

## N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM  
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT  
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED  
IN THE INTEREST OF MAKING AVAILABLE AS MUCH  
INFORMATION AS POSSIBLE

"Made available under NASA sponsorship  
in the interest of early and wide dis-  
semination of Earth Resources Survey  
Program information and without liability  
for any use made thereof."

8.0 - 10174  
JSC-13143  
NA  
165626

"AS-BUILT" DESIGN SPECIFICATION  
FOR  
EOD-LARSYS PROCEDURE 1  
Job Order 71-695  
(TIRF 77-0008)

(E80-10174) AS-BUILT DESIGN SPECIFICATION  
FOR EOD-LARSYS PROCEDURE 1 (Lockheed  
Electronics Co.) 307 p HC A14/MF A01

N80-28775

CSCL 05B

Unclas  
G3/43 00174

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

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION  
SCIENCE AND APPLICATIONS DIRECTORATE



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

October 1977

LEC-11293

JSC-13143

"AS-BUILT" DESIGN SPECIFICATION

FCR

EOD-LARSYS PROCEDURE 1

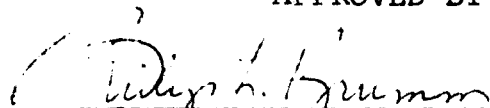
Job Order 71-695

(TIRF 77-0008)

Prepared By

B. E. Wills  
C. T. Gardner  
P. J. Aucoin, Jr.

APPROVED BY

  
Philip L. Krumm, Supervisor  
Applications Software Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division

Science and Applications Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

LYNDON B. JOHNSON SPACE CENTER

HOUSTON, TEXAS

October 1977

LEC-11293

## CONTENTS

Section	Page
1. SCOPE . . . . .	1-1
2. APPLICABLE DOCUMENTS . . . . .	2-1
3. SYSTEM DESCRIPTION . . . . .	3-1
3.1 <u>HARDWARE DESCRIPTION</u> . . . . .	3-3
3.2 <u>SOFTWARE DESCRIPTION</u> . . . . .	3-4
3.2.1 SOFTWARE COMPONENT NO. 1 (DOTDATA) . . . . .	3-4
3.2.2 SOFTWARE COMPONENT NO. 2 (LABEL) . . . . .	3-11
3.2.3 SOFTWARE COMPONENT NO. 3 (ISOCLS) . . . . .	3-24
3.2.4 SOFTWARE COMPONENT NO. 4 (SELECT) . . . . .	3-30
3.2.5 SOFTWARE COMPONENT NO. 5 (CLASSIFY) . . . . .	3-35
3.2.6 SOFTWARE COMPONENT NO. 6 (DISFLAY) . . . . .	3-42
3.2.7 SOFTWARE COMPONENT NO. 7 (UTILITY ROUTINES) . . . . .	3-51
4. OPERATION . . . . .	4-1
5. TEST PROCEDURE . . . . .	5-1
Appendices	Page
A SUBPROGRAM DESCRIPTIONS	
A.1 <u>SOFTWARE SUBPROGRAM NO. 1 (FLDTYP)</u> . . . . .	A-1
A.2 <u>SOFTWARE SUBPROGRAM NO. 2 (DOTDAT)</u> . . . . .	A-4
A.3 <u>SOFTWARE SUBPROGRAM NO. 3 (DOTS)</u> . . . . .	A-7
A.4 <u>SOFTWARE SUBPROGRAM NO. 4 (SET13)</u> . . . . .	A-12
B SUBPROGRAM DESCRIPTIONS	
B.1 <u>SOFTWARE SUBPROGRAM NO. 1 (ALLKIN)</u> . . . . .	B-1

Appendices

	Page
B.2 <u>SOFTWARE SUBPROGRAM NO. 2 (CLSMAP)</u> . . . . .	B-7
B.3 <u>SOFTWARE SUBPROGRAM NO. 3 (DOTDST)</u> . . . . .	B-13
B.4 <u>SOFTWARE SUBPROGRAM NO. 4 (KNEAR)</u> . . . . .	B-17
B.5 <u>SOFTWARE SUBPROGRAM NO. 5 (LABDOT)</u> . . . . .	B-22
B.6 <u>SOFTWARE SUBPROGRAM NO. 6 (LABEL)</u> . . . . .	B-25
B.7 <u>SOFTWARE SUBPROGRAM NO. 7 (LABLR)</u> . . . . .	B-28
B.8 <u>SOFTWARE SUBPROGRAM NO. 8 (MANORD)</u> . . . . .	B-33
B.9 <u>SOFTWARE SUBPROGRAM NO. 9 (MIXMAP)</u> . . . . .	B-36
B.10 <u>SOFTWARE SUBPROGRAM NO. 10 (SET14)</u> . . . . .	B-39
B.11 <u>SOFTWARE SUBPROGRAM NO. 11 (MAPHND)</u> . . . . .	B-41
B.12 <u>SOFTWARE SUBPROGRAM NO. 12 (ASCEND)</u> . . . . .	B-44
B.13 <u>SOFTWARE SUBPROGRAM NO. 13 (CRDSCN)</u> . . . . .	B-47
B.14 <u>SOFTWARE SUBPROGRAM NO. 14 (FILERD)</u> . . . . .	B-51
B.15 <u>SOFTWARE SUBPROGRAM NO. 15 (STOMAP)</u> . . . . .	B-54
B.16 <u>SOFTWARE SUBPROGRAM NO. 16 (DSPTAP)</u> . . . . .	B-57
B.17 <u>SOFTWARE SUBPROGRAM NO. 17 (NAMSTA)</u> . . . . .	B-61
B.18 <u>SOFTWARE SUBPROGRAM NO. 18 (REORDER)</u> . . . . .	B-64
B.19 <u>SOFTWARE SUBPROGRAM NO. 19 (LABMAN)</u> . . . . .	B-67
B.20 <u>SOFTWARE SUBPROGRAM NO. 20 (SUNFCS)</u> . . . . .	B-71
B.21 <u>SOFTWARE SUBPROGRAM NO. 21 (CLRKEY)</u> . . . . .	B-74
B.22 <u>SOFTWARE SUBPROGRAM NO. 22 (CNDMAP)</u> . . . . .	B-77

C SUBPROGRAM DESCRIPTIONS

C.1 <u>SOFTWARE SUBPROGRAM NO. 1 (ISODAT)</u> . . . . .	C-1
C.2 <u>SOFTWARE SUBPROGRAM NO. 2 (RDDATA)</u> . . . . .	C-4

Appendices	Page
C.3 <u>SOFTWARE SUBPROGRAM NO. 3 (SETUP7)</u> . . . . .	C-7
C.4 <u>SOFTWARE SUBPROGRAM NO. 4 (SUNFAC)</u> . . . . .	C-10
C.5 <u>SOFTWARE SUBPROGRAM NO. 5 (PSPLIT)</u> . . . . .	C-11
D    SUBPROGRAM DESCRIPTIONS	
D.1 <u>SOFTWARE SUBPROGRAM NO. 1 (SETUP4)</u> . . . . .	D-1
D.2 <u>SOFTWARE SUBPROGRAM NO. 2 (WGCHK)</u> . . . . .	D-5
D.3 <u>SOFTWARE SUBPROGRAM NO. 3 (INTWGT)</u> . . . . .	D-10
E    SUBPROGRAM DESCRIPTIONS	
E.1 <u>SOFTWARE SUBPROGRAM NO. 1 (REDIF2)</u> . . . . .	E-2
E.2 <u>SOFTWARE SUBPROGRAM NO. 2 (SETUP2)</u> . . . . .	E-5
F    SUBPROGRAM DESCRIPTIONS	
F.1 <u>SOFTWARE SUBPROGRAM NO. 1 (SETUP3)</u> . . . . .	F-1
F.2 <u>SOFTWARE SUBPROGRAM NO. 2 (DTCHK)</u> . . . . .	F-9
F.3 <u>SOFTWARE SUBPROGRAM NO. 3 (REDIF3)</u> . . . . .	F-12
F.4 <u>SOFTWARE SUBPROGRAM NO. 4 (DSPLY2)</u> . . . . .	F-17
F.5 <u>SOFTWARE SUBPROGRAM NO. 5 (DOTPCT)</u> . . . . .	F-25
F.6 <u>SOFTWARE SUBPROGRAM NO. 6 (DOTSUM)</u> . . . . .	F-30
F.7 <u>SOFTWARE SUBPROGRAM NO. 7 (DSPLAY)</u> . . . . .	F-40
F.8 <u>SOFTWARE SUBPROGRAM NO. 8 (TAPLAB)</u> . . . . .	F-43
G    SUBPROGRAM DESCRIPTIONS	
G.1 <u>SOFTWARE SUBPROGRAM NO. 1 (TAPERD)</u> . . . . .	G-1
G.2 <u>SOFTWARE SUBPROGRAM NO. 2 (RDDOTS)</u> . . . . .	G-4
G.3 <u>SOFTWARE SUBPROGRAM NO. 3 (WRDOT)</u> . . . . .	G-11
H    DOT DATA FORMAT. . . . .	H-1
I    VERIFICATION RUN (Partial Printout) . . . . .	I-1

## 1. SCOPE

This document contains the "As-Built" Design Specifications for implementing Procedure 1 in EOD-LARSYS. Software for two new processors were written and routines for four existing processors were modified.

EOD-LARSYS is operational on the Univac 1108 EXEC II version in Building 12. The batch oriented system is operated and maintained according to IDSD procedures.

This document assumes the reader is familiar with both Procedure 1 and the LARSYS system.

## 2. APPLICABLE DOCUMENTS

- User Documentation EOD-LARSYS, Lockheed Electronics Company, Inc., HASD, Houston, Texas, November 1975
- Job Order: 63-1557-1695
- Requirements Document, Ref: 642-2240, dated February 7, 1977
- TIRF: 77-0008
- Design Specification for LARSYS Procedure 1, Lockheed Electronics Company, Inc., HASD, Houston, Texas, April 1977



### 3. SYSTEM DESCRIPTION

This document describes the EOD-LARSYS Procedure 1. In particular, this section is intended to outline the sequence of executing the various processors to accomplish the task of classifying a LACIE segment. These capabilities described are those of first priority, and it is expected that other features will be added in the near future. The system is designed to provide RT&E with a means for experimenting with a new technique for classifying Landsat data for LACIE.

The data is preprocessed by ERIPS. (ERIPS merges tapes received from Goddard Space Flight Center and creates a multitemporal/multi-pass tape.) Using an ERIPS tape as input to the DOTDATA processor, a dot data file is output. The files contain both type 1 and type 2 dots. Type 1 dots are used both as starting vectors for the clustering processor (ISOCLS) and as labeling vectors for the labeling processor (LABEL). Type 2 dots are used as a bias correction factor in computing the classification results output by the DISPLAY processor.

After delineating DO/DU fields by card input, ISOCLS clusters the segment using the starting vectors from the dot data file to initialize the clustering process. Sun angle correction is provided. An unconditional cluster map and a set of 'unlabeled' stats is output.

The 'unlabeled' stats, cluster map, and dot data file are input to the LABEL processor. Using one of two procedures, k-nearest neighbor or all-of-a-kind, the stats are 'labeled'. A conditional or mixed cluster map may be output to be later displayed on the PMIS DAS or Image-100.

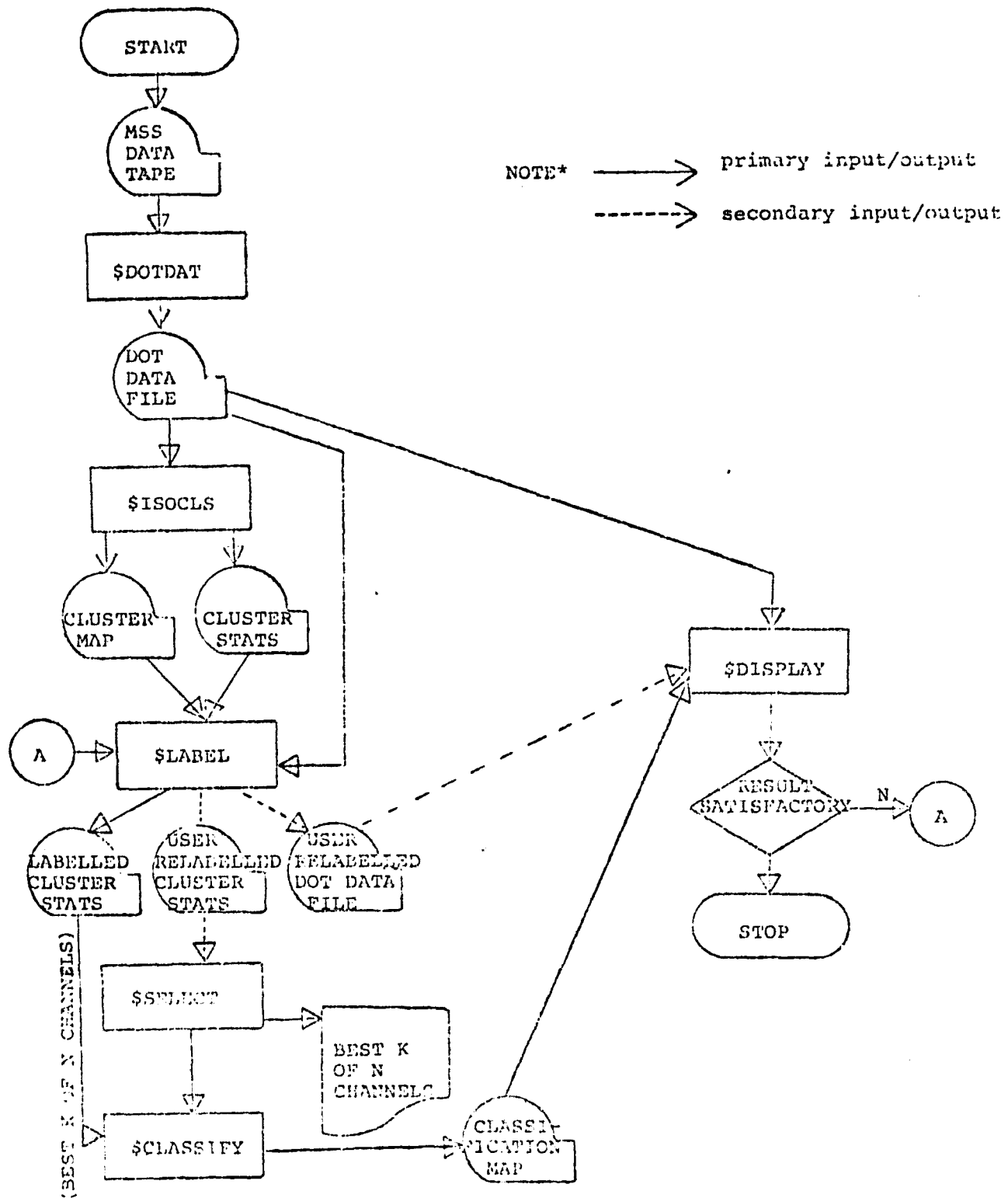


Figure 1 - Procedure I Processor Flow

Using the 'labeled' stats, and allowing the user the capability of setting the inter-subclass weights by categories, the best k of n channels may be selected by the feature selection processor SELECT.

In CLASSIFY, each class in the 'labeled' stats may be assigned to a category by the analyst or by the system. The a-priori probability for each category may be computed using the cluster population from the stats. Using the k best channels, the sum-of-density classifier assigns each pixel in the LACIE segment to a given subclass.

In DISPLAY, the bias correction dots are used in computing the bias correction for the classified area. Two performance tables relating to the bias correction dots are output. One table contains a category dot summary performance; the other, an individual dot summary performance.

Given that the analyst is not satisfied with the classification results, he may relabel the dots and/or relabel the stats (by a labeling procedure or by card input) and restart the process at any given point.

### 3.1 HARDWARE DESCRIPTION

N/A

## 3.2 SOFTWARE DESCRIPTION

### 3.2.1 SOFTWARE COMPONENT NO. 1 (DOTDATA)

In implementing Procedure 1, the means for allowing the user the capability to label certain MSS data points (known as dots) was added to the system by the processor DOTDATA. The main function of this processor is to output a file containing the dots of interest. This file is an interface for three processors, ISOCLS, LABEL, and DISPLAY.

The dots are defined by field cards. By user control, any subset of the 209 possible grid points may be selected. Optionally, the dots may be labeled at the initial creation of the file or the dots may be labeled in the LABEL processor. All the categories of interest do not have to be defined in this processor.

If a dot is to be retyped, the file must be recreated by executing this processor again.

The files contain both type 1 and type 2 dots. Type 1 dots (starting and labeling dots) are written on one file; type 2 dots (bias correction dots), on a second file.

By an OPTION control card, the user may request that the spatial and spectral information relating to each dot on the file be printed on the line printer.

#### 3.2.1.1 Linkages

The processor DOTDATA uses the FORTRAN-V compiler, Univac software system routines, EOD-LARSYS utility routines, and common blocks INFORM, GLOBAL and DOTVEC, and the following processor subprograms: DOTDAT, DOTS, FLDTYP, and SET13.

See Appendix A for description of subprograms.

### 3.2.1.2 Interfaces

N/A

### 3.2.1.3 Inputs

Requires an MSS data tape. See section 3.2 of User Documentation EOD-LARSYS. Requires three types of card input.

- Processor Card

<u>Keyword</u>	<u>Function</u>
(Col. 1) \$DOTDATA	Loads into the system all the routines needed for executing this processor.

- New Control Cards

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
(Col. 1) CHANNEL	DATA= $n_1, n_2, \dots, n_{30}$ (Default: None)	Integer numbers separated by commas referring to the channels on the MSS data tape.
DATAFILE	UNIT= $n$ , FILE= $m$ (Default: $n=3$ $m=1$ )	$n$ is the FORTRAN unit number assigned to the MSS data tape. $m$ is the file number of the data to process.
DOTFIL	OUTPUT/UNIT= $n$ , FILE= $m$ (Default: $n=8$ $m=1$ )	$n$ is the FORTRAN unit number assigned to the dot data file output by this processor. $m$ is the number of the file to output.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
OPTION	PRINT (Default: no line printer output)	Initiates the printing of the dot data file information.
HED1	Any 60 characters (Default: Lyndon B. Johnson Space Center)	First line of the heading on line printer output.
HED2	Any 60 characters (Default: Houston, Texas)	Second line of the heading on line printer output.
DATE	Any 12 characters (Default: Present date)	Date in the heading on line printer output.
COMMENT	Any 60 characters (Default: None)	Comment printed with heading on line printer output.
*END*		Indicates the end of the control card inputs.
\$END*		Indicates the end of all the card inputs.

- Field Cards

By the field card, the user defines the grid points to extract from the MSS data tape. The definition and order of the field card(s) determines the position of the dots on the dot data file, DOTFIL. The analyst will need to know the position of the dots for defining starting vectors in ISOCLS and for labeling/relabeling the dots in LABEL.

At the time of defining the fields, the type for each dot is defined by the TYPE card. By option, the analyst may label each dot by a CLASSNAME card. If this card is omitted, the unlabeled dots must be labeled by the control card DOTLABEL or excluded from the set by the control card EXCLUDE in the labeling processor, LABEL.

An example is given to illustrate a field data set expected by this processor:

```
*END*
TYPE      1
CLASSNAME WHEAT (optional)
LABEL1    (10,10),(10,10),(196,10)
LABEL2    (10,10),(10,20),(196,20)
CLASSNAME NONWHT (optional)
LABEL3    (10,10),(10,50),(100,50)
TYPE      2
CLASSNAME WHEAT (optional)
BIAS1     (10,10),(10,40),(196,40)
CLASSNAME NONWHT (optional)
BIAS2     (10,10),(10,70),(196,70)
$END*
```

Two files are written. File 1 contains 38 WHEAT dots, followed by 10 NONWHT dots. All of which are type 1 dots. File 2 contains 19 WHEAT dots followed by 19 NONWHT dots. All of which are type 2 dots.

If the CLASSNAME cards were omitted, file 1 would contain 48 unlabeled, type 1 dots. File 2 would contain 19 unlabeled type 2 dots.

In both cases, the reference number for the dots in file 1 defined by LABEL1 field card is 1 thru 19; LABEL2 field card

is 20 thru 38; and LABEL3 field card is 39 thru 48. The reference number for the dots in file 2 defined by BIAS1 field card is 1 thru 19 and by BIAS2 field card is 20 thru 38.

#### 3.2.1.4 Outputs

DOTFIL, a multi-file unformatted FORTRAN written tape, is output as an interface to the processors ISOCLS, LABEL, and DISPLAY. The default unit for DOTFIL is logical unit F (FORTRAN unit 8). See Appendix H for format of the tape.

The line printer output is:

- Summary of field information
- Spatial and spectral information for each dot (optional)
- Summary of user options

#### 3.2.1.5 Storage Requirements

Requires less than 53K of core.

#### 3.2.1.6 Description

The main function of DOTDATA is to create the dot data file, DOTFIL. Optionally, the spatial and spectral information of each dot in the DOTFIL may be printed on the line printer.

DOTDAT is the driver routine. From this routine, the setup routine SET13 is called. SET13 reads and analyzes the control cards and prints a summary of the user selected options.

DOTS is the coordinator and controller of the routines needed to create and output the file, DOTFIL.

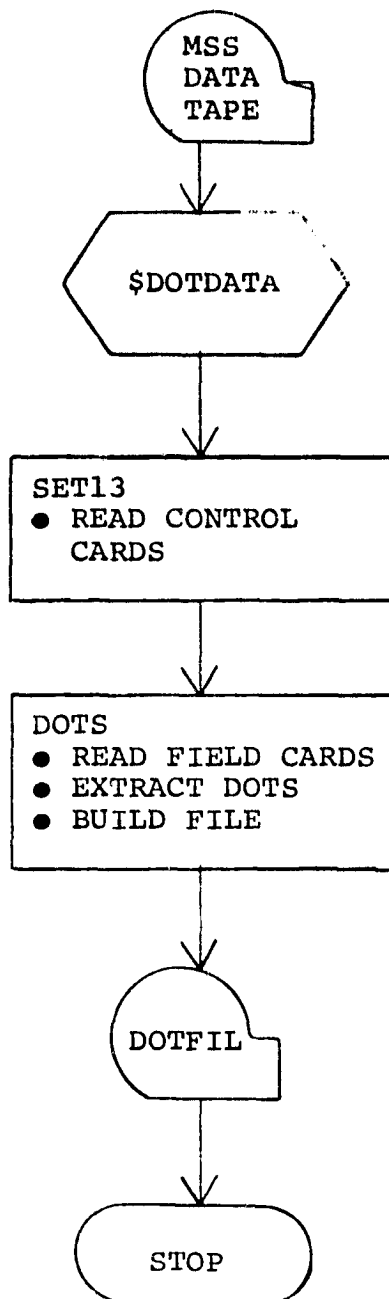
From DOTS, FLDTYP is called to initiate the reading of the field cards. In returning from FLDTYP, the sample number, line number,



type, category (optional), and data vector are stored in the array DATA. DATA is passed to the routine DOTFIL; this routine outputs the dot data file.

~~3-9~~  
//

3.2.1.7 Flowchart



### 3.2.2 SOFTWARE COMPONENT NO. 2 (LABEL)

To aid the analyst in supervising the labeling of the statistics obtained from the clustering processor ISOCLS, a new technique for labeling the statistics was implemented.

Two procedures for labeling the statistics is provided. Optionally, the analyst may select either the k- nearest neighbor procedure or the all-of-a-kind procedure.

The labels in the dot data file, DOTFIL, and/or previously labeled statistics file, SAVTAP, may be relabeled by control card input and the updated file(s) output to tape. Optionally, 1) a conditional or mixed cluster map may be output to tape, 2) an unconditional cluster map may be output in the format acceptable by the DISPLAY processor, 3) the spatial and spectral information of the relabeled DOTFIL may be output to the line printer, and 4) the statistics of the SAVTAP file may be output to the line printer.

#### 3.2.2.1 Linkages

The processor LABEL uses the Fortran-V compiler, Univac software system routines, EOD-LARSYS utility routines, common blocks INFORM, GLOBAL, and LABS, and the following processor subprograms: LABEL, KNEAR, ASCEND, SET14, CRDSCN, FILERD, STOMAP, LABLR, DSPTAP, MIXMAP, CNDMAP, NAMSTA, LABDOT, LABMAN, REORDER, MANORD, DOTDST, SUNFCS, ALLKIN, CLSMAP, CLRKEY, and MAPHND.

See appendix B for description of subprograms.

#### 3.2.2.2 Interfaces

Optionally, the three files listed below may be input:

- Statistics file (SAVTAP) from ISOCLS or STAT processor
- Cluster map file (MAPFIL) from ISOCLS processor
- Dot data file (DOTFIL) from DOTDATA processor

For format of the files see section 4.1 and 5.1 of the User Documentation EOD-LARSYS and appendix H of this document.

### 3.2.2.3 Inputs

Requires 3 types of card input.

- Processor card

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
(col. 1) \$LABEL		Loads into the system all the routines needed for executing this processor

- New control cards

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
(col. 1) CHANNEL	(col. 11-72) STAT= $n_1, n_2, \dots, n_{30}$ DATA= $m_1, m_2, \dots, m_{30}$ (Default: $n_i = \{ \text{all channels on SAVTAP file} \}$ $m_i = \{ \text{all channels on DOTFIL file} \}$	$n_i$ and $m_i$ are integer numbers separated by commas referring to the channels on the SAVTAP file and the DOTFIL file, respectively.
DOTFILE	INPUT/FILE= $m$ , UNIT= $n$	Defines the unit and file for the input dot data file, DOTFIL. $n$ is the Fortran unit number assigned to the input DOTFIL file. $m$ is the number of the input file to process.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
DOTFILE	OUTPUT/FILE=m,UNIT=n (Default: NONE)	Defines the unit and file to which the re-labeled DOTFIL file is output. n is the Fortran unit number assigned to the output DOTFIL file. m is the number of the file to output.
STATFILE	INPUT/FILE=m,UNIT=n (Default: NONE)	Defines the unit and file for the input SAVTAP file. n is the Fortran unit number assigned to the input SAVTAP file. m is the number of the file to process.
STATFILE	OUTPUT/FILE=m,UNIT=n (Default: NONE)	Defines the unit and file to which the labeled/re-labeled SAVTAP file is output. n is the Fortran unit number assigned to the output SAVTAP file. m is the number of the file to output.
MAPTAP	OUTPUT/FILE=m,UNIT=n (Default: no DISPLAY interface tape will be output.)	Defines the unit and file to which the unconditional cluster map, MAPTAP, is output. n is the Fortran unit number assigned to the output MAPTAP file.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
		If executing back to back with DISPLAY, n must be assigned to unit 2. m is the number of the file to output.
MAPFILE	INPUT/FILE=m,UNIT=16 (Default: m=1)	Defines the unit and file for the input unconditional cluster map, MAPFIL, that is output by ISOCLS during the clustering process. n=16 is the Fortran unit number assigned to the input MAPFIL file. (n must be assigned to unit 16 if executing back to back with ISOCLS). m is the number of the file to process.
MAPFILE	OUTPUT/FILE=m,UNIT=n (Default: n=16 m=1)	Defines the unit and file to which the conditional and/or mixed cluster map is output. n is the Fortran unit number assigned to the output MAPFIL file. m is the number of the file to output. <sup>a</sup>
DOTLABEL	category name, n <sub>1</sub> ,n <sub>2</sub> ,...,n <sub>250</sub> (Default: NONE)	The DOTFIL is labeled or relabel by this card. Category name is the label the analyst is assigning to the

<sup>a</sup>If both type of maps are output, the conditional map is output on file m; the mixed map, on file m+1.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
		dots $n_j$ , $j=1, \dots, i$ . The category name may be composed of a maximum of 6 characters. $n_i$ are integer numbers separated by commas referring to the position of the dot on the DOTFIL file.
STATLABEL	class name, $n_i$ , $n_2, \dots, n_{250}$ (Default: NONE)	The SAVTAP file may be manually relabeled by this card. $n_j$ ( $j=1, 2 \dots i$ ) are the number of the subclasses on the SAVTAP that are to be regrouped into another class. Class name is the name of the class to which the subclasses $n_j$ are to be reassigned. Class name must match a name on the SAVTAP file.
DISTANCE	L1 (Default: L1 distance)	The $L_1$ distance is used in computing the distance between the means of the cluster and the labeling dots.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
DISTANCE	L2 (Default: L <sub>1</sub> distance)	The L <sub>2</sub> distance is used in computing the distance between the means of the cluster and the labeling dots.
OPTION	COND (Default: NONE)	A conditional cluster map is output.
OPTION	MIXED (Default: NONE)	A mixed cluster map is output.
THRESHOLD	T (Default: T=25.0)	T is the threshold parameter used in creating the conditional cluster map. T is a floating point number.
NEAREST	K (Default: K=1)	K is the number of dots to be used in the k-nearest neighbor procedure. K is an integer number. K ≤ 250.
PROCED	NAME (Default: N=K-NEAREST)	NAME is an alpha word. NAME=K-NEAREST (Use the k-nearest neighbor procedure.) NAME=ALL (Use the all-of-a-kind procedure.)



<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
		NAME=MANUAL (Use the manual procedure of relabeling the DOTFIL or SAVTAP file.)
MODULE	(blank)	Initiates the input of the module STAT card deck. The deck must immediately follow this card.
EXCLUDE	$n_1, n_2, \dots, n_{250}$ (Default: all dots on the DOTFIL are used.)	$n_i$ are integer numbers referring to the dots on the DOTFIL that are to be excluded in all calculations (i.e., dots within a DO/DU area).
SUNANG	$m_1, m_2, \dots, m_i$ (Default: no sun angle correction is be applied.)	$m_j$ are integer sun angle numbers used in computing the $L_1$ or $L_2$ distances. A sun angle must be input for each acquisition of interest. An acquisition is assumed to be a 4 channel pass. Example: If the distance is computed using 16 channels, 4 sun angles ( $m_1, m_2, m_3, m_4$ ) must be input.
SUNANG	FILE (Default: No sun angle correction is applied.)	Sun angles will be extracted from the DOTFIL.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
OPTION	STATS (Default: Stats not printed)	Means and covariances for labeled or relabeled statistics on SAVTAP file are printed.
OPTION	DOTS (Default: Relabeled DOTFIL is not printed)	Spatial and spectral information of relabeled DOTFIL is printed.
HED1	Any 60 characters (Default: Lyndon B. Johnson Space Center)	First line of the heading on line printer output.
HED2	Any 60 characters (Default: Houston (Texas))	Second line of the heading on line printer output.
DATE	Any 12 characters (Default: Present date)	Date in the heading on line printer output.
COMMENT	Any 60 characters (Default: NONE)	Comment printed with heading on line printer output.
*END*		Indicates the end of the control card inputs.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
\$END*		Indicates the end of all the card inputs.

- FIELD CARDS

If 1) the all-of-a-kind procedure is selected, 2) a conditional or mixed cluster map is output, or 3) a DISPLAY interface tape is output, a MAPFIL tape must be input, and a field card defining the area of the unconditional cluster map (input MAPFIL) must also be input. The vertices must reflect the sample number and line number of the MSS data tape used to create the unconditional cluster map. That is, the field card must be the identical to the field card input to ISOCLS.

If 1) a procedure is not to be executed, 2) a MAPFIL file not output, or 3) a MAPTAP file is not output, a field card is not input.

Example of the data set:

- MAPFIL tape is being input

\*END\*

FIELD (1,1), (1,1), (196,1), (196,117), (1,117)

\$END\*

- MAPFIL tape is not being input

\*END\*

\$END\*

#### 3.2.2.4 Outputs

Optionally the following files are output:

- Labeled statistic file (SAVTAP) by using a procedure (See section 4.1 of User Documentation EOD-LARSYS for format of file)

- Labeled statistic file (SAVTAP) by control card input
- Relabeled dot data file (DOTFIL) (See Appendix G for format of file)
- A conditional cluster map (See section 5.1 of User Documentation EOD-LARSYS for format of file)
- A mixed cluster map.
- An unconditional cluster map in the format acceptable by the DISPLAY processor (See Appendix C of User Documentation EOD-LARSYS for format of file)

4. Line printer summary of the following is output:

- Summary of selected options
- Table of  $L_1$  or  $L_2$  distances
- Summary of the labeling dots within a cluster for the all-of-a-kind procedure
- Summary of the labeling dots for the k nearest dots to a cluster for the k-nearest neighbor procedure
- Spatial and spectral information of relabeled DOTFIL
- Means and covariances of labeled or relabeled statistics.

### 3.2.2.5 Storage Requirements

Requires about 53k of core.

### 3.2.2.6 Description

A distance table containing the  $L_1$  or  $L_2$  distance of all the type 1 dots and the mean of the clusters is computed.

$$L_1 = \frac{\sum_{i=1}^n |x_i - u_i|}{n}$$

$$L_2 = \sqrt{\frac{\sum_{i=1}^n (x_i - u_i)^2}{n}}$$

where:

$x_i$  =  $i^{\text{th}}$  element of the dot vector

$u_i$  =  $i^{\text{th}}$  element of the mean vector

$n$  = number of channels

Using the table as input to the k-nearest neighbor procedure or all-of-a-kind procedure, the statistics generated during clustering is labeled.

For the k-nearest neighbor procedure, all the labels of the k-nearest labeling dots to a given cluster are polled. The label with the majority of the dots labels the cluster. If a tie occurs, then k-1 dots are considered.

For the all-of-a-kind procedure, all of the labeling dots within a cluster are polled. If all the dots are of one category, the cluster labels that category. If the cluster contains dots for more than one category, the label with the majority of the dots labels the cluster. If there are no labeling dots within a cluster, the labeling defaults to the k-nearest neighbor procedure.

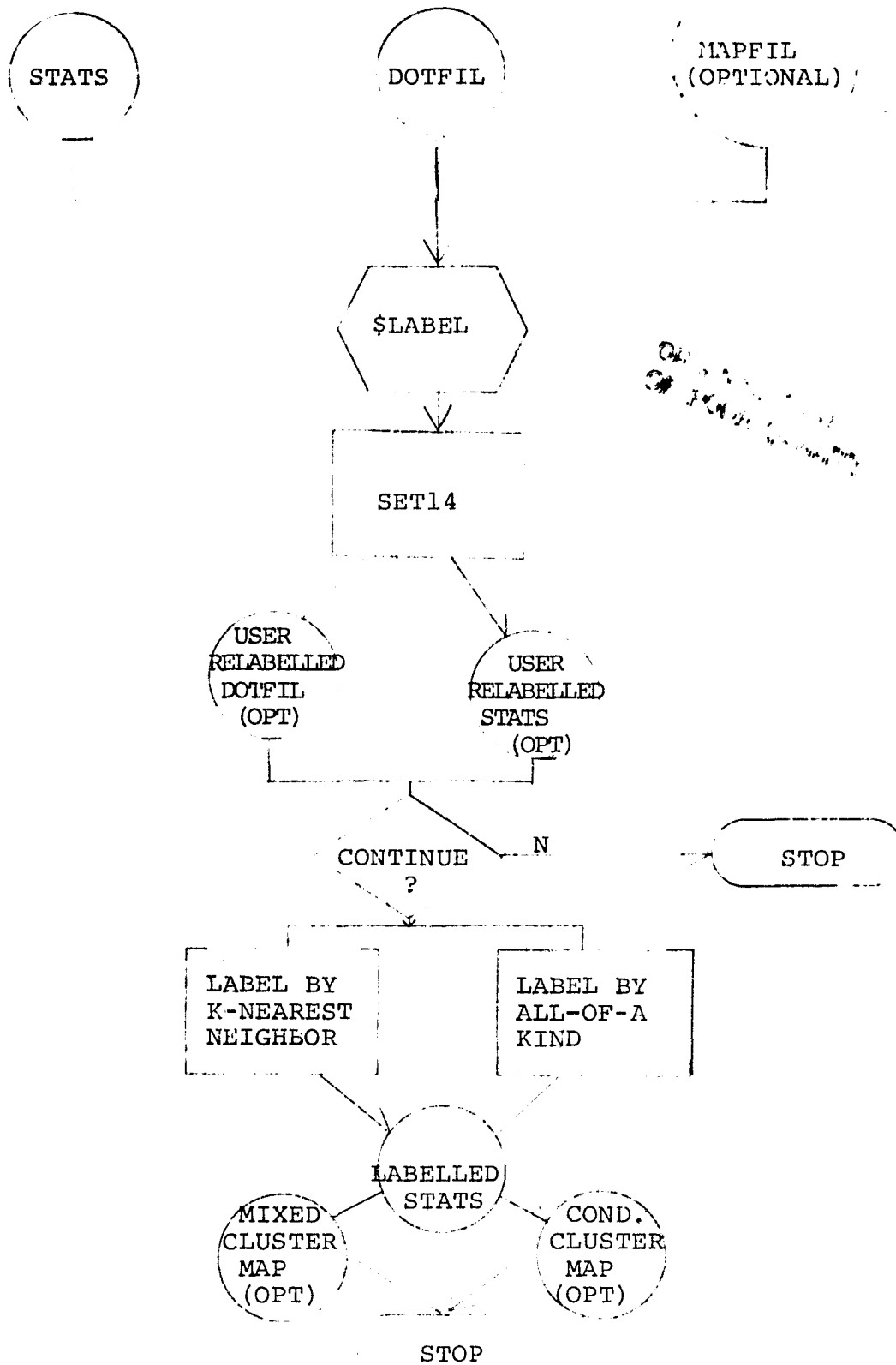
Optionally, a conditional cluster map may be output. A cluster is tagged as conditional if the distance between the nearest identically labeled labeling dot and mean of the cluster is greater than the analyst input threshold value  $t$ .

Optionally, a mixed cluster map may be output. A cluster is tagged as mixed if the labeling dots within a cluster are of more than one category.

Optionally, an unconditional cluster map may be output in the format acceptable to the DISPLAY processor. Information used in

the thresholding procedure in DISPLAY is dummed. If thresholding of the clustered data is desired, it can be performed by exercising the conditional map option in this processor.

3.2.2.7 FLOW CHART



### 3.2.3 SOFTWARE COMPONENT NO. 3 (ISOCLS)

Several additions were made to the ISOCLS (clustering) processor of the LARSYS system in support of Procedure 1 requirements. These additions are described in a general sense as follows:

- a. On option starting dots (pixels) from the dot data file DOTFIL may be used to begin clustering.
- b. The analyst may identify "designated other" and "designated unidentifiable" pixel sets (fields) by field card input. The pixels in these fields are not included as inputs to the clustering algorithm. They are assigned special cluster numbers and mean vectors for display purposes.
- c. On option, using a sun angle correction table, the pixel radiance values may be modified. [The correction table is built in.] The radiance value correction applies only for clustering purposes.

The user may input the sun angles by cards or request that these angles be extracted from the header record of a universal formatted MSS data tape (ERIPS unload tape).

#### 3.2.3.1 Linkages

The processor ISOCLS uses the FORTRAN-V compiler, Univac software system routines, EOD-LARSYS utility routines and common blocks PASS, GLOBAL and ISOLNK.

See Appendix C for description of the subprogram SUNFAC and modifications to the following subprograms: PSPLIT, SUTUP7, RDDATA, and ISODAT.

#### 3.2.3.2 Interfaces

Files generated by other processors and used as input to ISOCLS are:



- a. On option, a statistics file (SAVTAP) to provide starting cluster mean vectors.
- b. On option, input initial cluster "centers" by cards.
- c. On option, starting dots from DOTFIL to initialize the cluster processing.

Format descriptions of these files are included in the User Documentation EOD-LARSYS in sections 4.1 and 3.1.4.3, and Appendix H in this document, respectively.

### 3.2.3.3 Inputs

Require an MSS data tape and card input. See section 3.2 and 9.5, respectively, of User Documentation EOD-LARSYS. Added control cards and field cards are note below:

- New Control Cards

<u>Keyword</u>	<u>Parameters</u>	<u>Function</u>
DOTFILE	INPUT/UNIT=n,FILE=m (Default: Self-initializing starting)	Defines the FORTRAN unit number n and file number m of the dot data file DOTFIL containing the starting vectors.
DOTS	$n_1, n_2, \dots, n_{60}$ (Default: Dots will not be used for starting vectors.)	$n_i$ are integer numbers separated by a comma specifying the dots to be used as starting vectors.
SUNANG	TAPE (Default: No sun angle correction applied.)	Sun angles are extracted from the ERIPS unload MSS data tape.

<u>Keyword</u>	<u>Parameters</u>	<u>Function</u>
SUNANG	$n_1, n_2, \dots, n_j$ $n_j$ are integer numbers, $j \leq 8$ . (Default: No sun angle correction applied.)	$n_j$ are the sun angles to be used in computing the sun angle corrections for use in the clustering algorithm. A sun angle must be input for each set of 4 channels input on the CHANNEL control card.

● Field Cards

ISOCLS recognizes DO/DU fields. All the DO/DU field cards (for all classes) must be input before the fields to be clustered. These fields must immediately follow the \*END\* card. The CLASSNAME card follows the last DO/DU field card.

Example:

- If DO/DU fields are being defined

```

*END*
DESIGNATED    OTHER
OTHER         (1,1), (1,1), (40,1), (40,20), (1,20)
DESIGNATED    UNIDENTIFIABLE
UNIDEN        (1,1), (5,7), (8,7), (8,10), (5,10)
CLASSNAME     WHEAT
WHT1          (1,1), (1,1), (196,1), (196,117), (1,117)
$END*
  
```

- If no DO/DU fields are being defined

```

*END*
CLASSNAME     WHEAT
WH1           (1,1), (1,1), (196,1), (196,117), (1,117)
$END*
  
```

#### 3.2.3.4 Outputs

- a. Statistics file (SAVTAP) and on option, punched card deck.
- b. Cluster map tape (MAPFIL), on option
- c. Printout of cluster results, consisting of the following data items by class:

- Cluster numbers and symbols
- Cluster mean vectors (by channel)
- Cluster standard deviations by channel
- Inter-cluster distances
  
- Number of pixels per cluster
- Number of clusters
- Cluster map by field for each class

#### 3.2.3.5 Storage Requirements

Require 17449 words of core plus 19345 words of core for the resident programs.

#### 3.2.3.6 Description

The implementation of the additions to ISOCLS are generally described. For further details, please refer to the subprogram documentation which is included.

- a. By expanding the INVEC data vector of the control card is accepted. Specification of unit and file number on this card triggers a call to routine RDDOTS to load the starting vectors. These vectors are stored internally for use by ISODAT. The unit and file number is stored in an extension to labeled common block GLOBAL.
- b. Pixel fields to be described as "other" or "unidentifiable" are input by analyst-furnished classname and field definition cards to be read by RDDATA. The classname cards are punched as:

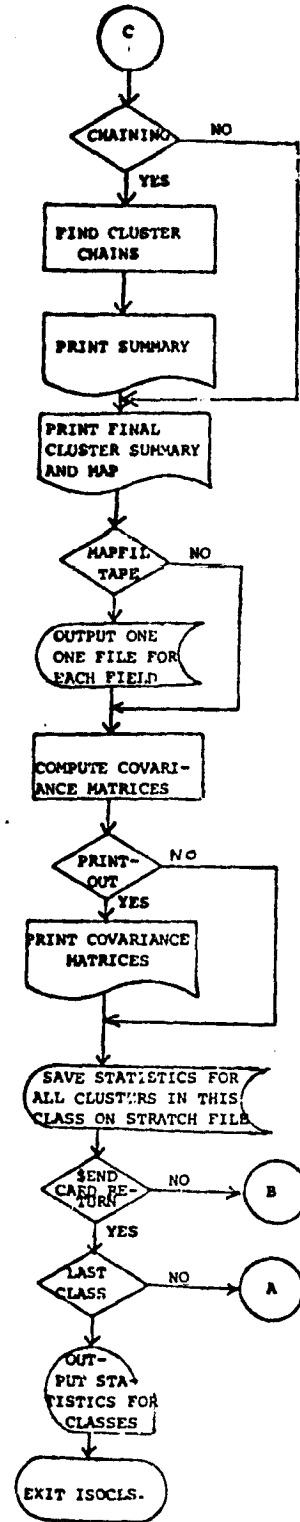
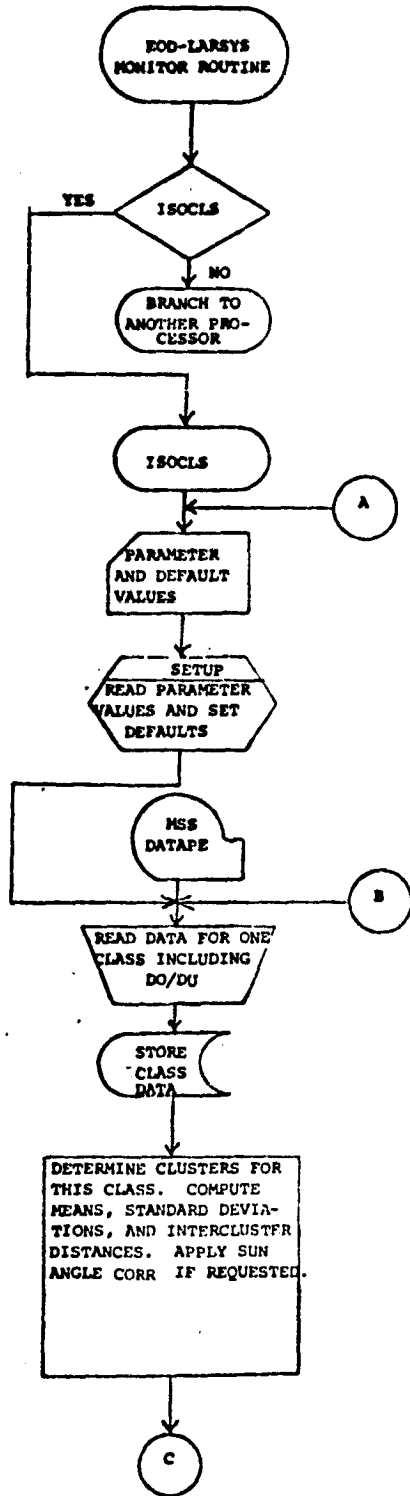
DESIGNATE OTHER or  
DESIGNATE UNIDENTIFIABLE.

The DO/DU pixels are flagged and assigned special cluster numbers. They are not included as input to the clustering algorithm. This requires a modification to RDDATA and ISODAT.

Field definition information is stored internally for use by ISODAT. A trigger indicating that DO/DU fields have been used is set and stored in an extension of the labeled common block PASS.

- c. To handle sun angle input, SETUP7 is modified (by adding 'SUNANG' to the INVEC data vector) to read a sun angle control card. This card either contains in the parameter field of the card, the sun angles for each pass or the word TAPE. TAPE triggers the unpacking of sun angles from the MSS data tape. If the SUNANG control card is not present, a default to no sun angle correction occurs. The trigger indicating that the sun angles have been unpacked appears in labeled common ISOLNK.
- d. The sun angle correction table is stored as DATA in subprogram SUNFAC. The correction is computed by 1) looking up the gain versus sun angle for each channel, and 2) multiplying the pixel radiance value for the channel by the gain.

3.2.3-7 FLOWCHART



### 3.2.4 SOFTWARE COMPONENT NO. 4 (SELECT)

The SELECT processor is modified to optionally provide an automatic assignment of interclass weights. The weights for class<sub>i</sub>, class<sub>j</sub> = 1.0 for i ≠ j and the weights for class<sub>i</sub>, class<sub>j</sub> = 0.0 for i = j. The breakdown of class<sub>i</sub>, class<sub>j</sub> pairs into the correct set of intersubclass pairs is provided by the processor. Information concerning the class-subclass association is extracted from the input statistics file, SAVTAP.

The modifications to the SELECT processor involve two subprograms, SETUP4 and WGTCHK. These modifications are made to enable use of the processor in the evaluation of the LACIE Procedure 1 crop classification technique. For application of the SELECT processor to LACIE Procedure 1, the user must consider classes equivalent to categories.

#### 3.2.4.1 Linkages

The SELECT processor uses the Univac 1108 (EXEC2) operating system and associated system routines, the FORTRAN V compiler, EOD-LARSYS utility routines, and EOD-LARSYS common blocks FSL, GLOBAL, and INFORM.

