"AS-BUILT" DESIGN SPECIFICATION FOR FIELD STATISTICS (FIELDSTAT)

Job Order 71-475

(TIRF 78-0011)

Prepared By

Lockheed Electronics Company, Inc.

Systems and Services Division

Houston, Texas

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

September 1978

LEC-12804

JSC-14487

"AS-BUILT" DESIGN SPECIFICATION
FOR
FIELD STATISTICS (FIELDSTAT)

Job Order 71-475

(TIRF 78-0011)

Prepared By
D. K. McCarley

APPROVED BY

for James A Wilkinson
Philip L. Krumm, Acting Supervisor
Scientific Applications Section

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

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

September 1978 LEC-12804

CONTENTS

Section Page

1. INTRODUCTION. 1-1

2. APPLICABLE DOCUMENTS 2-1

3. SYSTEM DESCRIPTION. 3-1

3.1 GENERAL. 3-1

3.2 DATA FLOW DIAGRAMS 3-2

3.2.1 IMUNLD DATA FLOW 3-2

3.2.2 FLDSTT DATA FLOW 3-2

3.3 FILE DESCRIPTIONS 3-2

3.3.1 IMAGE UNLOAD TAPE FILE (Input only). 3-2

3.3.2 SCREENED IMAGE DISK FILE - SGNMACQDT.IMØ (Output/Input) 3-6

3.3.3 IMUNLD.DAT DISK FILE (Input only) 3-6

3.3.4 HEDREC.SIT DISK FILE (Input only) 3-7

3.3.5 FLDSTT.DAT DISK FILE (Input only) 3-7

3.3.6 FSTLBL.DAT DISK FILE (Input only) 3-9

3.3.7 GROUND TRUTH DISK FILE - SGNMGENDT.GTØ (Input only) 3-9

3.3.8 DOT PIXEL STATISTICS DISK FILE - SGNMACQDT.DPØ (Output only). 3-11

3.3.9 PURE PIXEL STATISTICS DISK FILE - SGNMACQDT.PPØ (Output only). 3-14

3.3.10 MIXED PIXEL STATISTICS DISK FILE - SGNMACQDT.MPØ (Output only). 3-15

3.3.11 SUBPIXEL STATISTICS DISK FILE - SGNMACQDT.SPØ (Output only). 3-15

3.3.12 SCRATCH FILE - TMPSTT.TMP (Input/Output). . 3-17

Section Page

3.4 DETAILED DESIGN. 3-19

3.4.1 IMAGE UNLOAD MAIN PROGRAM (IMUNLD). 3-19

3.4.2 IMUNLD SUBROUTINES 3-24

3.4.3 FIELDSTATS MAIN PROGRAM (FLDSTT) 3-45

3.4.4 FLDSTT SUBROUTINES 3-71

4. OPERATIONS. 4-1

4.1 OPERATOR'S GUIDE 4-1

4.1.1 IMUNLD PROGRAM 4-1

4.1.2 FLDSTT PROGRAM 4-1

4.2 USER'S GUIDE 4-2

4.2.1 BATCH PROCEDURES 4-2

4.2.2 ONLINE PROCEDURES 4-5

1. INTRODUCTION

This document describes the implementation details of the Field Statistics (FIELDSTAT) Program which is designed to provide statistics describing the signatures of the crop classes (as determined by ground truth) in a Large Area Crop Inventory Experiment (LACIE) segment. Included are details on the Image Unload Program which transfers raw image data from magnetic tapes to a disk data base during a screening and tagging process. The screened image data and existing ground truth data are correlated by the FIELDSTAT program and ground truth spectral value statistics files are produced.

2. APPLICABLE DOCUMENTS

The following documents, of the exact issue noted, formed the basis of the software described in this document.

o TIRF 78-0011 dated February 3, 1978.

o Field Statistics (FIELDSTAT) Implementation Specification Document for Accuracy Assessment (LEC 12172) dated May, 1978.

3. SYSTEM DESCRIPTION

3.1 GENERAL

The first program run is the Image Unload program, IMUNLD. It reads from a disk file named IMUNLD.DAT three items of information: the auxiliary disk device number, the magnetic tape device ID, and the magnetic tape number of the image unload tape. It then reads the blind site segment numbers from a disk file named HEDREC.STT. Then it begins scanning the input tape and, when a desired segment is encountered, a disk file is created whose type is IMØ and whose name is the nine-character segment number and acquisition date combination. Next, each of the 117 scan line records is transferred from the tape to the disk file and, during the transfer, each of the 196 pixels in a scan-line is passed through a screening process. If the pixel is not classified as good, it is tagged before being stored on the output disk file. When all of that file has been transferred, the disk file is closed. A disk file is created for each acquisition on the tape from a blind site segment. A report is produced which identifies each disk file created, the portion of the three-part data base where it was stored, and the results of the screening process.

