.5	109	8	(157, 240) (164, 240) (164, 259) (157, 259)
7 -2.5	7 -2.5	7 -2.5	88	2	(188, 240) (195, 240) (195, 259) (188, 259)
8 -3.5	8 -3.5	8 -3.5	72	0	(217, 240) (224, 240) (224, 259) (217, 259)
9 WATER	9 WATER	9 WATER	176		(314, 240) (335, 240) (335, 259) (314, 259)

LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

09 APR 76

CMI NO. 37 TEST AREA

SUMMARY OF OUTPUT FOR FILZOM

OUTPUT FILE 1 ON UNIT 19 -- LARSYS II FORMAT.

NUMBER OF RECORDS THAT HAVE BEEN READ FROM LOGICAL UNIT 21 = 251.
NUMBER OF UNCLASSIFIED PIXELS THAT HAVE BEEN RECLASSIFIED DUE TO THE HOLE TEST (-3) FOR FILTER 7 =
TOTAL NUMBER OF HOLE PIXELS THAT HAVE BEEN RECLASSIFIED 1057.
NUMBER OF ISLAND PIXELS THAT HAVE BEEN RECLASSIFIED DUE TO THE ISLAND(-3) FOR FILTER 7 = 731.
TOTAL NUMBER OF ISLAND PIXELS THAT HAVE BEEN RECLASSIFIED = 6509.
TOTAL NUMBER OF ZOOMED OUTPUT POINTS = 55000.

PERCENT TABLE

SUBCLASS NAME	TOTAL POINTS	PERCENTAGES
+1.5	666	1.21
+0.75	1060	1.93
+0.25	1327	2.41
-0.25	3205	5.97
-0.75	15909	28.93
-1.5	8560	15.56
-2.5	8367	15.21
-3.5	2991	5.53
WATER	13332	24.24
HOLE	3	.01
UNIDEN	0	.00
OTHER	0	.00

FOR FILTER 7 THE AVERAGE CMI FOR ALL LAND PIXELS IS -1.306.

ORIGINAL PAGE IS
OF POOR QUALITY



4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

ORIGINAL PAGE IS
OF POOR QUALITY



47
48
49
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129



4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
10 #####
11 #####
12 #####
13 #####
14 #####
15 #####
16 #####
17 #####
18 #####
19 #####
20 #####
21 #####
22 #####
23 #####
24 #####
25 #####
26 #####
27 #####
28 #####
29 #####
30 #####
31 #####
32 #####
33 #####
34 #####
35 #####
36 #####
37 #####
38 #####
39 #####
40 #####
41 #####
42 #####
43 #####
44 #####
45 #####
46 #####
47 #####
48 #####
49 #####
50 #####
51 #####
52 #####
53 #####
54 #####
55 #####
56 #####
57 #####
58 #####
59 #####
60 #####
61 #####
62 #####
63 #####
64 #####
65 #####
66 #####

ORIGINAL PAGE IS
OF POOR QUALITY



47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129



130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192

ORIGINAL PAGE IS
OF POOR QUALITY



123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

TIME FOR FILZOR = 9.752 MINUTES.

ORIGINAL PAGE IS
OF POOR QUALITY

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

APPENDIX E
EXCERPT FROM DISPLAY PROCESSOR, USER DOCUMENTATION
EOD-LARSYS WITH MODIFICATIONS

APPENDIX E

EXCERPT FROM DISPLAY PROCESSOR, USER DOCUMENTATION EOD-LARSYS WITH MODIFICATIONS

The DISPLAY processor reads a classification tape (MAPTAP) output by CLASSIFY and performs the following functions:

- Provides a line-printer map of each classified field on the MAPTAP. Training and test fields within the classified image are outline.
- Produces classification summaries for each classified field. These summaries provide a break-down on the number of pixels classified and the number of pixels thresholded from each subclass, class, and category.
- Optionally produces an intensive test site classification summary for one zone type versus all 'other' zone types. The user specified zone may be a category, class or subclass.
- Allows the user to 'designate' fields to be excluded from the classification summaries. Fields may be designated 'unidentifiable' or 'other'. Pixels within the unidentifiable fields are counted and are not considered in the classification summaries. Pixels within the designated 'other' fields are counted as a separate zone type regardless of how they were classified. In the intensive test site report these pixels are included in the 'other' category.

All pixels within the 'designated' areas are printed with the pound, '#', symbol.