See Appendix D for the detailed description of modifications to the subprograms SETUP4 and WGTCHK.

#### 3.2.4.2 Interfaces

The SELECT processor interfaces with the LABEL processor via the labeled statistics file, SAVTAP, which is output by the LABEL processor. The statistics file is also the interface between SELECT and the STAT or ISOCLS processors.

#### 3.2.4.3 Inputs

The modifications to the SELECT processor include an additional

input option on the OPTION control card. The new input option is required to initiate the automatic (processor provided) assignment of interclass weights. See below for the format and function of the new option.

● Revised Control Card

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
(col. 1) OPTION	(col. 11-72) CLSWT (Default: The weights are assigned to intersubclass pairs.)	The processor determines the class-subclass correspondence (after any grouping of subclasses to form new subclasses if the GROUP control card is used) and assigns a weight=1.0 to the subclass pairs associated with all interclass pairs. Intra-class subclass pairs are given a weight=0.0

NOTE(1): The WEIGHTS control card remains available to allow the user to set weights for specific subclass pairs. If used, the input subclass pair weights override the processor-set subclass pair weights.

NOTE(2): The "WEIGHTS OTHERS" capability is not available when this option is exercised. If input, it is ignored by the processor.

#### 3.2.4.4 Outputs

The message for automatic assignment of interclass weights is added to the line printer output of user requests. There is no change to file output by the SELECT processor.

#### 3.2.4.5 Storage Requirements

The current storage allocation (overlay) map for EOD-LARSYS results in a utilization of 7696 storage locations in the Instruction Bank, and 15309 locations in the Data Bank, plus 19345 locations for the resident programs.

#### 3.2.4.6 Description

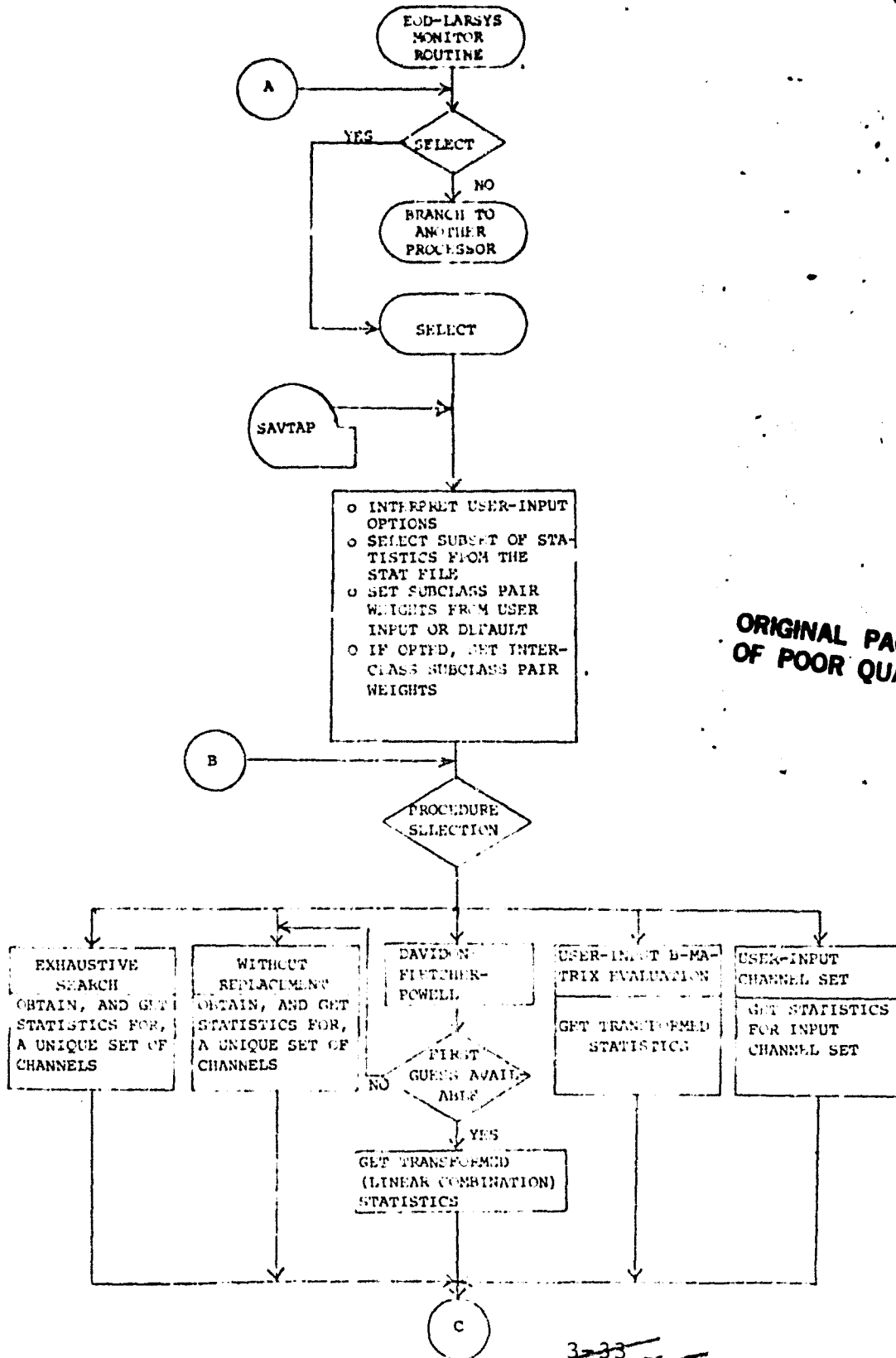
Initialized via the OPTION CLSWT control card, SETUP4 sets an internal flag (WTKEY=1) indicating that subclass pair weights are to be program-determined on an interclass basis. From the class (or category) and subclass information from the input statistics file, SAVTAP, the subclass-to-class association will be determined. In determining the class-subclass association, any new subclasses created by the user via the GROUP control card is recognized.

Using the existing storage location, SUBRAY, SETUP4 builds a table of subclass pair weights. The weight table is initialized by a newly added internal subroutine, INTWGT. All subclass pairs which represent an interclass pair are given a weight=1.0. All other subclass pairs (i.e., intraclass subclass pairs) are given a weight=0.0.

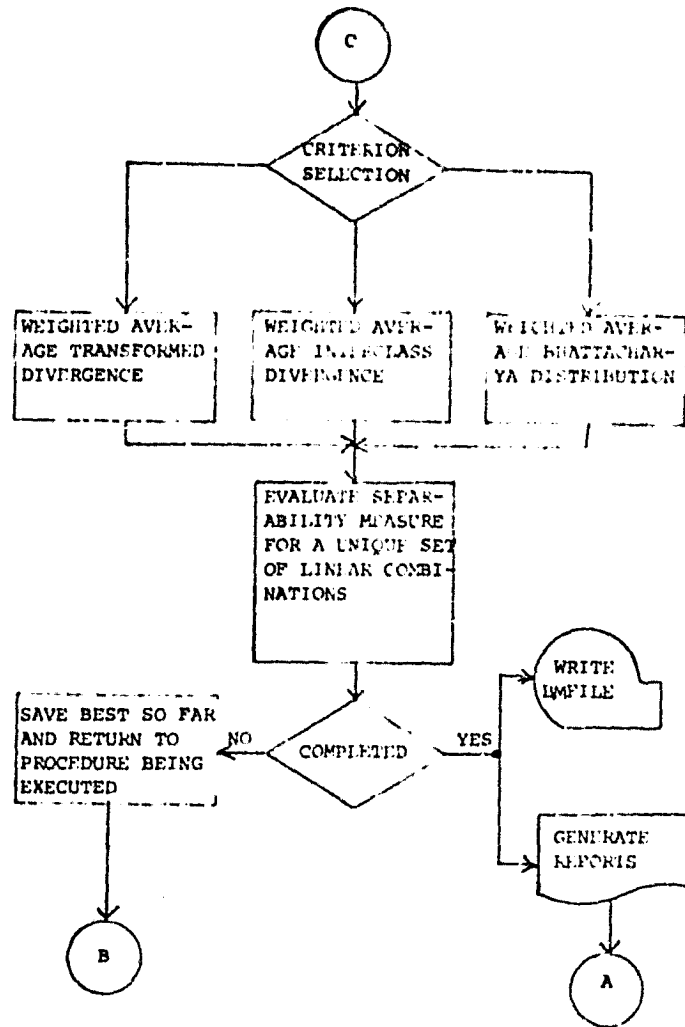
As usual, the user may input any subclass pair weights via the WEIGHTS control card. Any subclass pair weights input via the WEIGHTS card overrides the processor-set weights for those subclass pairs. This function is provided by the existing subprogram, WGTCHK, as usual.



3.2.4.7 FLOWCHART



ORIGINAL PAGE IS OF POOR QUALITY



Concluded - Select Processor Flowchart

### 3.2.5 SOFTWARE COMPONENT NO. 5 (CLASSIFY)

The classification processor, CLASSIFY, is modified as follows:

1. To allow an option for obtaining subclass a priori values using subclass population data from the input file, SAVTAP.
2. To allow the system to assign the category names using the class names from the input statistics file, SAVTAP, as the assigned category names. This modification is to adopt the processor to LACIE Procedure 1 techniques.

Both options are an addition to the usual capability of analyst-input of a priori probability values at the subclass, class, or category level via the APRIORI control card and of category name input via the CATEGORY control card.

#### 3.2.5.1 Linkages

The CLASSIFY processor uses the Univac 1108-EXEC 2 operating system and associated system routines, the FORTRAN V compiler, and EOD-LARSYS utility routines and common blocks. See Appendix E for the description of the modified subprograms.

#### 3.2.5.2 Interfaces

There are no modifications to CLASSIFY interfaces. As usual, the CLASSIFY processor obtains the class names, subclass statistics (mean vectors and covariance matrices) and related subclass data from either an input statistics file, SAVTAP, or the Module STAT card deck which are created by either the STAT processor or the ISOCLS processor. The CLASSIFY processor also interfaces with the SELECT processor to obtain a transformation matrix (the B-matrix) via either a file, BMFILE, or a B-MATRIX card deck which are both created by the SELECT processor.

The input statistics file, SAVTAP, is either assigned to UNIVAC unit A or to the users choice of available units. The input transformation matrix file, BMFILE, is always assigned to UNIVAC unit H.

### 3.2.5.3 Inputs

- MSS Data

The MSS (multi-spectral scanner) data to be classified is input to the CLASSIFY processor via the input file, DATAPE, as usual. The file must be in either of two specific formats, LARSYS II or Universal, and is either assigned to Univac unit C, or to the user's choice of units as designated on the DATAFILE control card.

- Revised Processor Control Card

Changes to the control card input for the CLASSIFY processor to initiate the modifications described in Section 3.2.5 are on the APRIORI control card and CATEGORY control card. Either the modified or standard cards may be input, depending on the options desired.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
(col. 1)	(col. 11-72)	
CATEGORY	FILE (Default: No categories are defined and the standard classifier will be applied)	Initiates the assigning of the category names using the class names from the input statistics file, SAVTAP, and invokes the category classifier.

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
APRIORI	FILE (Default: Subclass a priori is not be computed from the statistics file, SAVTAP.	The subclass apriori probability values are computed using subclass or cluster point popula- tions from the statistics file, SAVTAP.

- Field definition cards

There are no modifications to the field definition input to CLASSIFY. As usual, the area(s) on the input data tape to be classified by the CLASSIFY processor are defined to the processor on input "field definition" card(s). The area(s) to be classified are defined in terms of sample, line coordinates of each vertex of the field, up to a maximum of ten (10) vertices for an irregularly shaped field. Also contained on the field definition card(s) are the incrementation of samples and/or lines to be performed in reading the data for the defined field.

The format of the "field definition" card is provided in Section 3.1.3 of the user documentation EOD-LARSYS document no. LEC-3984.

#### 3.2.5.4 Output

There are no modifications to the output provided by CLASSIFY. As usual, the classification processor, CLASSIFY, outputs the classification results on a file, MAPTAP, assigned to Univac Unit B. The primary results output on the MAPTAP file are the subclass identification of each pixel of the area classified, and the value of the probability density function for each pixel of the area classified.

The classification processor outputs via the line printer a classification map, which illustrates the classified identity of each pixel by symbols, in correct spatial relationship according to sample, line coordinates of each pixel. Also output to the line printer are the class/subclass training field coordinates, the class/subclass statistics (mean vectors and covariance matrices) and a listing of the subclasses considered for classification along with apriori probability values for each subclass.

#### 3.2.5.5 Storage Used

With the present EOD-LARSYS overlay (MAP) structure, the CLASSIFY processor requires 4058 locations in the Instruction Bank, 24559 locations in the Data Bank, for a total of 28617 locations, plus 19345 locations for resident programs.

#### 3.2.5.6 Description

To accomplish the additional option described in Section 3.2.5, the modifications to the CLASSIFY processor involve two subprograms, REDIF2 and SETUP2.

Subprogram REDIF2 is modified to read and decode the revised APRIORI control card and CATEGORY control card described in Section 3.2.5.3. If the word FILE is detected in the parameter field (card columns 11 through 72) of the APRIORI control card, internal flags are set (APRKEY=-777777, APRFLG=APRKEY) to indicate that the subclass population data from the statistics file, SAVTAP, is to be used for computing subclass apriori probability values. If the word FILE is not detected in the parameter field of the input APRIORI card, it is assumed that the card contains floating point (decimal) numbers, and the parameter field of the card is read as is usually done - i.e., scanned for decimal numbers separated by a comma, and storing each number found into the apriori probability value array.

If the flag (APRFLG) is set by virtue of detecting the word FILE in the parameter field of the APRIORI control card, subprogram SETUP2 obtains the subclass population data from the statistics file, SAVTAP via the array KEPPTS. The computation of subclass apriori probability values are performed in SETUP2 and the computed values are stored in the apriori probability value array, APRIOR. The computation of the subclass apriori probability values are performed as follows:

$$P_i = \frac{N_i}{K}$$

where

$P_i$  = apriori probability for subclass  $i$

$N_i$  = number of pixels in subclass  $i$

$K$  = total number of pixels in all subclasses

If the word FILE is detected in the parameter field of the CATEGORY control card, each class defined on the SAVTAP file becomes a category. For each class name there is a corresponding category of like name.

If the word FILE is not detected in the parameter field of the CATEGORY card, it is assumed that a category is being defined by analyst input, as usual.

In the subprogram SETUP2 three existing arrays are initialized as follows, when CATEGORY FILE card is input:

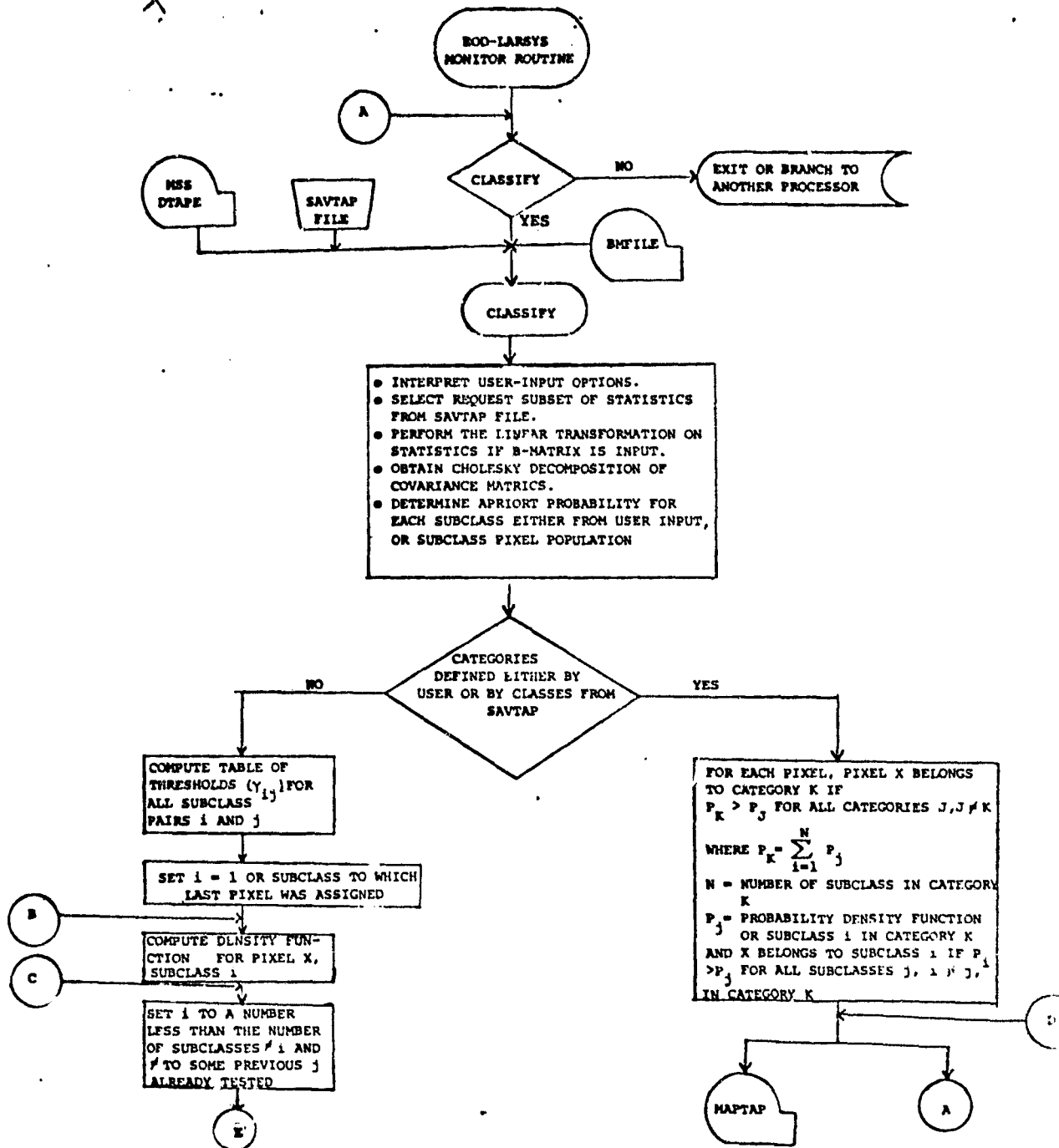
KCLSNA = class names from SAVTAP file

NOCTCL = 1 for number of classes in category I

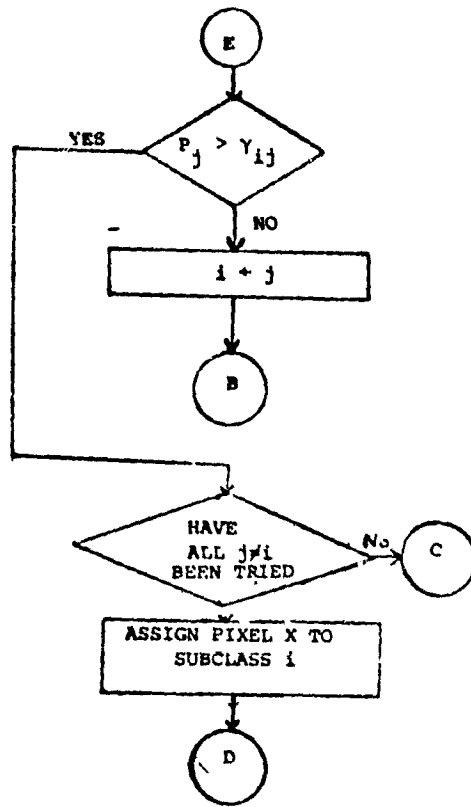
CATNAM = class names from SAVTAP file

and NOCAT = NOCLS2 where NOCAT is the number of categories and NOCLS2 is the number of class names on the SAVTAP file.

3.2.5.7 FLOWCHART







~~3-41~~  
43

### 3.2.6 SOFTWARE COMPONENT NO. 6 (DISPLAY)

The classification results display processor, DISPLAY, is modified to meet LACIE Procedure 1 image processing requirements, as follows:

1. to accept a new processor control card
2. to accept the dot data file, DOTFIL
3. to provide a dot classification performance summary by dot categories which also includes:
  - a tabulation of both the uncorrected proportion and the bias corrected proportion of each dot category in the total area classified
  - an "alpha" table which tabulates proportions for each labeled category of bias correction dots (type 2 dots), which for each category are the ratios of the number of dots with the given analyst-labeled category that were classified into each possible category, to the total number of dots classified into each category.
4. to provide a dot classification performance summary for each dot on the analyst's specified file (DOTFIL).

#### 3.2.6.1 Linkages

The DISPLAY processor requires the Univac EXEC2 operating system, the Fortran V compiler, the EOD-LARSYS utility routines, and common blocks GLOBAL and DISPL.

See appendix F for description of modifications to the following subprograms: SETUP3, REDIF3, DSPLY2, and DSPLY; for description of subprogram TAPLAB; and for description of internal subprograms DOTSUM, DTCHK, and DOTPCT.

### 3.2.6.2 Interfaces

As a result of these modifications, the DISPLAY processor optionally accepts as input the dot data file, DOTFIL, created by the DOTDATA processor. The file is assigned either to Univac logical unit F (Fortran unit 8) or to an analyst specified unit. (See appendix G for format of file)

As usual, the DISPLAY processor requires the input of a classification results file, MAPTAP, created by the CLASSIFY processor. The file must be assigned to Univac logical unit B (Fortran unit 2) (See Appendix D in EOD-LARSYS User Procedures LEC 3984 for format of file).

### 3.2.5.3 Inputs

Requires card input described in section 12.4 of User Documentation EOD-LARSYS and the following card:

#### ● NEW CONTROL CARD

<u>Keyword</u>	<u>Parameter(s)</u>	<u>Function</u>
(col. 1) DOTFILE	(col. 11-72) INPUT/UNIT=n,FILE=m (Default: UNIT=8, FILE=1 No dot data classification performance summaries)	Initiates the input of dot data file, DOTFIL, from the designated (or default, if not designated) unit and/or file, and initiates the output of the dot data classification performance summaries.  The parameter, m, designates the file to be processed by DISPLAY, and n designates the Fortran unit number assigned to the input file, DOTFIL.

- FIELD CARDS

If the Procedure I option is to be exercised, the only kinds of fields that may be input are DO/DU fields (designates other or designated unidentifiable).

No test fields may be input; the training fields are the bias correction dots on the dot data file, DOTFIL.

The format of the DO/DU field cards and the method of input is given in sections 3.1.3 and 12.4.4 of the User Documentation EOD-LARSYS, IEC 3984.

#### 3.2.6.4 Outputs

The DISPLAY processor normally provides an output tape for display on the PMIS DAS (Data Analysis Station), as an option.

Also, the DISPLAY processor normally provides the following output to the line printer:

- a line-printer map of each classified field on the input classification results file (MAPTAP).
- classification summaries for each classified field, showing the number of pixels classified into and thresholded from each subclass, class, and category.
- classification summaries of training fields, or optionally test fields. The summaries are presented by subclass, class, and category.

As a result of the modifications described in section 3.2.6, the DISPLAY processor optionally provides dot data classification performance summaries (instead of the normal output classification summaries described above) if the DOTFILE control card described in section 3.2.6.3 is input to the DISPLAY processor.

The line printer output of dot data performance summaries is as follows:

- A tabulation of individual dots (pixels) showing the sample/line coordinates of each dot, the analyst-labeled category of each dot, and the classification category of each dot.
- A classification performance summary by analyst-labeled dot data category. This performance summary includes the uncorrected segment (classified field) proportion for each dot category, the bias corrected segment (classified field) proportion for each dot category, and the bias correction "alpha" table. In addition, this summary provides the total number of dots (pixels) in each analyst-labeled category, the percent correct classification for each analyst-labeled category, the number of analyst-labeled dots which were thresholded, and the number of dots (pixels) of a given analyst-labeled category which were classified into the labeled category and classified into categories other than the given labeled category.

#### 3.2.6.5 Storage Requirements

With the current overlay (MAP) structure, the DISPLAY processor requires 16012 locations in the Data Bank, 6732 locations in the Instruction Bank, for a total of 32744 locations, plus 19345 locations for the resident programs.

#### 3.2.6.6 Description

The modifications to the DISPLAY processor are such that the analyst may either exercise the normal processor capabilities as documented in the User Documentation EOD-LARSYS, LEC 3984, or the LACIE Procedure I capabilities. The difference between the two capabilities is in the type and format of classification performance tables output. See Section 3.2.6.4 for a description of the output tables.

The processor as modified uses existing available storage, normally occupied by training field data, to store the dot coordinates and category labels, from the input dot data file.

The subprogram, REDIF3, is modified to read and de-code the new DOTFILE control card, and to set a flag which indicates that processing of the dot data is to be performed by DISPLAY.

Subprogram DSPLY2 is modified to construct the individual dot classified category table, using existing storage to construct the table, and also to build the dot classification described in section 3.2.6.4. The modifications to DSPLY2 are primarily the addition of two new internal subprograms. Subroutine DOTPCT, which constructs the table of dot classification results, and subroutine DOTSUM, which provides the computations and output of the two dot classification summaries.

In subroutine DOTSUM, the computations of the uncorrected category proportions, the "alpha" table, and the bias corrected category proportions are as follows:

$$\begin{aligned} P_{u_i} &= \text{uncorrected segment (classified field) proportion} \\ &\quad \text{for category } C_i \\ &= \frac{\text{number of dots (pixels) classified as } C_i}{\text{total number pixels in classified field}} \\ &= \frac{N_{C_i}}{N_T} \end{aligned}$$

for  $i = 1, 2, \dots, \text{NOCAT}$  categories of analyst-labeled dots.

$$\text{Alpha}_{i,j} = \alpha_{ij} = \frac{\text{Number of dots labeled } C_j, \text{ classified as } C_i}{\text{Number of dots classified } C_i}$$

for  $i, j, = 1, 2, \dots$ , NOCAT categories of labeled dots

$$Pc_i = \text{bias corrected proportion for category } C_i$$

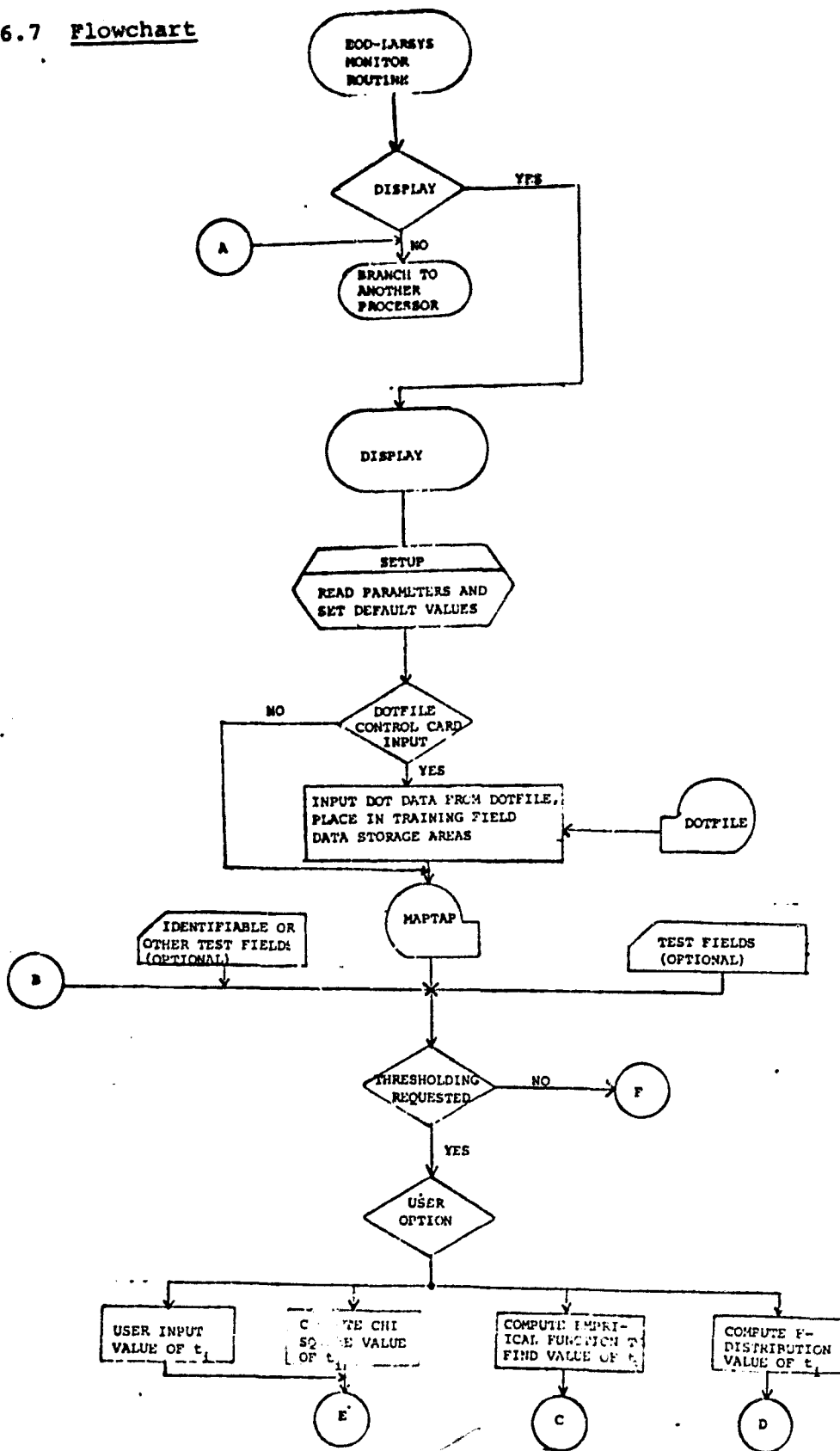
$$= \sum_{j=1}^{\text{NOCAT}} (\alpha_{ij} * Pu_j)$$

The computation of the percent correct classification of the dots is performed as follows:

$$\frac{\text{Number of dots (pixels) labeled } C_1, \text{ classified as } C_i}{\text{Total number labeled dots in } C_1}$$

$$= \frac{N_{C_i}}{N_{T_{C_i}}}$$

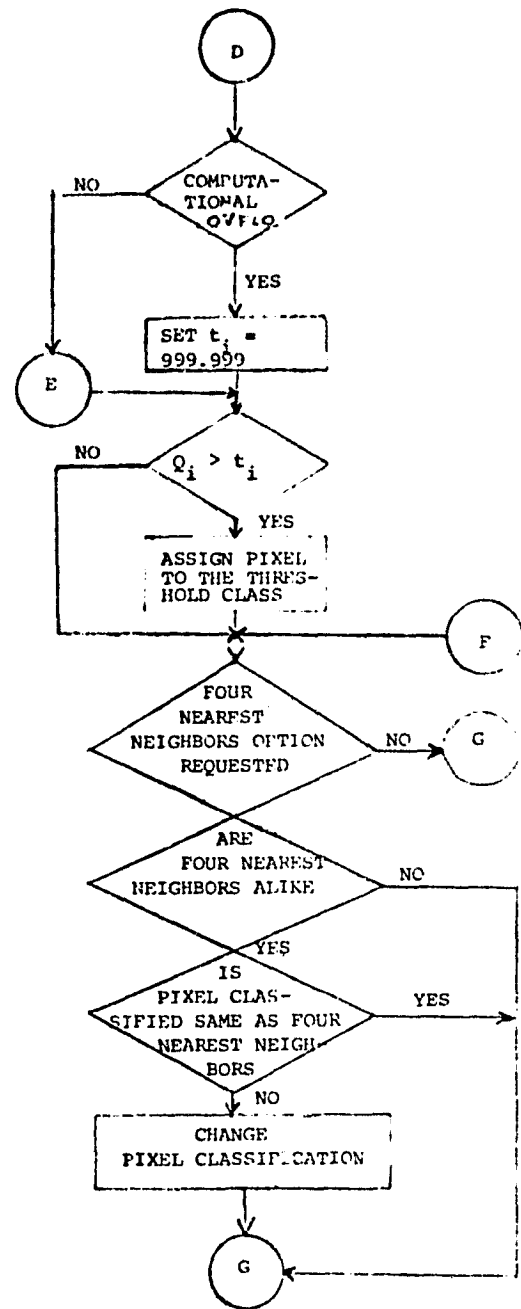
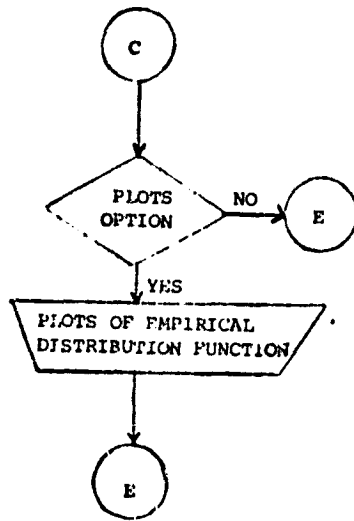
3.2.6.7 Flowchart



3-48

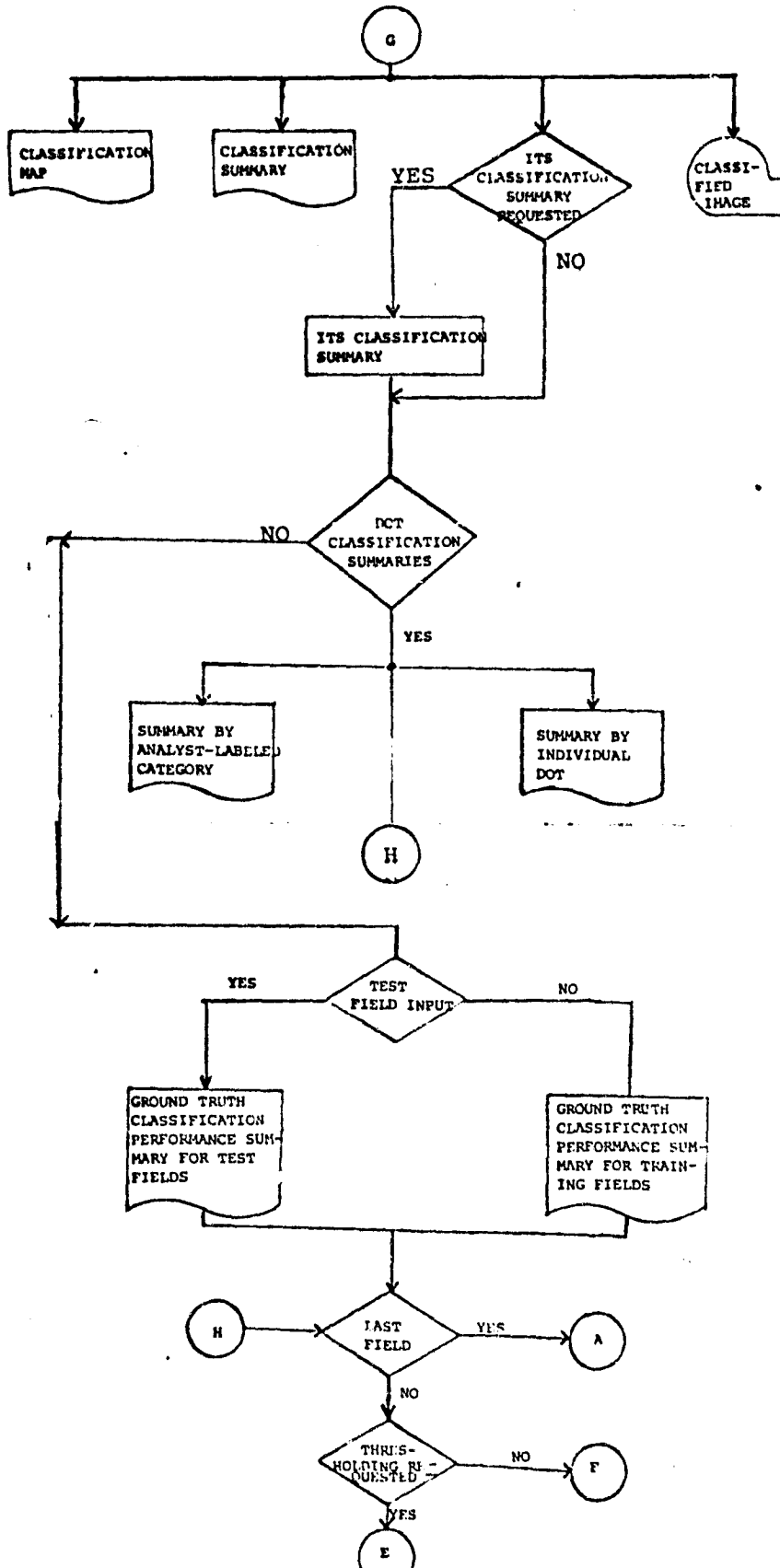
EO





ORIGINAL PAGE IS  
OF POOR QUALITY

Continued - DISPLAY Processor flowchart



3750 Concluded - DISPLAY Processor flowchart

52

T

### 3.2.7 SOFTWARE COMPONENT NO. 7 (UTILITY ROUTINES)

Three EOD-LARSYS system utility subprograms are described in this document; TAPERD, RDDOTS, and WRTDOT.

- TAPERD  
Routine TAPERD reads the MSS data tape (DATAPE) in either the Universal or LARSYS 2 format. (Most of the processors in the EOD-LARSYS system invoke this routine.)
- RDDOTS  
Routine RDDOTS reads the dot data tape (DOTFIL) for spectral spatial information concerning the dots of the Procedure 1 concept.
- WRTDOT  
Routine WRTDOT outputs the DOTFIL. In general, two types of files are written, the first containing labeling/starting dots, and the second containing bias correction dots.

#### 3.2.7.1 Linkages

The utility subprograms use the Fortran V compiler, Univac software system routines, other EOD-LARSYS utility routines, and common blocks GLOBAL and INFORM. See appendix G for description of modifications to subprogram TAPERD and for description of subprograms RDDOTS and WRTDOT.

#### 3.2.7.2 Interfaces

Interfaces with other routines mainly via calling arguments, but also may reference common blocks INFORM and GLOBAL.

#### 3.2.7.3 Inputs

- TAPERD  
Accepts a multi-file Universal or LARSYS 2 formatted tape. The tape may be created either by the EOD-LARSYS system or by a computer other than UNIVAC. (See appendix A and B, respectively in User Documentation EOD-LARSYS for formats.)

- RDDOTS

Accepts a multi-file unformatted Fortran created tape in the format defined in appendix H.

#### 3.2.7.4 Outputs

- WRTDOT

Outputs a multi-file unformatted Fortran written tape in the format defined in appendix H.

#### 3.2.7.5 Storage Requirements

The subprograms TAPERD, RDDOTS, and WRTDOT overlay one another. The largest of the 3 routines is TAPERD which requires 8700 locations of core.

#### 3.2.7.6 Description

- TAPERD

The TAPHDR section is modified to unpack the sun angles from the header record of the MSS data tape, DATAPE. Also the pixel start number, stop number, and skip factor for line and sample is extracted from the header record.

- RDDOTS

The subprogram RDDOTS reads the interface tape DOTFIL for 3 processors, LABEL, ISOCLS, and DISPLAY. Since ISOCLS and DISPLAY need different portions of the DOTFIL information, a switch, passed via a calling argument, controls which portions of the file are returned to the calling routine.

- WRTDOT

The subprogram WRTDOT outputs the dot data file, DOTFIL, for the processors DOTDATA and LABEL.

#### 4. OPERATION

Batch oriented system, EOD-LARSYS, is operational on the Univac 1108 Exec 2 located in Building JSC12. Guidelines for job submittal are outlined in the IDSD Procedures Manual. Guidelines for program procedures for executing Procedure 1 are outlined in two documents, section 3 of this document and the User Documentation EOD-LARSYS.

## 5. TEST PROCEDURE

To verify the accuracy of the added modifications to the EOD-LARSYS system, a segment in Rice, Kansas was processed.

The entire segment was clustered with the clustering algorithm being self-initialized. The resultant unlabeled statistics were labeled by the all-of-a-kind procedure. The labeled statistics were then used by the sum-of-density classifier to classify the LACIE segment. Using bias correction dots (subset of the 209 dots), the classification results were adjusted to allow for analyst interpretation of a given dot.

Note, the shown results are to demonstrate programming accuracy only.

See appendix I for verification run.

EOD-LARSYS PROCEDURE 1

TIRF 77-0008

Job Order 71-695

APPROVAL SHEET

TEST CONDUCTOR: Barbara Hills DATE: 6/13/77  
COGNIZANT ENGINEER: Barbara Hills DATE: \_\_\_\_\_  
REQUIREMENTS: J.C. Muntz DATE: 6/13/77  
USER: Stuart Yu DATE: \_\_\_\_\_  
NASA TECHNICAL MONITOR: R.P. Heydon DATE: 6/13/77  
MAINTENANCE & OPERATIONS: \_\_\_\_\_ DATE: \_\_\_\_\_  
LACIE QUALITY ASSURANCE: \_\_\_\_\_ DATE: \_\_\_\_\_  
QUALITY ASSURANCE: \_\_\_\_\_ DATE: 6-13-77  
DEVELOPER : Barbara Hills  
P.J. Assis Jr.  
C.J. Gardner

REMARKS: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

APPENDIX A  
SUBPROGRAM DESCRIPTIONS



APPENDIX A

TABLE OF CONTENTS

- A.1 Software for Subprogram No. 1 (FLDTYP)
- A.2 Software for Subprogram No. 2 (DOTDAT)
- A.3 Software for Subprogram No. 3 (DOTS)
- A.4 Software for Subprogram No. 4 (SET13)

## A.1 SOFTWARE SUBPROGRAM NO. 1 (FLDTYP)

The subprogram FLDTYP initiates the reading of the field data set defined in section 3.2.1.3 and signals the calling routine DOTS when all the fields for a given type have been processed.

### A.1.1 LINKAGES

FLDTYP is called by the routine DOTS and calls the routine LAREAD.

### A.1.2 INTERFACES

Interfaces with other routines through the common blocks INFORM and DOTVEC.

### A.1.3 INPUTS

Calling sequence: CALL FLDTYP(FIELDS, STAMNT, \$, \$, \$, IPT, VERTEX)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
FIELDS	(4,1)	Out	Array for storing the field data:
STAMNT	1	In	Parameter for a go to Fortran statement
\$	1	In	Exit route if a field card is encountered
\$	1	In	Exit route if a TYPE card is encountered

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
\$	1	In	Exit route if a \$END* card is encountered
IPT	1	In	Pointer for FIELDS array
VERTEX	1	Out	Array for storing the field vertices

#### A.1.4 OUTPUTS

Stores the type and category names in the common block DOTVEC.

#### A.1.5 STORAGE REQUIREMENTS

Storage used: Code 212<sub>8</sub> Data 123<sub>8</sub>.

#### A.1.6 DESCRIPTION

The subprogram FLDTYP signals the calling routine when a TYPE card, field card or \$END\* card is encountered by exiting through calling arguments. If a TYPE card is encountered, the subprogram exits by calling argument number 4; if a field card is encountered, the subprogram exits by calling argument number 3; if a \$END\* card is encountered, the subprogram exits by calling argument number 5.

#### A.1.7 FLOW CHART

See figure A-1 for flow chart of the subprogram FLDTYP.

#### A.1.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

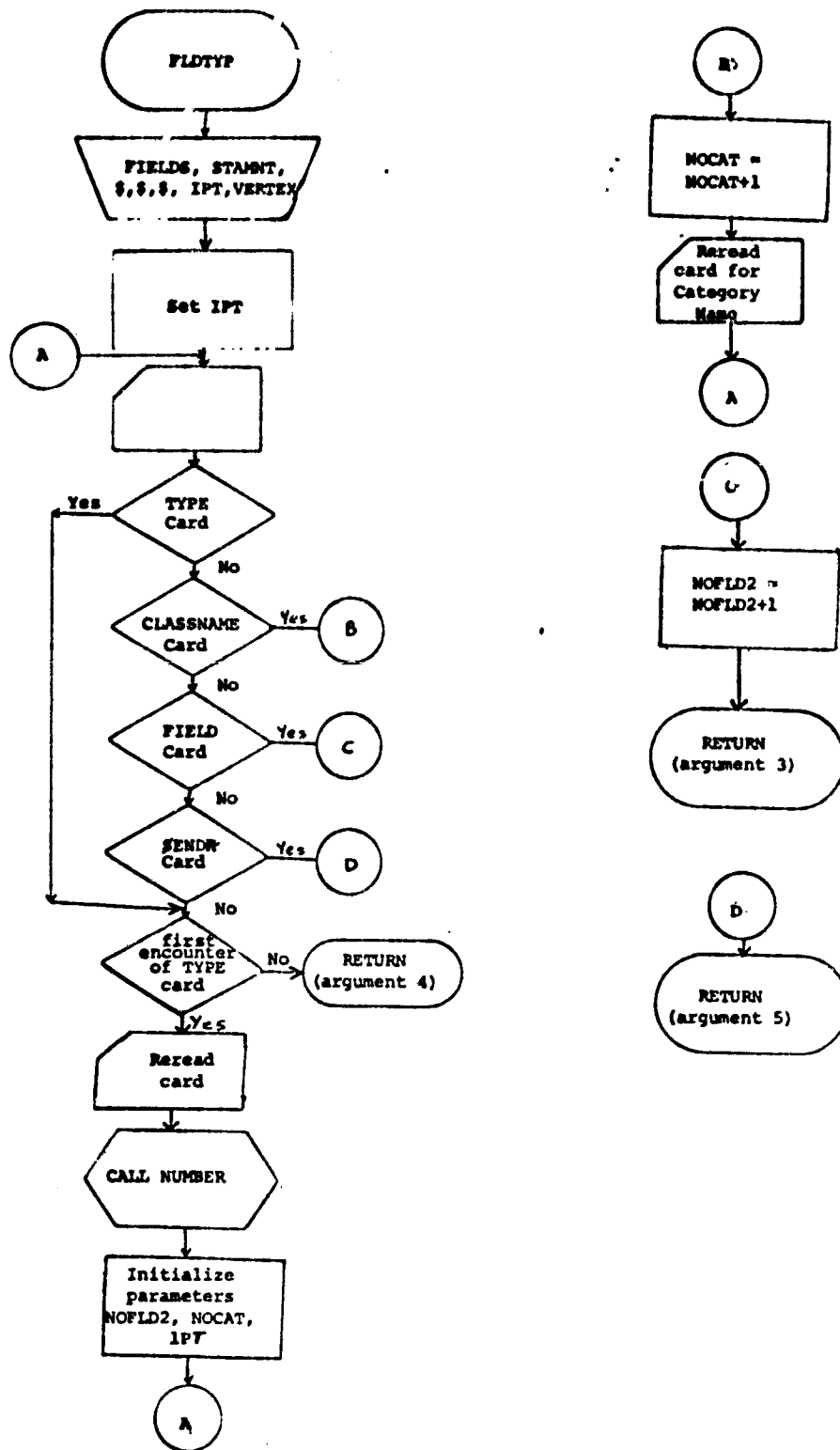


Figure A-1.- Flow chart for the subprogram FLDTYP

A-5  
62

## A.2 SOFTWARE SUBPROGRAM NO. 2 (DOTDAT)

The subprogram DOTDAT is the driver routine for the DOTDATA processor.

### A.2.1 LINKAGES

DOTDAT is called by the monitor routine MONTOR and calls the routines SET13 and DOTS.

### A.2.2 INTERFACES

N/A

### A.2.3 INPUTS

Calling sequence: CALL DOTDAT (ARRAY, TOP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10,600	In	Array in blank common dimensioned by the parameter top
TOP	1	In	TOP = 10,600

### A.2.4 OUTPUTS

N/A

### A.2.5 STORAGE REQUIREMENTS

Storage Used: Code - 6<sub>8</sub> Data - 35<sub>8</sub>

### A.2.6 DESCRIPTION

The subprogram DOTDAT calls SET13 to read and analyze the control cards. DOTS is called to coordinate the functions required to output the dot data file, DOTFIL.

A.2.7 FLOW CHART

See figure A-2 for flow chart of the subprogram DOTDAT.