The second program that is run is the Fieldstats program, FLDSTT. The FLDSTT and IMUNLD programs are independent except that FLDSTT uses as input the files that are output from IMUNLD. FLDSTT reads from a disk file named FLDSTT.DAT several items of information: the crop code translations file ID, the ground truth data file ID, the LACIE segment number, an output report option flag, and from one to fifteen acquisition dates. The program reads crop code translations and brightness and greenness coefficients from the crop code translations file. It assigns the specified ground truth data file, and then, for each acquisition date input, that date is combined with the segment number to create the file names used for one input disk file (image unload - type. IMØ) and four output disk files. The four files output are the dot pixel statistics (type.DPØ), the pure pixel statistics (type.PPØ), the mixed pixel statistics (type.MPØ), and the subpixel statistics (type.SPØ). One scan line of data is read from the image unload file and from the ground truth file. For each of the 196 pixels in the scan line, counts and accumulations are made (on a per crop code basis) and kept on an interim scratch disk file. When all 117 scan lines have been processed, the data for the four output files are calculated and stored, and the desired output reports are generated. Then the process is repeated for the next acquisition date.

3.2 DATA FLOW DIAGRAMS

3.2.1 IMUNLD DATA FLOW

Figure 3-1 illustrates the data flow for the IMUNLD program.

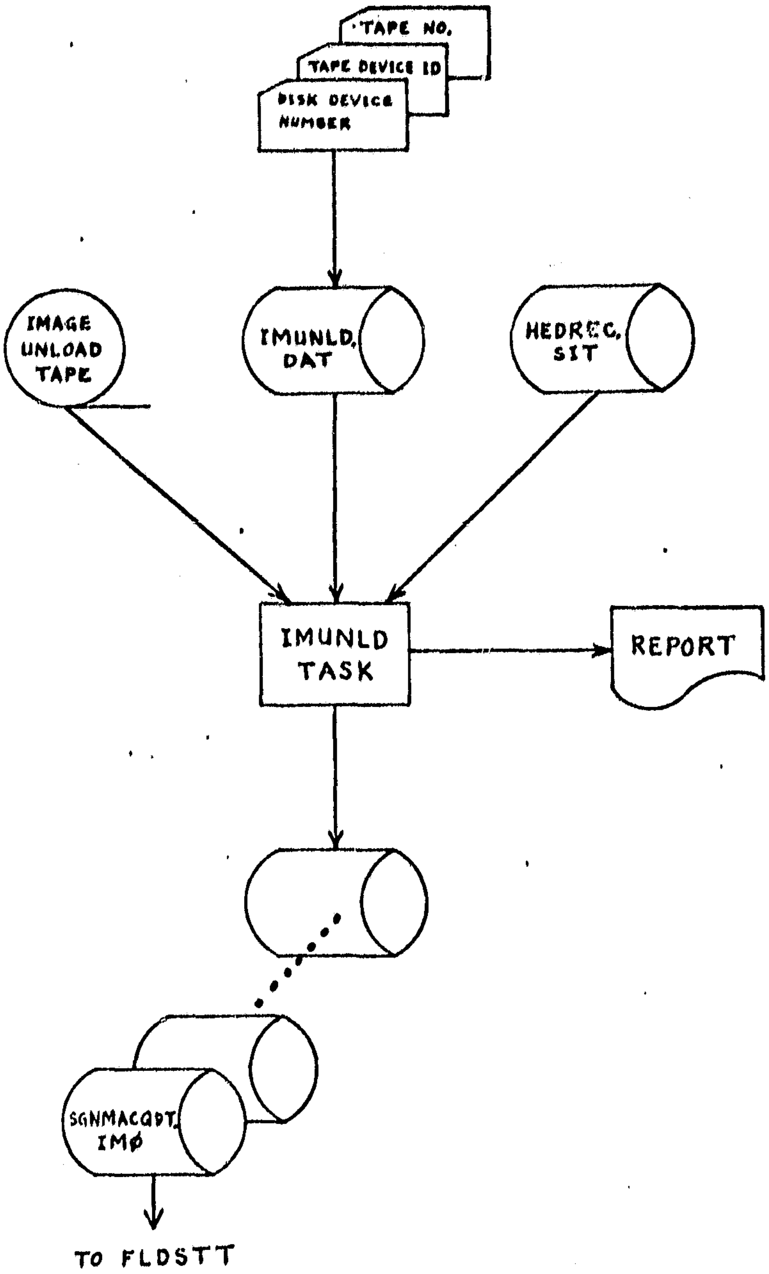
3.2.2 FLDSTT DATA FLOW

Figure 3-2 illustrates the data flow for the FLDSTT program.