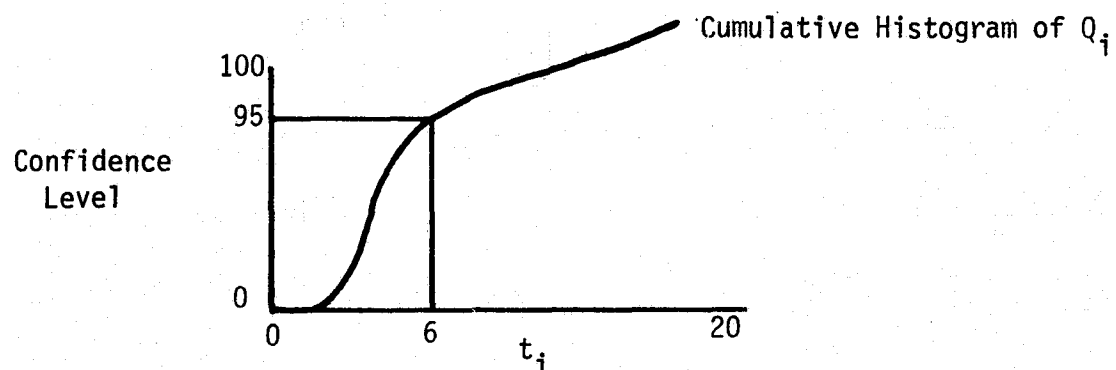
See section 12.4.4³ for sample input of 'designated' fields.

- If thresholding is requested, assigns a pixel to the threshold class if:

$$Q_i > t_i$$

where: Q_i = the value of the quadratic form $(x-\mu_i) K_i^{-1} (x-\mu_i)^T$
as computed by CLASSIFY for subclass i
 μ_i = mean vector for subclass i
 K_i = covariance matrix for subclass i
 t_i = threshold value for subclass i

- Allows the threshold value t_i to be determined in one of three ways.
 1. User input. The user inputs the exact threshold value. See control cards (THRESHOLD) and (OPTION THRESHOLD VALUE).
 2. Chi square. The user inputs confidence levels for each subclass on the THRESHOLD card and includes the OPTION CHI-SQUARE. The program obtains the chi-square threshold value from an internal chi-square functional routine.
 3. Empirical. The user inputs confidence levels for each subclass on the THRESHOLD card and includes the OPTION EMPIRICAL card. The program determines the empirical distribution function for each sub-class from the cumulative histogram of the quadratic form for correctly classified pixels in the ground truth areas, (i.e. training or test fields).



From this example curve the user input of a 95% confidence level for subclass i would result in a threshold value of 6.0.

- Produces plots of the empirical distribution function when the OPTION PLOT is exercised.
- Optionally performs a four nearest neighbor spatial filtering on the classified image. This algorithm takes into consideration that, in many instances, a pixel is most likely to be like its nearest neighbors. When the option is exercised (via the OPTION FILTER control card), the four nearest neighbors of each pixel are examined. If all the neighbors are classified the same and the pixel in question is classified differently, then it is assumed that the pixel was classified incorrectly and its classification is changed.

Line 1	C
2	C X C
3	C

The pixel classified as X will be changed to C. See reference 9³ for more information on this algorithm.

- Optionally outputs the classified image on to tape in either LARSYS-2 or Universal format. (This option is exercised via the FORMAT control card.)
- Provides classification performance summaries for ground truth areas within the classified image. The following six performance summaries are available to the user. 'Fields' in these reports are training fields used in STAT or ISOCLS and passed to DISPLAY on the MAPTAP, or they may be 'test' fields input directly to DISPLAY (see section 12.4.4)³
 1. Field by subclass
 2. Field by class
 3. Field by category
 4. Class by subclass
 5. Class by class
 6. Class by category

12.1 INPUT FILES

The only input file required for DISPLAY is the MAPTAP (Section 4.4)³ file output by CLASSIFY. This file must be assigned to logical unit B (Fortran unit 2). If DISPLAY is executed back-to-back with CLASSIFY the file may be assigned to FAstrand. However, if DISPLAY is to be executed later, the file must be assigned to tape and saved when CLASSIFY is executed.

12.2 OUTPUT FILES

The DISPLAY processor will optionally generate a tape of the classified image for display on the DAS. The control card "FORMAT" (Section 12.4.3)³ allows the user to exercise this option. When requested the tape should be assigned to a 9-track tape drive for compatibility with the DAS tape drives. The tape assignment must be made to logical unit N (Fortran unit 16).

The modified DISPLAY processor will optionally generate a tape containing the classified image and pertinent information about each zone. The option is exercised via the FIZKEY control card. The tape assignment must be made to logical unit I (FORTRAN unit 11). If more than one execution of DISPLAY is desired, multiple files can be output on the INFIZ tape.

12.3 SCRATCH FILES

The random access drum file is used as a scratch file in DISPLAY. No assignment is necessary.

12.4 CARD INPUT

All formats referred to in this section are defined in section 3.

12.4.1 PROCESSOR CARD

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
\$DISPLAY	FILE = N (Default N = 1)	Designates to the monitor routine which processor is to be executed. Parameter value N is the file number on the MAPTAP file to be displayed.