A.2.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

~~A-5~~  
64

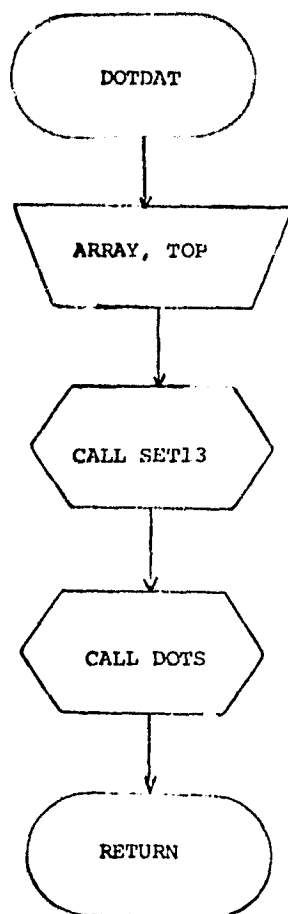


Figure A-2.— Flow chart for the subprogram DOTDAT

A-6 65

### A.3 SOFTWARE SUBPROGRAM NO. 3 (DOTS)

The subprogram DOTS is the coordinator and supervisor of the functions required to output the dot data file, DOTFIL. This file is an interface for the processors ISOCLS, LABEL and DISPLAY.

#### A.3.1 LINKAGES

DOTS is called by the routine DOTDAT and calls the routines TAPHDR, FLDINT, LINERD, FDLINT, WRTFLD, WRTDOT, and internal subprogram PRTDOT.

#### A.3.2 INTERFACES

Interfaces with other routines through the common blocks INFORM, GLOBAL, DOTVEC, and ISOLNK.

#### A.3.3. INPUTS

Calling sequence: CALL DOTS (DATA, FIELDS, VERTEX, TOP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DATA	5000	In	Array for storing the dot data.
FIELDS	1000	In	Array for storing the field data
VERTEX	1000	In	Array for storing the field vertices.
TOP	1	In	Parameter. TOP=10,600.



#### A.3.4 OUTPUTS

Optionally, prints the spatial and spectral information of the dot data file, DOTFIL, on the line printer.

#### A.3.5. STORAGE REQUIREMENTS

Storage used: Code: 754<sub>8</sub> Data: 23633<sub>8</sub>

#### A.3.6 DESCRIPTION

DOTS calls the routine FLDTYP to coordinate the reading of the field cards for a given type. For each field in type i information such as line number, sample number, type number, category number, and dot (grid point) is collected. A maximum of  $n \leq 5000 / (\text{number of channels}) \leq 250$  dots may be collected for type i.

When all the fields for type i have been processed, the routine DOTFIL will output the data to a file. A file is written for each type i.

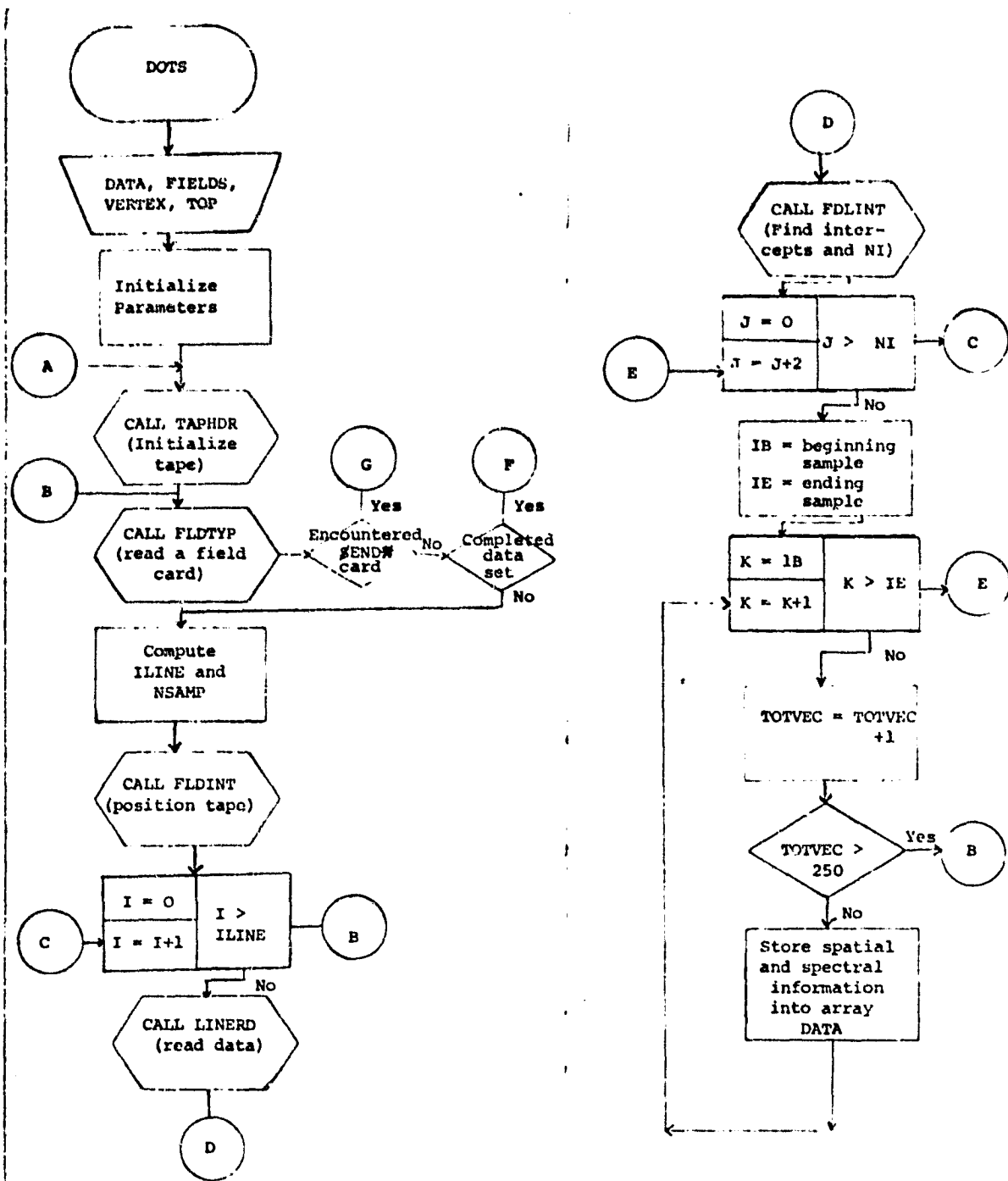
If applicable, the sun angles from the input imagery data tape will be output on the DOTFIL tape.

#### A.3.7 FLOWCHART

See figure A-3a for flow chart of the subprogram DOTS and figure A-3b for flow chart of the internal subprogram PRTDOT.

#### A.3.8 LISTING

Available in the Data Technique Laboratory, JSC Building 17.



ORIGINAL PAGE IS  
OF POOR QUALITY

Figure A-3a.- Flow chart for subprogram DOTS

7-8 68

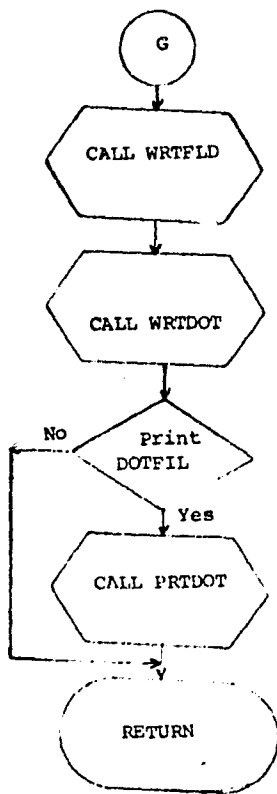
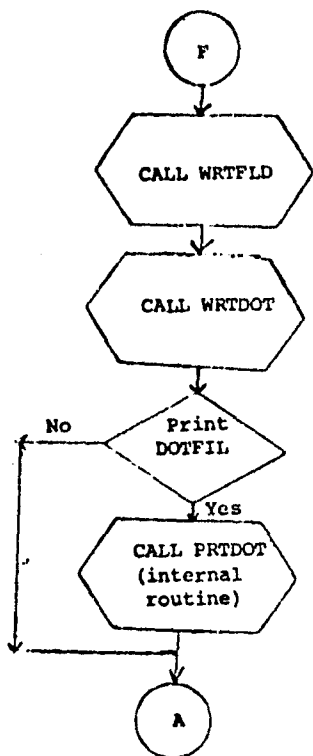


Figure A-3a.- Flow chart for subprogram DOTS (Concluded)

A-1069

ORIGINAL PAGE IS  
OF POOR QUALITY

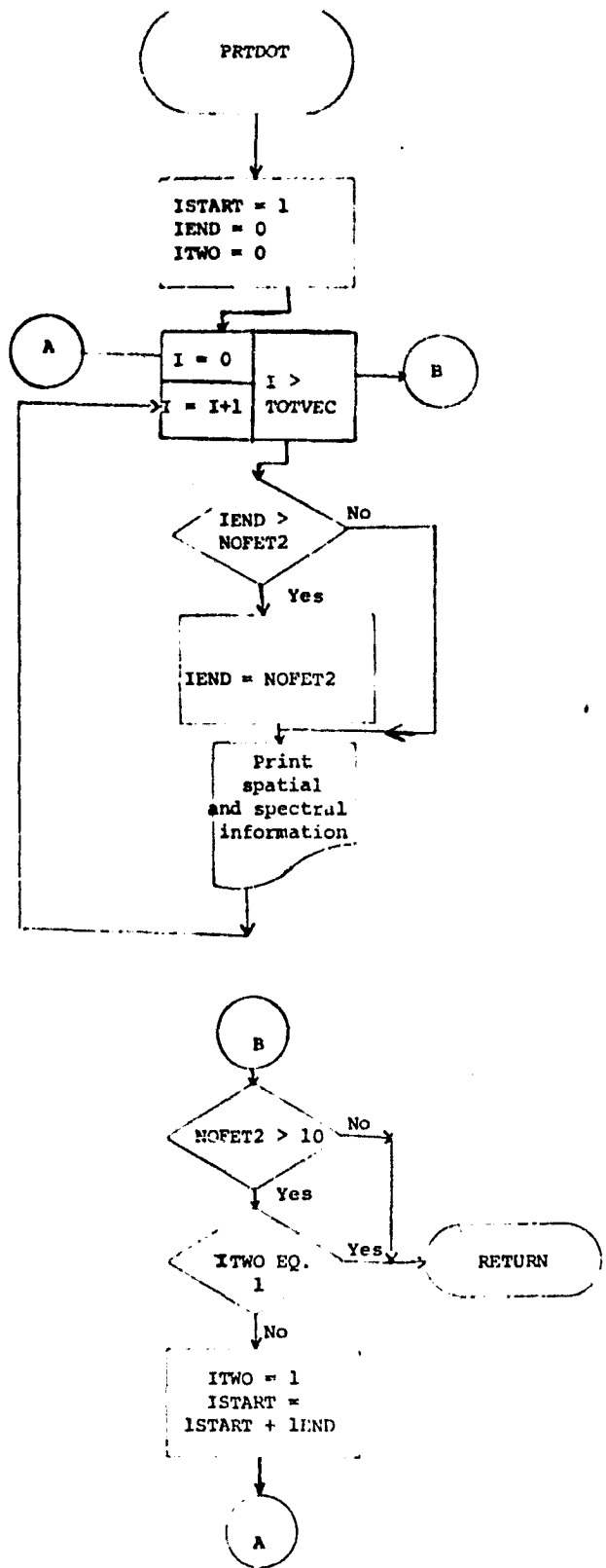


Figure A-3b.— Flow chart for internal subprogram PRD DOT

#### A.4 SOFTWARE SUBPROGRAM NO. 4 (SET13)

The subprogram SET13 reads and analyzes the control cards for the DOTDATA processor.

##### A.4.1 LINKAGES

SET13 is called by DOTDAT and calls the routines NUMBER, ORDER, NXTCHR, and FIND.

##### A.4.2 INTERFACES

Interfaces with other routine through the common blocks GLOBAL, INFORM, and DOTVEC.

##### A.4.3 INPUTS

- Calling sequence: CALL SUBROUTINE SET13  
(No calling arguments are used)
- Control Cards  
For description of control cards acceptable by this routine, see section 3.2.1.3.

##### A.4.4 OUTPUTS

A line printer summary of the selected options is printed.

Parameter values such as file switches and option switches will be stored in the common blocks.

##### A.4.5 STORAGE REQUIREMENTS

Storage used: Code - 62<sub>g</sub>, Data - 241<sub>g</sub>.

##### A.4.6 DESCRIPTION

Each control card will be read and analyzed for format errors. If the control card is valid, the appropriate parameters and

switches will be set. When applicable, if a control card for a specific option is not input, a default value will be supplied.

#### A.4.7 FLOWCHARTS

See figure A-4 for flow chart of the subprogram SET13.

#### A.4.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

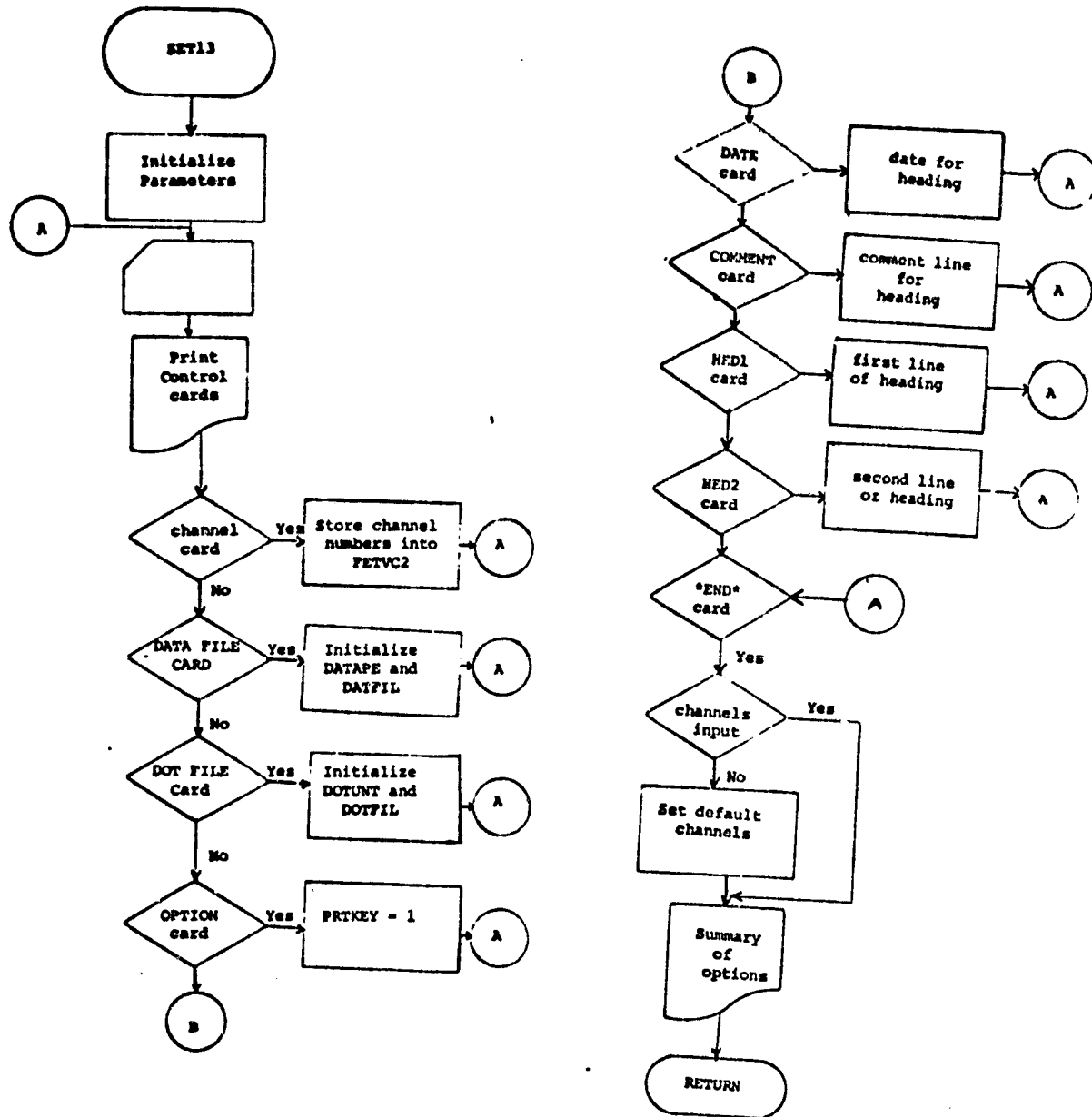


Figure A-4.- Flow chart for the subprogram SET13

APPENDIX B  
SUBPROGRAM DESCRIPTIONS



## APPENDIX B

### TABLE OF CONTENTS

B.1	Software for Subprogram No. 1 (ALLKIN)
B.2	Software for Subprogram No. 2 (CLSMAP)
B.3	Software for Subprogram No. 3 (DOTDST)
B.4	Software for Subprogram No. 4 (KNEAR)
B.5	Software for Subprogram No. 5 (LABDOT)
B.6	Software for Subprogram No. 6 (LABEL)
B.7	Software for Subprogram No. 7 (LABLR)
B.8	Software for Subprogram No. 8 (MANORD)
B.9	Software for Subprogram No. 9 (MIXMAP)
B.10	Software for Subprogram No. 10 (SET14)
B.11	Software for Subprogram No. 11 (MAPHND)
B.12	Software for Subprogram No. 12 (ASCEND)
B.13	Software for Subprogram No. 13 (CRDSCN)
B.14	Software for Subprogram No. 14 (FILERD)
B.15	Software for Subprogram No. 15 (STOMAP)
B.16	Software for Subprogram No. 16 (DSPTAP)
B.17	Software for Subprogram No. 17 (NAMSTA)
B.18	Software for Subprogram No. 18 (REODER)
B.19	Software for Subprogram No. 19 (LABMAN)
B.20	Software for Subprogram No. 20 (SUNFCS)
B.21	Software for Subprogram No. 21 (CLRKEY)
B.22	Software for Subprogram No. 22 (CNDMAP)

## B.1 SOFTWARE SUBPROGRAM NO. 1 (ALLKIN)

The subprogram ALLKIN labels the statistics obtained from the clustering processor ISOCLS by all-of-a-kind procedure.

### B.1.1 LINKAGES

The subprogram ALLKIN is called by the routine LABLR and if applicable, calls the routine KNEAR.

### B.1.2 INTERFACES

Interfaces with other routines through the common blocks INFORM and LABS.

### B.1.3 INPUTS

Calling sequence: CALL ALLKIN (DOTS, SUBVEC, SUBNO, CATVEC, MEAN)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Function</u>
DOTS	(SIZE, TOTDT2) SIZE = 4 + NOFEAT TOTDT2 = total dots on DOTFIL	IN	Array containing the dot data information
SUBVEC	NOSUB2 NOSUB2=No. of clusters	IN	Array containing the sequence no. used for ordering the output statis- tics
SUBNO	NOSUB2	OUT	Array containing the no. of clusters in each category
CATVEC	NOSUB2	OUT	Array containing the category no. each cluster was assigned during this labeling procedure

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Function</u>
MEANS	(NOFET2,1) NOFET2=No. of channels	IN	Array containing the means of the unlabeled statis- tics

#### B.1.4 OUTPUTS

A line printer summary of the labeling dots within a cluster will be printed.

#### B.1.5 STORAGE REQUIREMENTS

Storage used: Code - 772<sub>g</sub>      Data - 2262<sub>g</sub>

#### B.1.6 DESCRIPTION

For a given cluster, the labeling dots will be scanned and all the dots assigned to that cluster during the clustering process will be grouped.

The labels for the group of dots will be polled. If the labels are of one category, the cluster will be labeled that category. If more than one category is encountered, the category with the majority of the dots will label the cluster. If a given cluster does not contain any labeling dots within the cluster, the k-nearest neighbor procedure will be applied.

#### B.1.7 FLOWCHARTS

See figure B-1 for flow chart of the subprogram ALLKIN and figure B-1a for internal subprogram CHECK.

#### B.1.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

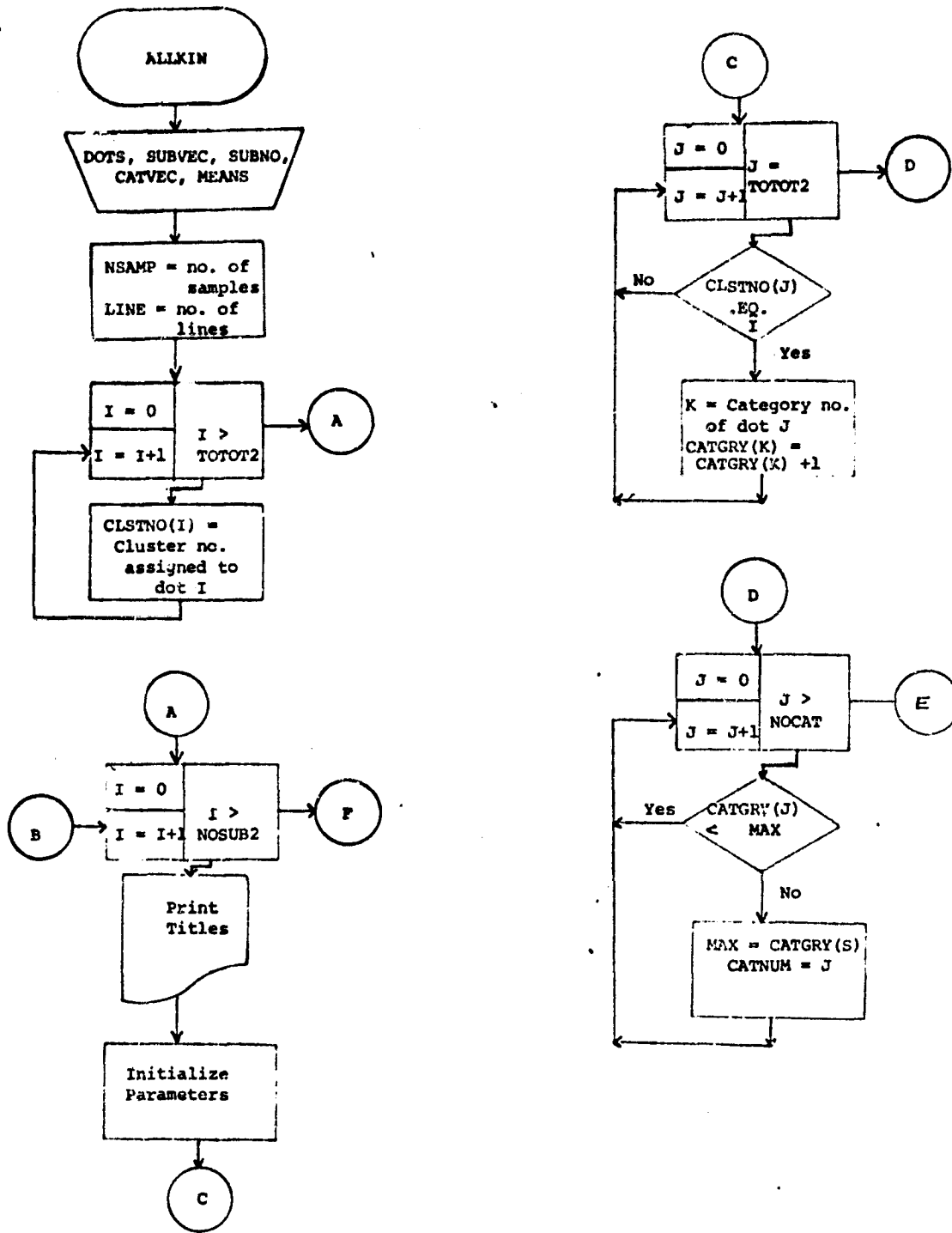


Figure B-1.— Flow chart of subprogram ALLKIN

ORIGINAL PAGE IS  
OF POOR QUALITY

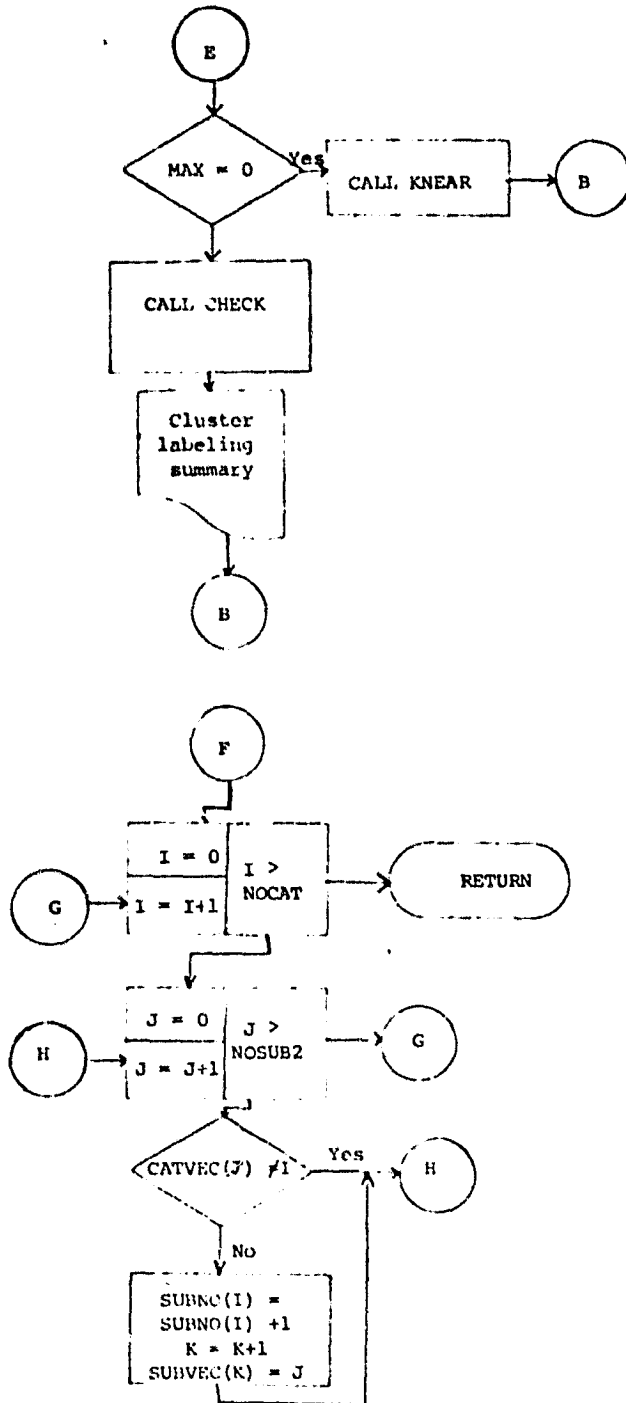
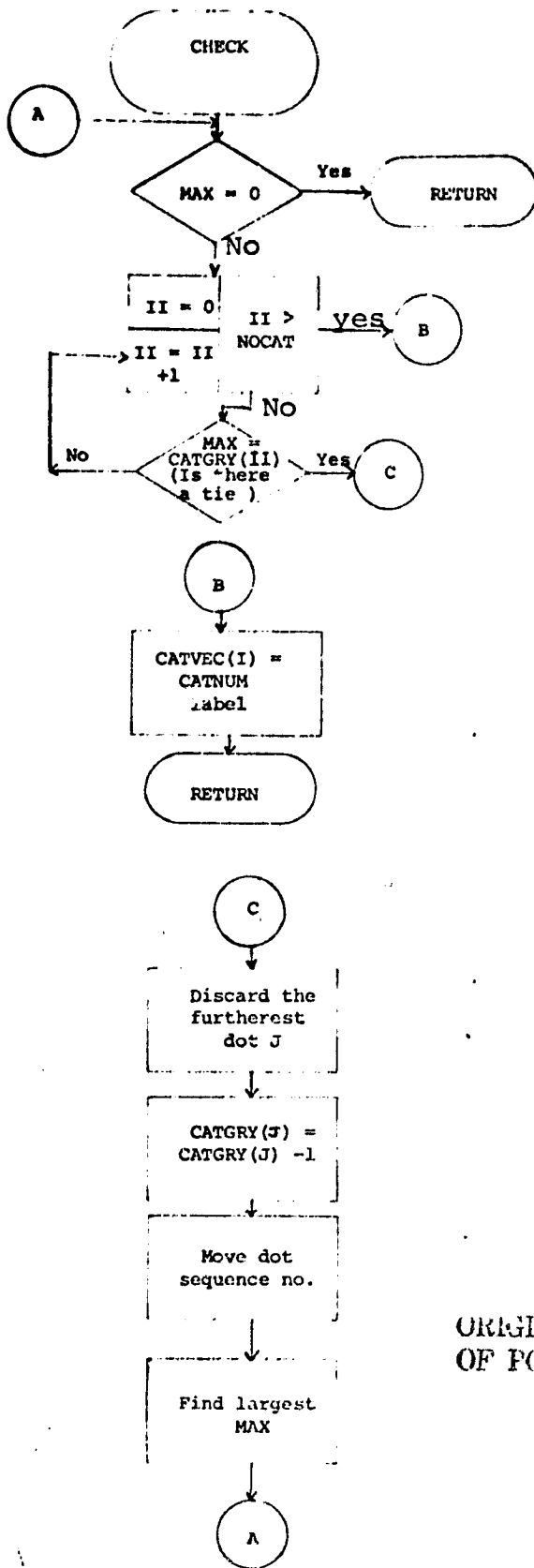


Figure B-1.— Flow chart of subprogram ALLKIN (continued)



ORIGINAL PAGE IS  
OF POOR QUALITY

Figure B-1a.— Flow chart for internal subprogram CHECK

## B.2 SOFTWARE SUBPROGRAM NO. 2 (CLSMAP)

The subprogram CLSMAP outputs conditional and/or mixed cluster maps.

### B.2.1 LINKAGES

The subprogram CLSMAP is called by the routine LABLR and calls the routines WRTHED, WRTLN, MAPHND, and CLRKEY.

### B.2.2 INTERFACES

Interfaces with other routines through the common blocks INFORM, GLOBAL and LABS.

### B.2.3 INPUTS

Calling sequence: CALL CLSMAP (CATSUB, SWTCH, SUBNO, SUBVEC, SUBDES, CATVEC)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
CATSUB	NOSUB2 NOSUB2=No. of clusters	In	Array that flags the mixed and conditional clusters
SWTCH	1	In	SWTCH=1 - conditional map is output SWTCH=2 - mixed map is output
SUBNO	NOCAT	In	Array containing the no. of clusters in a category
SUBVEC	NOSUB2	In	Array containing the sequence no. used for ordering the output statistics



SUBDES	NOSUB2	In	Array containing the cluster names
CATVEC	NOSUB2	In	Array containing the category no. each cluster was assigned

#### B.2.4 OUTPUTS

A conditional and/or cluster map will be output on the analyst specified unit. The default unit will be FORTRAN unit 16 (logical unit N). (See section 5.1 in LARSYS document for format of file). If output of both a conditional and mixed cluster map is requested, the conditional map is output first; then, the mixed map.

A line printer symbol map of the conditional and/or mixed cluster map will be printed.

#### B.2.5 STORAGE REQUIREMENTS

Code-1034<sub>g</sub> Data 7360<sub>g</sub>

#### B.2.6 DESCRIPTION

For the line printer conditional or mixed map, each unconditional labeled cluster is assigned a category symbol and each conditional or mixed cluster is assigned a unique symbol.

For the output PCF tape the unconditional labeled clusters are arranged in alphabetical order, followed by the conditional or mixed clusters, arranged by cluster number in ascending order. By a table look-up technique each cluster is assigned a color.

The previously stored unconditional map (MAPFIL) is read into core from drum a line at a time. Each pixel of the line is

assigned the appropriate predetermined symbol or color and output to the line printer and PCF tape, respectively.

#### B.2.7 FLOWCHART

See Figure B-2a and B-2b for flow chart of CLSMAP.

#### B.2.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

3-9  
24

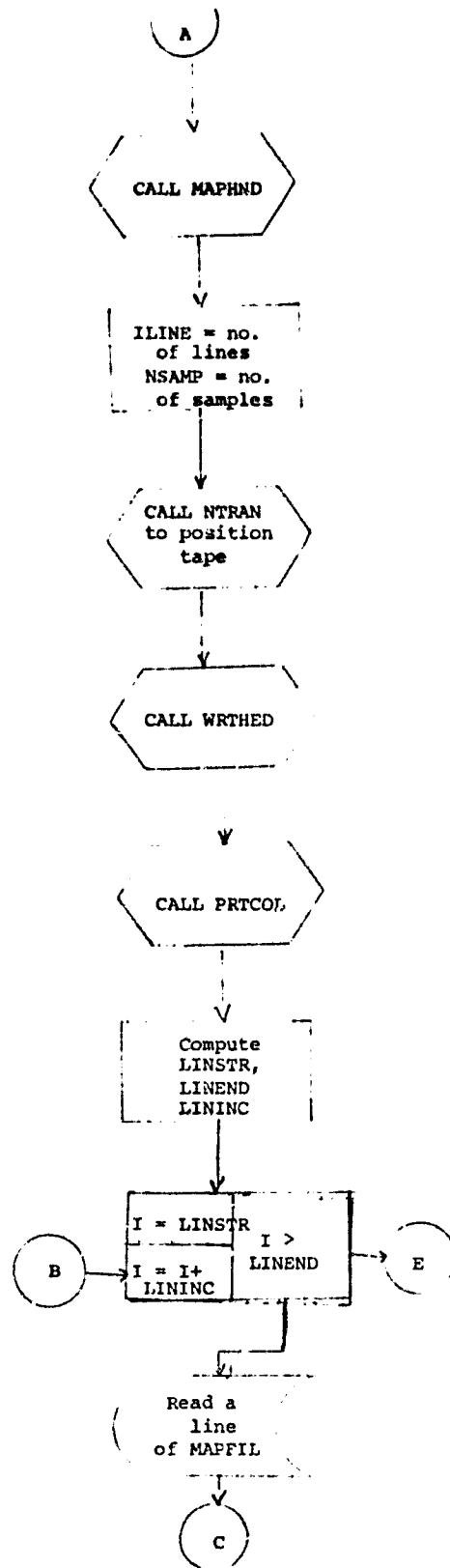
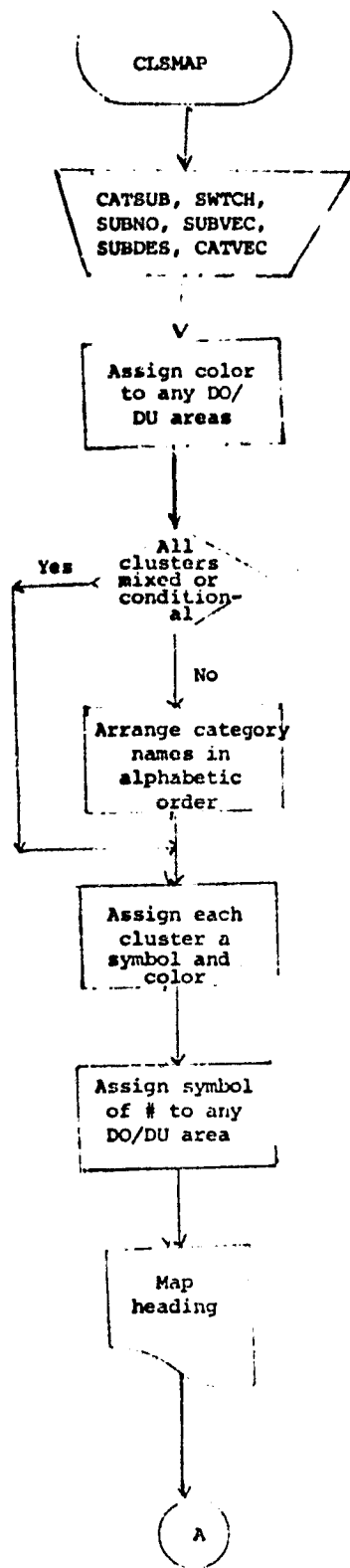


Figure B-2a.-- Flow chart for subprogram CLSMAP

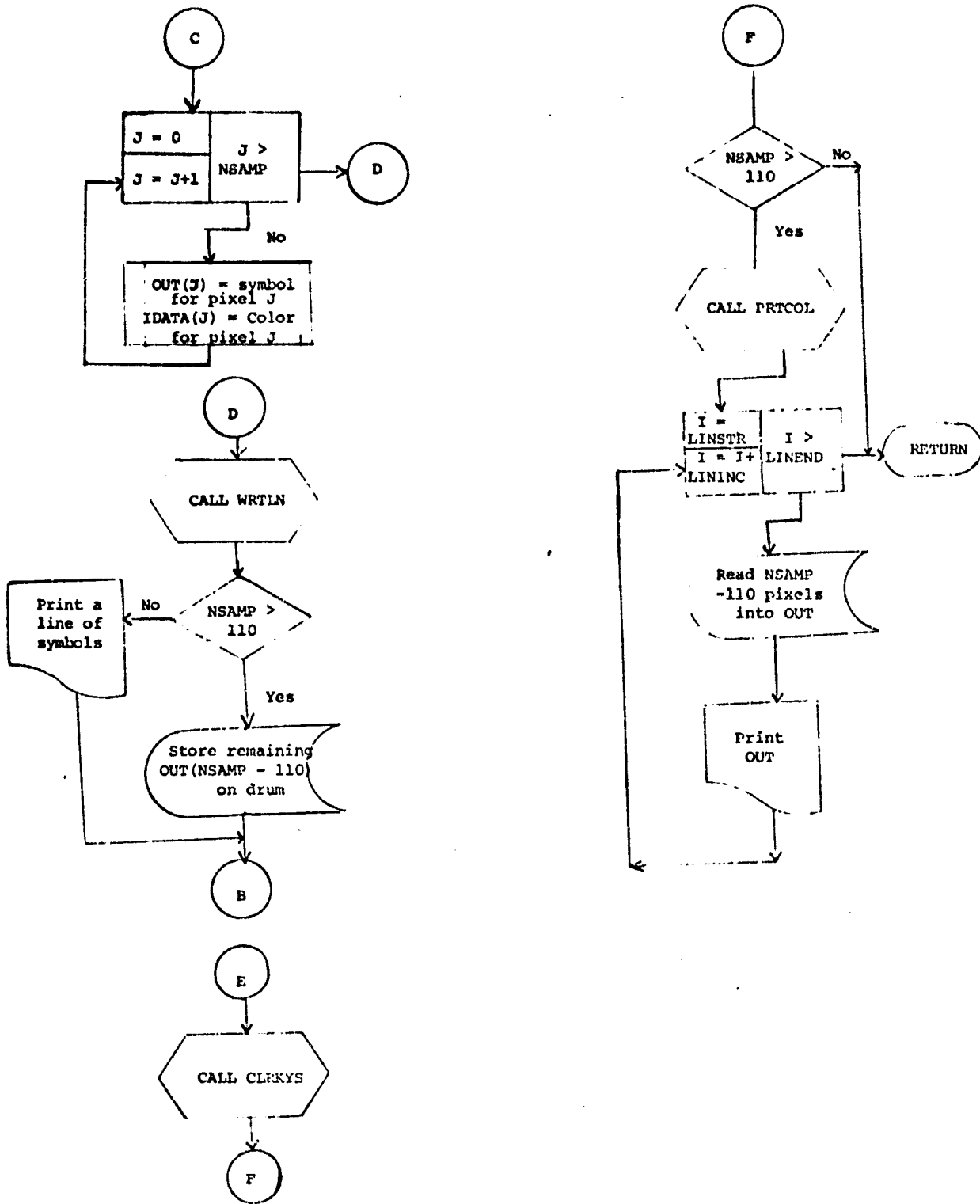


Figure B-2a.— Flow chart for subprogram CLSMAP (Continued)

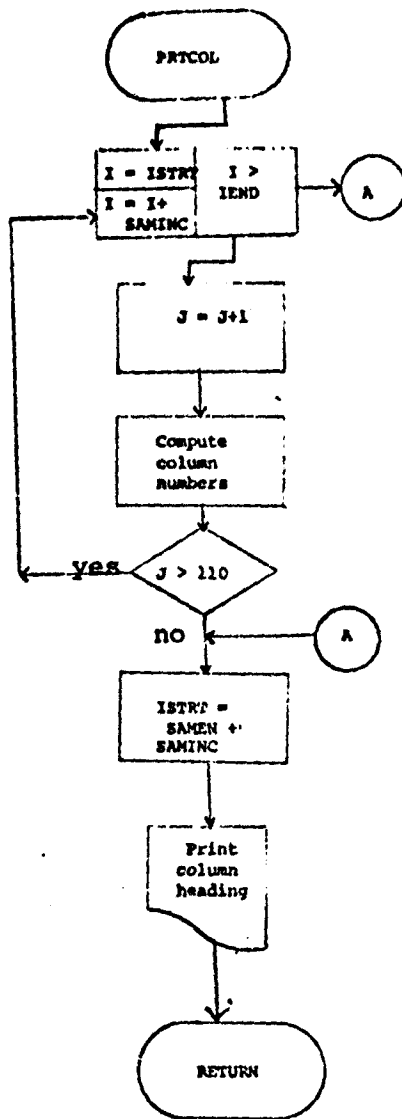


Figure B-2b.— Flow chart for PRTCOL internal subprogram

B-12  
87

### B.3 SOFTWARE SUBPROGRAM NO. 3 (DOTDST)

The subprogram DOTDST computes the inter-cluster dot distance. The distances for all the labeling dots are computed and saved on drum for later referencing.

#### B.3.1 LINKAGES

The subprogram DOTDST is called by the routine LABLR and calls the routine SUNFCS.

#### B.3.2 INTERFACES

Interfaces with other routines through the common blocks GLOBAL, INFORM, and LABS.

#### B.3.3 INPUTS

Calling sequence: CALL DOTDST (MEANS, DOTS, TABLE, TOP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
MEANS	(NOFEAT,NOSUB2)	In	Array containing the unlabeled means.
DOTS	(SIZE,250)	In	Array containing the dot data.
TABLE	(TOTDT2,1)	In	Array containing the distances of all the dots for each cluster.
TOP	1	In	Parameter that defines the maximum no. of words in TABLE.

#### B.3.4 OUTPUTS

The inter-distances table will be printed.

#### B.3.5 STORAGE REQUIREMENTS

Code-455<sub>g</sub>      Data-504<sub>g</sub>

#### B.3.6 DESCRIPTION

The distance (L1 or L2) from each dot and a cluster is computed using a default sun angle correction of 60° or using an analyst specified sun angle correction.

When the distances for a given cluster are computed, they are stored on high speed drum a group at a time. (Group = no. of labeling dots). The number of writes to drums depends upon the number of cluster means.

To print the intercluster-dot distance table, the distances are read into core, then, printed on the line printer.

#### B.3.7 FLOWCHART

See Figure B-3 for flowchart of DOTDST.

#### B.3.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

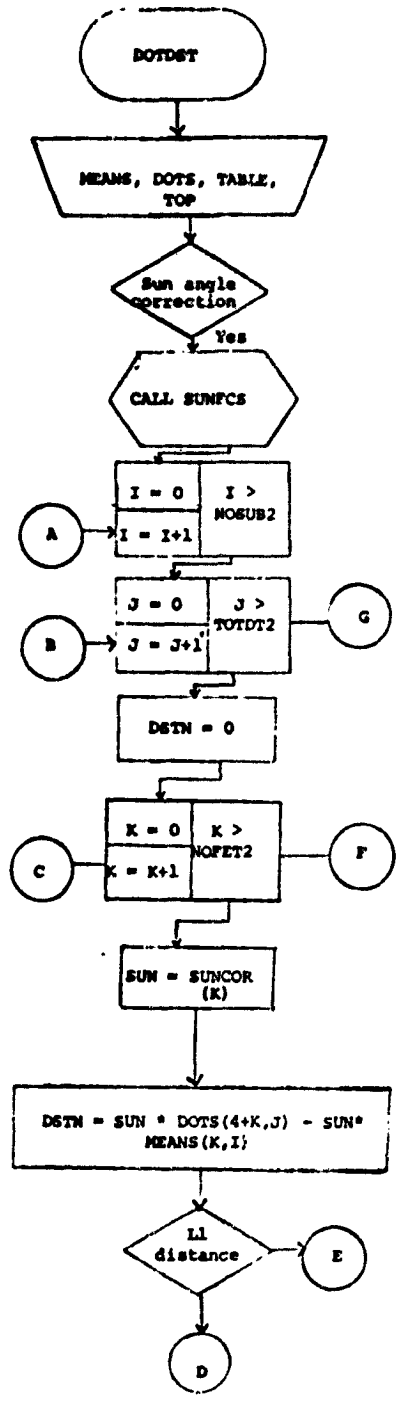
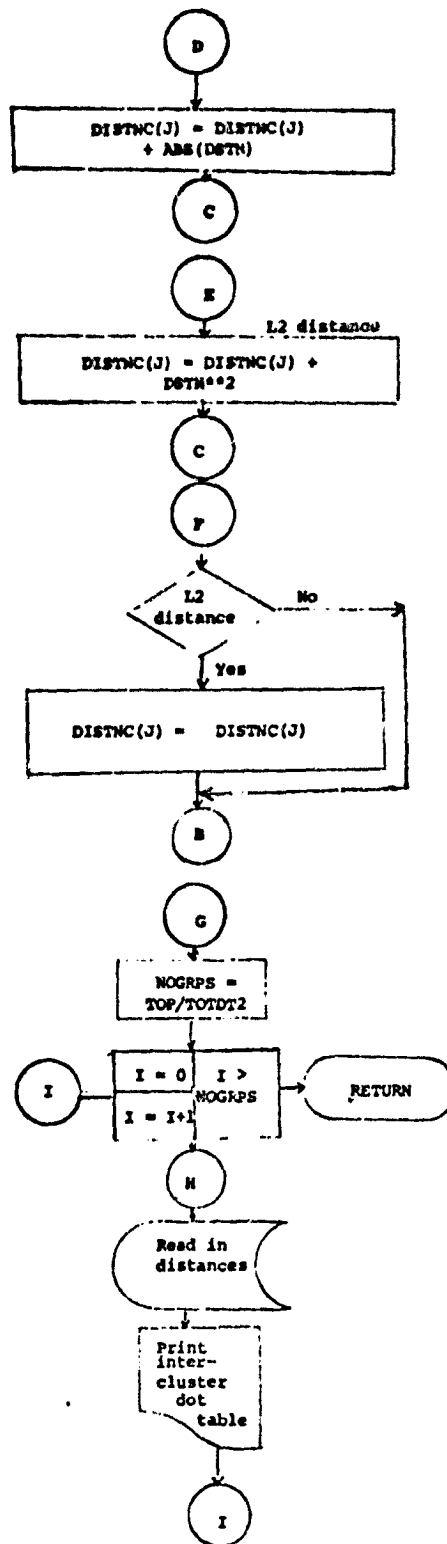


Figure B-3.- Flow chart for DOTDST subprogram

~~B-15~~  
90





ORIGINAL PAGE IS  
OF POOR QUALITY

Figure B-3 - Flowchart for subprogram  
DOTDST (concluded)

#### B.4 SOFTWARE SUBPROGRAM NO. 4 (KNEAR)

The subprogram KNEAR labels the statistics obtained from the clustering processor ISOCLS by the k-nearest neighbor procedure.

##### B.4.1 LINKAGES

The subprogram KNEAR is called by the routine LABLR and does not call any LARSYS routines.

##### B.4.2 INTERFACES

Interfaces with other routines through the common blocks GLOBAL, INFORM, and LABS.

##### B.4.3 INPUTS

Calling sequence: CALL KNEAR (DOTS, SUBVEC, SUBNO, CATVEC, ITER, TAB1, SWTCH, CATNUM, CLUNUM, MEANS)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DOTS	(SIZE,TOTDT2)	In	Array containing the dot data information
SUBVEC	NOSUB2	In	Array containing the sequence no. used for ordering the output statistics.
SUBNO	NOSUB2	Out	Array containing the no. of clusters in each category
CATVEC	NOSUB2	Out	Array containing the category no. each cluster was assigned during this labeling procedure

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ITER	1	In	Number of iterations
TAB1	1	In	Starting drum address for retrieving distances
SWTCH	1	In	SWTCH = 0
CATNUM	1	In	Category number cluster CLUNUM was assigned
CLUNUM	1	In	Number of unlabeled cluster being labeled
MEANS	(NOFET2,1)	In	Array containing the means

#### B.4.4 OUTPUTS

A line printer summary of the k-nearest labeling dots is printed.