3.3 FILE DESCRIPTIONS

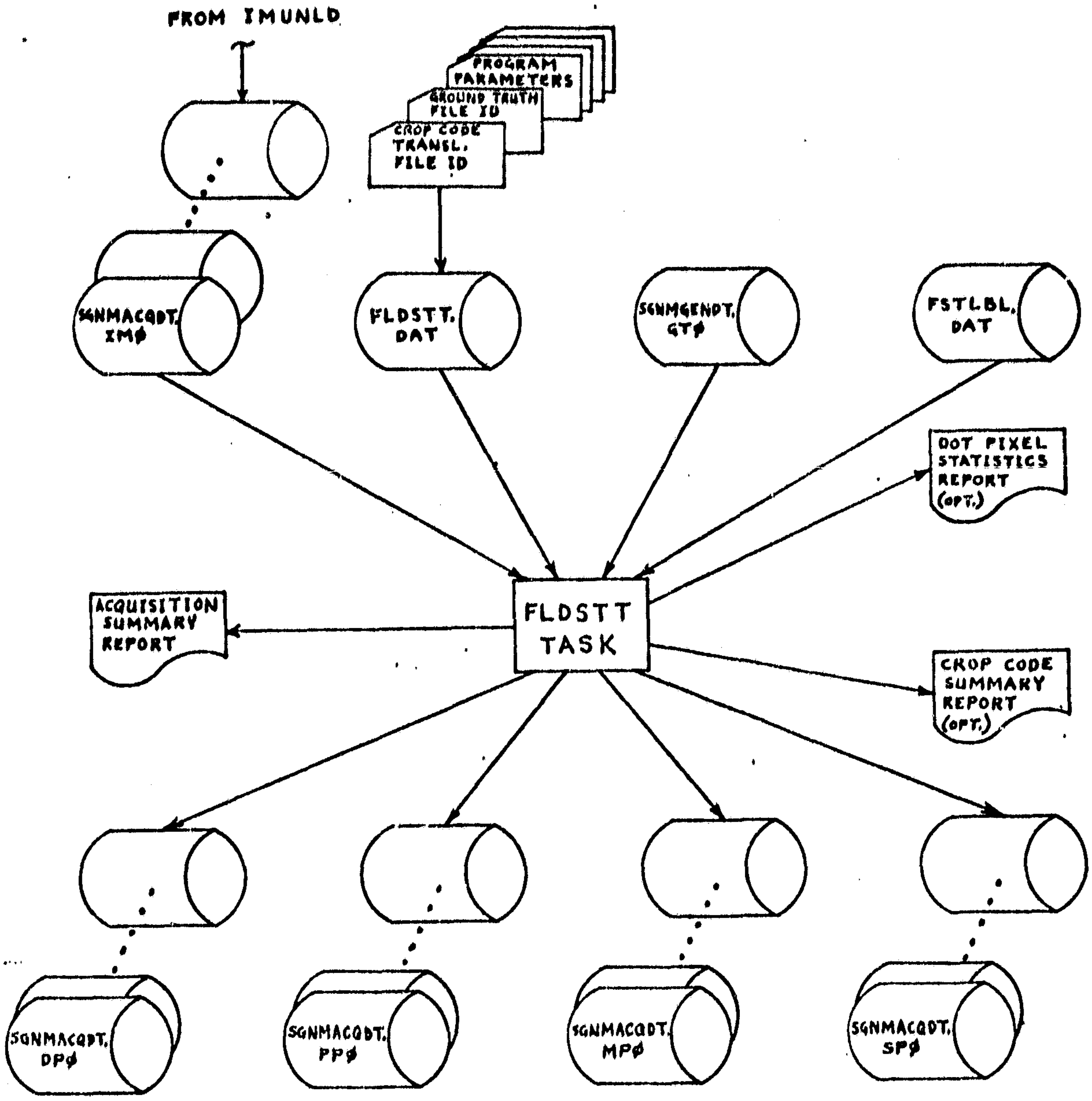
3.3.1 IMAGE UNLOAD TAPE FILE (Input only)

This tape contains the four channels of image data for an acquisition in one file. There may be as many as 100 files on the tape. Each file is in the Universal Format, consisting of one 3060-byte header record, 117 900-byte data records, and one end-of-file record. The end of tape is denoted by two consecutive end-of-file records. The information of interest in the header is described below.



IMUNLD DATA FLOW

FIGURE 3-1



FLDSTT DATA FLOW

FIGURE 3-2

BYTE	CONTENTS
1-32	Tape title in EBCDIC: LACIE PFC IMAGE UNLOAD TAPE
33-38	Tape number in EBCDIC
39-60	N/A
61-63	Date of this tape generation
61	Day of month in binary
62	Month number in binary
63	Last two digits of year in binary
64-66	N/A
67-68	Segment number in binary (bytes inverted)
69-2200	N/A
2201-2202	Sun angle in degrees in binary (bytes inverted)
2203-2248	N/A
2249-2253	Acquisition date in EBCDIC in YYDDD format
2254-3060	N/A

The information of interest in each data record (which represents one scan line) is described below.