12.4.2 SPECIAL SYSTEM DECKS

None

12.4.3 CONTROL CARDS

<u>Keyword</u>	<u>Parameter Value</u>	<u>Function</u>
OPTIONS	STATS (DEFAULT: No statistics printed)	Prints statistics for sub- classes used in the previous CLASSIFY run. These statis- tics are saved on the MAPTAP.
OPTIONS	PCT (DEFAULT: Perfor- mance summary printed for classes only).	Print a performance summary on a per field basis as well as per class basis. This is a performance summary for ground truth fields, (i.e. training or test fields within the classified image).
OPTIONS	NOMAP (DEFAULT: Map printed)	Do not print a map of the data. Only a performance summary is printed.
OPTIONS	FILTER (DEFAULT: Spatial filtering is not performed)	Perform four nearest neighbor spatial filtering on the classified image.
OPTION	FIZKEY (DEFAULT: No INFIZ tape is generated).	An INFIZ tape is generated.

<u>Keyword</u>	<u>Parameter Value</u>	<u>Function</u>
THRESHOLD	$T_1, T_2 \dots$ (DEFAULT: No thresholding)	<p>Use the threshold values $T_1, T_2 \dots$ for subclasses 1, 2, ... respectively. Thresholds must be positive floating point numbers. One value must be specified for each subclass on the MAPTAP. Thresholds may also be specified in the following format:</p> <p>$N_1 * T_1, N_2 * T_2, \dots$ where N_1 and N_2 are integers which specify how many consecutive times the corresponding thresholds should be used.</p> <p>For the CHI-SQUARE option and the EMPIRICAL option the numbers input on this card are the confidence levels. (i.e. $T_1 = .99$ means that the user wants to maintain 99% or reject 1%).</p> <p>For the THRESHOLD VALUE option, the numbers input on this card are the actual threshold values to be used for thresholding (i.e. $T_1 = 10.02$ means that the threshold value for subclass 1 is 10.02).</p>

<u>Keyword</u>	<u>Parameter Value</u>	<u>Function</u>
OPTIONS	CHI SQUARE (DEFAULT: None) *See footnote	Compute thresholds from the chi-square distribution, using the confidence levels input on the THRESHOLD control card.
OPTIONS	PLOT	Plot the empirical distribution functions obtained from the cumulative histograms of the quadratic form for each subclass.
OPTIONS	EMPIRICAL (DEFAULT: None) *See footnote	Compute the empirical threshold values using the percentages input on the THRESHOLD control card.
OPTION	THRESHOLD VALUES (DEFAULT: None) *See footnote	Use the numbers input on the THRESHOLD control card for the actual threshold value.
SYMBOLS	$S_1, S_2 \dots$ (DEFAULT: 1,2, -9 A,B,C,D, ... Z, 1,2,3,4)	Assign symbols S_1, S_2, \dots to subclasses 1,2, ... respectively.

*If the THRESHOLD control card is input, one of the three options (CHI-SQUARE, EMPIRICAL, or THRESHOLD VALUES) should be input also. If the OPTION card is omitted and the THRESHOLD card is input, chi-square is assumed.

<u>Keyword</u>	<u>Parameter Value</u>	<u>Function</u>
ACREAGE	TOTAL = X, CROP = Y, OTHER = Z (DEFAULT: None)	The total acreage in the intensive test site is X. Acreage of the crop named on the 'CROP' control card is Y. Acreage of all other crop types in the intensive test site is Z. X, Y, and Z are floating point numbers. This input is meaningful only if the 'CROP' control card is input also.
SITE	Any 24 characters (DEFAULT: Blanks)	Name of the intensive test site. This is used in printing the heading for the intensive test site report.
ANALYST	Any 18 characters (DEFAULT: Blanks)	Name of the data analyst. This is printed in the heading for the intensive test site report.
PROCEDURE	Any 60 characters (DEFAULT: Blanks)	Procedure used in classification of the intensive test site. This is printed in the heading for the intensive test site report.

<u>Keyword</u>	<u>Parameter Value</u>	<u>Function</u>
FORMAT	NAME (DEFAULT: No output classifica- tion map tape is generated by DISPLAY)	NAME = UNIVERSAL The output classification tape is generated in the Universal format. NAME = LARSYS The output classification tape is generated in the LARSYS-2 format
HED1	Any 60 characters beginning in column 11. (DEFAULT: LYNDON B. JOHNSON SPACE CENTER)	Replace first header line with the 60 characters in the parameter field.
HED2	Any 60 characters beginning in column 11. (DEFAULT: HOUSTON, TEXAS)	Replace second header line with the 60 characters in the parameter field.
COMMENT	Any 60 characters beginning in column 11. (DEFAULT: blanks)	Print a comment line using the 60 characters found in the parameter field.
DATE	Any 12 characters beginning in column 11. (DEFAULT: current date)	Replace the date in the standard heading with the 12 characters in the para- meter field.

<u>Keyword</u>	<u>Parameter Value</u>	<u>Function</u>
END	(none)	Signals the end of the control cards
\$END*	(none)	Signals the end of all control cards for the current processing function.