#### B.4.5 STORAGE REQUIREMENTS

Storage used: Code-613<sub>8</sub> Data- 1721<sub>8</sub>

#### B.4.6 DESCRIPTION

For a given cluster, a poll of the category labels of the k-nearest dots to the cluster is computed. The label with the majority of the dots labels the cluster. If a tie occurs, k-1 dots used. The default value of k is 1.

B.4.7 FLOWCHART

See figure B-4 for flowchart of KNEAR subprogram.

B.4.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

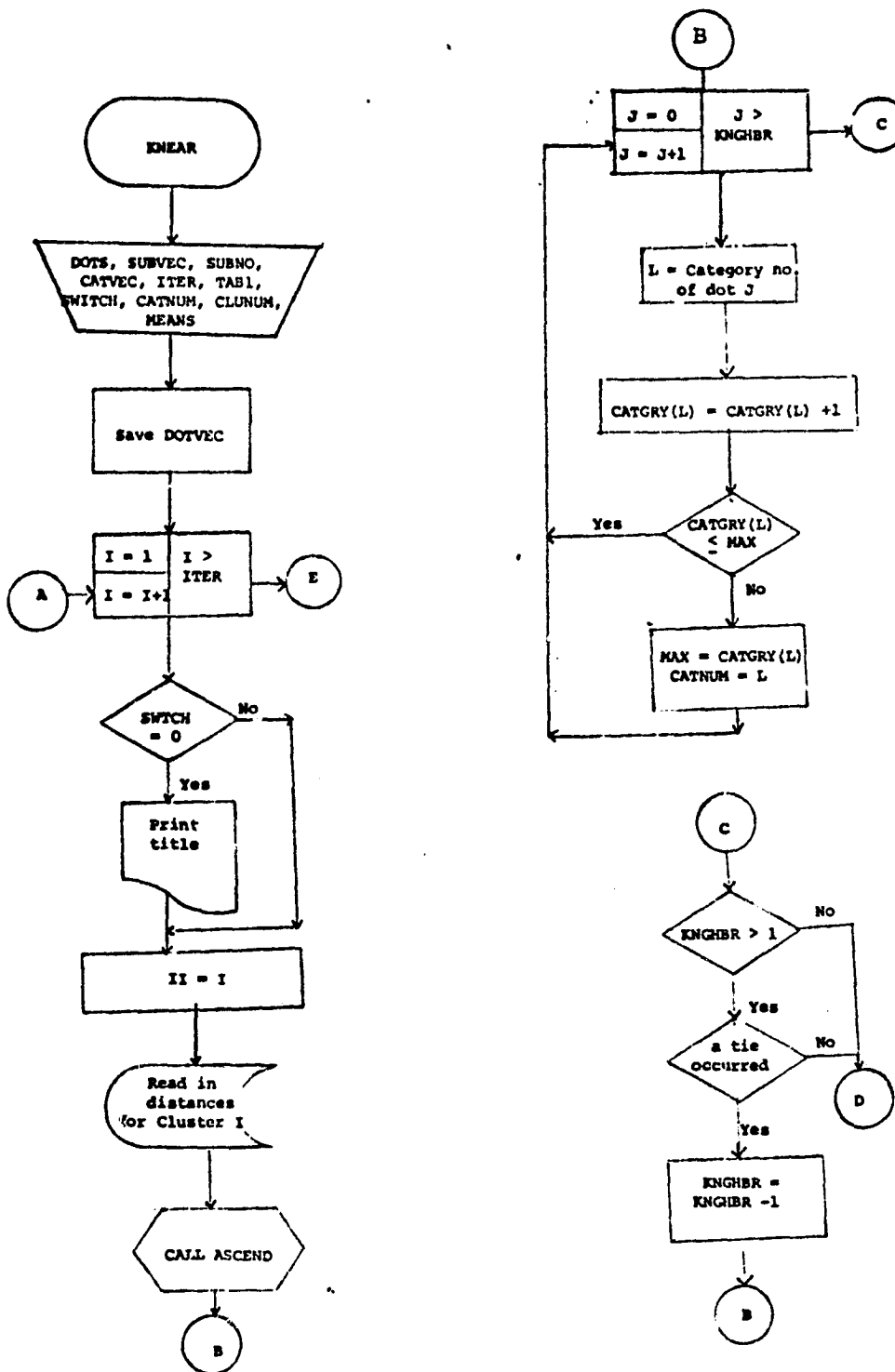


Figure B-4.—Flow chart for KNEAR subprogram

B-20  
95

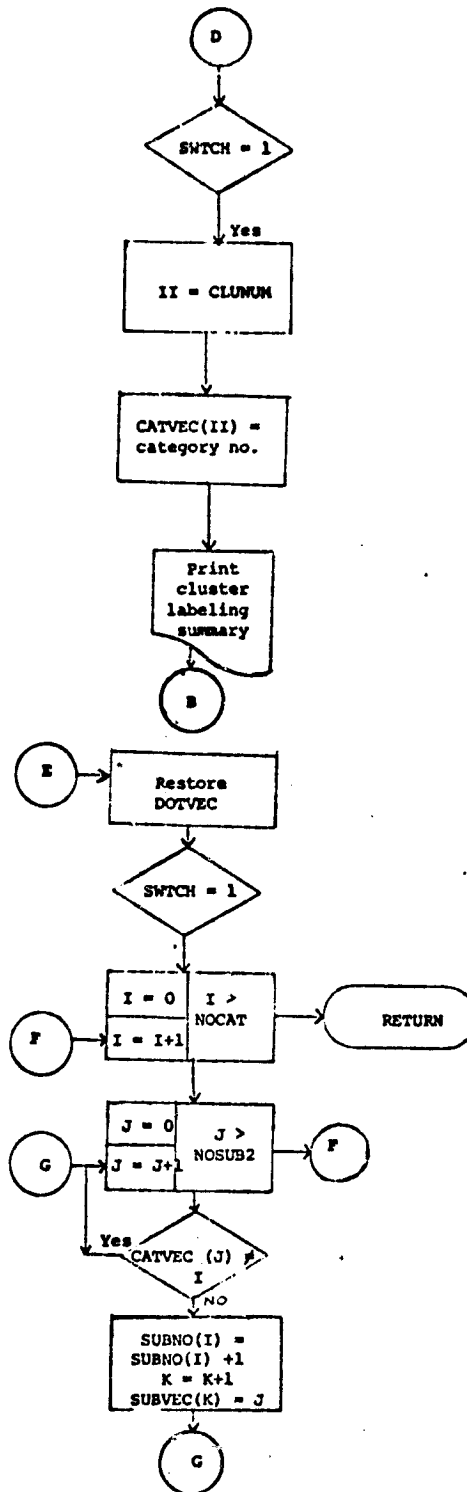


Figure B-4.- Flow chart for KNEAR subprogram (Continued)

## B.5 SOFTWARE SUBPROGRAM NO. 5 (LABDOT)

The subprogram LABDOT labels or relabels the dot data file, DOTFIL.

### B.5.1 LINKAGES

The subprogram LABDOT is called by the routine LABLR and does not call any LARSYS routines.

### B.5.2 INTERFACES

Interfaces with other routines through the common block LABS.

### B.5.3 INPUTS

Calling sequence: CALL LABDOT (DOTS)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DOTS	(SIZE,1)	In	Array containing the dot data information

### B.5.4 OUTPUTS

N/A

### B.5.5 STORAGE REQUIREMENTS

Storage used: Code-113<sub>g</sub>      Data-120<sub>g</sub>

### B.5.6 DESCRIPTION

The category names input through control cards are checked for new name entries. If a new category has been input, the arrays concerning the category information is adjusted.

Each dot number input through a control card in the setup routine is relabeled to the appropriate label. No dots may be deleted

from the set of dots being updated.

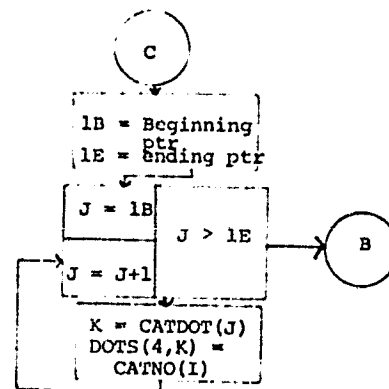
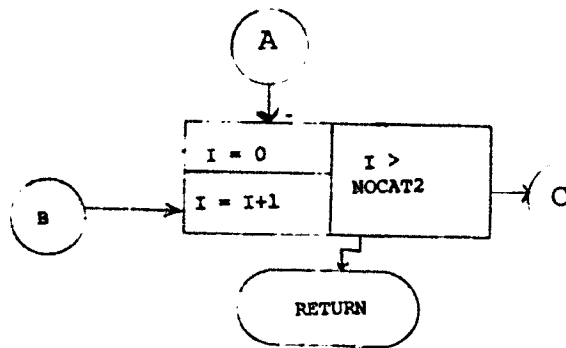
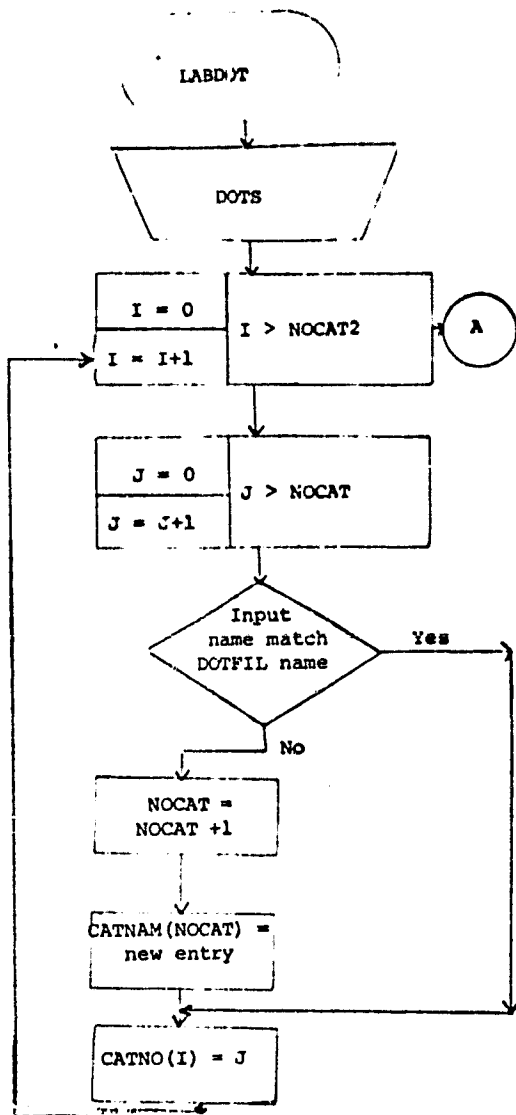
#### B.5.7 FLOWCHARTS

See figure B-5 for flowchart of LABDOT subroutine.

#### B.5.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.





ORIGINAL PAGE IS  
OF POOR QUALITY

Figure B-5.— Flow chart for subprogram LABDOT

## B.6 SOFTWARE SUBPROGRAM NO. 6 (LABEL)

The subprogram LABEL is the driver routine for the LABEL processor.

### B.6.1 LINKAGES

The subprogram LABEL is called by the monitoring routine MONTOR and calls routines SET14, LABLR, and FILERD.

### B.6.2 INTERFACES

Interfaces with other routine through the common blocks INFORM and LABS.

### B.6.3 INPUTS

Calling sequence: CALL LABEL (ARRAY, TOP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10,600	In	Array in blank common.
TOP	1	In	Parameter. TOP=10,600

### B.6.4 OUTPUTS

N/A

### B.6.5 STORAGE REQUIREMENTS

Storage used: Code-3731<sub>8</sub>      Data-53<sub>8</sub>

### B.6.6 DESCRIPTION

LABEL calls the subprogram SET14 to read and analyze the control cards, calls FILERD to read in the specified files, and calls LABLR to coordinate the functions required to execute the options specified by the analyst.

B.6.7 FLOWCHART

See Figure B-6 for flowchart of LABEL.

B.6.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

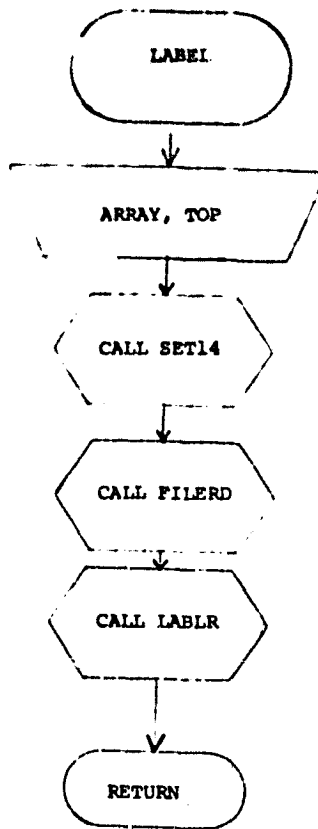


Figure B-6.- Flowchart for Subprogram LABEL.

*B-2T 109*

## B.7 SOFTWARE SUBPROGRAM NO. 7 (LABLR)

The subprogram LABLR is the coordinator and supervisor of the routines required to perform the operations specified by the analyst.

The analyst may either update the DOTFIL and/or SAVTAP files or label a given set of statistics by one of two procedures (k-nearest neighbor or all-of-a-kind) in one execution of the LABEL processor.

### B.7.1 LINKAGES

The subprogram LABLR is called by the routine LABEL and calls routines LABMAN, LABDOT, KNEAR, ALLKIN, CNDMAP, MIXMAP, MANORD, REODER, WRTDOT, DOTDST, MSTA, and DSPTAP.

### B.7.2 INTERFACES

Interfaces with routines through the common blocks INFORM, GLOBAL, and LABS.

### B.7.3 INPUTS

Calling sequence: CALL LABLR (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV, VERTX, MEANS)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10,600	In	Array in blank common
TOP	1	In	Parameter. TOP=10,600
NOFLD	1	In	No. of fields on the DOTFIL
TOTVRT	1	In	No. of vertices of fields on DOTFIL

VERTX	(2,TOTVRT)	In	Array containing the field vertices of fields on DOTFIL
MEANS	(NOFET2,1)	In	Array for storing the means

#### B.7.4 OUTPUTS

N/A

#### B.7.5 STORAGE REQUIREMENTS

Storage used: Code-1117<sub>8</sub>      Data-7504<sub>8</sub>

#### B.7.6 DESCRIPTION

Depending on the analyst selected options LABLR supervises the following functions:

- Relabeling of the SAVTAP file
- Relabeling of the DOTFIL file
- Execution of the k-nearest neighbor procedure
- Execution of the all-of-a-kind procedure
- Output of the conditional cluster map
- Output of the mixed cluster map
- Output of the DISPLAY interface tape, MAPTAP

#### B.7.7 FLOWCHART

See figure B-7a and B-7b for flowchart of LABLR subprogram.

#### B.7.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

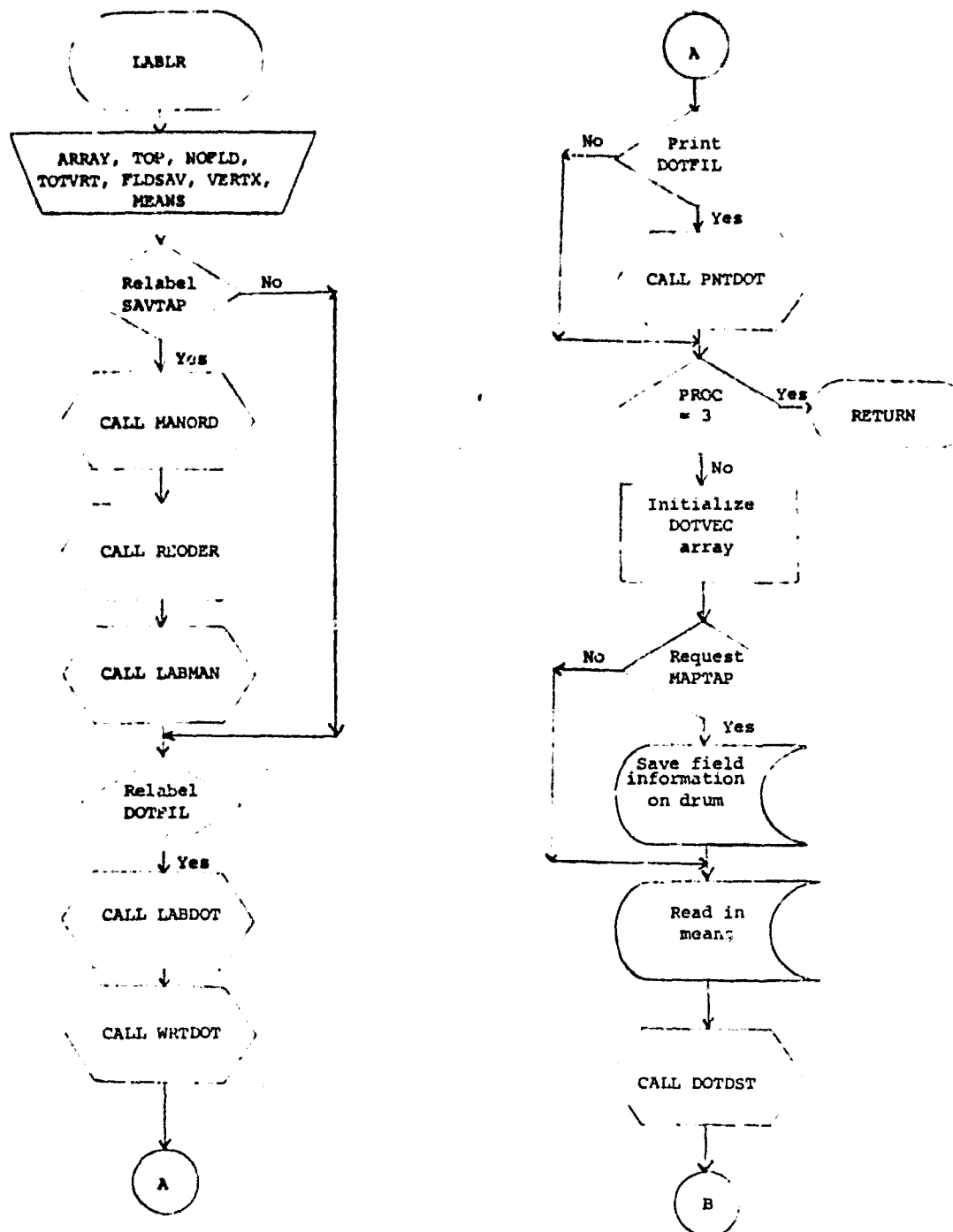


Figure B-7a.— Flow chart for subprogram LABLR.

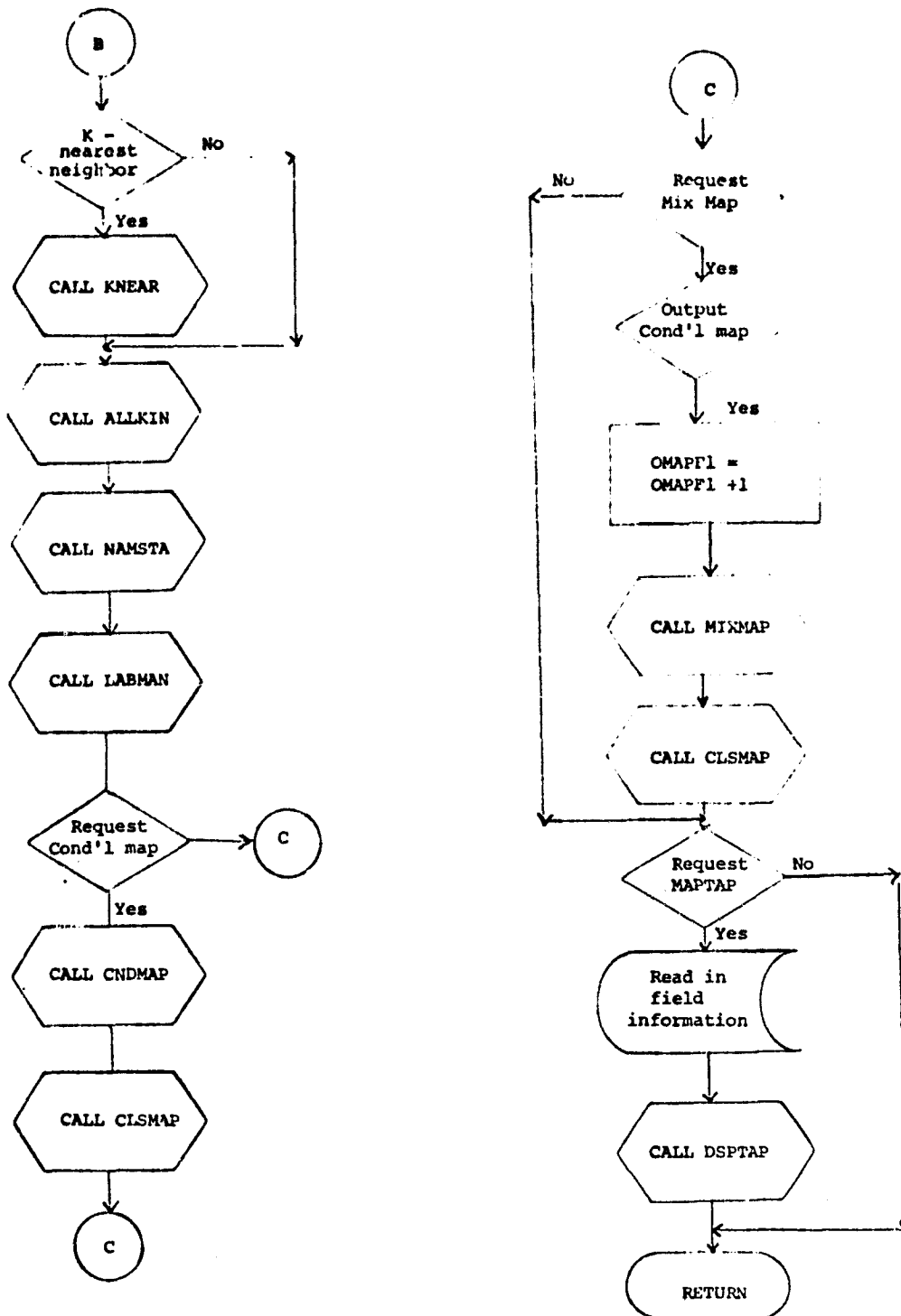


Figure B-7a.- Flow chart for subprogram LABLR (Continued)



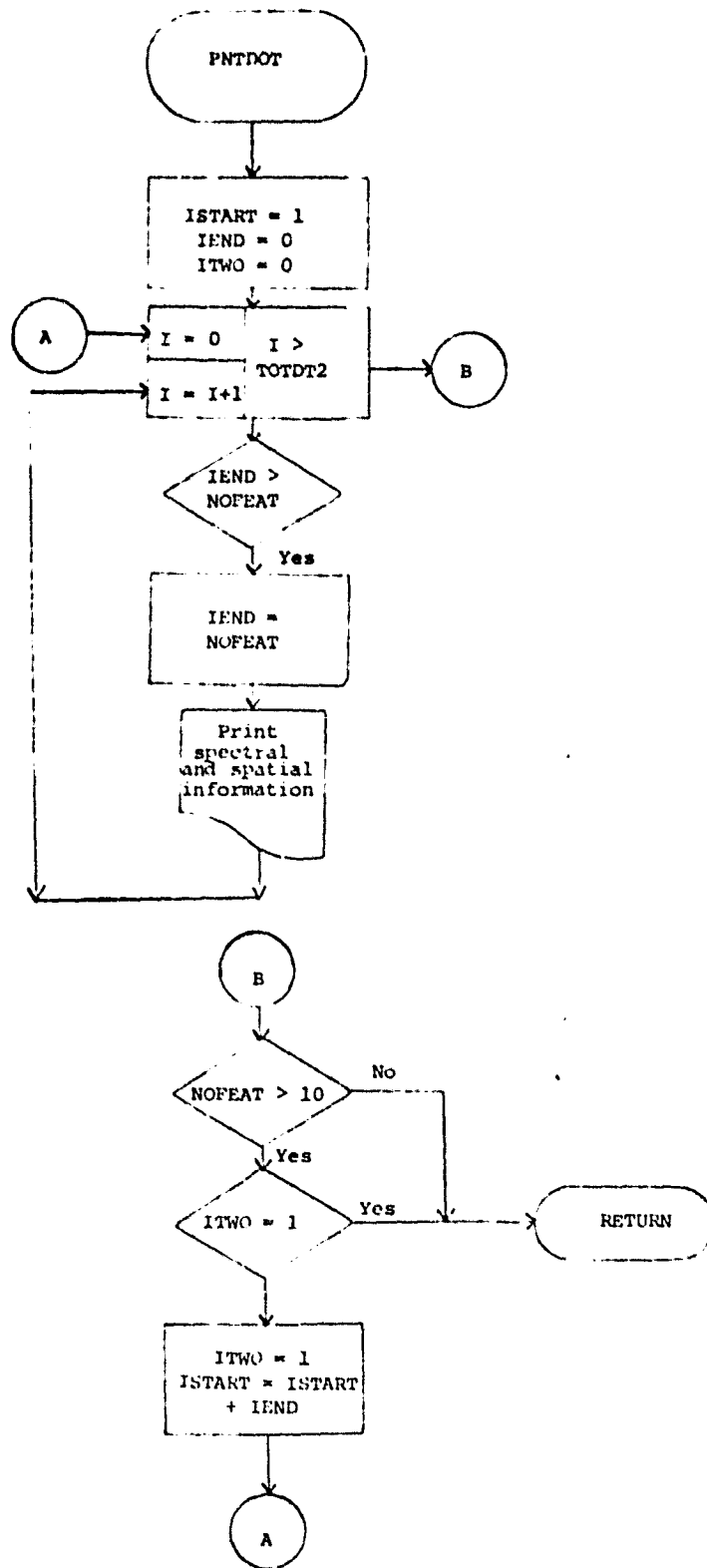


Figure B-7b.- Flow chart for internal subprogram PNTDOT

## B.8 SOFTWARE SUBPROGRAM NO. 8 (MANORD)

The subprogram LABMAN relabels the statistics file, SAVTAP, by input of the control card STATLABEL.

### B.8.1 LINKAGES

The subprogram MANORD is called by the routine LABLR and does not call any LARSYS routines.

### B.8.2 INTERFACES

Interfaces with other routines through the common block IABS.

### B.8.3 INPUTS

Calling sequence: CALL LABMAN (CLSNAM, CLSVEC, SUBVEC, NOCLS2, SUBNO, NOSUB2)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Dimension</u>
CLSNAM	NOCLS2	In	Array containing class names (category names) on SAVTAP file.
CLSVEC	NOSUB2	In	Array containing the class-subclass correspondence.
SUBVEC	NOSUB2	Out	Array containing the sequence no. used for ordering the output statistics.
NOCLS2	1	In	No. of class names on SAVTAP file.

SUBNO	60	Out	Array containing the no. of clusters in each class (category).
NOSUB2	1	In	No. of clusters

#### B.8.4 OUTPUTS

N/A

#### B.8.5 STORAGE REQUIREMENTS

Storage used: Code-235<sub>g</sub> Data-166<sub>g</sub>

#### B.8.6 DESCRIPTION

The analyst may manually relabel the statistics file by input of control cards. The classnames (categories) input must match a name on the statistics file.

The analyst may regroup any clusters into another class. For example, if the cluster WHT1 belonged to the class WHEAT and the analyst decided that the cluster should belong to the NONWHT class, he may force the cluster WHT1 to be assigned to the class NONWHT.

As a result of regrouping the clusters, the order of the statistics on the file are rearranged, but the regrouped cluster are not be renamed.

#### B.8.7 FLOWCHART

See figure B-8 for flowchart of MANORD subprogram.

#### B.8.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

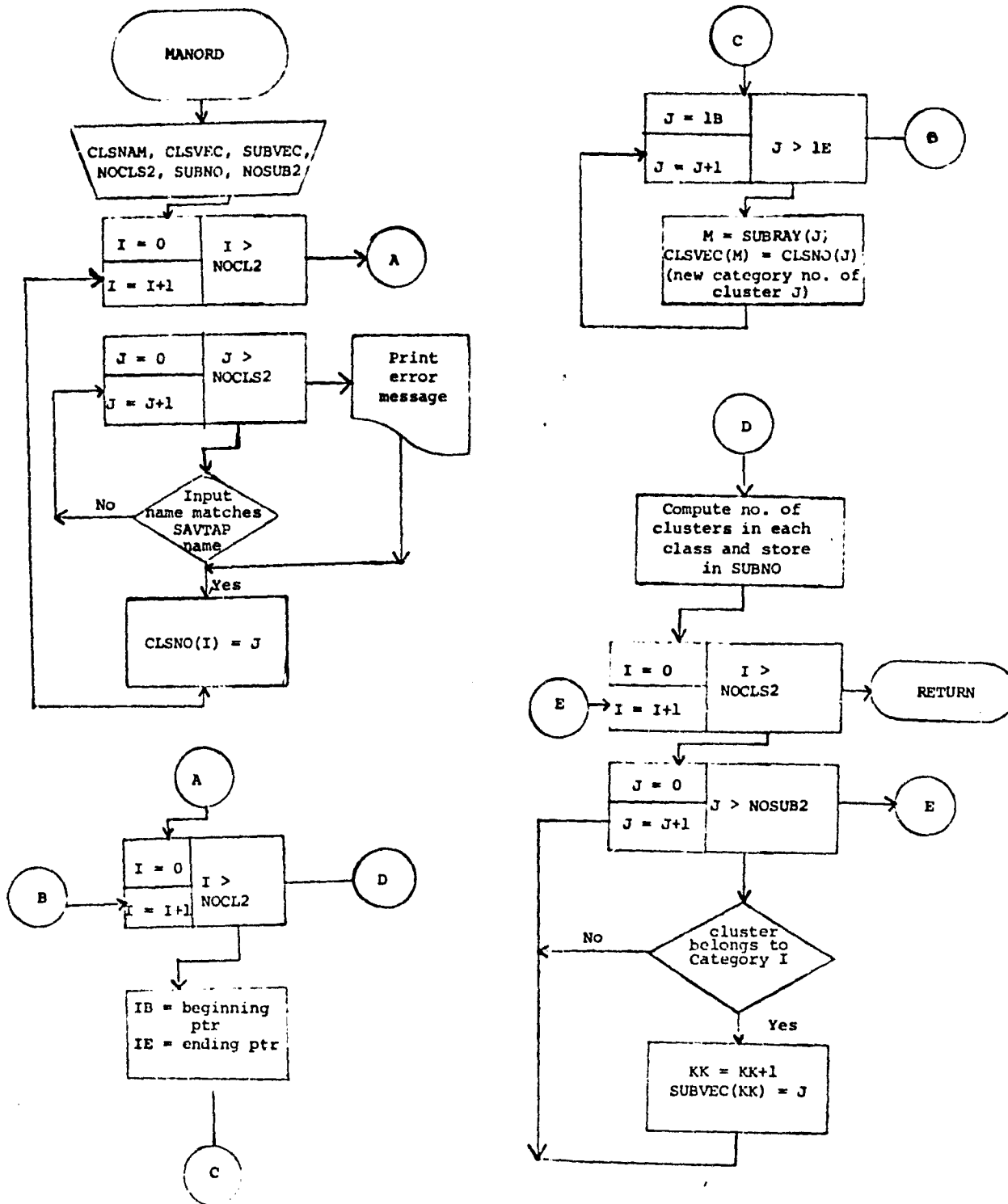


Figure B-8.— Flow chart for subprogram MANORD

## B.9 SOFTWARE SUBPROGRAM NO. 9 (MIXMAP)

The subprogram MIXMAP flags the conditional clusters.

### B.9.1 LINKAGES

The subprogram MIXMAP is called by the routine LABLR and does not call any LARSYS routines.

### B.9.2 INTERFACES

Interfaces with other routines through the common blocks GLOBAL and LABS.

### B.9.3 INPUTS

Calling sequence: CALL MIXMAP (DOTS, MIXSUB, NOSUB2, CATVEC)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DOTS	(SIZE,TOTDT2)	In	Array containing the dots.
MIXSUB	NOSUB2	Out	Array used for flagging a mixed cluster on the output mixed map.
NOSUB2	1	In	No. of clusters.
CATVEC	NOSUB2	In	Array containing the category no. of each cluster.

### B.9.4 OUTPUTS

N/A

### B.9.5 STORAGE REQUIREMENTS

Code-156<sub>8</sub>      Data-427<sub>8</sub>

#### B.9.6 DESCRIPTION

The cluster number assigned to each dot during clustering is retrieved from high speed drum. All the dots belonging to cluster I are collected, and their labels are polled. If the labels define more than one category, cluster I is flagged as a mixed cluster. A mixed cluster is assigned a cluster of 62-J where J = no. of mixed clusters encountered at that time.

#### B.9.7 FLOWCHART

See Figure B-9 for flowchart of MIXMAP.

#### B.9.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

~~B-37~~  
1123

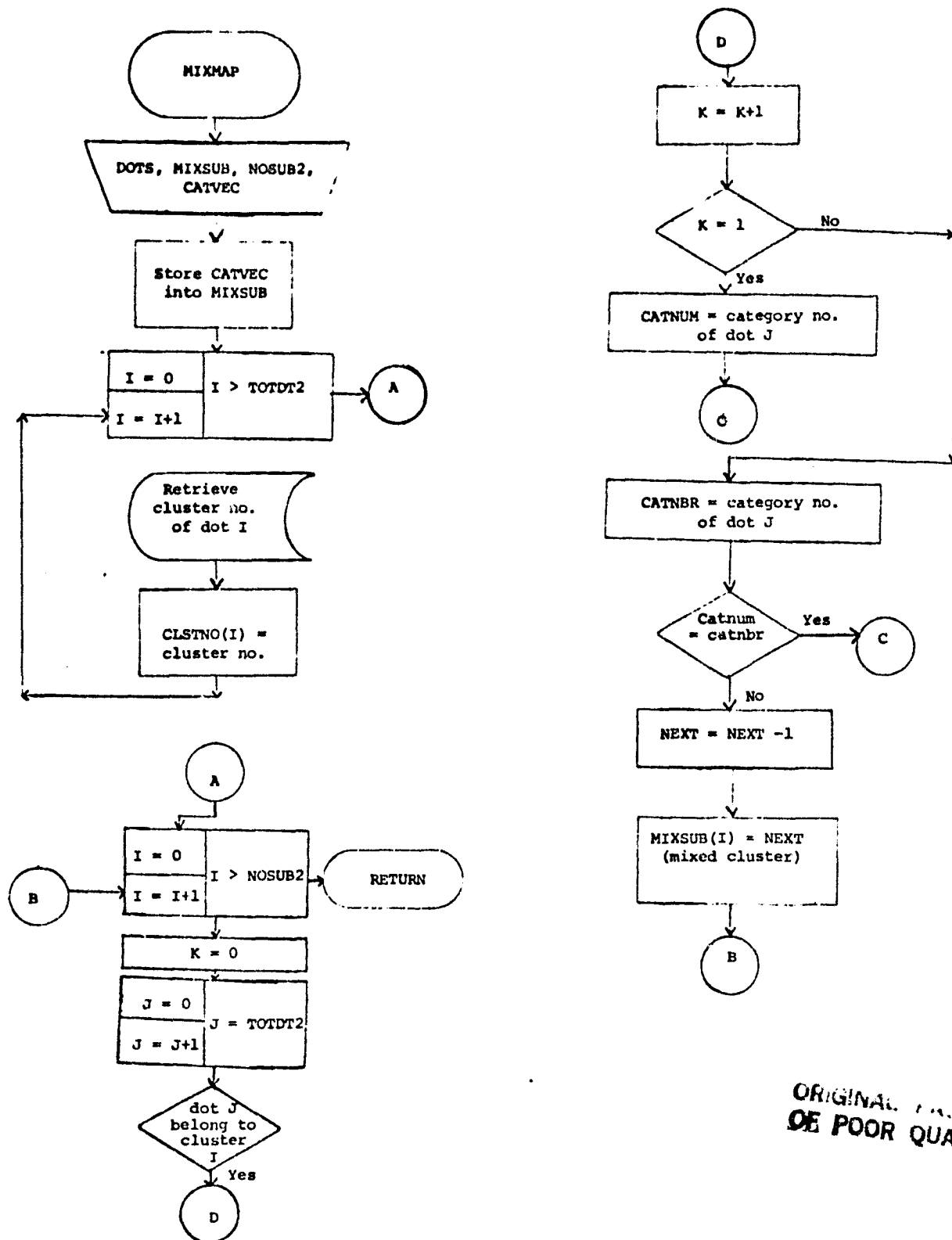


Figure B-9.— Flow chart for subprogram MIXMAP

ORIGINAL FILED  
OF POOR QUALITY

## B.10 SOFTWARE SUBPROGRAM NO. 10 (SET14)

The subprogram SET14 reads and analyzes the control cards for the LABEL processor.

### B.10.1 LINKAGES

The subprogram SET14 is called by the driver routine LABEL and calls the routines NUMBER, ORDER, NXTCHR, FIND, CRDSCN, FLTNUM, and CRDSTA.

### B.10.2 INTERFACES

Interfaces with other routines through the common blocks GLOBAL, INFORM, and LABS.

### B.10.3 INPUTS

- Calling sequence: CALL SET14 (ARRAY, TOP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10,600	In	Array in blank common.
TOP	1	In	Parameter.TOP=10,600

- Control Cards

For description of control cards acceptable by this routine, see section 3.2.2.3.

### B.10.4 OUTPUTS

A line printer summary of the selected options is printed.

Parameter values such as file switches and option switches are stored in the common blocks INFORM, GLOBAL, and LABS.



#### B.10.5 STORAGE REQUIREMENTS

Storage used: Code-2550<sub>8</sub>      Data-575<sub>8</sub>

#### B.10.6 DESCRIPTION

Each control card is read and analyzed for format errors. If the control card is valid, the appropriate parameters and switches are set. When applicable, if a control card for a specific option is not input, a default value is supplied.

#### B.10.7 FLOWCHART

See figure B-10 for flowchart of SET14 subprogram.

#### B.10.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

## B.11 SOFTWARE SUBPROGRAM NO. 11 (MAPHND)

The subprogram MAPHND prints out the heading for the conditional and mixed cluster maps.

### B.11.1 LINKAGES

The subprogram MAPHND is called by the routine CLSMAP.

### B.11.2 INTERFACES

Interfaces with other routines through the common block INFORM.

### B.11.3 INPUTS

Calling sequence: CALL MAPHND (NOCAT, CLSSYM, CATNAM, KATNO, SUBDES, CATSUB)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Function</u>
NOCAT	1	In	No. of categories.
CLSSYM	NOSUB2	In	Array containing the symbols.
CATNAM	NOCAT	In	Array containing the category names.
KATNO	NOSUB2		Array containing the category no. each cluster was assigned.
SUBDES	NOSUB2	In	Array containing the cluster names.
CATSUB	NOSUB2	In	Array containing the flagged conditional or mixed clusters

#### B.11.4 OUTPUTS

N/A

#### B.11.5 STORAGE REQUIREMENTS

Storage used: Code-175<sub>g</sub> Data-130<sub>g</sub>

#### B.11.6 DESCRIPTION

All the clusters for a category are grouped together. Information such as the category name, cluster number, cluster name, and symbol for the cluster is printed on the heading of the conditional and/or mixed cluster map.

#### B.11.7 FLOWCHART

See Figure B-11 for flowchart of MAPHND subprogram.

#### B.11.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

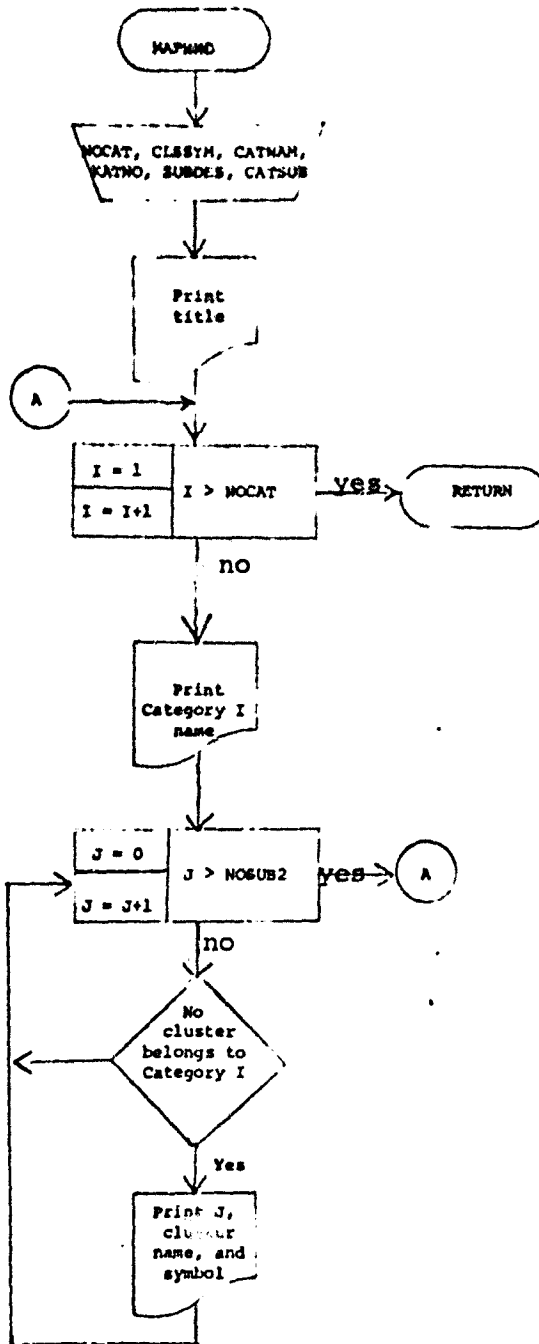


Figure B-11.- Flow chart for MAPIIND subprogram

~~B-43~~

118

## B.12 SOFTWARE SUBPROGRAM NO. 12 (ASCEND)

The subprogram ASCEND sorts an array of floating point numbers in descending order.

### B.12.1 LINKAGES

The subprogram ASCEND is called by the routine KNEAR and does not call any LARSYS routines.

### B.12.2 INTERFACES

Interfaces with other routines via calling arguments.

### B.12.3 INPUTS

Calling sequence: CALL ASCEND (SCN, LNCAT, PTR1, PTR2)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
SCN	LNCAT	In/Out	Array to be sorted.
LNCAT	1	In	No. of elements to sort.
PTR1	LNCAT	In/Out	Array containing information relating to SCN.
PTR2	LNCAT	In/Out	Array containing information relating to SCN.

### B.12.4 OUTPUTS

N/A

### B.12.5 STORAGE REQUIREMENTS

Storage used: Code-174<sub>8</sub> Data-27<sub>8</sub>

#### B.12.6 DESCRIPTION

The subprogram ASCEND operates on 3 arrays, a floating point array and 2 integer arrays. The floating point array is arranged in decreasing order; the 2 integer arrays are sorted based on the ordering of the floating point array.

#### B.12.7 FLOWCHART

See Figure B-7 for flowchart of ASCEND subprogram.

#### B.12.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

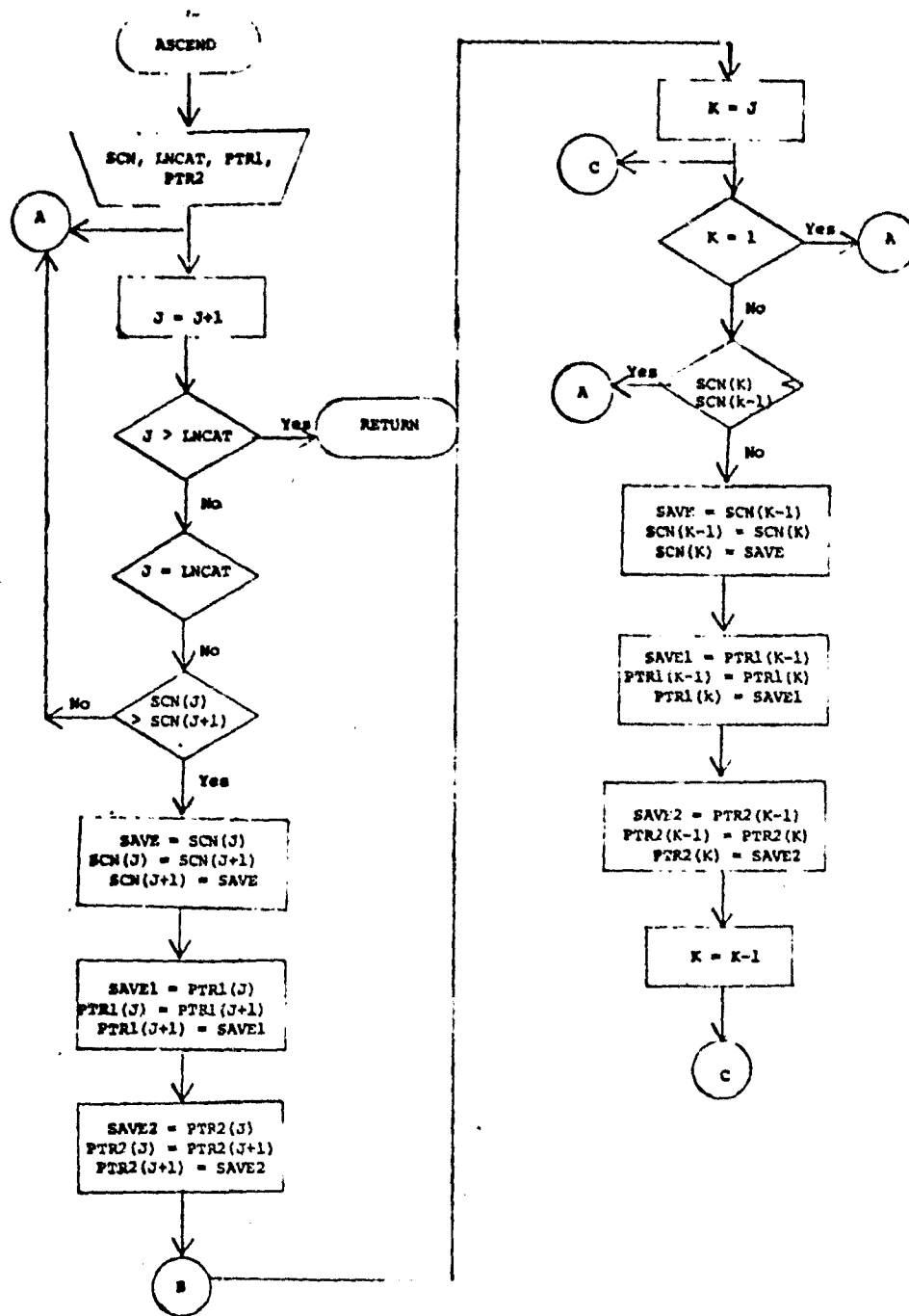


Figure B-12.— Flow chart for ASCEND subprogram

B-46  
121

### B.13 SOFTWARE SUBPROGRAM NO. 13 (CRDSCN)

The function CRDSCN interprets the control cards DOTLABEL and STATLABEL.

#### B.13.1 LINKAGES

The subprogram CRDSCN is called by the routine SET14 and calls the routines NXTCHR, FIND, and NUMBER.

#### B.13.2 INTERFACES

N/A

#### B.13.3 INPUTS

Calling sequence: CALL CRDSCN (CARD, GRPDEX, GRPNAM, GROUPS, NOGRP, GRPTR)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
CARD	62	In	Array containing the string of characters from control card DOTLABEL or STATLABEL.
GRPDEX	60	Out	Array containing the pointers for retrieving information from GROUPS array.
GRPNAM	60	Out	Array containing the input category names.



GROUPS	120	Out	Array containing the no. of clusters associated with an input category name, followed by the actual cluster no.
NOGRP	60	Out	No. of category names encountered on input card at a given time.
GRPTR	1	In	Pointer for storing into the GROUPS array.