BYTE	CONTENTS
1-72	N/A
73-268	Channel one values for pixels 1-196 in binary
269-464	Channel two values
465-660	Channel three values
661-856	Channel four values
857-900	N/A

3.3 '. SCREENED IMAGE DISK FILE - SGNMACQDT.IMØ (Output/Input)

The screened image file is a sequential, unformatted, FILES 11 disk file in Universal Format and is essentially a copy of the corresponding (segment number and acquisition date) file on the image unload tape. The distinction in the data records is that the channel four value for a pixel (which normally ranges from 0-127) is increased by 128 if the screening process determines that the pixel is not good. The header record is copied exactly from the tape with the following exceptions:

a. The new EBCDIC file title (bytes 1-32) is FIELDSTAT SCREENED IMAGE FILE.

b. The segment number bytes (67 and 68) are in their proper order.

c. The current date is stored in bytes 61-63.

3.3.3 IMUNLD.DAT DISK FILE (Input only)

This sequential, formatted disk file is a card-image file used to supply general information for any particular run of the IMUNLD program. Its format is as follows:

RECORD	CHARACTER	CONTENTS
1	1	Device number of the disk onto which the IMUNLD is to store the screened image files. For example, if DB2: is the disk device used, then this record would contain a 2.
2	1-3	Magnetic tape device ID for input Image Unload tape, e.g., MT1.
3	1-6	The Image Unload tape number, e.g., 770080.

3.3.4 HEDREC.SIT DISK FILE (Input only)

This sequential, unformatted disk file contains all the valid blind site segment numbers in one record as binary integers. The first 64 words are the 64 segments pertaining to Part 1 of the Accuracy Assessment (AA) data base. The second 64 words are the 64 segments pertaining to Part 2 of the AA data base, and the third 64 words are the 64 segments pertaining to Part 3 of the AA data base.

3.3.5 FLDSTT.DAT DISK FILE (Input only)

This sequential, formatted disk file is a card-image file used to supply general information for any particular run of the FLDSTT program. Its format is as follows:

RECORD	CHARACTER	CONTENTS
1	1-30	File ID where the crop code translations and brightness and greenness coefficients are stored. The file name is FSTLBL.DAT, but this field should include the specific device and User File Directory (UFD) to enable location of the desired file, e.g., DB2:[131,ØØ1]FSTLBL.DAT.
2	1-30	File ID where the ground truth data are stored. After this file is assigned, then this character string is used to create output files, therefore it must be very exact in format. For example, the file ID DB2:[131,ØØ1]161177353.GTØ is acceptable (the acquisition date must be characters 18-22).
3	1-4	Segment number to be processed.

RECORD	CHARACTER	CONTENTS
3	6-7	Output report option flag. A two-digit decimal number specifying which output reports are desired. The number is the sum of the numbers assigned to the various reports. The Acquisition Summary report is always produced, whereas the Dot Pixel Statistics report (assigned number 1) and the Crop Code Summary report (assigned number 2) are optional.
3	9-10	A two-digit decimal number, greater than zero and less than sixteen, which specifies how many acquisition dates are to be processed during this run of the FLDSTT program.
4	1-5	First acquisition date to be processed. Represented in standard YYDDD format. As many acquisition dates as are indicated on record three should be specified in successive fields on records four, five, and six.
4	11-15	Second acquisition date.
4	21-25	Third acquisition date.
4	31-35	Fourth acquisition date.
4	41-45	Fifth acquisition date.
5	1-5	Sixth acquisition date.
5	11-15	Seventh acquisition date.
5	21-25	Eighth acquisition date.
5	31-35	Ninth acquisition date.

RECORD	CHARACTER	CONTENTS
5	41-45	Tenth acquisition date.
6	1-5	Eleventh acquisition date.
6	11-15	Twelfth acquisition date.
6	21-25	Thirteenth acquisition date.
6	31-35	Fourteenth acquisition date.
6	41-45	Fifteenth acquisition date.

3.3.6 FSTLBL.DAT DISK FILE (Input only)

This sequential, formatted disk file is a card image file used to supply crop code translations and brightness and greenness coefficients to the FLDSTT program. The first sixteen cards each contain sixteen five-character fields. The first three characters of each field are the numeric crop code, and the last two characters are the alphanumeric translations. These fields must be in ascending numerical sequence, 000-255. The seventeenth card contains the four brightness coefficients beginning at character one and separated by commas. The eighteenth card contains the four greenness coefficients beginning at character one and separated by commas.

3.3.7 GROUND TRUTH DISK FILE - SGNMGENDT.GTØ (Input only)

The ground truth file is a sequential, unformatted, FILES 11 disk file in Universal format, with one 3060-byte header record and 351 540-byte data records. The only information in the header record that is of interest to the FLDSTT program is the segment number, stored in word thirty-four, which is checked against the specified segment number from the FLDSTT.DAT file.

In ground truth data, which records actual crop codes, each pixel is divided into six subpixels (three rows by two columns), so that one scan line across the 196 pixels actually records values for 2*196 subpixels, and three scan lines are needed to record values for the complete 196 pixels in each of the 117 pixel rows. Consequently, there are 3*117=351 data records, each of which contains crop code values for two subpixels out of each of the 196 pixels in the scan line. The information in the data records is laid out as follows:

RECORD	BYTE	CONTENTS
1	1-72	N/A
1	73	Crop code for upper left subpixel of pixel one in row one.
1	74	Crop code for upper right subpixel of pixel one in row one.
1	75	Crop code for upper left subpixel of pixel two in row one.
.	.	.
.	.	.
.	.	.
1	464	Crop code for upper right subpixel of pixel 196 in row one.
1	465-540	N/A
2	1-72	N/A
2	73	Crop code for middle left subpixel of pixel one in row one.
.	.	.
.	.	.
.	.	.
2	464	Crop code for middle right subpixel of pixel 196 in row one.
2	465-540	N/A
3	1-72	N/A

RECORD	BYTE	CONTENTS
3	73	Crop code for lower left subpixel of pixel one in row one.
.	.	.
.	.	.
.	.	.
3	464	Crop code for lower right subpixel of pixel 196 in row one.
3	465-540	N/A
4	1-72	N/A
4	73	Crop code for upper left subpixel of pixel one in row two.
.	.	.
.	.	.
.	.	.
351	464	Crop code for lower right subpixel of pixel 196 in row 117.
351	465-540	N/A

It should be noted that the value of each crop code is biased by -128 if looked at as an eight-bit computer word.

3.3.8 DOT PIXEL STATISTICS DISK FILE - SGNMACQDT.DPØ (Output only)

The dot pixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header (two bytes per word) and eleven 270-word data records. The format of the header record is as follows:

WORD	CONTENTS (binary single word integer except as noted)
1-5	FIELDSTATS (ASCII characters)

WORD	CONTENTS
6-8	Date of file generation: Word 6 = day of the month Word 7 = month number Word 8 = value of the last two digits of the year
9	Segment number
10-11	Acquisition date (double word binary integer)
12	Sun angle in degrees
13	Number of good pixels
14	Number of garbled pixels
15	Number of cloud pixels
16	Number of shadow pixels
17	Number of water pixels
18-19	Channel one means (real)
20-21	Channel two means (real)
22-23	Channel three means (real)
24-25	Channel four means (real)
26-27	Gamma (real)
28	Number of one-class pixels
29	Number of two-class pixels
30	Number of three-class pixels
31	Number of four-class pixels
32	Number of five-class pixels
33	Number of six-class pixels
34	Number of different crop codes encountered in good pixels.

WORD	CONTENTS
35-290	Crop code occurrance array. Applicable only to the pure pixel, mixed pixel, and subpixel statistics files. Each word corresponds to crop codes 0-255, in that order. A zero in the word means that crop code did not occur. A value, I, in the Jth word means that crop code J-1 did occur, and the statistics for that crop code are recorded in record I (header record excluded) of the file.
291-360	Zero fill.

Each data record contains nineteen twelve-word blocks of information in the first 228 words of the record and zero fill in the last forty-two words. Each block of information pertains to one of the designated 209 dot pixels and is laid out as follows:

WORD	CONTENTS (all are single-word binary integers)
1	Dot ID, range 1-209
2	Channel one spectral value, range 0-255
3	Channel two spectral value, range 0-255
4	Channel three spectral value, range 0-255
5	Channel four spectral value, range 0-127
6	Subpixel one crop code, range 0-255
7	Subpixel two crop code, range 0-255
8	Subpixel three crop code, range 0-255
9	Subpixel four crop code, range 0-255
10	Subpixel five crop code, range 0-255
11	Subpixel six crop code, range 0-255
12	Screening label, range 1-47

3.3.9 PURE PIXEL STATISTICS DISK FILE - SGNMACQDT.PPØ (Output only)

The pure pixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header record and from zero to 256 630-word data records. The header record is identical to the dot pixel statistics file header record. There will be one data record for each different crop code found in the good pixels of the segment. A pure pixel is a good pixel whose subpixels all have the same crop code. For a crop code detected in the segment, but having no pure pixels with that crop code, the data record will contain all zeroes except the first word, which will contain the crop code value. For crop codes that did have pure pixels, the data records have the following format:

WORD	CONTENT (single word binary integer except as noted)
1	Crop code
2-3	Channel one means (real)
4-5	Channel two means (real)
6-7	Channel three means (real)
8-9	Channel four means (real)
10-11	Channel one standard deviation (real)
12-13	Channel two standard deviation (real)
14-15	Channel three standard deviation (real)
16-17	Channel four standard deviation (real)
18-19	Channels 1x2 correlation (real)
20-21	Channels 1x3 correlation (real)
22-23	Channels 1x4 correlation (real)
24-25	Channels 2x3 correlation (real)
26-27	Channels 2x4 correlation (real)

WORD	CONTENT
28-29	Channels 3x4 correlation (real)
30	Pixel count
31-181	Brightness histogram, 151 words representing counts of values 0-150
182-297	Greenness histogram, 116 words representing counts of values (-30) -85
298-630	Zero fill.