#### B.13.4 OUTPUTS

N/A

#### B.13.5 STORAGE REQUIRED

Storage used: Code-345<sub>g</sub>      Data-732<sub>g</sub>

#### B.13.6 DESCRIPTION

The function CRDSCN interprets an alphanumeric string, one character per word. The first 1 to 7 characters are assumed to be alpha characters, followed by at least one numeric character.

The alpha characters define a category name plus a comma, the numeric characters, cluster number(s).

Three arrays are output. One array contains the labels (category names); one, the no. of the clusters or dot to relabel; and the other, index pointers.

For the following control cards

DOTLABEL        WHEAT, 3, 4, 7

DOTLABEL        NONWHT, 10, 12

the array will contain:

GRPNAM(1) = WHEAT	GRPDEX(1) = 1	GROUPS (1) = 3
GRPNAM(2) = NONWHT	GPPDEX(2) = 5	GROUPS (2) = 3
		GROUPS (3) = 4
		GROUPS (4) = 7
		GROUPS (5) = 2
		GROUPS (6) = 10
		GROUPS (7) = 12

#### B.13.7 FLOWCHART

See figure B-13 for flowchart of CRDSCN subprogram.

#### B.13.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

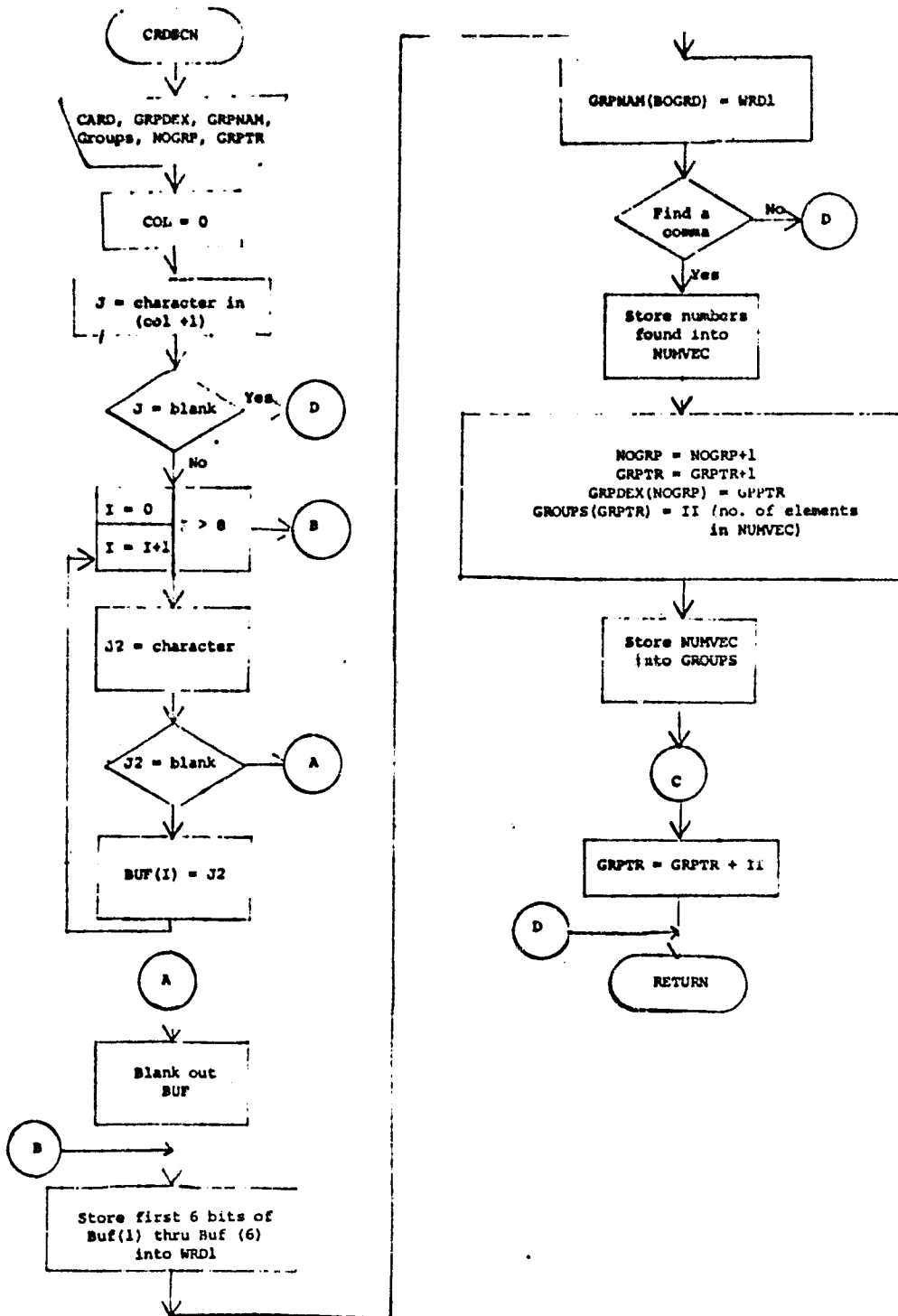


Figure B-13.— Flow chart for subprogram CRDSCN

## B.14 SOFTWARE SUBPROGRAM NO. 14 (FILERD)

The subprogram FILERD computes the needed high speed drum addresses and reads in the input files.

### B.14.1 LINKAGES

The subprogram FILERD is called by the driver routine LABEL and calls the routines LAREAD, STOMAP, REDSAV, RDDOTS, and WRTFLD.

### B.14.2 INTERFACES

Interfaces with other routines through the common blocks INFORM, GLOBAL, and LABS.

### B.14.3 INPUTS

Calling sequence: CALL FILERD (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV, VERTX)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10600	Out	Used for scratch area and contains the dot data information.
TOP	1	In	Parameter TOP=10
NOFLD	1	Out	No. of fields on DOTFIL.
TOTVRT	1	Out	No. of field vertices on DOTFIL
FLDSAV	(4,NOFLD)	Out	Array containing the field information on DOTFIL.
VERTX	(2,TOTVRT)	Out	Array containing the field vertices on DOTFIL.

#### B.14.4 OUTPUTS

N/A

#### B.14.5 STORAGE REQUIREMENTS

Storage used: Code-304<sub>8</sub>      Data-105<sub>8</sub>

#### B.14.6 DESCRIPTION

If a MAPFIL file is input, the subprogram FILERD initiates the reading of the field card, computes the dimensions of the input file, reads in the file, and stores the entire image on high speed drum.

If a SAVTAP file is input, the statistics are read into the upper partition of the array. Only the means and covariances are stored on high speed drum.

If a DOTFIL file is input, the file is read in and stored starting in the location of ARRAY previously occupied by the covariances.

Based on the number of files read in, drum addresses that will be needed in other routes are computed and stored in the common block LABS.

#### B.14.7 FLOWCHART

See figure B-14 for flowchart of FILERD subprogram.

#### B.14.8 LISTING

Available in the Data Technique Laboratory, JSC Building 17.

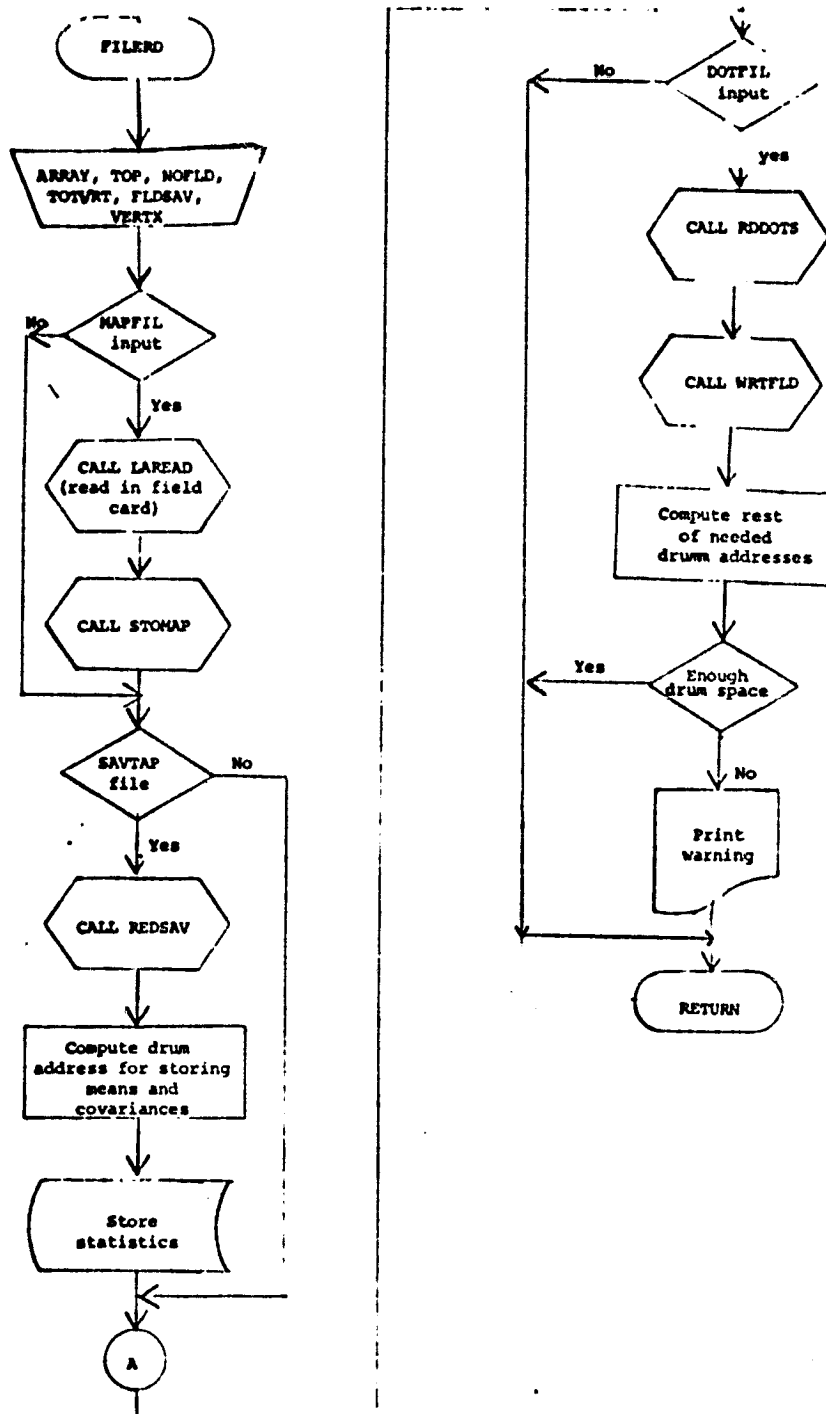


Figure B-14.— Flow chart for FILERD subprogram

## B.15 SOFTWARE SUBPROGRAM NO. 15 (STOMAP)

The subprogram STOMAP reads the input MAPFIL tape and stores the image on high speed drum.

### B.15.1 LINKAGES

The subprogram STOMAP is called by the routine FILERD and calls the routines TAPHDR, FLDINT and LINERD.

### B.15.2 INTERFACES

Interfaces with other routines through the common block GLOBAL.

### B.15.3 INPUTS

- MAPFIL tape

Accepts the cluster or classified map output by ISOCLS or DISPLAY. See section 5.1 of User Documentation EOD-LARSYS.

- Calling arguments: CALL STOMAP (ILINE, NSAMP, HIST, LIMIT, BEGIN1)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ILINE	1	In	No. of lines on MAPFIL
NSAMP	1	In	No. of samples per scan line.
HIST	LIMIT	In	Scratch area
LIMIT	1	In	Parameter dimensioning HIST
BEGIN1	1	In	Beginning drum address for storing MAPFIL

#### B.15.4 OUTPUTS

N/A

#### B.15.5 STORAGE REQUIREMENTS

Storage used: Code-241<sub>8</sub>      Data-57<sub>8</sub>

#### B.15.6 DESCRIPTION

Based on the size of both the array HIST, and the input file the subprogram STOMAP computes how many scan lines of the MAPFIL file can be read into core at one time. When the buffer is filled, the data is dumped to high speed drum. (The number of dumps to the drum is a function of the number of lines that can reside in core).

#### B.15.7 FLOWCHART

See Figure B-15 for flowchart of subprogram STOMAP.

#### B.15.8 Listing

Available in the Data Techniques Laboratory, JSC Building 17.



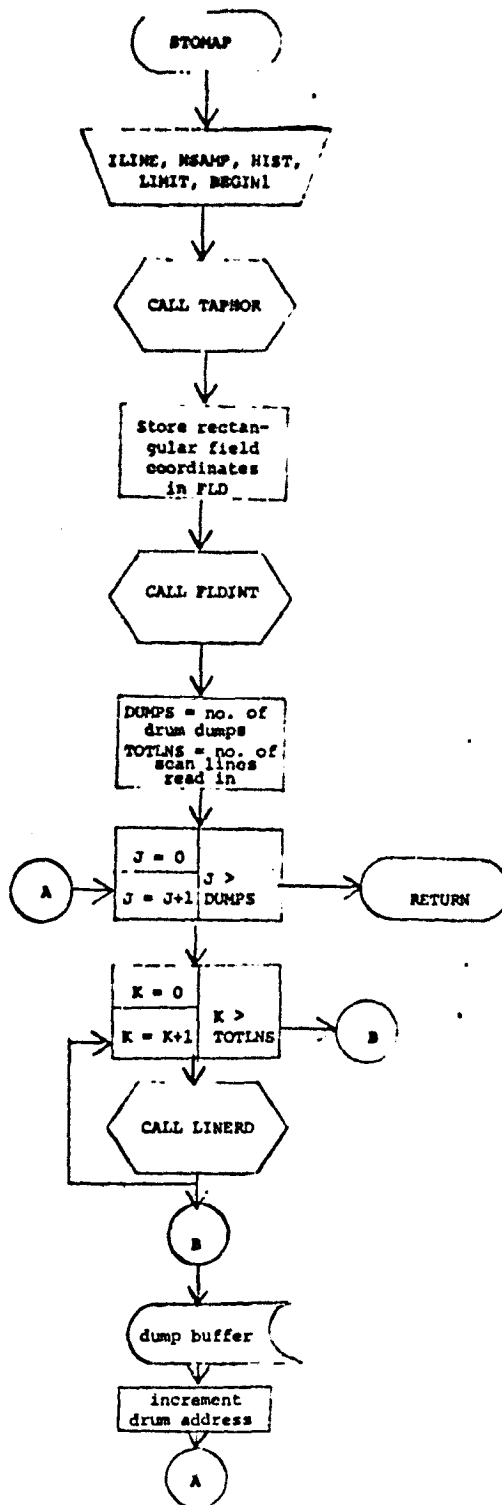


Figure B15.— Flow chart for subprogram STOMAP

## B.16 SOFTWARE SUBPROGRAM NO. 16 (DSPTAP)

Subprogram DSPTAP writes an unformatted unconditional cluster map tape to be used as input to the DISPLAY processor to obtain certain general performance summaries. The layout and structure of this tape is described in the User Documentation EOD-LARSYS manual in the SYSTEM INTERNAL FILES section (MAPTAP). It has the same layout as the classification map tape normally written by the CLASSIFY processor. However, only category/subclass information is written.

The trigger to initiate writing of this tape is set by inclusion of the MAPTAP control card as input to the LABEL processor.

### B.16.1 LINKAGES

Subprogram DSPTAP is called by routine LABLR. DSPTAP calls the UNIVAC tape routine FSBSFL and the drum read routine RREAD.

### B.16.2 INTERFACES

Interfaces with other routines through common blocks INFORM, GLOBAL, and LABS.

### B.16.3 INPUTS

Calling sequence: DSPTAP (SUBNO, SUBDES, FLDSAV, VERTX, CATVEC, SUBVEC, MEANS, COVAR, TOP, DATA, NOFLD, TOTVRT)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
SUBNO	NOCAT	In	Number of sub-classes in each category.
SUBDES	NOSUB2	In	Subclass names

FLDSAV	(4,NOFLD)	In	For each training field 1. field name 2. class no. of field 3. subclass no. of field 4. no. of field vertices
VERTX	(2,TOTVRT)	In	Field vertices
CATVEC	60	In	Category no. for each subclass
SUBVEC	60	In	Re-ordering of original subclass order by category number
MEANS	(NOFET2,NOSUB2)	In	Subclass mean vectors
COVAR	(VARSZ2,NOSUB2)	In	Subclass covariance matrices, by row as lower triangular
TOP	1	In	Upper limit of ARRAY vector, used as general buffer storage. TOP set in MONTOR routine, currently 10600
DATA	(PTS,NOLINE) NOLINE=No. of lines read into core	In	Original cluster number the pixel was assigned during clustering (in ISOCLS)

NOFLD	1	In	Total no. of fields
TOTVRT	1	In	Total no. of vertices

#### B.16.4 OUTPUT

An unformatted tape (MAPTAP) is written.

#### B.16.5 STORAGE REQUIREMENTS

Code-1005<sub>8</sub>      Data-4223<sub>8</sub>

#### B.16.6 DESCRIPTION

An unformatted unconditional cluster map tape is written for use by the DISPLAY processor. The layout is the same as that for the MAPTAP written by the CLASSIFY processor. However, only category and subclass information is provided, there is no inclusion of class/subclass and category/class associations. The information is that obtained as a result of cluster labeling in the LABEL processor, either by the k-nearest neighbor or all-of-a-kind criteria.

#### B.16.7 FLOWCHART

See Figure B-16 for flowchart of DSPTAP subprogram.

#### B.16.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

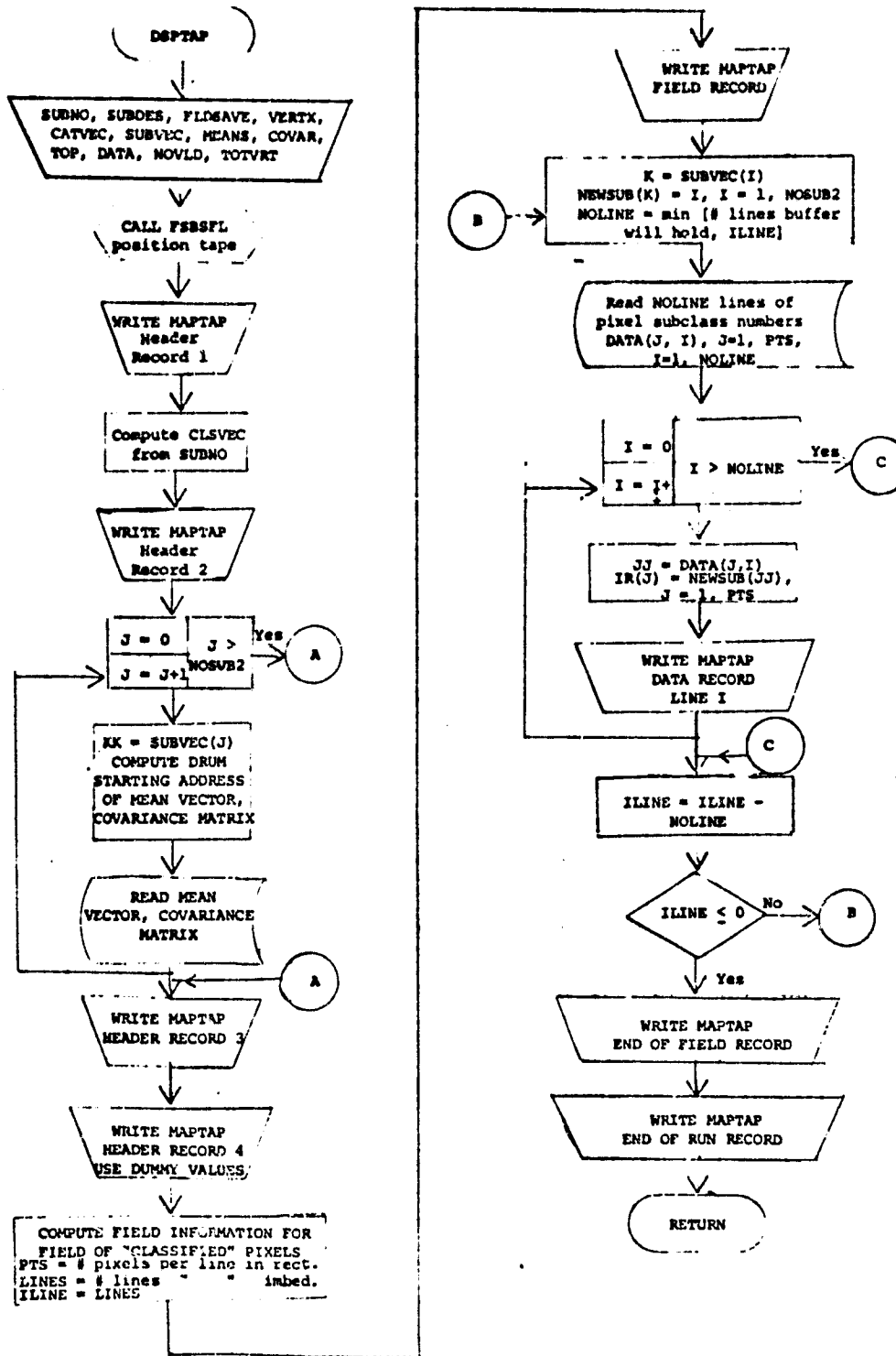


Figure B-16.-- Flow chart for subprogram DSPTAP

B-60  
135

REPL. PAGE IS  
OF LOW QUALITY

## B.17 SOFTWARE SUBPROGRAM NO. 17 (NAMSTA)

Subprogram NAMSTA provides cluster names based on labeling results in the LABEL processor. These names are based on category names and sequence numbering. A check is made to ensure that each category is involved in the naming of at least one cluster. If not, that category is deleted from the total count of number of categories.

### B.17.1 LINKAGES

Subprogram NAMSTA is called by routine LABLR.

### B.17.2 INTERFACES

Interfaces with other routines through common blocks INFORM and LABS.

### B.17.3 INPUTS

Calling sequence: NAMSTA (SUBNAM, SUBVEC, N, CATVEC, SUBNO)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
SUBNAM	60	Out	Cluster (subclass) names
SUBVEC	60	In	Not used in NAMSTA, re-ordering of original subclass names by category
N	60	In	Not used
CATVEC	60	In	Category no. of the subclasses
SUBNO	60	In	No. of subclasses in each category

#### B.17.4 OUTPUT

N/A

#### B.17.5 STORAGE REQUIREMENTS

Code-203<sub>8</sub>            Data 40<sub>8</sub>

#### B.17.6 DESCRIPTION

Cluster names are stored in vector SUBNAM on the basis of the category assignment to each cluster. These names are six characters long. The first three characters are derived from the category name, and the last three correspond to sequence numbers, starting with '001' for each category. In the event that a category has no associated clusters, SUBNO and NOCAT are adjusted to remove that category from subsequent consideration.

#### B.17.7 FLOWCHART

See Figure B-17 for flowchart of NAMSTA.

#### B.17.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

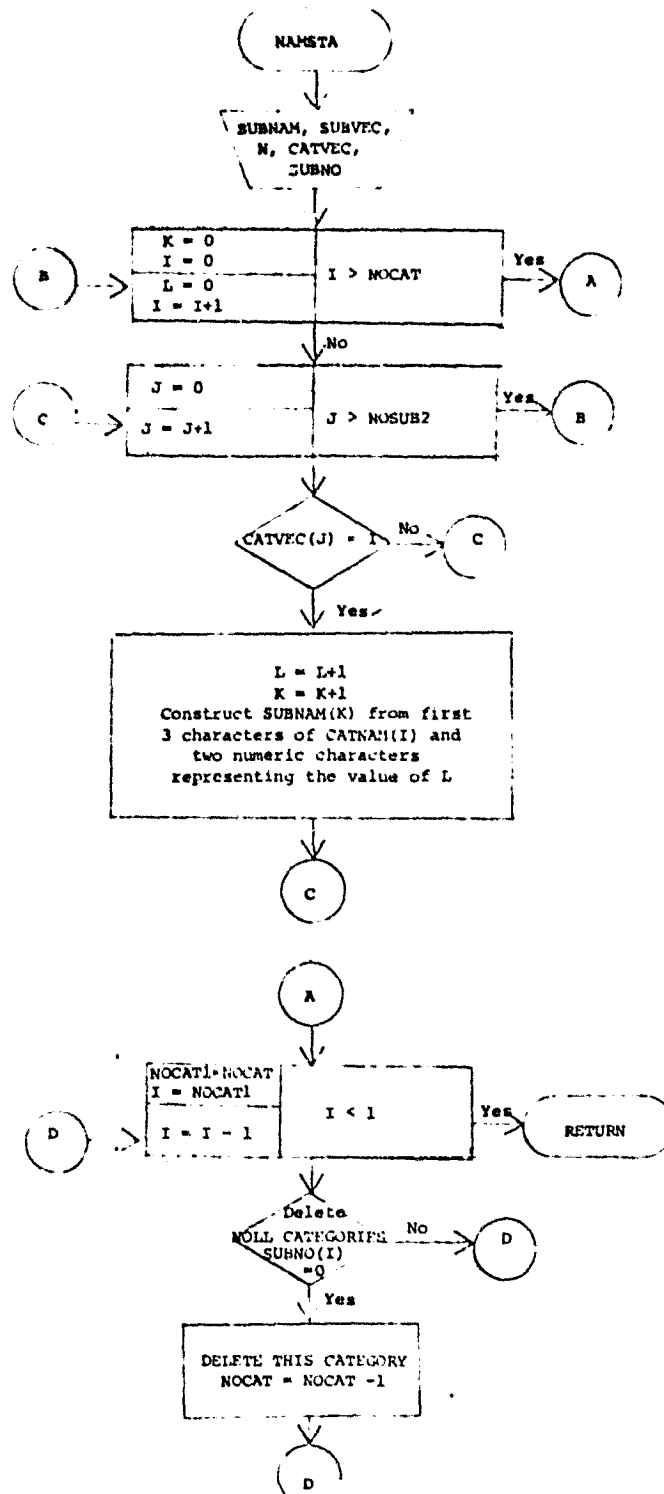


Figure B-17.-- Flow chart for subprogram NAMSTA

B-65  
138



## B.18 SOFTWARE SUBPROGRAM NO. 18 (REORDER)

Subprogram REORDER provides a re-ordering of subclass names and number of associated pixels, as part of the manual relabeling of stats option in the LABEL processor.

### B.18.1 LINKAGES

Subprogram REORDER is called by routine LABLR.

### B.18.2 INTERFACES

Interfaces with other routines through the common block INFORM.

### B.18.3 INPUTS

Calling sequence: REORDER (ARRAY, SUBVEC, N)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	TOP (currently 10600)	In/Out	Subclass names, starting at index SUBDS2
SUBVEC	60	In	Reordering of subclass by category
N	60	Out	No. of pixels in each subclass, reordered using SUBVEC

### B.18.4 OUTPUTS

N/A

### B.18.5 STORAGE REQUIREMENTS

Code-107<sub>8</sub>      Data-124<sub>8</sub>

#### B.18.6 DESCRIPTION

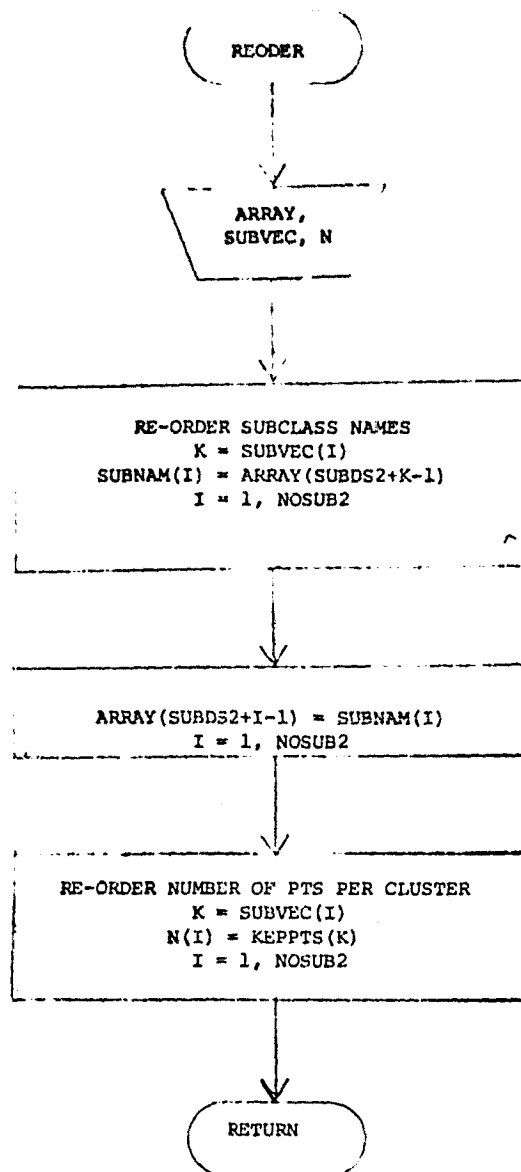
The subclass names and number of pixels per subclass are reordered according to SUBVEC. The new ordering places all subclasses associated with the first category first in order, all subclasses associated with the second category second in order, and so on.

#### B.18.7 FLOWCHART

See Figure B-18 for flowchart of the REORDER subprogram.

#### B.18.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.



REORDER PAGE IS  
FOR QUALITY

Figure B-18.— Flow chart for subprogram REORDER

## B.19 SOFTWARE SUBPROGRAM NO. 19 (LABMAN)

Subprogram LABMAN, under the manual relabeling of stats option of the LABEL processor, writes a statistics tape (SAVTAP) and optionally punches a card deck.

### B.19.1 LINKAGES

Subprogram LABMAN is called by the LABLR routine. LABMAN calls the UNIVAC tape file positioning on two occasions, one for manual stat relabeling, the other for the LABEL routine FSBSFL and the drum read routine RREAD.

### B.19.2 INTERFACES

Interfaces with no routines other than LABLR, and that only through the calling sequence.

### B.19.3 INPUTS

Calling sequence: LABMAN (UNIT, FILE, NOCLS, TOTSUB, NOFEAT, TOTFLD, TOTVRT, FETVEC, FLDSAV, VERTEX, CLSNMS, NOSUBS, SUBNM, N, STADRS, VARSIZ, PUNCH, SUBVEC, PRNSTS)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
UNIT	1	In	Statistics tape unit number
FILE	1	In	Statistics tape file no. for this output file
NOCLS	1	In	No. of classes
TOTSUB	1	In	No. of subclasses
NOFEAT	1	In	No. of channels (features)
TOTFLD	1	In	No. of fields

FETVEC	NOFEAT	In	Features
FLDSAV	NOFLD2*4	In	Field information by field (name, class, dummy entry, no. of vertices in fields
CLSNMS	NOCLS2	In	Class names
NOSUBS	1	In	No. of clusters in each class
SUBNM	NOSUB2	In	Reordered subclass names
N	60	In	Reordered no. of pixels in each subclass
STADRS	1	In	Starting address for drum storage access
VARSIZ	1	In	Storage size for lower triangular part of covariance matrix
PUNCH	1	In	Trigger to punch card deck
SUBVEC	60	In	Reordering of subclasses by category
PRNSTS	1	In	Trigger to print statistics summary

#### B.19.4 OUTPUT

An unformatted statistics tape (SAVTAP) is output. On options, a punch card deck and/or a line printer summary is performed.

#### B.19.5 STORAGE REQUIREMENTS

Code-1222<sub>8</sub>      Data-1350<sub>8</sub>

#### B.19.6 DESCRIPTION

A statistics tape is written under the SAVTAP format to reflect the manual relabeling of clusters under this option of the LABEL processor. On option, a punched card deck is produced.

#### B.19.7 FLOWCHART

See Figure B-19 for flowchart of LABMAN.

#### B.19.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

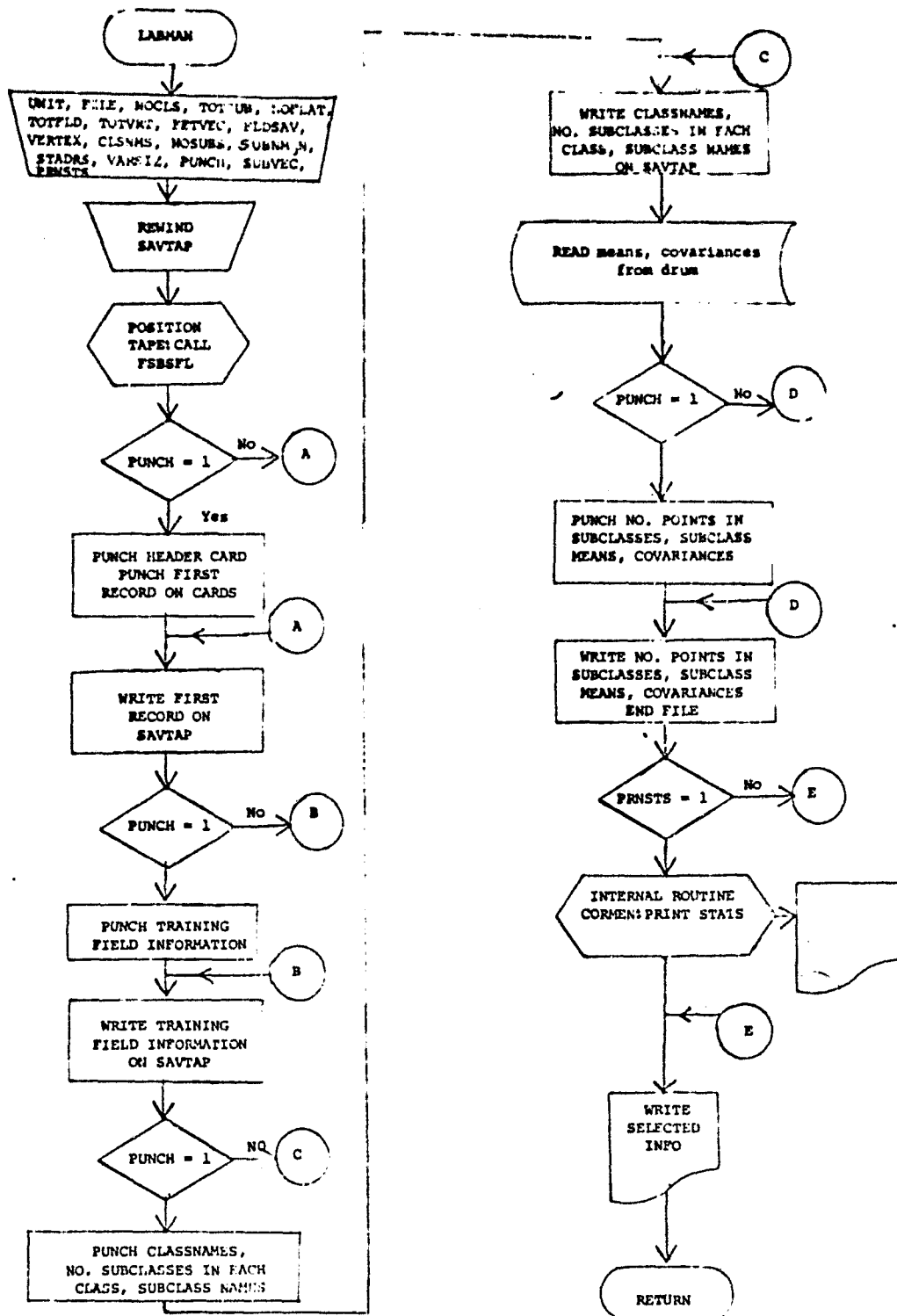


Figure B-19.— Flow chart for subprogram LABMAN

B-70  
145

## B.20 SOFTWARE SUBPROGRAM NO. 20 (SUNFCS)

Subprogram SUNFCS computes sun angle gain corrections for pixel radiance values for each channel based on input sun angles.

### B.20.1 LINKAGES

Subprogram SUNFCS is called by routine DOTDST.

### B.20.2 INTERFACES

No common blocks are provided.

### B.20.3 INPUTS

Calling sequence: SUNFCS (SUNCOR, SUNANG, FETVEC, NOFEAT, ISUNC, ISUNT)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
SUNCOR	30	Out	Corrections corresponding to channels in FETVEC
SUNANG	8	In	Sun angles
FETVEC	30	In	Channels
NOFEAT	1	In	No. of channels
ISUNC	1	In	Flag indicating sun angles came from cards.
ISUNT	1	In	Flag indicating sun angles were extracted from MSS data tape

### B.20.4 OUTPUT

N/A

~~B-71~~  
146



#### B.20.5 STORAGE REQUIREMENTS

Code-254<sub>8</sub>

Data-576<sub>8</sub>

#### B.20.6 DESCRIPTION

The sun angle corrections are computed from an internal table covering four channels at a haze factor of 25 percent. The corrections correspond to the channels contained in FETVEC. If the ISUNT flag is on, the first sun angle in SUNANG is assumed to correspond to pass 1. If the ISUNC flag is on, the first sun angle in SUNANG is assumed to correspond to the first pass in FETVEC.

The sun angle corrections, which are basically gain factors, are used in the computation of cluster mean-dot distances.

#### B.20.7 FLOWCHART

See Figure B-20 for flowchart of SUNFCS.

#### B.20.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

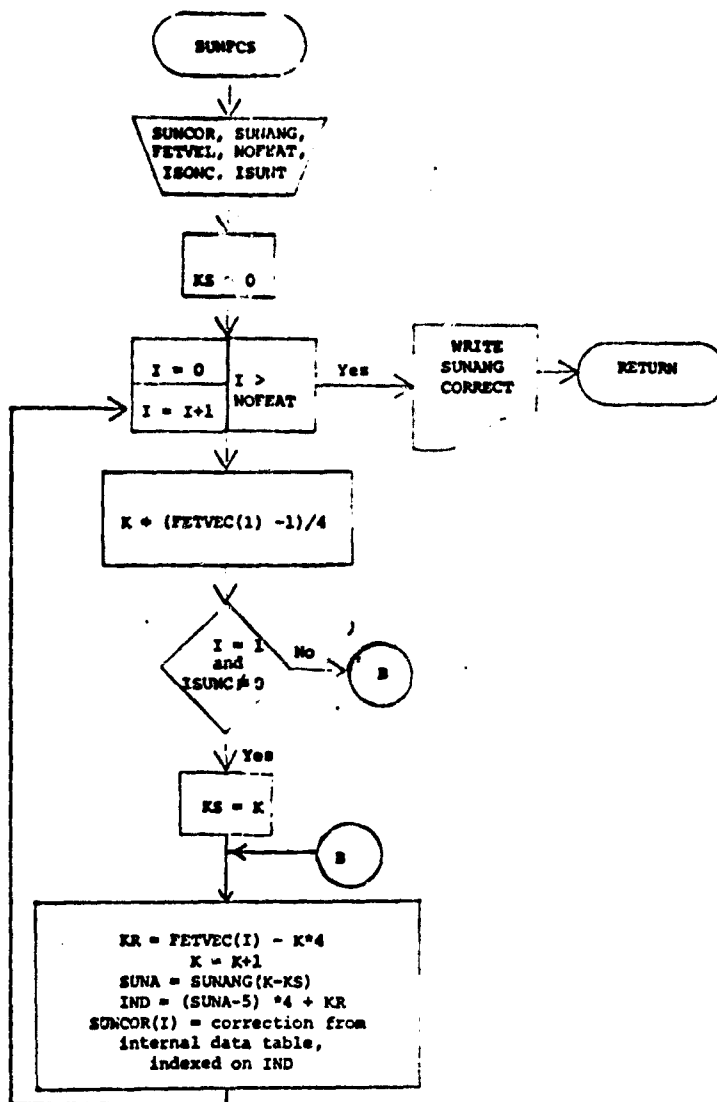


Figure B-20.— Flow chart for subprogram SUNFCS

B-73  
148

## B.21 SOFTWARE SUBPROGRAM NO. 21 (CLRKEY)

Subprogram CLRKEY writes the color keys on the mixed and conditional cluster map tapes.

### B.21.1 LINKAGES

Subroutine CLRKEY is called by routine CLSMAP. CLRKEY calls WRTLN, an entry point in TAPWR1.

### B.21.2 INTERFACES

There are no common blocks associated with CLRKEY.

### B.21.3 INPUTS

Calling sequence: CLRKEY (XSIZ, IDATA, NOSUB2, CH, MEANS, NC)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
XSIZ	1	In	No. of pixels/ line (maximum 200)
IDATA	(XSIZ,CH)	Out	Colors output per line
NOSUB2	1	In	No. of subclasses
CH	1	In	Set equal to 1; in the multi- channel case, set equal to NC+1
MEANS	(NC,NOSUB2)	In	Colors corresponding to subclasses.
NC	1	In	Set equal to 1, the no. of channels

B-14  
149

ORIGINAL PAGE IS  
OF POOR QUALITY

#### B.21.4 OUTPUT

The color keys are added to the mixed or conditional map tape.  
(See appendix H for format.)

#### B.21.5 STORAGE REQUIREMENTS

Code-247<sub>8</sub>          Data-51<sub>8</sub>

#### B.21.6 DESCRIPTION

For each cluster (subclass), a 10 x 10 color number is written on the conditional and/or mixed cluster map tapes. This cluster number is obtained from an internal table.

#### B.21.7 FLOWCHART

See Figure B-21 for flowchart of CLRKEY program.

#### B.21.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

~~B-75~~  
150

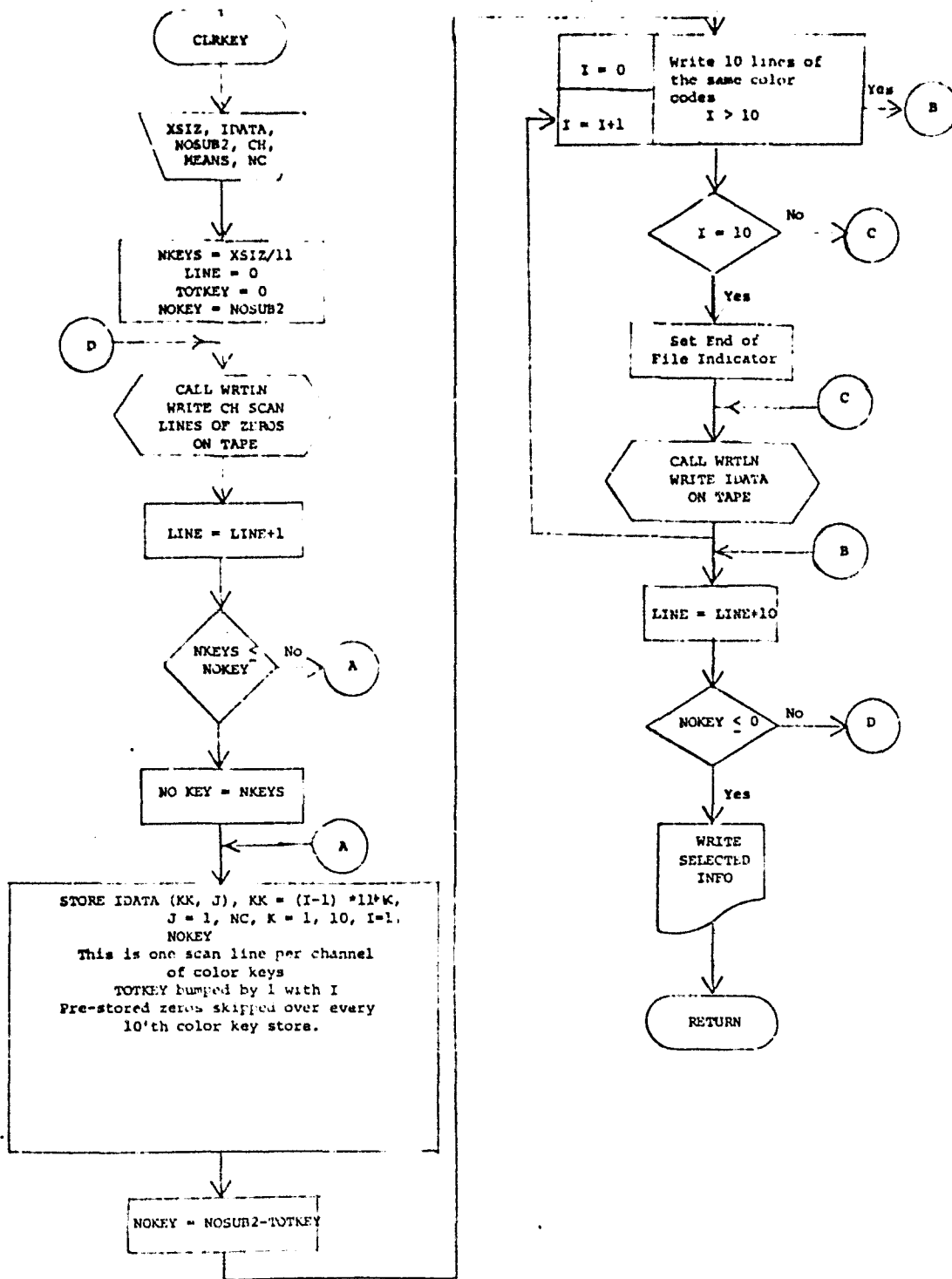


Figure B-21.— Flow chart for subprogram CLRKEY

B-16  
151

## B.22 SOFTWARE SUBPROGRAM NO. 22 (CNDMAP)

Subprogram CNDMAP flags conditional clusters.

### B.22.1 LINKAGES

Subprogram CNDMAP is called by routine LABLR. CNDMAP calls the UNIVAC drum read routine RREAD.

### B.22.2 INTERFACES

Interfaces with other routines through common blocks INFORM and LABS.

### B.22.3 INPUTS

Calling sequence: CNDMAP (DOTS, CNDSUB, CATVEC)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DOTS	(4,TOTDT2)	In	Dot data-for each dot 1. sample number 2. line number 3. type 4. category no
CNDSUB	60	Out	For each subclass: set equal to entry in CATVEC if not conditional, other- wise set equal to 62 minus no. of conditional sub- classes flagged previously
CATVEC	NOSUB2	In	Category no. of each subclass

#### B.22.4 OUTPUT

N/A

#### B.22.5 STORAGE REQUIREMENTS

Code-142<sub>8</sub>          Data-424<sub>8</sub>

#### B.22.6 DESCRIPTION

For each cluster I, the subprogram CNDMAP reads into core from drum that portion of the distance table for cluster I. The minimum distance between the labeled cluster I and dot of like label is found and compared with a threshold value. If the distance is greater than the threshold, the cluster is flagged as conditional. This flagging is done by assigning it a category number equal to 62 minus the number of clusters already flagged.

#### B.22.7 FLOWCHART

See Figure B-22 for flowchart of CNDMAP.

#### B.22.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

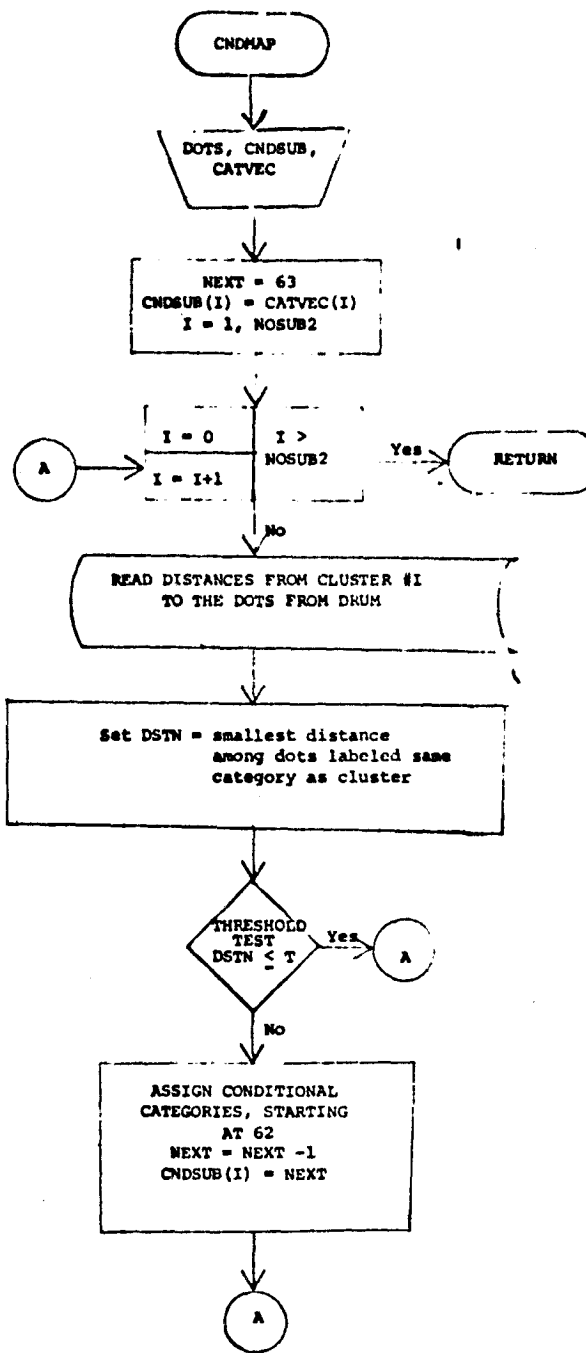


Figure B-22.— Flow chart for subprogram CNDMAP

B-79

154



APPENDIX C  
SUBPROGRAM DESCRIPTIONS

APPENDIX C

TABLE OF CONTENTS

C.1 Software for Subprogram No. 1 (ISODAT)  
C.2 Software for Subprogram No. 2 (RDDATA)  
C.3 Software for Subprogram No. 3 (SETUP7)  
C.4 Software for Subprogram No. 4 (SUNFAC)  
C.5 Software for Subprogram No. 5 (PSPLIT)

## C.1 SOFTWARE SUBPROGRAM NO. 1 (ISODAT)

The new capabilities of subprogram ISODAT are:

1. on option, call the utility routine RDDOTS to obtain cluster starting vectors,
2. on option, apply a sun angle correction by channel to each pixel during clustering,
3. remove "designated other" or "designated unidentifiable" pixels from the clustering algorithm input.