3.3.10 MIXED PIXEL STATISTICS DISK FILE - SGNMACQDT.MPØ (Output only)

The mixed pixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header record and from zero to 256 630-word data records. The header record is identical to the dot pixel statistics file header record. There will be one data record for each different crop code found in the good pixels of the segment. A mixed pixel for a certain crop code is a good pixel with at least one subpixel having that crop code. Note that this definition includes pure pixels as a subset of mixed pixels. Each data record has exactly the same format as a pure pixel statistics file data record.

3.3.11 SUBPIXEL STATISTICS DISK FILE - SGNMACQDT.SPØ (Output only)

The subpixel statistics file is a sequential, unformatted, FILES 11 disk file with one 360-word header record and from zero to 256 630-word data records. The header record is identical to the dot pixel statistics file header record. There will be one data record for each different crop code found in the good pixels of the segment. Each data record will have the following format:

WORD	CONTENTS (double word binary integer except as noted)
1	Crop code (single word binary integer)
2-3	Channel one means (real)
4-5	Channel two means (real)
6-7	Channel three means (real)
8-9	Channel four means (real)
10-11	Channel one standard deviation (real)
12-13	Channel two standard deviation (real)
14-15	Channel three standard deviation (real)
16-17	Channel four standard deviation (real)
18-19	Channels 1x2 correlation (real)
20-21	Channels 1x3 correlation (real)
22-23	Channels 1x4 correlation (real)
24-25	Channels 2x3 correlation (real)
26-27	Channels 2x4 correlation (real)
28-29	Channels 3x4 correlation (real)
30-31	Subpixel count
32-333	Brightness histogram, 151 integers representing counts of values 0-150
334-565	Greenness histogram, 116 integers representing counts of values (-30) -85
566-630	Zero fill

3.3.12 SCRATCH FILE - TMPSTT.TMP (Input/Output)

The program FLDSTT uses a direct access, unformatted, scratch disk file, that has 1280 words per record. The first 256 records are reserved for interim storage of counts and accumulations on a per crop code basis. Records 257 through 267 are used for interim storage of the dot pixel statistics file data records and are formatted exactly like those records. The format of any one of the 256 crop code statistics records is as follows:

WORD	CONTENTS (single word binary integers except as noted)
1	Crop code
2,3	Sum of pure pixel channel one values (double word binary integer (DWBI))
4,5	Sum of pure pixel channel two values (DWBI)
6,7	Sum of pure pixel channel three values (DWBI)
8,9	Sum of pure pixel channel four values (DWBI)
10,11	Sum of channels 1x1 products (DWBI) - pure pixels
12,13	Sum of channels 1x2 products (DWBI) - pure pixels
14,15	Sum of channels 1x3 products (DWBI) - pure pixels
16,17	Sum of channels 1x4 products (DWBI) - pure pixels
18,19	Sum of channels 2x2 products (DWBI) - pure pixels
20,21	Sum of channels 2x3 products (DWBI) - pure pixels
22,23	Sum of channels 2x4 products (DWBI) - pure pixels
24,25	Sum of channels 3x3 products (DWBI) - pure pixels
26,27	Sum of channels 3x4 products (DWBI) - pure pixels
28,29	Sum of channels 4x4 products (DWBI) - pure pixels
30	Pure pixel count

WORD	CONTENTS
31-181	Pure pixel brightness counters, 151 integers representing values 0-150
182-297	Pure pixel greenness counters, 116 integers representing values (-30) -85
298-593	Repeat of words 2-297 except for mixed pixels
594-621	Repeat of words 2-29 except for subpixels
622-1157	Repeat of words 30-297 except DWBI for subpixels.

3.4 DETAILED DESIGN

3.4.1 IMAGE UNLOAD MAIN PROGRAM (IMUNLD)

3.4.1.1 Linkage

This program calls the special subroutines BEGRPT, FBLD, and FSTART in addition to system subroutines.

3.4.1.2 Interface

BEGRPT, FBLD, FSTART

3.4.1.3 Input

An Image Unload tape, a blind site segment numbers disk file, and a program parameters disk file.

3.4.1.4 Output

Screened image disk files for use by at least the FLDSTT program, and a report showing, for each screened image disk file produced, the file name, the part of the AA data base into which it was stored, and the count of good, garbled, cloud, shadow, and water pixels.

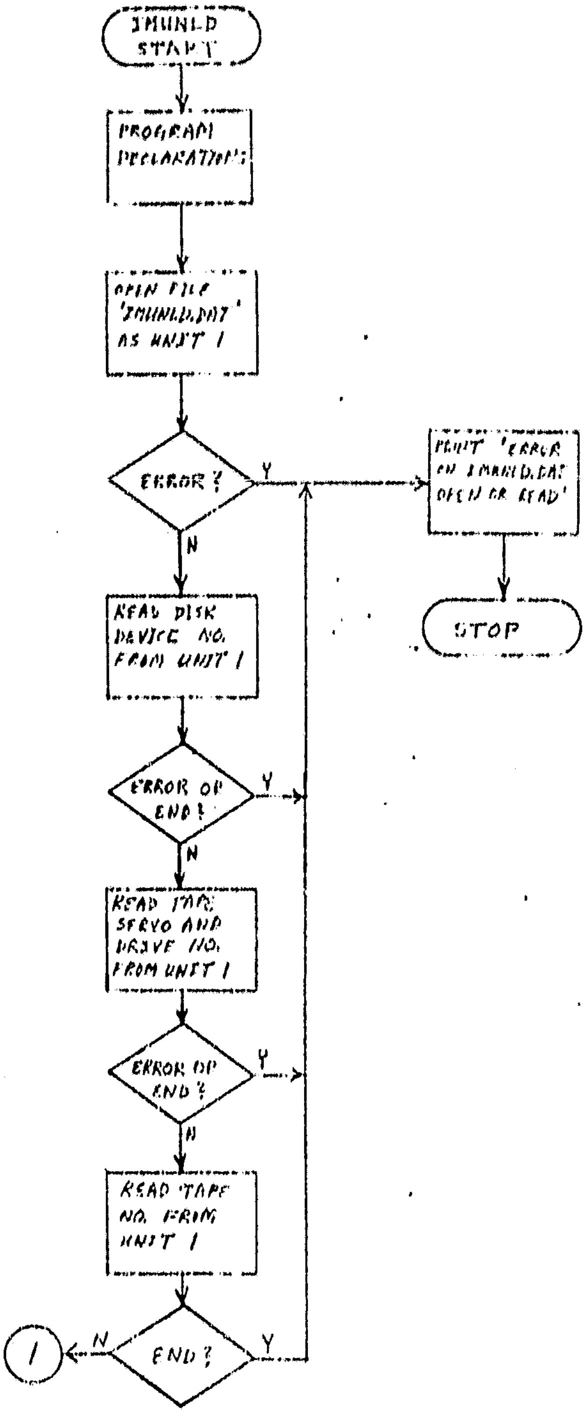
3.4.1.5 Storage

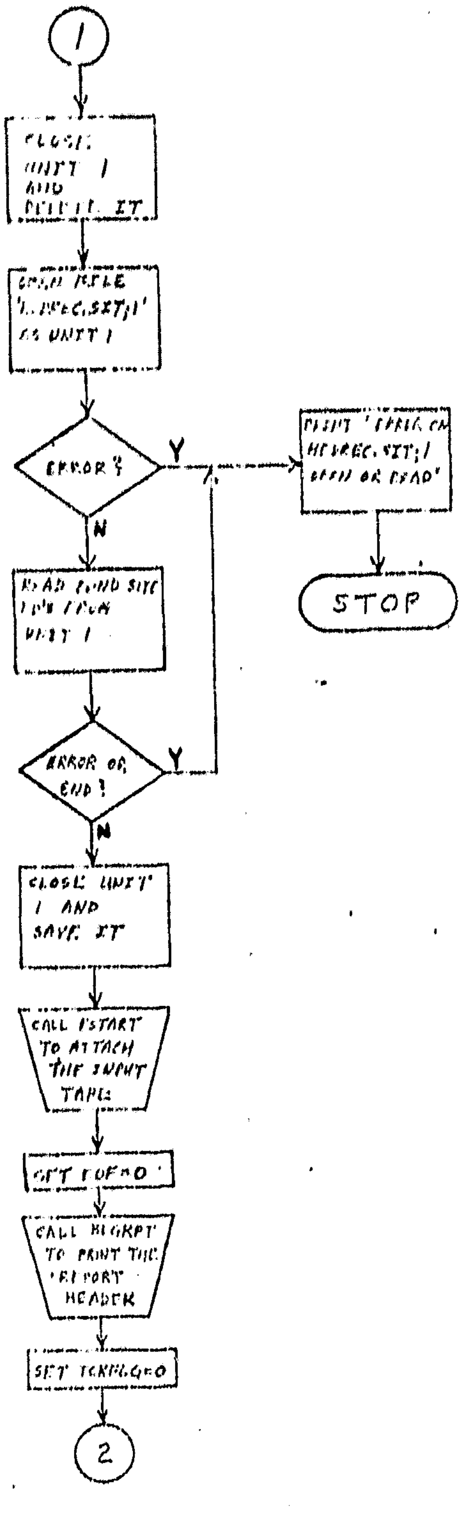
15,334 words.