### C.1.1 LINKAGES

The Subprogram ISODAT is called by the routine ISOCLS, and calls routines PRINT, PSPLIT, CLDIST, and DELETE.

### C.1.2 INTERFACES

Subprogram ISODAT interfaces with other routines through common blocks GLOBAL, PASS, and ISOLNK.

### C.1.3 INPUTS

- Calling sequence: CALL ISODAT (C, IPLACE, MEANS, N, STDEV, CLD, FLDINF, AVP, AMN)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
C	(NOFEAT, NOPTS)	Out	Storage for one drum read of pixels for clustering
IPLACE	NOPTS	Out	NOPTS is the number of points for one drum write of pixel cluster numbers.

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
MEANS	(NOFEAT,MAXCLS)	In/Out	Starting mean values for clustering. MAXCLS is the maximum number of clusters.
N	MAXCLS	Out	Number of pixels per cluster.
STDEV	(NOFEAT,MAXCLS)	Out	Standard deviations per cluster/channel.
CLD	(LMAXCLS,MAXCLS)	In	Cluster distance matrix.
FLDINF	(IPT)	Out	Field information (vertices, vertex location).
AVP	(NOFEAT,MAXCLS)	Out	Temporary storage for computation of cluster means
AMN	(NOFEAT,MAXCLS)	In	Temporary storage for computation of cluster standard deviations.

#### C.1.4 OUTPUTS

No additional outputs.

#### C.1.5 STORAGE REQUIREMENTS

Code-2176<sub>8</sub>      Data-4400<sub>8</sub>

#### C.1.6 DESCRIPTION

A dot file trigger in labeled common ISOLNK invokes a call to RDDOTS for starting vectors. The unit and file numbers will have been loaded in labeled common GLOBAL. Sun angle correction triggers in labeled common ISOLNK causes the sun angles (stored in ISOLNK) to be used to find sun angle corrections from an internal table. Application of the correction consists of multiplying the pixel channel radiance values by the corrections in the pixel assignment to clusters, and the computation of maximum standard deviations.

The DO/DU pixels are identified from the input DO/DU field information. As regular field pixels are extracted from the MSS data tape in routine RDDATA, any DO/DU pixels contained therein are identified and the radiance values are set to 0 or 255 for subsequent identification and deletion in the clustering procedure.

#### C.1.7 FLOWCHART

N/A

#### C.1.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

## C.2 SOFTWARE SUBPROGRAM NO. 2 (RDDATA)

The new capabilities of subprogram RDDATA are:

1. read DESIGNATED OTHER and DESIGNATED UNIDENTIFIABLE class-name cards and the associated field definition cards, and calculate the field information (number of vertices, vertex locations, number of pixels) for both classes,
2. compute information concerning the DO/DU pixels for use by other subprograms. This includes the number of DO/DU pixels in each pixel field for the current class, and drum addresses for DO/DU pixels,
3. assign printing symbols to DO and DU pixels, as well as special mean values (0 or 255).

### C.2.1 LINKAGES

The subprogram RDDATA is called by the routine ISOCLS, and calls the routines TAPERD and LAREAD.

### C.2.2 INTERFACES

Subprogram RDDATA interfaces with other routines through common blocks GLOBAL, PASS, and ISOLNK.

### C.2.3 INPUTS

- Calling Sequence: CALL RDDATA (ARRAY, TOP, IDATA, IDIM, LAST)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	TOP	In	Starting location for field information.
TOP	1	In	Number of available words in ARRAY.

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
IDATA	IDIM	In	Buffer for pixel value storage per scan line.
IDIM	1	In	Maximum dimension of IDATA.
LAST	1	Out	Flag indicating (when set to a value of 1) that all classes for one set of control cards have been processed.

#### C.2.4 OUTPUTS

No additional outputs.

#### C.2.5 STORAGE REQUIREMENTS

Code-2316<sub>8</sub>      Data-1350<sub>8</sub>

#### C.2.6 DESCRIPTION

Designated "other" and/or "unidentifiable" pixels are identified from field information. The radiance values are reset to values of 0 and/or 255. This serves as the flag in subsequent processing.

The field description from cards are held internally.

The trigger for DO/DU processing is added to labeled common PASS and set in RDDATA. Two special print symbols (\* and #) for DO and DU pixels are provided.

~~CS~~  
1/21

Special cluster numbers are assigned to DO and DU pixels. These numbers have values 1 greater and 2 greater than the number of regular clusters.

C.2.7 FLOWCHART

N/A

C.2.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.



### C.3 SOFTWARE SUBPROGRAM NO. 3 (SETUP7)

The new capabilities of subprogram SETUP7 are:

1. read an optional DOTFILE control card for unit and file number, and set option trigger for subsequent use by ISODAT
2. read an optional SUNANG control card, set the sun angle correction trigger for use by ISODAT, and extract the sun angles if these are present on the card.

#### C.3.1 LINKAGES

The subprogram SETUP7 is called by the routine ISOCLS. Routine SETUP7 calls card reader/interpreter routines FIND, NXTCHR, NUMBER, and FLTNUM.

#### C.3.2 INTERFACES

Subprogram SETUP7 interfaces with other routines through common blocks GLOBAL, PASS, and ISOLNK.

#### C.3.3 INPUT

- Calling sequence: CALL SETUP7 (ARRAY, TOP, ITIME)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	TOP-CORBAS	Out	Stores means or a statistics Module STAT deck.
TOP	1	In	Maximum dimension of ARRAY (normally 10,600)
ITIME	1	In	Count of number of times SETUP7 called

● New Control Cards:

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
DOTFIL	INPUT/UNIT=n, FILE=m (Default: Self- initialized start- ing)	Defines the FORTRAN unit number n and file number m of the dot data file DOTFIL con- taining the starting vectors.
SUNANG	TAPE (Default: No sun angle correction applied)	Sun angles will be ex- tracted from the MSS data tape.
SUNANG	$n_1, n_2, \dots, n_j$ $n_j$ are integers, $j \leq 8$ (Default: No sun angle correction applied)	$\{n_j\}$ are the sun angles to be used in computing the sun angle correc- tions for use in the clustering algorithm. A sun angle must be in- put for each set of 4 channels appearing on the CHANNEL control card.

C.3.4 OUTPUTS

No additional outputs.

C.3.5 STORAGE REQUIREMENTS

Code-2410<sub>8</sub>      Data-671<sub>8</sub>

C.3.6. DESCRIPTION

The DOTFILE and SUNANG control cards are handled by ex-  
tending data vector INVEC to include the character strings

~~C-8~~  
164

DOTFIL and SUNANG. The unit and file number of DOTFILE are stored in GLOBAL labeled common. Sun angles appearing on the SUNANG card are placed in ISOLNK as sun angle card input. The sun angle correction triggers appear in ISOLNK and are set by SETUP7.

C.3.7 FLOWCHART

N/A

C.3.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

#### C.4 SOFTWARE SUBPROGRAM NO. 4 (SUNFAC)

Subprogram SUNFAC is identical to subprogram SUNFCS. Please refer to the description of this subprogram in Appendix B, number B-20.

##### C.4.1 LINKAGES

Subprogram SUNFAC is called by routine ISODAT.

##### C.4.2 DESCRIPTION

The sun angle gain corrections computed by SUNFAC are used in computing pixel/cluster mean distances in the clustering routine PSPLIT and in determining the largest standard deviations in routine ISODAT.

## C.5 SOFTWARE SUBPROGRAM NO. 5 (PSPLIT)

Subprogram PSPLIT performs much of the calculative activity associated with the ISOCLS processor. For each iteration of the clustering algorithm, it assigns pixels to clusters, and computes the new mean and standard deviation vectors.

### C.5.1 LINKAGES

PSPLIT is called by routine ISODAT. It calls the UNIVAC drum read/write routines in RANDIO.

### C.5.2 INTERFACES

PSPLIT interfaces with other routines through common blocks PASS and ISOLNK.

### C.5.3 INPUTS

Calling sequence: PSPLIT (MEANS, STDEV, N, CLD, C, IPLACE, AVP, AMN, MEN)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
MEANS	(NOFEAT, MAXCLS)	In/Out	Cluster mean vectors
STDEV	(NOFEAT, MAXCLS)	Out	Cluster standard deviations
N	(MAXCLS)	Out	No. of pixels in each cluster
CLD	(MAXCLS, MAXCLS)	In	Inter-cluster distances, not used
C	(NOFEAT, NOPTS)	Out	Buffer holding pixel radiance values after drum read

IPLACE	(NOPTS) NOPTS=buffer size for number of pixels	Out	Cluster no. pixel is assigned
AVP	(NOFEAT,MAXCLS) NOFEAT=number of channels (max = 30) MAXCLS=62 (max number of clusters)	Out	Intermediate results in calcu- lation of cluster standard deviations
AMN	(NOFEAT,MAXCLS)	Out	Intermediate results in calcu- lation of cluster means
MEN	(NOFEAT,MAXCLS)	Out	Not used; in ISODAT call, the same storage as MEANS

#### C.5.4 OUTPUTS

N/A

#### C.5.5 STORAGE REQUIREMENTS

Code-1002<sub>8</sub>      Data-137<sub>8</sub>

#### C.5.6 DESCRIPTION

As the pixel radiance vectors come in from drum, these are first tested for 0 or 255 values over all channels (DO/DU pixels). DO and DU pixels are assigned cluster numbers greater than the ordinary cluster numbers. Any pixel not DO or DU is assigned to a cluster based on minimum  $L_1$  distance from the means provided to PSPLIT. Sun angle correction can be applied to this distance

computation by user option. After all pixels are assigned to clusters, the cluster means and standard deviations are recomputed.

In order to save calculation time, PSPLIT has two basically identical cluster assignment loops, one having incorporated the sun angle correction and the other without it. Thus, for runs without sun angle correction, this multiplication (with default values of 1) is not performed.

#### C.5.7 FLOWCHART

N/A

#### C.5.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

APPENDIX D  
SUBPROGRAM DESCRIPTIONS



APPENDIX D

TABLE OF CONTENTS

- D.1 SOFTWARE FOR SUBPROGRAM NO. 1 (SETUP4)
- D.2 SOFTWARE FOR SUBPROGRAM NO. 2 (WGTCHK)
- D.3 SOFTWARE FOR SUBPROGRAM NO. 3 (INTWGT)

## D.1 SOFTWARE SUBPROGRAM NO. 1 (SETUP4)

The SETUP4 subprogram of the SELECT processor is modified to recognize the new option, "CLSWT", on the OPTION control card. If the new option is input, SETUP4 initializes the table of weights for interclass subclass pairs (=1.0) and for intraclass subclass pairs (=0.0). Also, SETUP4 is modified to include a new internal subprogram, subroutine INTWGT which accomplishes the function of initializing interclass subclass-pair weights.

### D.1.1 LINKAGES

The SETUP4 subprogram is called from the processor driver program, SELECT. Subprogram SETUP4 continues to reference the following subprograms: BMFIL, BSTCHK, CRDSTA, EVLCHK, FIND, GRPSCN, NUMBER, NXTCHR, ORDER, PRTFLD, REDSAV, WGTCHK, and WGTSCN. An additional call to internal subroutine INTWGT is included in this modification of SETUP4. (See section D.3 for description of subroutine INTWGT).

### D.1.2 INTERFACES

SETUP4 interfaces with the driver program, SELECT, and other subprograms via the named common blocks FSL, GLOBAL, and INFORM.

### D.1.3 INPUTS

Calling sequence: CALL SETUP4 (ARRAY, TOP, STOPFG, JTIME, SUBRAY, SUBSIZ)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10600	In	Working array in blank COMMON.
TOP	1	In	The maximum usable location in ARRAY. TOP=10600, set in MONTOR.

STOPFG	1	Out	Flag set=1 to indicate that the \$END* card has been read.
JTIME	1	In	Counter for the no. of times SETUP4 is called from SELECT. When JTIME=1, the line printer output is labeled with the HEADER array contents. For JTIME≠1, the heading is not placed on the output.
SUBRAY	12000	In	Working array, provided from the SELECT program.
SUBSIZ	1	In	Maximum usable location in SUBRAY, set in SELECT program (=12000).

● Revised Control Card

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
(col. 1)	(col. 11-72)	
OPTION	CLSWT	SETUP4 sets a flag, WTKEY=1, to indicate that internal sub-program INTWGT is to initialize the table of weights in SUBRAY for interclass subclass pairs.

#### D.1.4 OUTPUTS

One addition to line printer output is provided by the modified SETUP4 subprogram. When the OPTION CLSWT control card is input, the printer listing of user requests will include:

"USE AUTOMATIC INTERCLASS SUBCLASS WEIGHTS"

There are no other changes to the output. The subclass-pair weights determined in SETUP4 are listed in the INTERCLASS SEPARABILITY TABLE, which is part of the usual printer output from the SELECT processor.

#### D.1.5 STORAGE REQUIREMENTS

Storage required: Data-791 Code-990

#### D.1.6 DESCRIPTION

When the OPTION control card is recognized by SETUP4, a transfer is made to statement number 60, where there are tests for three possible options - STATS, PUNCH, or CLSWT. Unique internal flags are set for each of the options detected on an OPTION card. If CLSWT (actually, "C" as a first character) is deleted, the flag WTKEY is set = 1. The flag, WTKEY, is tested in order to output the appropriate listing of user requests (see section D.1.4, OUTPUTS), and to initiate a call to an internal (to SETUP4) subprogram, subroutine INTWGT. The call to INTWGT accomplishes the automatic initialization of interclass subclass-pair weights in a SETUP4 working array, SUBRAY.

The flag, SETWGT, is the flag normally set by SETUP4 when a WEIGHTS card is read (SETWGT=2). Both SETWGT and WTKEY are tested for the necessity of calling the subprogram, WGTCHK. If either SETWGT=2 or WTKEY=1 or both, subroutine WGTCHK is called.

In the call to WGTCHK, the subclass-pair weights initialized by subroutine INTWGT are passed in array SUBRAY and two unused storage locations, ARRAY (WGHS14) and ARRAY(WGHS14+1), are used to pass the flags SETWGT and WTKEY to subroutine WGTCHK. Subroutine WGTCHK will place any input (via a WEIGHTS card) subclass-pair weights into the array, overriding weights placed there by subroutine INTWGT, then perform its usual function of reducing the two dimensional subclass-pair weights array to a linear array and passing it to SETUP4 beginning at ARRAY (WGHS14). At this point, control is returned to the processor driver program, SELECT, as usual.

#### D.1.7 FLOWCHART

N/A

#### D.1.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

## D.2 SOFTWARE SUBPROGRAM NO. 2 (WGTCHK)

Subprogram WGTCHK of the SELECT processor, as modified, performs one of two functions:

- a. The usual capability of initializing all intersubclass pair weights =1.0, followed by overriding these preset weights with the weights in the input argument, WGHT, for subclass pairs (and honoring the OTHERS weight assignment, for all unspecified subclass-pairs). The weight table returned to SETUP4 via the argument, WEIGHT, contains for each intersubclass-pair either a weight =1.0, or a weight = the weight specified on a WEIGHTS card input to SETUP4, or a weight = the weight specified on an OTHERS card input to SETUP4.
- b. Bypass the initialization of subclass pair weights to 1.0 and the OTHERS weight assignment, if any. Store into the weight table returned to SETUP4 via the argument, WEIGHT, either the weight(s) for specific intersubclass-pairs, as named on WEIGHTS card(s) input to SETUP4, or the weights for interclass subclass pairs as initialized (=1.0) in internal subroutine INTWGT and sent to WGTCHK via the argument, WRKRY. All remaining intersubclass-pair weights in the WEIGHT table will be set = 0.0 (via initialization in subroutine INTWGT).

The choice of functions to be performed by WGTCHK is derived from the value of a flag, WTKEY, sent to WGTCHK via calling argument WEIGHT(2).

### D.2.1 LINKAGES

Subprogram WGTCHK is called by the SETUP4 subprogram. WGTCHK does not reference any lower level programs.

### D.2.2 INTERFACES

Subprogram WGTCHK does not use a common block. Interface with the SETUP4 subprogram is via the calling arguments.

### D.2.3 INPUTS

Calling sequence: CALL WGTCHK (WEIGHT, CLSNAM, NAMPR, WGHT, WPTR, WRKRY, NOCLS2)

<u>Parameters</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
WEIGHT	NOCLS2* (NOCLS2-1)/2 (NOCLS2=No. of subclasses)	In/Out	Output table of weights for subclass pairs. WEIGHT (1) is equivalent to ARRAY(WGHS14) in the calling program SETUP4. On input, WEIGHT(1) is used to convey the flag, SETWGT, from SETUP4 to WGTCHK. WEIGHT(2) is used to convey the flag, WTKEY, from SETUP4 to WGTCHK.
CLSNAM	NOCLS2 (=no. of subclasses)	In	Final set of subclass names from SAVTAP file, after any grouping of subclasses performed due to GROUP control card.
NAMPR	(2,WPTR)	In	Subclass names or name pairs input to SETUP4 on WEIGHTS card(s). NAMPR(1,WPTR)= subname <sub>1</sub> ,

				<p>NAMPR(2,WPTR)=  subname<sub>2</sub>,  if input, other-  wise=blank.  NAMPR(1,WPTR) may  also contain  "OTHERS", from  WEIGHTS OTHERS  card.</p>
WGHT	400		In	<p>Subclass, sub-  class pair, or  "OTHERS" weights  from WEIGHTS  card(s) input to  SETUP4 via the  WGTSCN subprogram.</p>
WPTR	1		In	<p>The number of  weights input to  SETUP4 on "WEIGHTS"  control card(s).</p>
WRKRY	(NOCLS2, .NOCLS2) NOCLS2=No. of subclasses		In	<p>Working array. If  WTKEY=WEIGHT(2)+1,  contains initializa-  tion weights for  intersubclass pairs  (=1.0), and interim  storage of subclass  pair weights, from  WGHT array. If WTKEY  (WEIGHT(2))=1, WRKRY  contains interclass  subclass pair weights  sent in from sub-  routine INTWGT.</p>



NOCLS2	1	In	The total no. of subclasses for all classes.
--------	---	----	--

#### D.2.4 OUTPUTS

The modifications to WGTCHK makes no change in line printer or file output.

#### D.2.5 STORAGE REQUIREMENTS

WGTCHK requires 296 locations in the Instruction Bank, 69 locations in the Data Bank, for a total of 365 locations.

#### D.2.6 DESCRIPTION

The calling sequence for WGTCHK is not modified. However, the use of the first argument in the calling sequence, WEIGHT, is modified so that on input to WGTCHK, WEIGHT(1) = SETWGT (from SETUP4) and WEIGHT(2) = WTKEY (from SETUP4). Two flags internal to WGTCHK, by the same names (i.e., SETWGT and WTKEY), are initialized from WEIGHT(1) and WEIGHT(2) immediately upon entry to WGTCHK.

If WTKEY=1, subprogram WGTCHK performs its usual function; i.e., pre-initializing in the working array, WRKRY, all intersubclass-pair weights either =1.0, or set to the value on the WEIGHTS OTHERS card, if it is input to SETUP4. Following pre-initialization in the working array, WRKRY, if SETWGT=2 the subprogram overrides the preset weights in the working array with weights from the input WGHT array, using subclass names in the input NAMPR array to direct the storage of input weights. When the working array is completed, the subclass-pair weight values from the working array are transferred to the weight table (WEIGHT) and output by WGTCHK via calling argument.

If WTKEY=1, WGTCHK assumes that the working array, WRKRY, contains subclass-pair weights as initialized in subroutine INTWGT(SETUP4). Pre-initialization of WRKRY is bypassed, and the flag, SETWGT, is tested. If SETWGT=2, the weights in the working array are overridden by weights from the input WGHT array, using subclass names in the input NAMPR array to direct the storage of weights into the working array, WRKRY. If SETWGT $\neq$ 2, the weights in WRKRY remain as initialized in subroutine INTWGT. The weights from WRKRY are then transferred to the weight table (WEIGHT) and output by WGTCHK via calling argument.

#### D.2.7 FLOWCHART

N/A

#### D.2.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

### D.3 SOFTWARE SUBPROGRAM NO. 3 (INTWGT)

The subprogram INTWGT is an internal subroutine in the SETUP4 subprogram. The function of INTWGT is to set=1.0 the weights of all interclass subclass-pairs, and to set=0.0 the weights of all intraclass subclass-pairs.

#### D.3.1 LINKAGES

Subroutine INTWGT is called by the SETUP4 subprogram. INTWGT does not reference any lower-level subprograms.

#### D.3.2 INTERFACES

INTWGT interfaces with subroutine SETUP4 via the calling arguments. No common blocks are utilized by INTWGT.

#### D.3.3 INPUTS

CALL INTWGT (SUBWGT, CLSUB, NOSUB2, NOCLS2)

<u>Parameter</u>	<u>Dimension</u>	<u>Type</u>	<u>In/Out</u>	<u>Definition</u>
SUBWGT	(NOSUB2, NOSUB2)	Real	In	Working array for storage of interclass subclass-pair weights and intraclass subclass-pair weights. Equivalent to SUBRAY(W1) in SETUP4.
CLSUB	NOCLS2	Integer	In	Input array containing the no. of subclasses in each class. Equivalent to ARRAY(NOSUB2) in SETUP4.

NOSUB2	1	Integer	In	The total no. of subclasses in all classes
NOCLS2	1	Integer	In	The total no. of classes

#### D.3.4 OUTPUTS

All output of subroutine INTWGT is via the first calling argument (See Section D.3.3 above.)

#### D.3.5 STORAGE REQUIREMENTS

INTWGT is internal to subroutine SETUP4. Approximately 50 storage locations are required by subroutine INTWGT alone.

#### D.3.6 DESCRIPTION

In SETUP4 when the flag, WTKEY=1, is set due to an input OPTION CLSWT control card, subroutine INTWGT is called by SETUP4 to perform the functions of setting interclass subclass-pair weights =1.0, and intraclass subclass pair weights = 0.0.

INTWGT utilizes a portion of the working array, subray, to contain the subclass-pair weights returned to SETUP4. The portion of SUBRAY utilized by subroutine INTWGT is the same part of the array that is utilized as a working array (WRKRY) in subroutine WGTCHK. Within INTWGT, the array is known as SUBWGT, dimensioned (NOSUB2, NOSUB2), where NOSUB2 = the total number of subclasses, after any grouping required by input GROUP control card(s).

The method of setting the subclass-pair weights in INTWGT is as follows:

- initialize the weight for all subclass pairs = 0.0  
 $SUBWGT(I,J) = 0.0 \quad I=1, NOSUB2; J=1, NOSUB2$   
 $SUBWGT(J,I) = 0.0$

2. given that NOCLS2 = total number of classes  
 and  $N_{K1} = \text{CLSUB}(K1) =$  number of subclasses in class K1  
 and  $N_{K2} = \text{CLSUB}(K2) =$  number of subclasses in class K2  
 then for  $K1 = 1, 2, 3, \dots, \text{NOCLS2}-1$   
 and  $K2 = K1+1, K1+2, \dots, \text{NOCLS2}$

the subclass numbers, I, for class K1 are as follows:

$$I_i = i + \sum_{L=0}^{K1-1} N_L; \quad i = 1, 2, 3, \dots, N_{K1}$$

( $N_0 = 0$ )

likewise, the subclass numbers, J, for class K2 are as follows:

$$J_j = j + \sum_{M=1}^{K2-1} N_M; \quad j = 1, 2, 3, \dots, N_{K2}$$

( $N_1 =$  Number subclasses in class 1)

3. SUBWGT (I,J) = 1.0  
 SUBWGT (J,I) = 1.0
4. Return to SETUP4, with interclass subclass-pairs = 1.0 and  
 intraclass subclass-pairs = 0.0, in SUBWGT.

#### D.3.7 FLOWCHART

N/A

#### D.3.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

APPENDIX E  
SUBPROGRAM DESCRIPTIONS

APPENDIX E

TABLE OF CONTENTS

- E.1 SOFTWARE FOR SUBPROGRAM NO. 1 (REDIF2)
- E.2 SOFTWARE FOR SUBPROGRAM NC. 2 (SETUP2)

## E.1 SOFTWARE SUBPROGRAM NO. 1 (REDIF2)

Subprogram REDIF2 of the CLASSIFY processor was modified to, (1) accept a revised APRIORI control card and (2) set a flag (APRFLG) to indicate that in the subprogram SETUP2, the apriori probability values for each subclass are to be computed, using subclass population data from the statistics file, SAVTAP.

### E.1.1 LINKAGES

Subprogram REDIF2 is called by subprogram SETUP2. Subprogram REDIF2 calls the following subprograms:

BMFIL, CATSCN, CRDSTA, FIND, FLTNUM, GRPSCN, NUMBER, NXTCHR, ORDER, and TDATE.

### E.1.2 INTERFACES

The common blocks used by subprogram REDIF2 are: CLASS, GLOBAL, INFORM. The variable, APRFLG, in common block CLASS was involved in the modification to subprogram REDIF2.

### E.1.3 INPUTS

- Calling sequence: Call REDIF2 (ARRAY, TOP, APRIOR, KATNO, BMATRX, PRIORI)

<u>Parameters</u>	<u>Dimension</u>	<u>In/out</u>	<u>Definition</u>
ARRAY	10 600	In	Variably dimensioned working storage
TOP	1	In	Maximum usable location in ARRAY - TOP = 10, 600
APRIOR	60	In	Storage for a priori probability values - Initialize = 0.0 in REDIF2



<u>Parameters</u>	<u>Dimension</u>	<u>In/out</u>	<u>Definition</u>
KATNO	60	In	Unused in REDIF2.
BMATRX	(BMCOMB, BMFEAT) BMCOMB = no. of linear com- binations in B-matrix BMFEAT = no. of channels used in B-matrix	Out	Storage location for B-matrix
PRIORI	60	Out	Storage for apriori probability values read from input APRIORI con- trol card.

- Revised control card

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
APRIORI	FILE	If the first nonblank character on the APRIORI card (parameter field) is an F, the flag, APRFLG, is set = -777777. This value of APRFLG will initiate in SETUP2 the computation of subclass apriori probability values from the statistics file's subclass pixel population data.

#### E.1.4 OUTPUTS

File output is unchanged by these modifications to REDIF2.

Line printer output is modified to include the message "\*\*\* CLSFY/ REDIF2 - BAD INPUT ON APRIORI CARD - DEFAULT APRIORI PROBABILITY VALUES WILL BE USED. This message is printed immediately following the listing of the APRIORI control card as is usually done, and is only printed when the first character on the APRIORI control card is neither a legitimate "F", nor a number in the range of 0, 1, ..., 9.

#### E.1.5 STORAGE REQUIREMENTS

Subroutine REDIF2 requires 708 locations in the Instruction Bank, 343 locations in the Data Bank, for a total of 1051 locations.

#### E.1.6 DESCRIPTION

REDIF2 modifications include: (1) a test for the character "F" as the first non-blank character on the APRIORI control card, (2) the initialization of the variable, APRFLG = -777777 if the character "F" is encountered.

If the first nonblank character is not an F, REDIF2 proceeds to the usual reading of apriori probability values from the parameter field of the APRIORI card. If the first character is not a legitimate numeric (number), an error message is printed (see section E.1.4). In the case of an error on the APRIORI control card the flag variable, APRKEY is set = 0.

#### E.1.7 FLOWCHART

N/A

#### E.1.8 LISTING

The listing is available in the Data Techniques Laboratory, JSC Building 17.

## E.2 SOFTWARE SUBPROGRAM NO. 2 (SETUP2)

Subprogram SETUP2 tests the value of the flag, APRFLG, which is returned from subprogram REDIF2 via common block CLASS and if APRFLG = -777777, SETUP2 uses subclass pixel population data obtained from the statistics file, SAVTAP, to compute the subclass apriori probability values which are stored in the APRIOR array, and subsequently used in classification.

### E.2.1 LINKAGES

Subprogram SETUP2 is called by the CLASSIFY processor driver program, CLSFY. Subprogram SETUP2 calls the following subprograms: FIND, NUMBER, REDIF2, REDSAV, and WRTBM.

### E.2.2 INTERFACES

Subprogram SETUP2 uses the named common blocks CLASS, GLOBAL, and INFORM. The variable, APRFLG, in common block CLASS was involved in the modifications to SETUP2.

### E.2.3 INPUTS

- Calling sequence: CALL SETUP2 (ARRAY, TOP, FLDFLG, APRIORI, BMATRX, KATNO)

<u>Parameter</u>	<u>Dimension</u>	<u>In/out</u>	<u>Definition</u>
ARRAY	10,600	In	Variably dimensioned working storage.
TOP	1	In	Maximum usable location in ARRAY - set = 10,600
FLDFLG	1	In	Unused in SETUP2
APRIOR	60	Out	Storage for the subclass apriori probability values.

<u>Parameter</u>	<u>Dimension</u>	<u>In/out</u>	<u>Definition</u>
BMATRIX	(BMCOMB, BMFEAT) BMCOMB = no. of linear com- binations in B -matrix BMFEAT = no. of channels used in the B -matrix.	OUT	Storage for the B - transformation matrix
KAT110	60	Out	Class-category correspondence.

#### E.2.4 OUTPUTS

There were no changes in either file output or line printer output as a result of these modifications to SETUP2.

#### E.2.5 STORAGE REQUIREMENTS

Storage required: Data-1000<sub>8</sub> Code-1442<sub>8</sub>

#### E.2.6 DESCRIPTION

The modifications to SETUP2 include:

1. A test on the flag, APREFLG = -777777, to trigger a transfer to the coding for computation of subclass apriori probability values using subclass pixel populations available in the array, KEPPTS (common block INFORM).
2. The computation and storage of subclass apriori probability values, in the following manner:

Total no. of subclass pixels

$$TKEPTS = \sum_{I=1}^{NOSUB2} KEPPTS(I)$$

apriori probability value for subclass I

$$\text{APRIOR}(I) = \frac{\text{KEPPTS}(I)}{\text{TKEPTS}}$$

I = 1, NOSUB2 (NOSUB2 = total no. of subclasses)

E.2.7 FLOW CHART

N/A

E.2.8 LISTING

The listing is available in the Data Techniques Laboratory,  
JSC Building 17.

APPENDIX F  
SUBPROGRAM DESCRIPTIONS

APPENDIX F

TABLE OF CONTENTS

F.1	Software for Subprogram No. 1 (SETUP3)
F.2	Software for Subprogram No. 2 (DTCHK)
F.3	Software for Subprogram No. 3 (REDIF3)
F.4	Software for Subprogram No. 4 (DSPLY2)
F.5	Software for Subprogram No. 5 (DOTPCT)
F.6	Software for Subprogram No. 6 (DOTSUM)
F.7	Software for Subprogram No. 7 (DSPLAY)
F.8	Software for Subprogram No. 8 (TAPLAB)

## F.1 SOFTWARE SUBPROGRAM NO. 1 (SETUP3)

The modifications to SETUP3 adapt SETUP3 to LACIE Procedure 1 image processing requirements. The modifications accomplish the following purposes:

1. Implement the input of dot data parameters from the DOTFIL file created by the DOTDATA processor, when the new DOTFIL control card is input.
2. Utilize existing storage in ARRAY normally occupied by training field information to store the dot data parameters
3. Perform a check for agreement between the set of dot category names and the set of classification category names from the input classification file, MAPTAP, and
4. Order the dot data in storage by ascending line number with the group of dots on a given line ordered by ascending sample number.

An added dot performance flag, DOTKEY and an associated error flag, DOTERR, IS TESTED. If DOTKEY>0 and DOTERR=0, activities related to dot performance summary processing is initiated in SETUP3. If either DOTKEY=0 or DOTERR>0, dot performance summary processing is not initiated.

If dot data processing is initiated and if there is any mismatch of category names between analyst-labeled dot categories (DOTFIL) and classification categories (MAPTAP), dot data processing is terminated, a message written to warn the user (see F.1.4(2) below) and the usual DISPLAY processor functions performed as detailed by the input control cards. Also, other error checks are made on the input DOTFIL data, before proceeding with dot processing (see items 5 and 7 in section F.1.4).



### F.1.1 LINKAGES

Subprogram SETUP3 is called by the DISPLAY processor driver program, DSPLAY. Subprogram SETUP3 makes calls to the following subprograms: FIND, NUMBER, REDIF3, WRTFLD, FSBSFL, internal subprograms SETKEY and FIXFLD and three new subprograms TAPLAB, RDDOTS, and internal subprogram DTCHK.

### F.1.2 INTERFACES

Subprogram SETUP3 references two common blocks, DISPL and GLOBAL. The modifications to SETUP3 resulted in the addition of two variables to the common block, DISPL. The two added variables are DOTKEY and DOTERR.

### F.1.3 INPUTS

Calling sequence: CALL SETUP3 (ARRAY, TOP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10 600	In	Variably dimensioned working array.
TOP	1	In	Maximum usable location in ARRAY. Set=10 600 in the EOD-LARSYS executive program, MONTOR.

### F.1.4 OUTPUTS

Modifications to SETUP3 include eight additions to line printer output:

1. In the user input summary of requests the following message was added:

*F-2*  
*195*

PRINT DOT DATA PERFORMANCE SUMMARIES FOR DOT DATA FROM  
FORTRAN UNIT XX, NO. XX AND TAPE (OR FILE) CCCCCC#11

If OPTION OUTLINE control card is input, the following additional line is printed:

OUTLINE THE DOTS ON THE CLASSIFICATION MAP

2. If each category name from the input dot data file, DOTFIL, does not agree with one of the category names from the input classification results file, MAPTAP, the following message will be printed (from internal subprogram DTCHK):

DOT DATA PERFORMANCE SUMMARIES WILL NOT BE PRODUCED - THE  
CATEGORY NAMES FROM MAPTAP AND DOTFILE DO NOT MATCH -  
CATEGORY NAMES FROM MAPTAP ARE: MCAT1, MCAT2, MCAT3, ...  
CATEGORY NAMES FROM DOTFILE ARE: DCAT1, DCAT2, DCAT3, ...

3. If each category name from DOTFIL matches each category name on MAPTAP, or if each category name on DOTFIL matches one of the categories on MAPTAP and there are additional categories on the MAPTAP which do not match a DOTFIL category name, the following message is printed (from internal subprogram DTCHK):

THERE IS OR IS NOT 1-1 CORRESPONDENCE BETWEEN THE CATEGORY  
NAMES FROM DOTFILE - DCAT1, DCAT2, ...  
AND THE CATEGORY NAMES FROM MAPTAP - MCAT1, MCAT2, MCAT3, ...

4. If the number of categories (NOCAT) on the input classification results file, MAPTAP, is zero(=0) (i.e., category classification not performed), the following message is printed:

\*\*\* CLASSIFICATION BY CATEGORY (ON MAPTAP) IS REQUIRED IN  
ORDER TO PROCESS THE DOT DATA \*\*\*

\*\*\* DOT PERFORMANCE SUMMARIES WILL NOT BE OUTPUT \*\*\*

5. If the number of categories (NDCAT) on the input dot data file, DOTFIL, is zero (=0), the following message is printed:

```
*** SETUP3 ** FROM DOTFILE, THE NO. OF DOT CATEGORIES=XX
**** DOT PERFORMANCE SUMMARIES WILL NOT BE PROVIDED ****
```

6. If the total working storage available in ARRAY (10600 locations) is exceeded by the usage of ARRAY in SETUP3, the amount of excess storage locations attempted is output in the message:

```
***** DISPLAY/SETUP3 - CORE OVERFLOW (TOP-TOP2) BY
XXXX -- EXECUTION TERMINATED *****
```

7. If the number of dots on the input dot data file, DOTFIL, exceeds 250 dots, the following message is printed:

```
**** DSPLAY/DSPLY2 **** NO. OF DOTS = XXXX --- EXCEEDS THE
MAX. ALLOWABLE (250)
**** DOT PERFORMANCE SUMMARIES WILL NOT BE PRODUCED ****
```

8. After reading the dot data from DOTFIL, using subroutine RDDOTS to read the file, a printout of the dot file contents is provided, as follows:

DOT FILE INFORMATION:

NO. DOT CATEGORIES = XX

TOTAL NO. DOTS = YYY

DOT CATEGORY NAMES:

DCAT1, DCAT2, DCAT3, ...

DOT RECORD:

<u>DOT NO.</u>	<u>SAMPLE</u>	<u>LINE</u>	<u>TYPE</u>	<u>CATEGORY</u>
1	XXA	YYA	2	1
2	XXB	YYB	2	2
3	XXC	YYC	2	2
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

### F.1.5 STORAGE REQUIREMENTS

SETUP3 requires 1530 locations in the Instruction Bank, 749 locations in the Data Bank, for a total of 2279 locations.

### F.1.6 DESCRIPTION

The modifications to subprogram SETUP3 are as follows:

1. The storage allocation in ARRAY for training fields (TRNSAV, TRNFLD, TRNVER) is used for final storage of the dot data information input by the subprogram RDDOTS. This storage area is used after the usual printout of SAVED TRAINING FIELDS. Treating each dot as a one pixel training field, the base addresses FLDSV2, VERTX2, and FIELD2, for three areas in ARRAY are recomputed, based on the total number of dots ( $\leq 250$ ) read from DOTFIL.

Base addresses for ARRAY storage are computed as follows:

FLDSV2 = TOP1, where TOP1 is the first available location in ARRAY, above the area in ARRAY used for test field information.

FIELD2 = FLDSV2 + 1000 (allowance for information on 250 dots in TRNSAV (4,250))

VERTX2 = FIELD2 + 1250 (allowance for information on 250 dots in TRNFLD (5,250))

TOP2 = VERTX2 + 500 (allowance for information on 250 dots in TRNVER (2,250))

2. The tests on TRNKEY are modified to include a test on the flag, DOTKEY (dot data processing flag).
3. Initiate the output of a correct list of user-requested options. (See section F.1.4(1)).

4. The dot data performance summary flag, DOTKEY=1, is used to initiate input of dot data from the DOTFIL, and also a comparison of category names input from the classification results file, MAPTAP, against the category names read from the dot data file, DOTFIL. If there is a one-for-one match on category names, dot data performance summaries will be produced by DISPLAY. If there is a mismatch between the two sets of category names, the message described in section F.1.4 (2) above will be written, the DOTKEY flag reset = 0, TRNKEY set = 0, and the error flag DOTERR set = 0. Dot data processing will not be done by the DISPLAY processor, when DOTKEY = 0 or DOTERR > 0.
5. TOP2 is the beginning address in ARRAY for a scratch area, which is used to temporarily contain the input dot data from DOTFIL by subroutine RDDOTS. The maximum size required is 1000 locations, for DOTS (4, NDOTS). Subroutine RDDOTS initializes DOTS (1,I) = dot sample number, DOTS (2,I) = dot line number, DOTS (3,I) = dot type, and DOTS (4,I) = dot category number (on DOTFIL). Subroutine RDDOTS also initializes the total number of dots (TOTVT2), the number of dot categories (NDCAT), and the dot category names (DOTCAT).
6. The existing array, CARD (62), in SETUP3 is used to temporarily contain the dot category names read from DOTFIL by subroutine RDDOTS (i.e., DOTCAT, is equivalenced to CARD).
7. An internal (to SETUP3) subprogram, subroutine DTCHK, checks the dot category names from DOTFIL against the classification category names from MAPTAP. If each dot category name is present in the set of classification category names, DTCHK orders the dot data in DOTS (4,NDOTS) (i.e., ARRAY (TOP2)) such that the dots are in sequential ascending line number order, with all dots on a given line ordered by ascending

pixel numbers. After ordering the dot data, DTCHK stores the dot data into the areas in ARRAY usually reserved for training field data. The training field data input from MAPTAP is overwritten in the process.

The training field storage areas in ARRAY are initialized with dot data information as follows:

a. Storage is allocated such that

TRNSAV(1,1) = ARRAY(FLDSV2),

TRNFLD(1,1) = ARRAY(FIELD2),

TRNVER(1,1) = ARRAY(VERTEX2)

b. The training field-related parameters, TOTVT2 and NOFLD2, are set = the number of dots

c. Dot data is moved from the temporary storage in DOTS (4,NDOTS) to the training field storage areas in ARRAY in the following manner:

TRNSAV(1,I) = unused in SETUP3

TRNSAV(2,I) = MAPTAP category number corresponding to dot I analyst-labeled category name

TRNSAV(3,I) = dot type (usually = 2, for bias-correction dots)

TRNSAV(4,I) = 1 = number of "vertices"

TRNVER(1,I) = dot sample number

TRNVER(2,I) = dot line number

For I = 1,2,3,..., NOFLD2 (= total number of dots).

Internal subprogram FIXFLD initializes TRNFLD as usual to contain the coordinates of the rectangular area surrounding each dot, such that

TRNFLD(1,I) = starting line number

TRNFLD(2,I) = ending line number

TRNFLD(3,I) = starting sample number

TRNFLD(4,I) = ending sample number

TRNFLD(5,I) = pointer to the location in TRNVER for the  
vertex of the field (dot)

TRNFLD is later utilized to outline the dots on the  
classification map.

8. The size of the performance table PCTAB is set = NOFLD2 =  
total number of dots (i.e., PCTSZ = NOFLD2 if DOTKEY > 0) and  
PCTAB utilizes the scratch area in ARRAY, beginning at ARRAY  
(TOP2), as usual.
9. After the ordering of the dot data, initialization of TRNSAV,  
TRNVER, and TRNFLD, the usual call to the subprogram SETKEY  
(if DOTKEY > 0), will be bypassed, and SETUP3 will set FLDKEY =  
0, DOTKEY = number of dot categories, and will take the  
normal return to DSPLAY.

#### F.1.7 FLOWCHART

N/A

#### F.1.8 LISTING

The listing for SETUP3 is available in the Data Techniques  
Laboratory, JSC Building 17.

## F.2 SOFTWARE SUBPROGRAM NO. 2 (DTCHK)

DTCHK is a new internal subroutine in subroutine SETUP3. The functions of this routine are:

1. To check the correspondence between DOTFILE's and MAPTAP's category names.
2. To order the dots by ascending line number, and for all dots on a given line, by ascending sample number.

### F.2.1 LINKAGES

The subprogram, DTCHK, is called by SETUP3 and is internal to SETUP3. The subprogram is not available to be referenced outside of SETUP3.

### F.2.2 INTERFACES

The interface with SETUP3 is by means of three calling arguments and five global variables.

### F.2.3 INPUTS

Calling sequence: CALL DTCHK (DOT, TRNSAV, TRNVER, DCATNM,  
NDOTS, NDCAT)



<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DOT	(4,NDOTS) where NDOTS = total number of dots ( <u>≤</u> 250)	In	DOT(1,I) = sample no. (2,I)=line no. (3,I)=dot type (4,I)=labeled dot (DOTFILE) cate- gory no.
TRNSAV	(4,NDOTS)	Out	TRNSAV (1,I) = unused (2,I) = category number (3,I) = dot type (4,I) = 1 = no. of "vertices"
TRNVER	(2,NOFLD2) where NOFLD2 = total no. of dots ( <u>≤</u> 250)	In	TRNVER(1,I) = sample no. (2,I)=line no.
DCATNM	NDCAT (NDCAT=no. of dot categories)	In	The dot category names
NDOTS	1	In	Total no. of dots from DOTFILE ( <u>≤</u> 250)
NDCAT	1	In	Total no. of dot categories from DOTFILE ( <u>≤</u> 60)

#### F.2.4 OUTPUTS

See the messages described in Section F.1.4 for the line printer outputs from DTCHK.

#### F.2.5 STORAGE REQUIREMENTS

DTCHK is internal to subroutine SETUP3. See Section F.1.5 for SETUP3 storage requirements, including DTCHK.

#### F.2.6 DESCRIPTION

If there is a match in category names from MAPTAP for each category name from DOTFIL, the type of correspondence (i.e., 1-1 or NOT 1-1) is printed out along with each set of category names. The NOT 1-1 correspondence is for the case of more MAPTAP (classification) categories than dot categories.

If there is not a match for each dot category name in the set of classification (MAPTAP) categories, an error message is printed out. DOTKEY and TRNKEY is reset to zero (=0) and DOTERR is set to one (=1). See Section F.1.4 for the message output.

#### F.2.7 FLOWCHART

N/A

#### F.2.8 LISTING

The listing is available in the Data Techniques Laboratory, JSC Building 17.

### F.3 SOFTWARE SUBPROGRAM NO. 3 (REDIF3)

The modifications to subprogram REDIF3 result in the following additional functions being performed:

1. A new control card, DOTFILE, is read and decoded
2. A new flag, DOTKEY, is set and used in other subprograms to indicate that dot data classification performance summaries are requested. DOTKEY is added to the common block DISPL.
3. A new flag, DOTERR, is initialized and used in other subprograms. DOTERR set to one if MAPTAP does not contain any dot category names. DOTERR is also added to COMMON/DISPL/.

#### F.3.1 LINKAGES

Subprogram REDIF3 is called by subprogram SETUP3. Subprogram REDIF3 calls the following subprograms: CHIN, FIND, FLTNUM, LAREAD, NXTCHR.

#### F.3.2 INTERFACES

Subprogram REDIF3 references the common blocks DISPL and GLOBAL. Two new variables, DOTKEY and DOTERR are added to the common block DISPL. DOTKEY is tested in subprograms SETUP3, DSPLY2, and DOTSUM, and used for initiating dot data processing. DOTERR may be reset or tested in subprograms SETUP3 and DISPLAY. It is used to inhibit dot data processing, when an error condition has been detected.

#### F.3.3 INPUTS

The calling sequence for subprogram REDIF3 is unchanged by these modifications.

Calling sequence: CALL REDIF3 (TSTSAV, TSTFLD, TSTVER, VDIM)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
TSTSAV	(4,250)	Out	Used for test field or designated field information: TSTSAV(1,N) = field name TSTSAV(2,N) = class number TSTSAV(3,N) = subclass TSTSAV(4,N) = no. of vertices for field N N = 1,2,3,... no. of fields
TSTFLD	(5,200)	Out	Used for storage of coordinates of the rectangular area surrounding a test or designated field.
TSTVER	VDIM	Out	Used for storage of the sample, line coordinates of the vertices for all input test or designated fields.
VDIM	1	In	Total no. of locations allocated for storage of all test or

or designated  
field vertices  
in the variably-  
dimensioned ARRAY  
in SETUP3.

● New control card

<u>Keyword</u>	<u>Parameter</u>	<u>Function</u>
DOTFILE	INPUT/UNIT=m, FILE=n (Default: UNIT=8, FILE=1)	Implements the reading of dot data from the speci- fied (or default) unit and file in SETUP3 and the out- put of dot performance summaries in DSPLY2. NOTE: If OUTPUT/UNIT=X, FILE=Y appears in the parameter field, it will be ignored.

F.3.4 OUTPUTS

The line printer output lists the new control card at the time it is read by REDIF3. If MAPTAP does not contain any dot category names, an error message is printed, as follows:

```
*** CLASSIFICATION BY CATEGORY (IN MAPTAP) IS REQUIRED IN ORDER  
    TO PROCESS THE DOT DATA ***
```

DOTERR is set to one (=1) in this case. If INPUT is not found on the DOTFILE card, the default unit and file will be used to read the DOTFILE. If an error in format is sensed on the DOTFILE card following INPUT, the error message:

```
*** ERROR IN DOTFILE CARD-TERMINATE EXECUTION ***
```

is printed out and exit is called.

### F.3.5 STORAGE REQUIREMENTS

REDIF3 requires 964 locations in the Instruction Bank, 411 locations in the Data Bank, for a total of 1375 locations.

### F.3.6 DESCRIPTION

The order of the modifications to subprogram REDIF3 are:

1. The keyword array, OPT, is redimensioned to 15 locations, and the new control card keyword, "DOTFIL", is added to OPT.
2. The new dot data processing flag, DOTKEY = 0 is initialized, as well as TSTKEY (=0), TRNKEY (=0), and DOTERR (=0).
3. For match-up of keyword DOTFIL in OPT array versus input control card keyword, a transfer to a new statement number 785 is added to the transfer list.
4. Statement number 785 and additional coding to test for the number of MAPTAP categories (NOCAT) > 0 coding to set the new flag, DOTKEY = 1, and a transfer to statement number 789 if NOCAT  $\leq$  0. At statement number 789 coding is added to set DOTERR = 1, TRNKEY = 0.
5. After existing statement number 870, if DOTKEY > 0, or DOTERR > 0, TSTKEY is set = 0 (following the test for input of test fields). Following the test for DOTKEY > 0, DOTERR > 0, the usual test for "designated" field input is performed. They may be either "DESIGNATED OTHER" or "DESIGNATED UNIDENTIFIABLE." For dot data processing, the test field flag, TSTKEY, is turned off (TSTKEY=0). The designated field flag DESKEY is on (DESKEY=1), if and only if any fields read in are identified as "DESIGNATED" fields.

### F.3.7 FLOWCHART

N/A

F.3.8 LISTING

The listing is available in the Data Techniques Laboratory,  
JSC Building 17.

F-16  
209

#### F.4 SOFTWARE SUBPROGRAM NO. 4 (DSPLY2)

DSPLY2 is modified to test the dot performance flag, DOTKEY. If DOTKEY > 0, a call to a new internal subprogram DOTPCT is initiated to create the dot classification performance table, PCTAB. Also, a call to an additional new internal subprogram, DOTSUM produces the two dot data classification summaries. The capability to border each dot on the classification line printer map, as is normally done with training fields, is preserved. The dots are outlined on the classification map if OPTION OUTLINE control card has been input to the DISPLAY processor.

##### F.4.1 LINKAGES

Subprogram DSPLY is called by the DISPLAY processor driver program DSPLAY. Subprogram DSPLY2 calls the following subprograms: DESIG, FLDBOR, MAPHD, PCT, PRSUM, WRTHED, WRTLN, PRTCOL, SETMRG (no longer effective), RWRITE, WRTHD, RREAD, and two new internal subprograms, DOTSUM and DOTPCT.

##### F.4.2 INTERFACES

Subprogram DSPLY2 references two common blocks DISPL and GLOBAL.

##### F.4.3 INPUTS

The calling sequence for DSPLY2 is unchanged as a result of these modifications. The usage of the three training field information array (allocated in ARRAY) in the calling arguments - TRNSAV, TRNFLD, and TRNVER - are revised by these modifications. Also, the usage of the classification performance table PCTAB (also allocated to an area in ARRAY), is modified for dot data performance tabulation. The revision to usage of the training field storage area is in the type of information placed in the three areas, and is only active for the dot performance summary option (i.e., DOTKEY > 0). Otherwise, the training field storage areas are available for their normal usage. The main change to usage



of PCTAB is the size (dimension) of the storage area used for dot classification tabulation. For data, PCTAB may be treated as a linear (one-dimensional) array, since there is only a one dot - one subclass correspondence possible in the classification of each dot. Once again, this usage of PCTAB is only activated by the request for dot performance summaries (DOTKEY > 0), and normal usage of PCTAB for training or test fields remains available when not in the dot processing mode in the DISPLAY processor.

Calling sequence: CALL DSPLY2 (TRNSAV, TRNFLD, TRNVER, TSTSAV, TSTFLD, TSTVER, PCTAB)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
TRNSAV	(4,NOFLD2) NOFLD2 = total no. of dots	In	Initialized in SETUP3 as follows if DOTKEY > 0): TRNSAV(1,I) = unused (scratch) TRNSAV(2,I) = MAPTAP (classification) category no., for the MAPTAP category matching the DOT's category name. TRNSAV(3,I) = DOT type (=1,or 2) TRNSAV(4,I) = no. of "vertices" for DOT I, I = 1,2,..., NOFLD2 = total no. of dots

TRNFLD	(5,NOFLD2)	In	Coordinates of rectangular area surrounding each dot. Initialized in SETUP3 as follows: TRNFLD(1,I) = starting line no. TRNFLD(2,I) = ending line no. TRNFLD(3,I) = starting sample no. TRNFLD(4,I) = ending sample no. TRNFLD(5,I) = loca- tion in TRNVER for vertex of dot I. I = 1,2,...,NOFLD2 = total no. of dots
TRNVER	(2,TOTVT2) (TOTVT2 = total no. of vertices for all dots = total number of dots	In	Array storage for training field (dot) vertex coordinates: TRNVER(1,I) = sample number TRNVER(2,I) = line number I = 1,2,..., total no. of dots = NOFLD2 TOTVT2
TSTSAV	(4,NOFLD3) NOFLD2 = total no. of test or designated fields (<200)	In	Array storage for test or designated fields, equivalent to TRNSAV for training fields.

TSTFLD	(5,NOFLD3)	In	Used for test or designated fields, equivalent to TRNFLD for training fields.
TSTVER	(2,TOTVT3) (TOTVT3 = total no. of vertices for all test or designated fields)	In	Used for test or designated fields, information equivalent to TRNVER for training fields.
PCTAB	(NOFLD2,1) NOFLD2 = total no. of dots	In	Storage area, beginning at ARRAY (PCTID3), for dot performance tabulation (PCTAB(I,1) = classified subclass for dot I, I = 1,2,...,NOFLD2, upon return from internal subprogram, DOTPCT).

The input of DESIGNATED fields to DSPLY2, in the test field storage areas - TSTSAV, TSTFLD, and TSTVER - is preserved (not changed), and if designated fields have been input to the DISPLAY processor (the flag DESKEY is set = 1), the usual processing of the classification record to mark pixels in the designated area(s) is performed in DSPLY2, with the call to subroutine DESIGN, as usual.

#### F.4.4 OUTPUTS

There are three additional messages printed from DSPLY2, as a result of these modifications when in the dot processing mode

DOTKEY > 0. The dot classification performance table, PCTAB, is checked after initialization in internal subprogram DOTPCT.

1. If the classification subclass number for any dot exceeds NOSUB3 (= the threshold "subclass" number), it is assumed that the dot has been classified "DESIGNATED", in which case the message is printed:

sample, line  
"DOT (XXX, YYY) IS IN THE DESIGNATED AREA"

2. If the classification subclass number for a dot is zero (=0), the dot has been found to have sample, line coordinates not included in the actual area classified, and the subclass for the dot set = 0 in internal subprogram DOTACT. In this case, the message is printed:

sample, line  
"DOT (XXX, YYY) IS NOT IN THE CLASSIFIED AREA"

3. If the number of dot categories matching MAPTAP (classification) categories is not equal to the total number of dot categories on the input DOTFILE, the message is printed (from internal subprogram DOTSUM):

"\*\*\*\* DSPLY2/DOTSUM -- DISCREPANCY IN DOTFILE INFORMATION \*\*\*\*  
NO. OF DOT CATEGORY LABELS MATCHING MAPTAP CATEGORY NAMES = XX  
NO. OF DOT CATEGORIES IS GIVEN AS = YY"

The classification map and the classification summary for the total area classified is output from DSPLY2, as usual.

In addition, as a result of these modifications, following the usual classification map and associated classification summary, two dot classification performance summaries are output, when in the dot processing mode (DOTKEY > 0). The two dot performance summaries - one by dot category, the other by individual dot - are output from the new internal subprogram, DOTSUM. See Section F.6.4 for a detailed description of the output dot classification performance summaries.

When in the dot processing mode (DOTKEY > 0), DSPLY2 ignores any "test" field processing - i.e., outlining of "test" fields on the classification map.

When in the dot processing mode (DOTKEY > 0) and when the OPTION OUTLINE control card has been input to the DISPLAY processor, the flag, TRNKEY, will have been set = 1, in which case DSPLY2 outlines each dot on the classification map with a border of asterisks ("\*"), as is normally provided for training fields when not in the dot processing mode.

#### F.4.5 STORAGE REQUIREMENTS

The subprogram, DSPLY2, including the two added internal subprograms, DOTPCT and DOTSUM, requires 2106 locations in the Instruction Bank, 13340 locations in the Data Bank, for a total of 15446 locations.

#### F.4.6 DESCRIPTION

The modifications to DSPLY2 are to add capabilities related to LACIE Procedure 1 image processing techniques. LACIE Procedure 1 techniques involve dots, which are one-pixel fields, category-labeled by photo-interpreters. Normal (non-LACIE Procedure 1) capabilities and options are preserved in DSPLY2, when not in the dot processing (LACIE Procedure 1) mode. To achieve the additional capabilities, DSPLY2 is modified as follows:

1. A test on the dot processing flags, DOTKEY and DOTERR is made, to bypass the usual initialization of the performance table, PCTAB, to zero. This is necessary due to the revised dimension of PCTAB when it is used for dot classification tabulations.

2. TRNSAV(4,I), I = 1, NOFLD2 (no: of dots) is set = 2, to satisfy the requirement of the field bordering subprogram, FLDBOR, for the number of vertices of a one-pixel field.
3. The IR array, used in the classification performance to contain the classification record of three adjacent scan lines, is reused in the new internal subprogram, DOTSUM, as a scratch area for several arrays used internally in DOTSUM. The equivalence of these arrays to portions of the IR array is done in DSPLY2, thereby making these arrays global and unnecessary to include in the calling sequence to DOTSUM.
4. In the usual scan line-by-scan line classification performance loop in DSPLY2, tests are added for dot processing (DOTKEY > 0 and DOTERR  $\neq$  1), in order to go through additional coding which tests for the presence of dots on the current scan line, and provides a call to the new internal subprogram, DOTPCT, to build the dot classification tabulation in PCTAB. Tests are made before calling DOTPCT, to see that there are dots on the current scan line.
5. The usual call to subprogram PCT, to build performance tables for either test or training fields, is by-passed when in the dot processing mode, but remains available as usual when not doing dot processing.
6. The bordering of dots on the classification map is performed by subprogram FLDBOR, when the training field flag, TRNKEY, is set = 1. (This will be the case when the OPTION OUTLINE card is input to the DISPLAY processor). Test field bordering is bypassed, by virtue of the test field flag, TSTKEY, having been set = 0, when in the dot processing mode.
7. Following the call to subroutine PRSUM, which prints the classification map, the dot classification table, PCTAB, is checked for dots which have subclass numbers greater than NOSUB3 (the thresholded subclass number), or equal zero (=0) which is the case when internal subprogram DOTPCT

determines that the dot coordinates are not compatible with the pixel coordinates of the classified area. Any dots in the classification table, PCTAB, having a subclass number greater than NOSUB3 is assumed to be either "DESIGNATED OTHER" (=NOSUB3+5) or "DESIGNATED UNIDENTIFIABLE" (=NOSUB3+4), in which case the dot classification subclass is set = 0 in PCTAB. For each dot that is already set = 0 in DOTPCT, a message to the user is printed out to the effect that the dot is not contained in the set of pixel which were classified. For dots set = 0 due to being within a "designated" area, a message to the user is printed out identifying the dot as being in a designated area. These dots will not be considered in the dot classification summaries provided later in internal subprogram DOTSUM.

8. Following the printout of the classification map by PRTSUM and the checking of dot classification described in (7) above, a call is made to the new internal subprogram, DOTSUM, which will provide the two dot classification summaries described in section F.4.4, using the dot classification tabulation in PCTAB as the basis for the summaries.

#### F.4.7 FLOWCHART

The flowchart for the new internal subprogram, DOTSUM is provided in section F.6.7. The flowchart for the new subprogram, DOTPCT, is provided in section F.5.7.

#### F.4.8 LISTING

The listing is available in the Data Techniques Laboratory, JSC Building 17.

## F.5 SOFTWARE SUBPROGRAM NO. 5 (DOTPCT)

DOTPCT is a new internal subprogram in subprogram DSPLY2. DOTPCT is called only if the DOTKEY flag is on (>0). The functions of this subprogram are:

1. If dot (sample, line) is not contained in the classified area, PCTAB (K,l) is set to zero. K refers to the appropriate dot (sample, line).
2. If dot (sample, line) is contained in the classified area, PCTAB (K,l) is set to its respective classified subclass number.

### F.5.1 LINKAGES

Subprogram DOTPCT is called by DSPLY2 and does not reference any other subprograms, nor may be referenced by any other routine external to DSPLY2.

### F.5.2 INTERFACES

The interface with subprogram DSPLY2 is by means of three calling arguments.

### F.5.3 INPUTS

Calling sequence: CALL DOTPCT (PCTAB, FIELD, IR)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
PCTAB	(NOFLD2,1) NOFLD2=no. of dots	Out	Table of subclass numbers for the dots. PCTAB(I,1)=classifi- cation subclass no. for dot I.



FIELD	(2,NOFLD2)	In	Table of dot coordinates. FIELD(1,I) = pixel no., FIELD(2,I)=line no. for dot I. (equivalent to TRNVER in DSPLY2)
IR	PTS PTS=no. of pixels on each classified scan line.	In	The classification record for one scan line. IR(J)= subclass no. for pixel J.

In addition to the calling arguments, subprogram DOTPCT utilizes global variables which have been initialized in DSPLY2, or sent to DSPLY2 already initialized. This set of global variables required by DOTPCT are:

1. PCTKEY = flag for initializing PCTAB to zero (=0) for all dots - this is done once on the first call to DOTPCT (i.e., first scan line), and the flag PCTKEY is then reset in DSPLY2 (PCTKEY > 0) so that pre-initialization of PCTAB is bypassed after the first call to DOTPCT.
2. SAMSTR = first pixel number in the classification record, IR.
3. SAMINC = pixel increment, in the classification record IR.
4. BCNT = starting position, in the dot "vertex" coordinate array, for dots having a line number equal to the current classification scan line number.
5. ECNT = ending position in the dot "vertex" coordinate array, for dots having a line number equal to the current classification scan line number.

#### F.5.4 OUTPUTS

The tabulation of subclass number for each dot is output by DOTPCT, via the calling argument, PCTAB.

#### F.5.5 STORAGE REQUIREMENTS

Subroutine DOTPCT is internal to subroutine DSPLY2. The storage requirements for DOTPCT are included in the storage requirements for DSPLY2 (see section F.4.5).

#### F.5.6 DESCRIPTION

DOTPCT is called from DSPLY2 once for each classified scan line input from MAPTAP in DSPLY2. Tests are made in DSPLY2 to determine that there are dots with the given scan line number, before calling DOTPCT. In DOTPCT, a check is made on the pixel number for each dot with the given scan line number, to ascertain that the dot is present in the actual pixels classified. This must be done due to the possibility of sample and/or line incrementation in the classified area by the CLASSIFY processor which generated the classification results file, MAPTAP.

For those dots present in the given classification record (IR), the subclass (number) at the dot's position in the classification record is extracted from the classification record (IR) and placed into the dot's position in the performance table, PCTAB.

For those dots having pixel coordinates which are not in the classification record, the subclass number is placed in PCTAB as a zero (=0).

The loop for checking the dots and initializing the performance table is controlled by two global variables determined in DSPLY2 - BCNT and ECNT. BCNT is the beginning dot number for dots having a line coordinate equal to the current classified scan line

number, and ECNT is the ending dot number for dots having a line coordinate equal to the current classified scan line number.

After FIELD (1,BCNT) through FIELD (1,ECNT) is processed (i.e., the dots BCNT through ECNT) a return is made to calling subprogram DSPLY2, with the classification performance for these dots tabulated and returned in the argument, PCTAB.

#### F.5.7 FLOWCHART

See Figure F-5 for flowchart of subprogram DOTPCT.

#### F.5.8 LISTING

The listing of subprogram DOTPCT is contained within the listing of subprogram DSPLY2, which is available in the Data Techniques Laboratory, JSC Building 17.

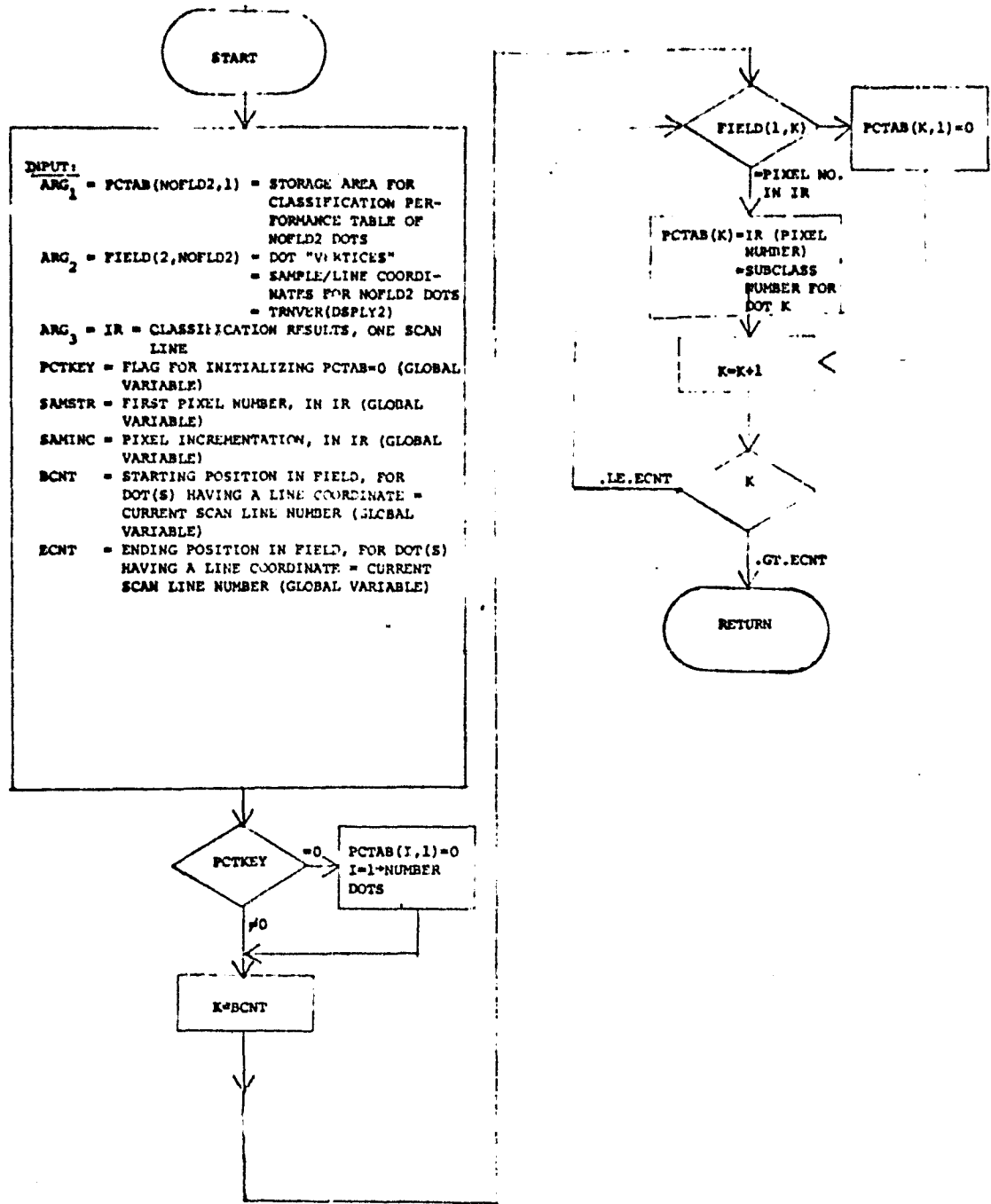


Figure F-5.- Flowchart for Subprogram DOTPCT

## F.6 SOFTWARE SUBPROGRAM NO. 6 (DOTSUM)

Subprogram DOTSUM is a new internal subprogram added to the DSPLY2 subprogram in the DISPLAY processor. The functions performed by the subprogram are as follows:

1. To compute analyst labeled category proportions over the classified area as follows:
  - a. Uncorrected proportions for the analyst-labeled categories
  - b. Bias corrected proportions for the analyst-labeled categories
2. To compute the "alpha" table
3. To output two classification performance tables as follows:
  - a. Summary by dot (analyst-labeled) category showing, for each analyst-labeled category:
    - 1.) Total number of dots labeled by analyst for the given category
    - 2.) Percent of correct classification of analyst-labeled dots (i.e., ratio of number of dots labeled  $C_I$ , classified  $C_I$  to the total number of dots labeled  $C_I$ )
    - 3.) Number of dots thresholded
    - 4.) Uncorrected category proportion
    - 5.) Bias corrected category proportion
    - 6.) Alpha values
    - 7.) Number of dots in the labeled category which were classified into each possible classification (MAPTAP) category.
  - b. Summary by individual dot showing:
    - 1.) Coordinate of dot (sample, line)

- 2.) Analyst-labeled category name
- 3.) Classifier's identification of the dot (category and subclass name)

### F.6.1 LINKAGES

Subprogram DOTSUM is internal to subprogram DSPLY2. The subprogram does not reference any other subprograms, and may not be referenced by other routines external to DSPLY2.

### F.6.2 INTERFACES

Certain variables from common block DISPL and DSPLY2 are utilized by subprogram DOTSUM, as global initializes. See the flowchart for DOTSUM (Figure F-6) for identification of these variables.

### F.6.3 INPUTS

Calling sequence: CALL DOTSUM (SUBTOT, THTOT, PCTAB, TRNSAV, TRNVER)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
SUBTOT	NOSUB3 (no. of subclasses +1)	In	Total no. of pixels classified into each of the subclasses, and thresholded total for all subclasses.
THTOT	NOSUB3	In	Total no. of pixels thresholded in each subclass, over the total area classified.
PCTAB	(NOFLD2,1) NOFLD2=total no. of dots	In	For DOT performance, PCTAB(I,1) = subclass for dot I

TRNSAV	(4,NOFLD2)	In	TRNSAV(1,N)=unused TRNSAV(2,N)=labeled category number TRNSAV(3,N)=dot type TRNSAV(4,N)=2 (no. of vertices)
TRNVER	(2,TOTVT2) TOTVT2=total no. of dots	In	Dot coordinates: TRNVER(1,I) = sample coordinate, TRNVER (2,I) = line coordi- nate of dot I

#### F.6.4 OUTPUTS

The dot data classification summaries are provided in place of the usual test or training field summaries. Two dot classification summaries are output:

- Dot category summary table
- Individual dot summary table

#### F.6.5 STORAGE REQUIREMENTS

The storage for DOTSUM is included in the total storage requirements of subroutine DSPLY2 (see Section F.4.5).

#### F.6.6 DESCRIPTION

The subprogram functions as follows:

1. Input via calling arguments and common block DISPL:
  - a. SUBTOT (NOSUB3) = total number of pixels classified into subclass 1,2,..., NOSUB2, THRESH

NOSUB2 = total number of subclasses  
NOSUB3 = total number of subclasses + 1

~~E-32~~  
2.25

- b. THTOT (NOSUB3) = total pixels thresholded, by subclass
- c. PCTAB (NOFLD2,4) = the subclass no. for each dot i, i = 1,2,...,NOFLD2 = total number of dots
- d. TRNSAV (4,NOFLD2):  
 TRNSAV(1,N) = unused  
 TRNSAV(2,N) = dot category number for the MAPTAP category matching the analyst-labeled category of the dot  
 TRNSAV(3,N) = dot type (=1 or =2)  
 TRNSAV(4,N) = 2 = number of "vertices" for dot
- e. From common block DISPL: NOCAT = number of classification categories; CATNAM(61) = classification category names; SUBCAT(60) = subclass-to-category correspondence (SUBCAT (I) = M, means subclass number I belongs in category number M); NOFLD2 = total number of dots; NDCAT = DOTKEY = number of dot categories, from DOTFILE.
2. Store dot type in TYPE (=TRNSAV(3,1)), initialize NDCAT = DOTKEY = number of dot categories and initialize NCLAT = NOCAT = number of classification categories, NCLCAT = NOCAT +1.
3. Move labeled category from TRNSAV(2,I) to TRNSAV(3,I), for I = 1, NOFLD2 (=total number of dots)
4. Transfer each dot "vertex" (sample, line coordinate) from TRNVER array to TRNSAV (1,I), TRNSAV (2,I) for I = 1, NOFLD2 dots [TRNSAV(1,I) = dot sample, TRNSAV(2,I) = dot line number, TRNSAV(3,I) = labeled category number, I = 1, NOFLD2 = total number of dots].
5. Compute total number of pixels classified in the total area classified, except for thresholded pixels and pixels in the "designated" area:

$$\text{PIXTOT} = \sum_{i=1}^{\text{NOSUB2}} \text{SUBTOT}(i)$$



6. Compute the total number of pixels classified into each analyst-labeled category:

for  $I = 1, 2, \dots, \text{NOSUB2}$  (total number of subclasses),  
 $\text{LBLTOT}(K) = \text{LBLTOT}(K) + \text{SUBTOT}(I)$ ,  
where  $K = \text{SUBCAT}(I) = \text{category}$   
corresponding to subclass  $I$ .

7. Compute dot totals classified, by classification category:  
Initialize  $\text{CLTOT}(K) = 0$ ,  $K = 1, 2, \dots, \text{NCLAT}(=\text{NOCAT})$ . Then,  
for  $I = 1, \text{NOFLD2}$ ,  $\text{SUBCL} = (\text{PCTAB}(I,1))$ ,  $1 \leq \text{SUBCL} \leq \text{NOSUB2}$ ,  
 $\text{CAT} = \text{SUBCAT}(\text{SUBCL})$ :  $\text{CLTOT}(\text{CAT}) = \text{CLTOT}(\text{CAT}) + 1$

8. Based on occurrence in the dot data array,  $\text{TRNSAV}$ , initialize an ordered set of dot category labels  $\text{DOTLBL}(J,1) = \text{CATNAM}(\text{LBL}) = \text{category name}$  and  $\text{DOTLBL}(J,2) = \text{LBL} = \text{category number of labeled dot}$  for  $J = 1, 2, \dots, \text{NDCAT}$ , and  $\text{LBL} = \text{TRNSAV}(3,I)$ ,  $I = 1, 2, \dots$ , no. dots, such that  $\text{DOTLBL}(J+1,2) > \text{DOTLBL}(J,2)$ .

9. Compute the uncorrected proportions of dot categories over the total area classified;  $\text{PUNC}(I) = \text{LBLTOT}(J) / \text{PIXTOT} * 100$ ;  $I = 1, \text{NDCAT}$ ;  $J = \text{DOTLBL}(I,2)$ .

10. Initialize  $\text{TRNSAV}(4,I)$ ,  $I = 1, \text{NOFLD2}$  (number of dots) to contain the classification category (number) for dot  $I$ :  
 $\text{SUBCL} = \text{PCTAB}(I,1)$ ,  $I = 1, 2, \dots, \text{NOFLD2}$ ; if  $\text{SUBCL} = \text{NOSUB3}$ ,  
 $\text{CAT} = \text{NOCLCAT} (= \text{NOCAT} + 1)$ ; if  $\text{SUBCL} \neq \text{NOSUB3}$ ,  $\text{CAT} = \text{SUBCAT}(\text{SUBCL})$ ;  $\text{TRNSAV}(4,I) = \text{CAT}$  (classified category number).

11. Initialize  $\text{LABCLS}(I,J) = 0$ ,  $I = 1, 2, \dots, \text{NODCAT}$ ;  $J = 1, 2, \dots, \text{NCLCAT} (= \text{NOCAT} + 1)$

12. Form a table showing (a) the labeled dot category, and (b) the total number of dots with a given label that have been classified into each classification category:

Initialize  $\text{LABCLS}(I,J) = 0$ ,  $I = 1, \text{NDCAT}$ ,  $J = 1, \text{NCLCAT}$ ; then  
for  $J = 1, \text{NOFLD2}$ ,  $\text{LB} = \text{TRNSAV}(3,J)$ ,  $\text{CLS} = \text{TRNSAV}(4,J)$ :  
 $\text{LABCLS}(\text{LB}, \text{CLS}) = \text{LABCLS}(\text{LB}, \text{CLS}) + 1$ .

13. Compute alpha table for labeled versus classified dots;  
 $\alpha(I,J) = \text{LABCLS}(J,I)/\text{CLTOT}(I)*100$ ,  $I = 1, \text{NOCAT}$  classification categories,  $J = 1, \text{NDCAT}$  labeled categories; set  $\alpha(I,J) = 0$ , where  $I > \text{NDCAT}$ .
14. Compute corrected proportions for labeled dot categories:  
 for  $I = 1, \text{NDCAT}$  (no. of dot categories),
- $$\text{PCORR}(I) = \sum_{J=1}^{\text{NDCAT}} [\alpha(J,I) * \text{PUNC}(J)] * 100$$
15. Tabulate the total number of analyst-labeled dots in each dot category:  
 initialize  $\text{TOTLBL} = 0$ ; then for  $I = 1, \text{NOFLD2}$  = total number of dots and  $\text{LBL} = \text{TRNSAV}(3,I)$  and for  $J$  where  $\text{DOTLBL}(J,2) = \text{LBL}$ ;  $\text{TOTLBL}(J) = \text{TOTLBL}(J) + 1$ .
16. Tabulate the number of dots in each category which were thresholded; initialize  $\text{TOTTHR}(J) = 0$ ,  $I = 1, \text{NDCAT}$ ; then for  $I = 1, \text{NDCAT}$  = no. of dot categories,  $\text{TOTTHR}(I) = \text{LABCLC}(I, \text{NCLCAT})$
17. Print first dot classification performance summary, by category: for  $I = 1, \text{NDCAT}$ ,
- labeled dot category name =  $\text{DOTLBL}(I,1)$
  - total number of analyst-labeled dots in each category =  $\text{TOTLBL}(I)$
  - percent correct classification =  $\text{LABCLC}(I,I)/\text{TOTLBL}(I)*100$ .
  - number of dots thresholded in the analyst-labeled category =  $\text{TOTTHR}(I)$
  - uncorrected labeled category proportion =  $\text{PUNC}(I)$
  - bias-corrected labeled category proportion =  $\text{PCORR}(I)$
  - classification summary and  $\alpha_{JI}$  for labeled category  $I$ :  
 for  $J = 1, \text{NCLCAT}$ ;  $\text{LABCLS}(I,J), \text{ALPHA}(J,I)$

18. Set SUBNAM(NOSUB3) = 'THRESH'
19. Print second dot classification performance summary, by individual dot: for I = 1, NOFLD2 = total number of dots.
  - a. dot identification = (sample,line) = TRNSAV(1,I),  
TRNSAV(2,I)
  - b. labeled category name = CATNAM[TRNSAV(3,I)]
  - c. classified category name = CATNAM[TRNSAV(4,I)]
  - d. classified subclass = SUBNAM[PCTAB(I,1)]
  - e. If dot is thresholded, in designated area, or not in the area classified, substitute "\*\*\*\*\*" for both category and subclass names to be output for the dot.

See figure F-6 for flowchart of internal subprogram DOTSUM.

#### F.6.7 FLOWCHART

N/A

#### F.6.8 LISTING

The listing for DOTSUM is contained in the listing of subprogram DSPLY2, which is available in the Data Techniques Laboratory, JSC Building 17.

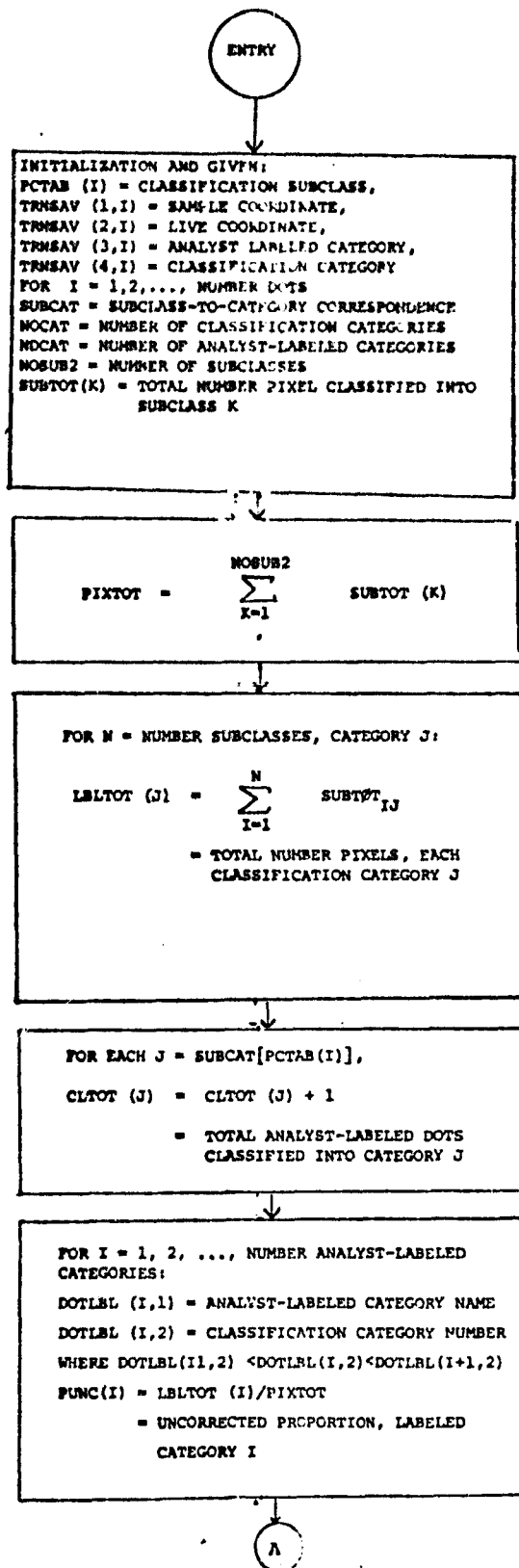


Figure F-6.- Flowchart for Subprogram DOTSUM

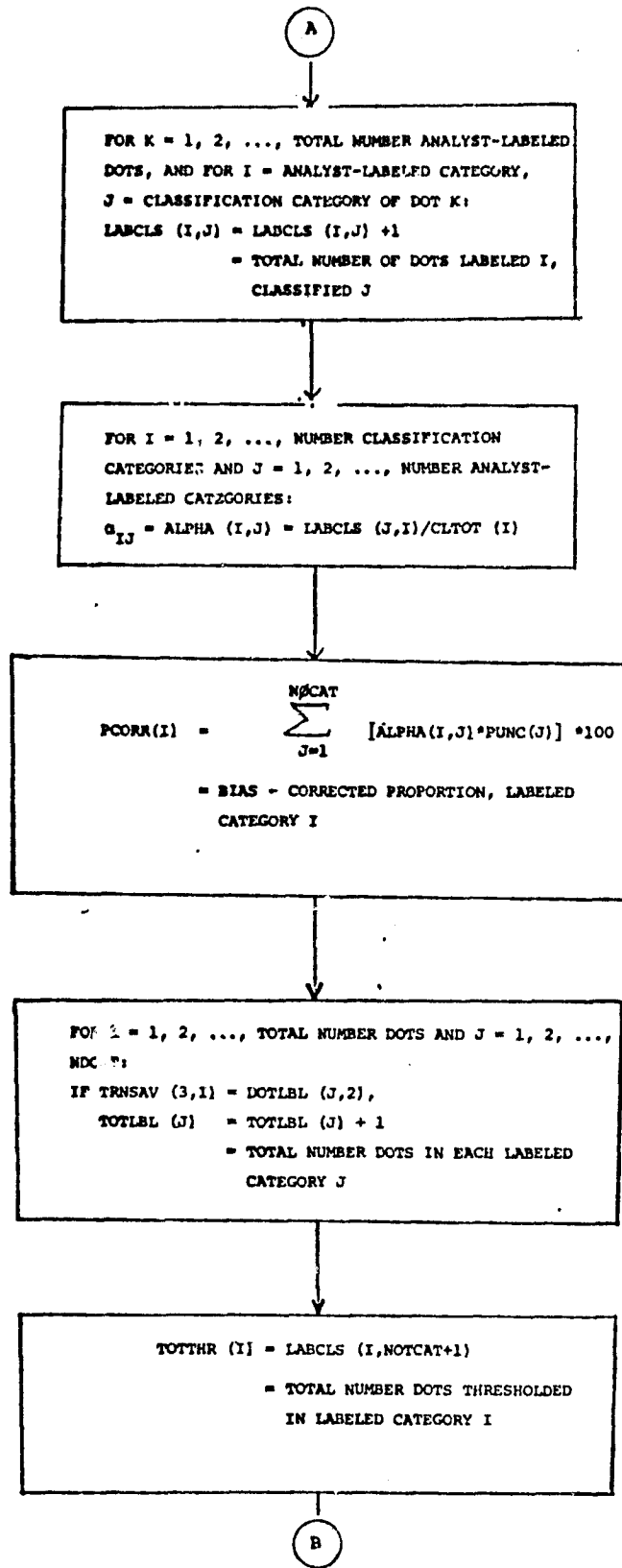


Figure F-6.- Continued

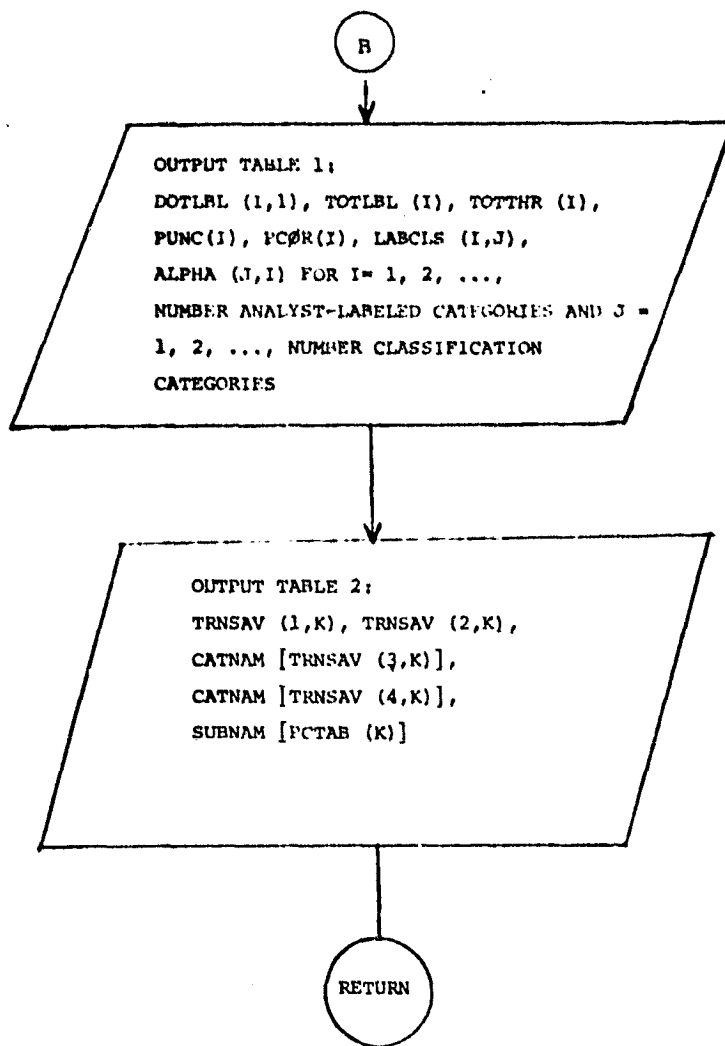


Figure F-6.-concluded

## F.7 SOFTWARE SUBPROGRAM NO. 7 (DSPLAY)

DSPLAY is the driver program for the DISPLAY processor. Minor modifications were required to incorporate LACIE Procedure 1 image data processing options. The primary impact of the modifications to DSPLAY are described in Section F.7.4 and F.7.6 below.

### F.7.1 LINKAGES

DSPLAY is called by the EOD-LARSYS executive program, MONTOR. DSPLAY calls SETUP3, DSPLY1, EMTHRS, DSPLY2, and PRTPCT.

### F.7.2 INTERFACES

The display program references the DISPL common block.

### F.7.3 INPUTS

Calling Sequence: CALL DSPLAY (ARRAY, TOP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
ARRAY	10,600	In	Variably dimensioned working array
TOP	1	In	Max. usable location in array. SET=10,600 in the EOD-LARSYS executive program, MONTOR.

### F.7.4 OUTPUTS

When not in the dot (LACIE Procedure 1) processing mode, a call is made to EMTHRS from DSPLAY, when test or training fields have been input to the DISPLAY processor. EMTHRS calculates the empirical thresholds, using data from defined test or training field storage areas in the large working array, ARRAY. The

empirical thresholds may be printed out (plotted) if the proper option is input to the DISPLAY processor.

When in the dot processing mode, however, the empirical threshold computations are not available to the user, due to the utilization of the training field storage areas for dot data storage, and due also to the fact that test fields are not allowed to be input when in the dot processing mode (designated fields may be input, however).

This constitutes a revision to the output normally available via DISPLAY. When not in the dot processing mode (no DOTFILE control card input), the usual empirical threshold options are available to the user.

Also, when not in the dot processing mode, a call to P RTPCT is made from DISPLAY. This results in classification performance summaries by subclass, class, and category for either test fields or training fields.

When in the dot processing mode, the test or training field performance summaries are not available to the user, due to the usage of the test and training field storage arrays described above.

The test or training field performance summaries are replaced by dot classification performance summaries, when in the dot processing mode in the DISPLAY processor.

#### F.7.5 STORAGE REQUIREMENTS

DISPLAY requires 224 locations in the Instruction Bank and 40 locations in the Data Bank for a total of 264 storage locations.



#### F.7.6 DESCRIPTION

DSPLAY is modified to bypass the call to the empirical threshold computation routine, EMTHRS, when in the dot processing mode (DOTKEY > 0 or DOTERR > 0), and to bypass the call to the test/training field classification performance output routine, PRTPCT, when in the dot processing mode. The modifications were made by inserting tests on the dot processing flags, DOTKEY and DOTERR, at the two appropriate places in DSPLAY.

This modification is made necessary because the dot data utilizes the usual training field storage areas, and test fields are not allowed to be input to the DISPLAY processor when dot performance summaries are requested via the input DOTFILE control card.

#### F.7.7 FLOWCHART

N/A

#### F.7.8 LISTING

The listing is available in the Data Techniques Laboratory, JSC Building 17.

## F.8 SOFTWARE SUBPROGRAM NO. 8 TAPLAB

TAPLAB is an assembly language (SLEUTHII) program for the Univac 1108/EXEC 2, which has a Fortran V calling sequence so that it may be called from a Fortran V program. The purpose of TAPLAB is to provide the operational label (i.e. tape reel number, or other label used on an "ASG" card in the equation "UNIT=LABEL") which is assigned to a logical tape unit (A,B,C,...).

### F.8.1 LINKAGES

Subprogram TAPLAB presumes the availability of the UNIVAC EXEC 2 system procedure (routine), TLABL\$.

### F.8.2 INTERFACES

Subprogram TAPLAB interfaces with the Fortran V calling program via two calling arguments. Subprogram TAPLAB references the UNIVAC/EXEC 2 system procedure, TLABL\$.

### F.8.3 INPUTS

Calling Sequence: CALL TAPLAB (UNIT, LABEL)

<u>Parameter</u>	<u>Dimension</u>	<u>Type</u>	<u>In/Out</u>	<u>Definition</u>
UNIT	1	Integer	In	The Fortran unit no. - legitimate numbers are (positive) 1-4,7-16, and 18-29. Fortran units 5,6 are system input, output units, respectively. Fortran unit 17 is reserved as system plot unit. Fortran unit 30 is the "reread" unit.

LABEL	1	Alphanu- meric or INTEGER 0's	Out	The operational label assigned to UNIT. If UNIT=5,6, 17, or 30 LABEL is set = XXXXXX and returned. If UNIT is unassigned as a tape unit, LABEL is returned all integer zero's in the six character positions in the word. If UNIT=1-4, 7-16, or 18-29 and the corresponding logical unit (A-D, E-N, O-Z) was assigned a tape unit by the ASG UNIT=LABEL card, LABEL = input tape label is returned as a 6-character alphanumeric.
-------	---	-------------------------------------	-----	---

#### F.8.4 OUTPUT

Subprogram TAPLAB provides to the calling program the tape label associated with (assigned to) the input (via first argument) Fortran unit number. The tape label is output via the second calling argument. The output tape label is either a six-character alphanumeric if the Fortran unit has been assigned a label, or all X's if the Fortran unit is unassigned, or all zero's if the Fortran unit is either 5,6,17, or 30.

### F.8.5 STORAGE REQUIREMENTS

Subprogram TAPLAB requires 68 storage locations.

### F.8.6 DESCRIPTION

TAPLAB tests the number in ARG1 for within range 1 to 4. If YES, the unit number is converted to the appropriate logical unit character A - D , and a jump made to the call to the UNIVAC System Procedure, TLABL\$. If NO, the number is tested for within range 7-16. If YES, the number is converted to the appropriate logical unit character E'- N and a jump made to the call to TLABL\$. If NO, the number is tested for within range 18-30. If YES, the appropriate conversion to logical unit O'-Z is made, and a jump to the call to TLABL\$. If the unit is none of the number 1-4, 7-16, or 18-30, a jump is made directly to ERUNIT where a label = XXXXXX is placed in ARG2 and a return made to the calling program.

If the unit is a legitimate number within the ranges described above, but was not assigned to a tape unit in the run, by means of the UNIVAC ASG UNIT=LABEL control card, the return from TLABL\$ is all zero's, and this is the label placed in ARG2 to return to the calling program.

A test is made on the first character of the label returned to TAPLAB from TLABL\$. If the first character position contains a zero, a blank character is inserted in the first character position of the label returned via ARG2. This is for the case of a 5-character tape label having been assigned to the Fortran unit.

TAPLAB utilizes registers A0, A7, A8, A9, and X11.

### F.8.7 FLOWCHART

N/A

F.8.8 LISTING

The listing is available in the EOD-LARSYS listing in the Data Techniques Laboratory, JSC Building 17.

APPENDIX G  
SUBROUTINE DESCRIPTIONS

APPENDIX G

TABLE OF CONTENTS

- G.1 Software for Routine No. 1 (TAPERD)
- G.2 Software for Routine No. 2 (RDDOTS)
- G.3 Software for Routine No. 3 (WRTDOT)

## G.1 SOFTWARE SUBPROGRAM NO. 1 (TAPERD)

The new capabilities of routine TAPERD are:

1. on option, unpack sun angles from the header record of the MSS Universal format input tape.
2. unpack the start and stop pixel number, pixel skip number, line skip number from the header record of the MSS Universal format input tape.

### G.1.1 LINKAGES

The routine TAPERD is called by the routines: DOTS, STOMAP, RDDATA, CLSFY2, LEARN, NDHST!, LNTRAN, HSTGRM, HISTGM. The corresponding processors are DOTDAT, LABEL, ISOCLS, CLASSFY, STAT, NDHIST, DATATR, HIST, and GRAYMP.

### G.1.2 INTERFACES

The subprogram TAPERD interfaces with other routines through common block ISOLNK.

### G.1.3 INPUTS

TAPERD has 3 entry points: TAPEDR, FLDINT and LINERD

- Calling sequence: CALL TAPHDR (DATAPE, DATFIL)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DATAPE	1	In	Tape unit number.
DATFIL	1	In	Number of EOF"s to be read over in positioning tape, minus one.

- Calling sequence: CALL FLDINT (FLDINF, FETVEC, NOFEAT)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
FLDINF	6	In	Rectangular field description for use



<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
FLDINF (Continued)			in unpacking pixels from MSS tape.
FETVEC	30	In	Contains channels requested.
NOFEAT	1	In	Number of channels in FETVEC.

● Calling sequence: CALL LINERD (IDATA, ENDTAP)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
IDATA	Depends upon buffer size in calling routine	Out	Array containing unpacked data.
ENDTAP	1	Out	EOF trigger: returns value of -1 if an EOF is encountered

#### G.1.4 OUTPUTS

N/A

#### G.1.5 STORAGE REQUIREMENTS

Code -  $16416_8$

Data -  $2420_8$

#### G.1.6 DESCRIPTION

The information to be unpacked from the Universal format MSS tape occupies the following byte positions on the tape header record:

<u>Byte Numbers</u>	<u>Description</u>
109-110	pixel start number

	<u>Byte Numbers</u>	<u>Description</u>
	110-111	pixel stop number
	1789-1790	pixel skip factor
	1791-1792	line skip factor
	2201-2202	sun angle for pass 1 (channels 1-4)
Tape	2203-2304	sun angle for pass 2 (channels 5-8) etc. (up to 8 passes)
Sun Angles	etc.	

This information is unpacked by simply extending the TAPHDR routine data vectors HWRD, BIT and NB, expanding the dimension of the ID vector, and expanding an existing unpacking code block. The information listed above will be placed in labeled common ISOLNK.

Please refer to the document "Universal Format Read Routine for UNIVAC 1108 Programs", Technical Memorandum, Project Number 1651V, August, 1973 for a full description of utility routine TAPERD.

#### G.1.7 FLOW CHART

N/A

#### G.1.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

## G. 2 SOFTWARE SUBPROGRAM NO. 2 (RDDOTS)

Routine RDDOTS reads information from the dot data tape.

### G.2.1 LINKAGES

Routine RDDOTS is called by routines FILERD, ISOCLS, and SETUP3. The corresponding processors are LABEL, ISOCLS, and DISPLAY. RDDOTS calls a file positioning routine FSBFL.

### G.2.2 INTERFACES

Interfaces with other routines through the common block GLOBAL. From this block is obtained the Fortran dot data tape unit number DOTUNT and the file number DOTFIL. The default unit number is 8 (UNIVAC number F).

### G.2.3 INPUTS

Calling sequence: RDDOTS (DOTS, DOTVEC, TOTDT3, TYPST, SIZES, TOTDT2, NOCAT, CATNAM, NOFET2, FETVC2, DOFEAT, FETVEC, NOSUN, ANGLE, NOFLD, TOTVRT, FLDSAV, VERTEX, KVAR)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DOTS	(SIZES, TOTDT2)	Out	Dot information, spectral, spatial, or both depending upon option value of TYPST
DOTVEC	TOTDT3	In	Depends upon value of TYPST: 1 - list of dots for which spectral information is wanted 2 - ignored, except for a check if TOTDT3 $\neq 0$ .

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
DOTVEC (cont)			3 - list of dots to be excluded, all other dots on the DOTFIL are passed back to calling program
TOTDT3	1	In	Depends upon value of TYPSTW: 1 - number of dots requested 2 - should be preset to 0 3 - number of dots to be excluded
TYPSTW	1	In	Option switch, must be preset to a value of 1,2, or 3 1 - spectral information requested 2 - spatial information requested (sample, line, type, and category numbers) 3 - both spatial and spectral information requested
SIZES	1	Out	Depends upon value of TYPSTW: 1 - set to NOFET2 2 - set to 4 3 - set to 4 + NOFET2
TOTDT2	1	Out	Depends upon value of TYPSTW: 1 - set to TOTDT3 2 - set to TOTDOT

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
TOTDT2 (continued)	1	Out	(number of dots on DOTFIL) 3 - set to TOTDOT-TOTDT3
NOCAT	1	Out	Number of categories, value extracted from DOTFIL
CATNAM	NOCAT	Out	Category names, values extracted from DOTFIL
NOFET2	1	In/out	Number of channels for which information is requested. If set to 0, reset to NOFEAT. If TYPSTW = 2, the value of NOFET2 is ignored.
FETVC2	NOFET2	In	Channel numbers for which information is requested, ignored if TYPSTW=2.
NOSUN	1	Out	Number of sun angles on DOTFIL
ANGLE	NOSUN	Out	Sun angle values on DOTFIL
NOFLD	1	Out	Number of fields on DOTFIL
TOTVRT	1	Out	Number of vertices on DOTFIL
FLDSAV	(4,NOFLD)	Out	Field information on DOTFIL

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
VERTEX	(2, TOTVRT)	Out	Vertex information on DOTFIL
KVAR	(SIZES, TOTDT2)	Out	Set if TYP SWT = 1, same information as DOTS, but in floating point.

#### G.2.4 OUTPUT

N/A

#### G.2.5 STORAGE REQUIREMENTS

Code - 762<sub>8</sub>      Data - 11767<sub>8</sub>

#### G.2.6 DESCRIPTION

The dot data tape is read and appropriate information is extracted. There are three modes of extraction:

1. dot spectral information
2. dot spatial information
3. both spectral and spatial information.

In all cases, the count information (first record) is made available.

Each file on the dot data tape consists of three records. Please refer to the flow chart provided in Figure G.2 for details concerning contents of these records, also the file description in Appendix H.

Dot spectral information consists of radiance values for those

dots and channels specified. Dot spatial information consists of :

1. sample number
2. line number
3. type number (1, for label/starting dots, 2, for bias dots)
4. category number

for those dots specified by the user.

#### G.2.7 FLOW CHART

See Figure G-1 for flow chart of the RDDOTS routine.

#### G. 2.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

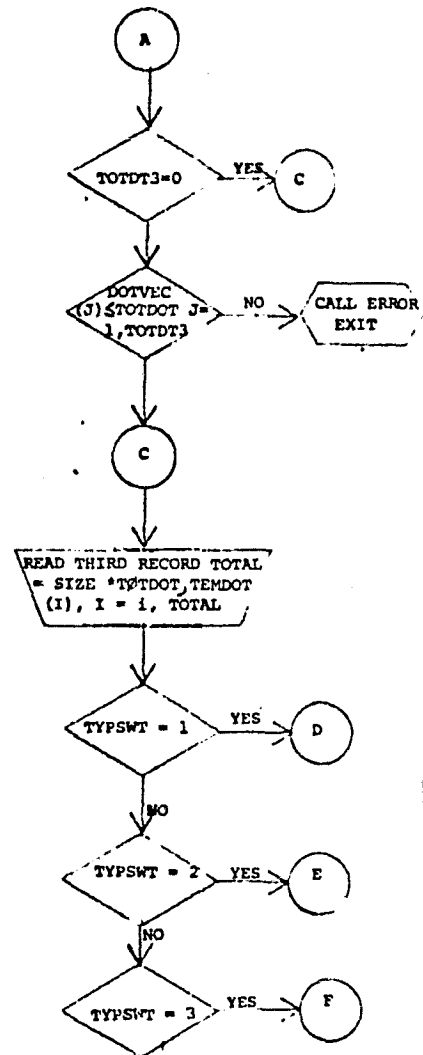
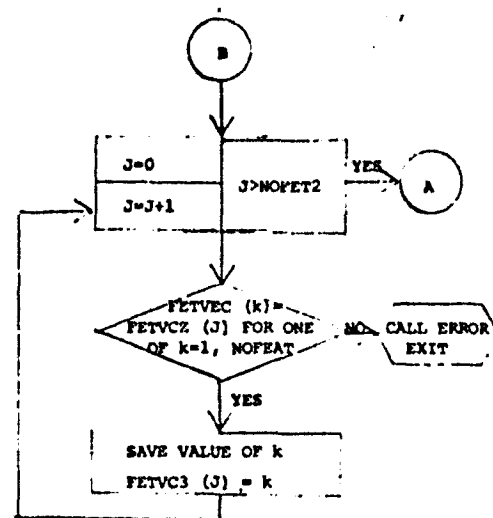
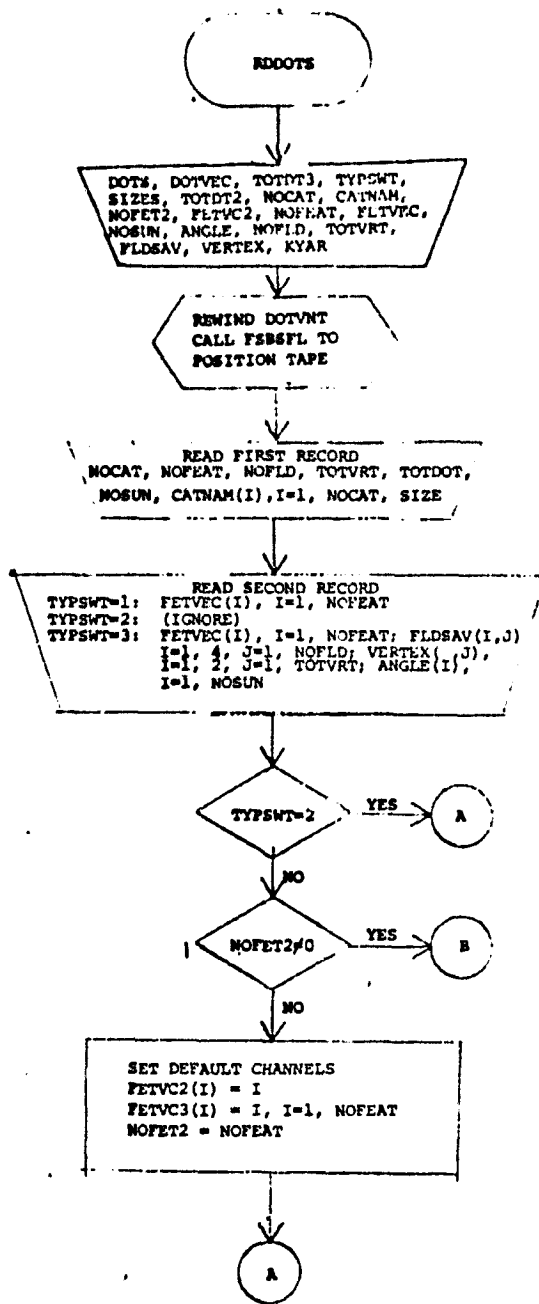


Figure G-1.- Flowchart for Subprograms RDDOTS

G-9

ORIGINAL PAGE IS  
OF POOR QUALITY



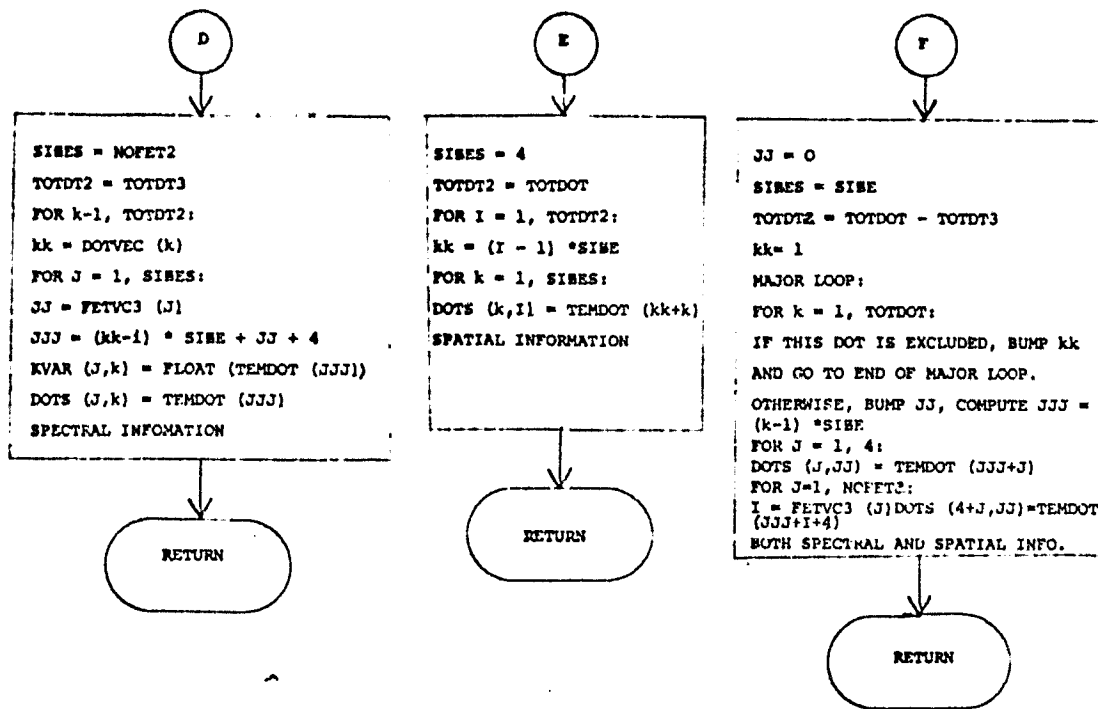


Figure G-1.- Continued

### G.3 SOFTWARE SUBPROGRAM NO. 3 (WRTDOT)

Subroutine WRTDOT writes the dot data tape. One file is written per call to WRTDOT.

#### G.3.1 LINKAGES

Routine WRTDOT is called by routine DOTS in the DOTDATA processor. WRTDOT calls the UNIVAC file positioning routine FSBSFL.

#### G.3.2 INTERFACES

No common blocks are provided for WRTDOT.

#### G.3.3 INPUTS

Calling sequence: WRTDOT (TOTDOT, NOSUN, FLDSAV, VERTEX, ANGLE, DOTS, NOCAT, CATNAM, SIZE, NOFET2, TOTVT2, NOFLD2, UNIT, FILE)

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Definition</u>
TOTDOT	1	In	Total number of dots to be written on DOTFIL
NOSUN	1	In	Number of sun angles to be written.
FLDSAV	(4,NOFLD2)	In	Field information to be written
VERTEX	(2, TOTVT2)	In	Vertices to be written
ANGLE	NOSUN	In	Sun angles to be written
DOTS	(SIZE, TOTDOT)	In	Dot information to be written
NOCAT	1	In	Number of categories
CATNAM	NOCAT	In	Category names
SIZE	1	In	4 + NOFET2
NOFET2	1	In	Number of features (channels) to be written on DOTFIL

<u>Parameter</u>	<u>Dimension</u>	<u>In/Out</u>	<u>Description</u>
FETVC2	NOFET2	In	Channel numbers
TOTVT2	1	In	Total number of vertices to be written
NOFLD2	1	In	Total number of fields to be written
UNIT	1	In	Unit number (Fortran) of DOTFIL
FILE	1	In	File number of DOTFIL for this file

#### G.3.4 OUTPUT

An unformatted binary tape is written, with three records per file.

#### G.3.5 STORAGE REQUIREMENTS

Code - 562<sub>8</sub>      Data 57<sub>8</sub>

#### G.3.6 DESCRIPTION

The contents of the output tape files are described in Appendix H. All the information needed to write a file is brought in through the calling sequence to subprogram WRTDOT.

#### G.3.7 FLOW CHART

See Figure G-2 for flow chart of WRTDOT.

#### G.3.8 LISTING

Available in the Data Techniques Laboratory, JSC Building 17.

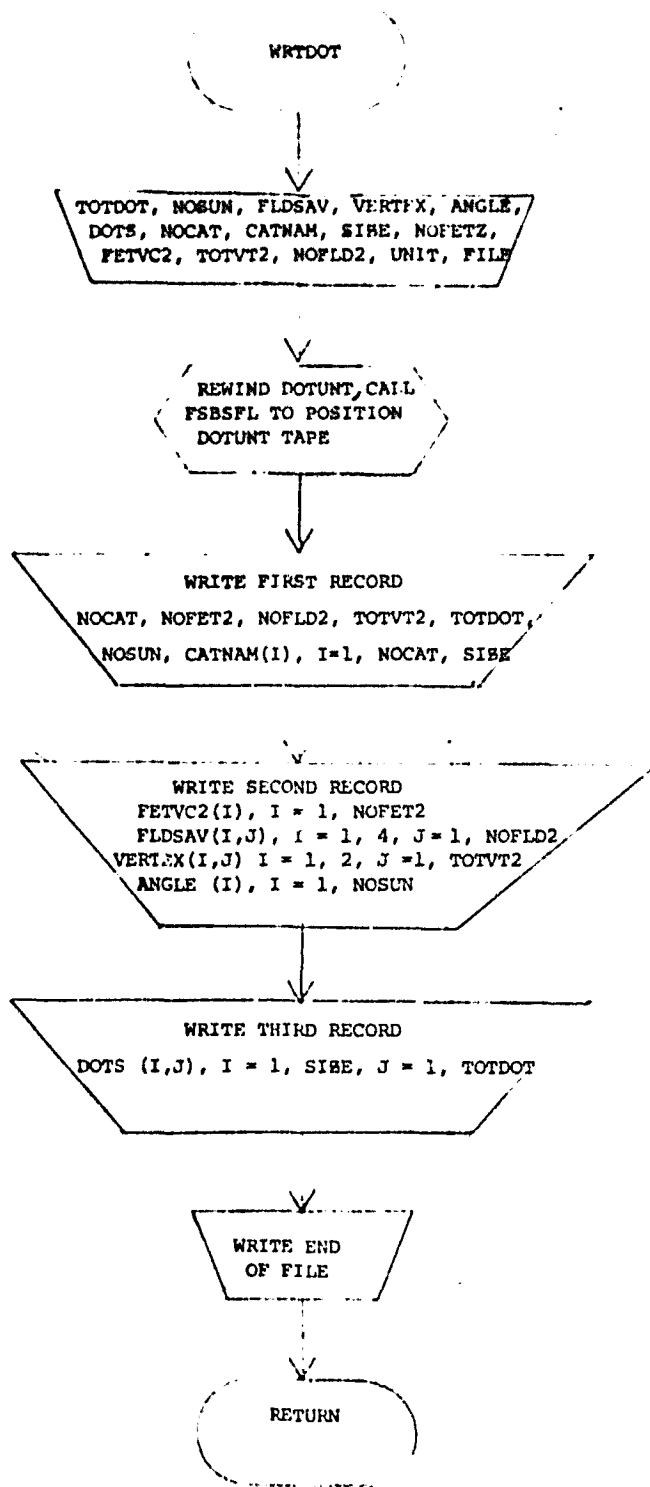


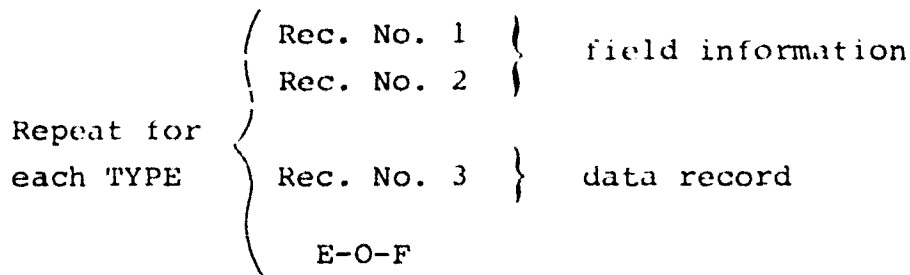
Figure G-2.- Flowchart of Subprogram WRTDOT

APPENDIX H  
DOT DATA FILE FORMAT

DOT DATA FILE FORMAT  
(DOTFIL)

The tape, DOTFIL is output by the DOTDATA processor. The records are written with an unformatted FORTRAN write.

A file is output for each TYPE of field(s). The file consists of the following records:



TYPE 1 consists of labeling dots; TYPE 2 consists of bias correction dots.

Rec. No. 1

WRITE (DOTUNT) NOCAT, NOFEAT, NOFLD, TOTVRT, TOTDOT, NOSUN,  
(CATNAM(I), I=1, NOCAT), SIZE

<u>Parameter</u>	<u>Dimension</u>	<u>Definition</u>
NOCAT	1	Number of category names.
NOFEAT	1	Number of channels.
NOFLD	1	Number of fields.
TOTVRT	1	Number of vertices.
TOTDOT	1	Number of dots
NOSUN	1	Number of sun angles.
CATNAM	NOCAT	Array containing the category names.
SIZE	1	4 + NOFEAT

Rec. No. 2

WRITE (DOTUNT) (FETVEC(I), I=1, NOFEAT)  
((FLDSAV(I,J), I=1,4), J=1, NOFLD),  
((VERTEX(I,J), I=1,2), J=1, TOTVRT),  
(ANGLE(I), I=1, NOSUN)

<u>Parameter</u>	<u>Dimension</u>	<u>Definition</u>
FETVEC	NOFEAT	Array containing the channel numbers.
FLDSAV	(4,NOFLD)	Array containing the field description.
VERTEX	(2,NOFLD)	Array containing the field vertices.
ANGLE	NOSUN	Array containing the sun angles.

Rec. No. 3

WRITE (DOTUNT) ((DOTS(I,J), I=1, SIZE), J=1, TOTDOT)

<u>Parameter</u>	<u>Dimension</u>	<u>Definition</u>
DOTS	(TOTDOT, SIZE)	Array containing the dot information. DOTS(1,I)=sample number for dot i. DOTS(2,I)=line number for dot i. DOTS(3,I)=type number for dot i. DOTS(4,I)=category number for dots i (optional)

Parameter

Dimension

Definition

DOTS(5,I)

.

=dot

.

vector :

DOTS(4+NOFEAT,I)



APPENDIX I  
VERIFICATION RUN

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

SDOTUATA

INPUT SUMMARY

CHANNEL DATA=1,2,3,4,5,6,7,8  
OUTFIL OUTPUT/UNIT=8,FILE=1  
OPTION PRINT  
\*END\*

USER HAS REQUESTED THE FOLLOWING OPTIONS :

SELECTED CHANNELS ARE 1 2 3 4 5 6 7 8  
PRINT DATA VECTORS

PROCEDURE I UNPACK INFO 45 45 60 60 60 60 60 60 1 196 54273 18669

INPUT IMAGE DATA TAPE INFORMATION

FORMAT UNIVERSAL  
NO. OF CHANNELS 20  
NO. OF PIXELS/LINE 196  
FIRST SCAN LINE NO. 1  
FIRST PIXEL REFERENCE PT 1

INPUT TEST FIELDS

FIELD	CLASS	SUBCLASS	VERTICES (SAMPLE, LINE)
1	HEAT		( 10, 10) ( 51, 10)
2	NONHEAT		( 10, 100) ( 51, 100)

NO. SAMPLE LINE TYPE CATEGORY DATA

NO.	SAMPLE	LINE	TYPE	CATEGORY	CH( 1)	CH( 2)	CH( 3)	CH( 4)	CH( 5)	CH( 6)	CH( 7)	CH( 8)
1	10	10	1	1	19	18	19	8	35	33	43	24
2	20	10	1	1	25	24	24	8	30	28	30	14
3	30	10	1	1	21	15	27	16	39	43	46	22
4	40	10	1	1	26	26	27	13	36	34	46	24
5	50	10	1	1	25	24	27	11	36	35	52	24
6	10	100	1	2	22	21	21	10	28	23	43	25
7	20	100	1	2	33	35	39	18	47	54	62	31
8	30	100	1	2	26	26	27	12	32	31	42	21
9	40	100	1	1	22	21	24	12	32	34	34	19
10	50	100	1	1	27	26	32	15	30	28	32	15

H-2  
261

INPUT TEST FIELDS  
SUBCLASS

VERTICES (SAMPLE, LINE)  
( 10: 20) ( 100: 20)  
( 10: 90) ( 196: 90)

FIELD CLASS  
1 WHEAT  
2 NONWHT

NO.	SAMPLE	LINE	TYPE	CATEGORY	DATA															
					CH(1)	CH(2)	CH(3)	CH(4)	CH(5)	CH(6)	CH(7)	CH(8)	CH(9)	CH(10)	CH(11)	CH(12)	CH(13)	CH(14)	CH(15)	CH(16)
1.	10	20	2	1	26	24	24	37	20	20	13	36	35	51	28					
2.	20	20	2	1	22	18	26	13	36	37	40	20								
3.	30	20	2	1	18	14	12	6	30	21	55	32								
4.	40	20	2	1	22	17	17	8	30	27	28	13								
5.	50	20	2	1	22	18	19	9	31	26	51	28								
6.	60	20	2	1	73	21	19	9	34	32	42	24								
7.	70	20	2	1	20	16	21	11	34	32	40	20								
8.	80	20	2	1	20	14	22	13	42	45	47	23								
9.	90	20	2	1	18	16	16	9	26	21	51	31								
10.	100	20	2	1	22	17	22	11	37	40	38	19								
11.	10	90	2	2	17	12	13	5	33	30	43	21								
12.	20	90	2	2	22	22	22	11	34	34	55	26								
13.	30	90	2	2	27	26	29	13	32	30	46	24								
14.	40	90	2	2	26	26	26	12	36	34	57	29								
15.	50	90	2	2	22	20	19	9	36	32	57	29								
16.	60	90	2	2	27	30	27	14	32	26	57	32								
17.	70	90	2	2	22	20	22	11	32	25	50	27								
18.	80	90	2	2	26	25	34	18	37	36	55	27								
19.	90	90	2	2	31	32	36	17	32	26	55	29								

ORIGINAL PAGE IS  
OF POOR QUALITY

262

20.	100	90	2	2	23	19	22	11	33	34	48	24
21.	110	90	2	2	23	17	27	14	36	36	43	21
22.	120	90	2	2	27	26	29	15	36	30	53	26
23.	130	90	2	2	31	32	32	17	29	24	57	29
24.	140	90	2	2	26	23	32	18	37	43	44	23
25.	150	90	2	2	18	14	24	14	34	35	39	20
26.	160	90	2	2	18	15	19	9	33	28	41	21
27.	170	90	2	2	37	30	36	18	36	37	42	20
28.	180	90	2	2	31	33	42	22	37	40	49	24
29.	190	90	2	2	30	26	44	20	39	42	45	23

TIME FOR OUTDATA 466 MINUTES

\$ISOCLS

INPUT SUMMARY

CHANNEL DATA=1,2,3,4,5,6,7,8  
LSTOP 8  
OPTION CLUSTER  
PCKMAT U  
\*END\*

YOU HAVE SELECTED THE FOLLOWING PARAMETER VALUES AND OPTIONS

STOP AFTER 8 ITERATION(S)  
ALLOW A MINIMUM OF 38 PIXELS PER CLUSTER  
PRINT A CLUSTER SUMMARY EVERY 20 ITERATION(S)  
PRINT A CLUSTER MAP EVERY 20 ITERATION(S)  
ALLOW A MAXIMUM OF 60 CLUSTERS PER CLASS  
THE STATISTICS FILE WILL BE WRITTEN AFTER 1 CLASS(ES) HAVE BEEN CLUSTERED  
CHANNELS ARE-- 1 2 3 4 5 6 7 8  
DUMIN = 3.200  
STUMAX = 4.500  
PERCENT = 1 80  
NSCOIS = 1 0  
NO. SUN ANGLES FROM CARDS = 0  
NO. SUN ANGLE TAPE SH = 0

WRITE A CLUSTER MAP OUTPUT TAPE IN UNIVERSAL FORMAT

INPUT IMAGE DATA TAPE INFORMATION

FORMAT UNIVERSAL  
NO. OF CHANNELS 20  
NO. OF PIXELS/LINE 196  
FIRST SCAN LINE NO. 1  
FIRST PIXEL REFERENCE PT 1

DESIGNATED OTHER OR UNIDENTIFIABLE FIELDS INPUT

DESIGN OTHER  
(1,1),(50,51),(61,52),(75,55),(45,54)



Handwritten alphanumeric characters, possibly a code or cipher, arranged in a grid-like pattern. The characters are mostly lowercase letters and numbers, with some symbols like asterisks and slashes. The text is oriented vertically on the page.

Handwritten numbers '17' and '26' at the bottom of the page.



POINTS PER CLUSTER IN THIS FIELD  
 CLUSTER SYMBOL POINTS

1	1	380
2	2	670
3	3	2186
4	4	1622
5	5	1074
6	6	3238
7	7	2229
8	8	533
9	9	379
10	A	1023
11	B	1659
12	C	1287
13	D	1714
14	E	1516
15	F	912
16	G	557
17	H	1794
18	I	618
19	J	479
20	*	62

867

THE STATISTICS FILE FOR 1 CLASSES AND 19 SUBCLASSES HAS BEEN WRITTEN  
 THE STATS WERE WRITTEN ON FILE 1  
 THE STATS FOR A PARTICULAR CLASS OR SUBCLASS SHOULD BE REFERRED TO IN LATER RUNS BY  
 THE FOLLOWING NAMES AND NUMBERS (WHICHEVER APPLICABLE)

CLASS 1 WHEAT SUBCLASSES (TOTAL= 19)

- 1 WHEG01
- 2 WHEG02
- 3 WHEG03
- 4 WHEG04
- 5 WHEG05
- 6 WHEG06
- 7 WHEG07
- 8 WHEG08
- 9 WHEG09
- 10 WHEG10
- 11 WHEG11
- 12 WHEG12
- 13 WHEG13
- 14 WHEG14
- 15 WHEG15
- 16 WHEG16
- 17 WHEG17
- 18 WHEG18
- 19 WHEG19

TIME FOR ISOCLS 4.923

268

SLABEL

INPUT SUMMARY

PROCED ALL  
DISTAN L2  
NEARES #  
OUTFIL INPUT/UNIT=0,FILE=1  
STATFI INPUT/UNIT=1,FILE=1  
MAPFIL OUTPUT/UNIT=1,FILE=2  
MAPFIL INPUT/UNIT=16,FILE=1  
MAPFIL OUTPUT/UNIT=16,FILE=2  
OPTION COND  
OPTION MIX  
THRESH 20.0  
\*END\*

USER HAS REQUESTED THE FOLLOWING OPTIONS :

CLUSTER/CLASSIFICATION TAPE IS BEING INPUT  
ALL-JF-A-KIND PROCEDURE WILL BE USED  
L2 DISTANCE WILL BE USED 20.000  
THRESHOLD DISTANCE #  
5-NEAREST DOTS WILL BE USED  
NO SUN ANGLE CORRECTION WILL BE APPLIED  
OUTFIL FILE IS BEING INPUT  
SAVTAP FILE IS BEING INPUT  
CONDITIONAL CLUSTER MAP WILL BE OUTPUT  
MIXED CLUSTER MAP WILL BE OUTPUT

I-10

INPUT IMAGE DATA TAPE INFORMATION

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

269

CLUSTER-DOT INTER-DISTANCE TABLE

DOTS	CLSTR (1)	CLSTR (2)	CLSTR (3)	CLSTR (4)	CLSTR (5)	CLSTR (6)	CLSTR (7)	CLSTR (8)	CLSTR (9)	CLSTR (10)	CLSTR (11)	CLSTR (12)	CLSTR (13)
1	40.75	35.02	15.07	8.93	24.51	13.86	6.95	19.17	37.83	29.58	15.28	7.46	27.76
2	44.66	35.69	18.23	18.72	37.46	17.05	20.89	19.33	39.48	29.75	18.32	14.87	35.80
3	37.00	34.53	6.48	15.95	12.42	22.95	10.73	22.59	33.37	21.32	14.30	19.90	18.34
4	27.00	22.29	14.01	17.98	19.89	15.12	12.70	15.09	23.60	19.04	8.23	18.24	16.95
5	27.72	29.87	12.76	19.79	17.38	13.59	13.53	29.60	24.76	15.27	4.81	20.37	16.25
6	13.20	34.39	25.07	19.72	33.27	10.47	17.88	49.25	35.84	17.05	18.95	19.31	13.14
7	13.00	34.39	32.30	42.87	24.30	12.92	40.30	45.25	34.94	17.48	31.35	19.80	17.86
8	38.09	33.30	18.66	8.87	25.89	16.18	14.09	9.27	26.51	17.48	14.11	15.35	22.84
9	38.25	33.38	19.40	11.77	28.39	22.00	12.09	8.41	35.85	13.74	20.76	19.89	28.48
10	38.09	28.26	18.65	18.99	35.53	27.85	23.41	5.91	31.78	25.87	26.59	21.00	31.64

DOTS	CLSTR (16)	CLSTR (17)	CLSTR (18)	CLSTR (19)
1	26.86	15.63	21.20	27.28
2	25.57	22.39	9.16	33.61
3	42.82	9.32	30.37	22.86
4	13.34	8.41	27.06	13.94
5	17.57	11.16	30.96	13.63
6	26.84	20.97	20.08	26.83
7	30.28	33.52	59.20	24.84
8	14.50	11.33	21.39	18.70
9	21.61	13.81	13.70	27.88
10	18.29	19.43	17.87	27.87

*Handwritten signature/initials*

13 JUN 77

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

LABELING BY ALL-OF-A-KIND PROCEDURE

LABEL :	NONMHT							
MEANS :	33.82	37.49	41.90	20.29	41.60	47.47	54.15	26.91
				DOTS		ANALYST LABEL		DISTANCE
				7.		NONMHT		13.06

TOTALS	
LABEL	NO.
WHEAT	0
NONMHT	1

ORIGINAL PAGE IS  
OF POOR QUALITY

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••  
LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT

MEANS : 32.93 35.45 39.47 19.37 30.42 25.57 51.57 29.49

DOTS	ANALYST LABEL	DISTANCE
4.	WHEAT	22.29
8.	NONWHT	23.30
5.	WHEAT	23.67
10.	WHEAT	26.26
6.	NONWHT	29.38

~~I-14~~

TOTALS	
LABEL	NU.
WHEAT	3
NONWHT	2

272

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

LABEL : WHEAT

MEANS : 21.86 16.80 24.21 12.61 39.57 44.18 47.72 23.68

DOTS ANALYST LABEL DISTANCE  
3. WHEAT 6.48

TOTALS  
LABEL NO.

WHEAT 1  
NONWHT 0

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••  
LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT  
MEANS : 22.29 19.18 21.42 12.55 32.31 30.09 44.17 23.42

DOTS	ANALYST LABEL	DISTANCE
8.	NONWHT	8.87
1.	WHEAT	8.93
6.	NONWHT	9.72
4.	WHEAT	9.98
5.	WHEAT	11.79

TOTALS

LABEL	NO.
WHEAT	3
NONWHT	2

115 274



LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

.. DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ..  
LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT

MEANS : 22.74 19.95 28.44 15.39 42.58 49.49 53.28 26.05

DOTS	ANALYST LABEL	DISTANCE
3.	WHEAT	12.42
5.	WHEAT	17.38
4.	WHEAT	19.89
7.	NONWHT	24.30
1.	WHEAT	24.51

TOTALS  
LABEL NO.  
WHEAT 4  
NONWHT 1

ORIGINAL PAGE IS  
OF POOR QUALITY

I-16

275

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

LABEL : NONWHT

MEANS : 22.55 20.36 21.27 10.40 30.85 26.22 51.46 29.32

0015 ANALYST LABEL DISTANCE  
6. NONWHT 10.47

TOTALS  
LABEL NU.

WHEAT 0  
NONWHT 1

1-1 276

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-UP-A-KIND PROCEDURE

LABEL : WHEAT

MEANS : 21.26 17.93 21.02 10.45 36.50 38.02 43.01 21.63

DOTS 1. ANALYST LABEL WHEAT DISTANCE 6.95

TOTALS  
LABEL NO.

WHEAT 1  
NONWHT 0

13 JUN 77

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

LABELING BY ALL-OF-A-KIND PROCEDURE

LABEL : WHEAT

MEANS : 26.06 25.23 28.10 13.82 31.54 30.60 34.02 16.91

DOTS	ANALYST LABEL	DISTANCE
2.	WHEAT	9.33
8.	NONWHT	9.27
10.	WHEAT	5.91

TOTALS  
LABEL NO.

WHEAT 2  
NONWHT 1

~~1-19~~

218

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••  
LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT

MEANS : 34.21 37.89 42.09 20.36 35.43 34.72 52.82 28.34

DOTS	ANALYST LABEL	DISTANCE
4.	WHEAT	23.60
5.	WHEAT	24.76
7.	NONWHT	24.94
8.	NONWHT	26.51
10.	WHEAT	31.78

TOTALS

LABEL NO.

WHEAT 3

NONWHT 2

L-20

279

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

LABEL : WHEAT

MEANS : 24.13 22.54 26.43 13.45 35.95 35.65 51.96 27.68

DOTS ANALYST LABEL DISTANCE

4: WHEAT 4.23  
5: WHEAT 4.81

TOTALS  
LABEL NUM

WHEAT 2  
NONWHT 0

~~I-21~~

082

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••

LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT

MEANS : 20.14 16.53 18.04 8.62 32.34 31.13 37.88 19.33

DOTS	ANALYST LABEL	DISTANCE
1:	WHEAT	7.96
9:	WHEAT	9.69
6:	NONWHT	13.31
2:	WHEAT	14.87
8:	NONWHT	15.35

TOTALS

LABEL	NU.
WHEAT	3
NONWHT	2

182  
~~22-1~~

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 73

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••  
LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

MEANS :	27.98	26.57	32.88	16.54	41.14	46.61	52.00	25.73
LABEL :	WHEAT							
DOTS	5	4	7	3	8			
ANALYST LABEL	WHEAT	WHEAT	NONWHT	WHEAT	NONWHT			
DISTANCE	16.25	13.95	17.68	16.36	25.84			

TOTALS

LABEL	NU.
WHEAT	3
NONWHT	2

*Handwritten:* 1-22 222



LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••  
LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL :	MEANS :	18.14	31.69	17.75	30.87	26.49	51.46	29.03	DOTS	ANALYST LABEL	DISTANCE
WHEAT	22.37								5:	WHEAT	15.27
									4:	WHEAT	16.04
									6:	NONWHT	17.05
									8:	NONWHT	17.48
									1:	WHEAT	20.58

TOTALS  
LABEL NO.  
WHEAT 3  
NONWHT 2

ORIGINAL PAGE IS  
OF POOR QUALITY

~~I-24~~  
283

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••

LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT

MEANS : 27.70 27.63 30.62 15.30 30.94 26.35 50.95 28.90

DOTS	ANALYST LABEL	DISTANCE
4.	WHEAT	12.51
5.	WHEAT	13.32
8.	NONWHT	13.96
6.	NONWHT	17.19
1.	WHEAT	22.46

TOTALS  
LABEL NO.

WHEAT 3  
NONWHT 2

1-25  
284

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

LABEL : WHEAT

MEANS : 21.15 17.09 25.75 13.85 31.88 31.24 34.16 17.17

DOTS ANALYST LABEL DISTANCE  
9. WHEAT 5.78

TOTALS  
LABEL NO.  
WHEAT 1  
NONWHT 0

~~1-26~~ 285

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••

LABELING BY S-NEAREST NEIGHBOR PROCEDURE

LABEL :	MEANS :	30.45	31.93	35.84	17.70	35.47	36.62	42.80	21.67	30TS	ANALYST LABEL	DISTANCE
WHEAT										4:	WHEAT	13.34
										8:	NONWHT	14.50
										5:	WHEAT	17.57
										10:	WHEAT	18.29
										9:	WHEAT	21.61

TOTALS  
LABEL NO.  
WHEAT 4  
NONWHT 1

286

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

•• DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ••  
LABELING BY S-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT  
MEANS : 23.84 21.56 29.47 15.63 36.62 38.75 43.59 21.86

DOTS	ANALYST LABEL	DISTANCE
4.	#WHEAT	8.41
3.	#WHEAT	9.32
5.	#WHEAT	11.16
8.	#NONWHT	11.33
9.	#WHEAT	13.81

TOTALS  
LABEL NO.  
#WHEAT 4  
#NONWHT 1

ORIGINAL PAGE IS  
OF POOR QUALITY

I-28 82-I 487

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

.. DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ..

LABELING BY 5-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT

MEANS : 21.40 17.91 19.40 9.04 28.75 26.16 27.52 13.18

DOTS	ANALYST LABEL	DISTANCE
2.	#WHEAT	9.16
9.	#WHEAT	13.70
10.	#WHEAT	17.87
6.	NONWHT	20.08
1.	#WHEAT	21.20

TOTALS

LABEL	NO.
WHEAT	4
NONWHT	1

I-29

285

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

LABELING BY ALL-OF-A-KIND PROCEDURE

.. DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE ..  
LABELING BY S-NEAREST NEIGHBOR PROCEDURE

LABEL : WHEAT  
MEANS : 28.89 29.64 35.17 17.85 35.72 35.17 53.66 28.73

DOTS	ANALYST LABEL	DISTANCE
5.	WHEAT	13.63
4.	WHEAT	13.94
8.	NONWHT	15.70
3.	WHEAT	22.66
7.	NONWHT	24.84

TOTALS  
LABEL NO.  
WHEAT 3  
NONWHT 2

I-30 289

THE STATISTICS FILE FOR 2 CLASSES AND 19 SUBCLASSES HAS BEEN WRITTEN  
 THE STATS FOR A PARTICULAR CLASS OR SUBCLASS SHOULD BE REFERRED TO IN LATER RUNS BY  
 THE FOLLOWING NAMES AND NUMBERS (\* HICHEVER APPLICABLE)

CLASS 1 WHEAT SUBCLASSES (TOTAL= 17)

- 1 WHE001
- 2 WHE002
- 3 WHE003
- 4 WHE004
- 5 WHE005
- 6 WHE006
- 7 WHE007
- 8 WHE008
- 9 WHE009
- 10 WHE010

- 11 WHE011
- 12 WHE012
- 13 WHE013
- 14 WHE014
- 15 WHE015
- 16 WHE016
- 17 WHE017

CLASS 2 NONWHT SUBCLASSES (TOTAL= 2)

- 18 NON001
- 19 NON002











13 JUN 77

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

CLASSIFY

UNIT=1, FILE=2  
STATFI DATA=1,2,3,4,5,6,7,8  
CHANNE STAT=1,2,3,4,5,6,7,8  
CATEGO FILE  
APRIOR FILE  
\*END\*

ORIGINAL PAGE IS  
OF POOR QUALITY

LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

SDISPLAY

DOTFIL INPUT/UNIT=8,FILE=2  
OPTION OUTLINE  
\*END\*

YOU HAVE SELECTED THE FOLLOWING OPTIONS:

PROCESS THE CLASSIFICATION RESULTS FROM MAPTAP ( \*DRUM\* ), UNIT 2 . FILE 2  
APPLY NO THRESHOLDING  
PRINT DOT DATA PERFORMANCE SUMMARIES FOR DOT DATA FROM FORTRAN UNIT \* , FILE NO. Z , TAPE (OR FILE) \*DRUM\*  
OUTLINE THE DOTS ON THE CLASSIFICATION MAP

13T 296

DOT FILE INFORMATION :

NO. DOT CATEGORIES = 2  
 TOTAL NO. OF DOTS = 29  
 DOT CATEGORY NAMES :

#HEAT NONWHT

DOT RECORD :

DOT NO.	SAMPLE	LINE	TYPE	CATEGORY
1	10	20	2	1
2	20	20	2	1
3	30	20	2	1
4	40	20	2	1
5	50	20	2	1
6	60	20	2	1
7	70	20	2	1
8	80	20	2	1
9	90	20	2	1
10	100	20	2	1
11	10	90	2	2
12	20	90	2	2
13	30	90	2	2
14	40	90	2	2
15	50	90	2	2
16	60	90	2	2
17	70	90	2	2
18	80	90	2	2
19	90	90	2	2
20	100	90	2	2
21	110	90	2	2
22	120	90	2	2
23	130	90	2	2
24	140	90	2	2
25	150	90	2	2
26	160	90	2	2
27	170	90	2	2
28	180	90	2	2
29	190	90	2	2











LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

13 JUN 77

CLASSIFICATION SUMMARY FOR FIELD  
TOTAL NUMBER OF SAMPLED POINTS 22932

SUBCLASS	PTS. BEFORE THRES.	PTS. AFTER THRES.	PCT. OF TOTAL CLSF.FLD.	PCT. OF TOTAL CLSF.FLD.	PTS. THRES.	PCT. OF SUBCLASS	PCT. OF TOTAL CLSF.FLD.	PCT. OF SUBCLASS
WREC01	691	691	3.01	3.01	0	100.00	0.00	0.00
WREC02	2311	2311	10.08	10.08	0	100.00	0.00	0.00
WREC03	1576	1576	6.87	6.87	0	100.00	0.00	0.00
WREC04	1003	1003	4.37	4.37	0	100.00	0.00	0.00
WREC05	2196	2196	9.58	9.58	0	100.00	0.00	0.00
WREC06	522	522	2.28	2.28	0	100.00	0.00	0.00
WREC07	361	361	1.57	1.57	0	100.00	0.00	0.00
WREC08	1042	1042	4.54	4.54	0	100.00	0.00	0.00
WREC09	1816	1816	7.92	7.92	0	100.00	0.00	0.00
WREC10	1254	1254	5.47	5.47	0	100.00	0.00	0.00
WREC11	1734	1734	3.20	3.20	0	100.00	0.00	0.00
WREC12	1612	1612	7.03	7.03	0	100.00	0.00	0.00
WREC13	1948	1948	4.13	4.13	0	100.00	0.00	0.00
WREC14	529	529	2.31	2.31	0	100.00	0.00	0.00
WREC15	1754	1754	7.63	7.63	0	100.00	0.00	0.00
WREC16	570	570	2.49	2.49	0	100.00	0.00	0.00
WREC17	472	472	2.06	2.06	0	100.00	0.00	0.00
NUM001	374	374	1.63	1.63	0	100.00	0.00	0.00
NUM002	3211	3211	14.00	14.00	0	100.00	0.00	0.00

PTS. THRESHOLDED IN DISPLAY 0  
PTS. THRESHOLDED IN CLASSIFY TOTAL 6

PCT. = .03

362

DOT DATA PERFORMANCE SUMMARY  
 (SUMMARY OF ANALYST LABELED VERSUS CLASSIFIED CATEGORIES)

ANALYST LABELED DOT CATEGORY	TOTAL NUMBER ANALYST LABELED DOTS	PERCENT CORRECTLY CLASSIFIED	NUMBER THRESHOLDED	UNCORRECTED CATEGORY PROPORTION	BIAS CORRECTED CATEGORY PROPORTION	DOT CLASSIFICATION SUMMARY (ALPHA)
				X		WHEAT      NONWHT
WHEAT	10	70.000	0	84.363	33.494	( 30.435) <sup>7</sup> ( 50.000) <sup>3</sup>
NONWHT	19	15.789	0	15.637	66.506	( 69.565) <sup>16</sup> ( 50.000) <sup>3</sup>

ORIGINAL PAGE IS  
 OF POOR QUALITY



(110, 90) (120, 90) (130, 90) (140, 90)  
CATEGORY CATEGORY CATEGORY CATEGORY  
L A B E L E D C L A S S I F I E D ( S U B C L ) L A B E L E D C L A S S I F I E D ( S U B C L ) L A B E L E D C L A S S I F I E D ( S U B C L ) L A B E L E D C L A S S I F I E D ( S U B C L )  
N O N W H T W H E A T ( W H E O U S ) N O N W H T W H E A T ( W H E O I 2 ) N O N W H T W H E A T ( W H E O I 2 ) N O N W H T W H E A T ( W H E O I 1 )

(150, 90) (160, 90) (170, 90) (180, 90)  
CATEGORY CATEGORY CATEGORY CATEGORY  
L A B E L E D C L A S S I F I E D ( S U B C L ) L A B E L F U C L A S S I F I E D ( S U B C L ) L A B E L E D C L A S S I F I E D ( S U B C L ) L A B E L E D C L A S S I F I E D ( S U B C L )  
N O N W H T W H E A T ( W H E O I 3 ) N O N W H T W H E A T ( W H E O I 0 ) N O N W H T W H E A T ( W H E O I 4 ) N O N W H T W H E A T ( W H E O I 1 )

(190, 90)  
CATEGORY  
L A B E L E D C L A S S I F I E D ( S U B C L )  
N O N W H T W H E A T ( W H E O I 4 )

~~1-46~~  
305

ORIGINAL PAGE IS  
OF POOR QUALITY