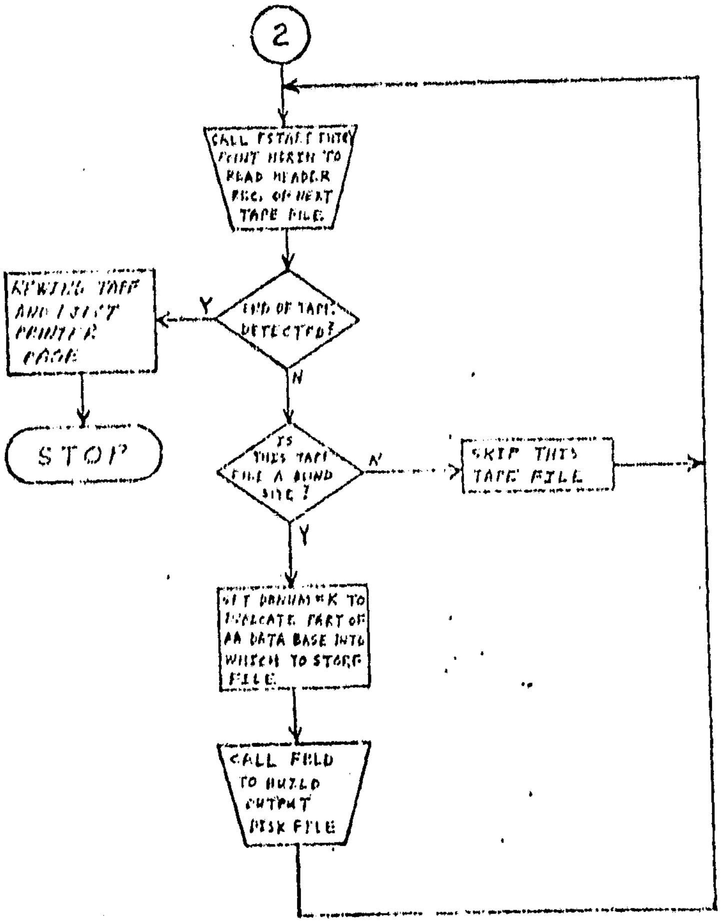
3.4.1.6 Description

This program scans files on an Image Unload tape and, for each blind site segment it finds, it creates a disk file of the same data after screening each pixel and flagging those which are not good.

3.4.1.7 Flowchart


IMUNLD START
PROGRAM DECLARATIONS
OPEN FILE 'IMUNLD.DAT' AS UNIT 1
ERROR?
Y
N
PRINT 'ERROR ON IMUNLD.DAT OPEN OR READ'
STOP
READ DISK DEVICE NO. FROM UNIT 1
ERROR OR END?
Y
N
READ TAPE SERVO AND DRIVE NO. FROM UNIT 1
ERROR OR END?
Y
N
READ TAPE NO. FROM UNIT 1
END?
N
Y
1


1
CLOSE UNIT 1 AND DELETE IT
OPEN FILE 'HEDREC.SIT;1' AS UNIT 1
ERROR?
Y
PRINT 'ERROR ON HEDREC.SIT;1 OPEN OR READ'
STOP
N
READ FUND SITE LIN FROM UNIT 1
ERROR OR END?
Y
N
CLOSE UNIT 1 AND SAVE IT
CALL ISTART TO ATTACH THE INPUT TAPE
SET EOF=0
CALL HDGRPT TO PRINT THE REPORT HEADER
SET ICNFLG=0
2


2
CALL FSTART ENTRY POINT HEREIN TO READ HEADER REC. OF NEXT TAPE FILE.
END OF TAPE DETECTED?
Y
REWIND TAPE AND EJECT PRINTER PAGE
STOP
N
IS THIS TAPE FILE A BUND SITE ?
N
SKIP THIS TAPE FILE
Y
SET DBNUM*K TO INDICATE PART OF AA DATA BASE INTO WHICH TO STORE FILE.
CALL FBLD TO BUILD OUTPUT DISK FILE

3.4.1.8 Listing

```
        PROGRAM IMUNLD
C
C       THIS PROGRAM TRANSFERS IMAGE UNLOAD GRIND SITE DATA FROM
C       TAPE TO DISK, SCREENING EACH PIXEL AND MARKING THE NON-GOOD
C       ONES BY ADDING 128 TO THE CHANNEL FOUR VALUE.
C
        IMPLICIT INTEGER(A-Z)
        PARAMETER SIZE=64,MAXDSK=3
        LOGICAL*1 FMNUM,ACCDTE(6),DEVICE
        COMMON SEGNUM,ACCDTE,TAPNO(3),DEVICE
        DIMENSION SITE(SIZE,MAXDSK),SEGNAM(2)
C
C
        OPEN(UNIT=1,NAME='IMUNLD.DAT',TYPE='OLD',ERR=3)
        GO TO 6
3       CONTINUE
        PRINT 4000
4000    FORMAT(' ERROR ØN IMUNLD.DAT ØPEN ØR READ'/1H1)
        STOP       'IMUNLD.DAT ERR'
6       CONTINUE
        READ(1,7,END=3,ERR=3)DEVICE
7       FORMAT(A1)
        READ(1,10,END=3,ERR=3)SERVR,DRIVE
10      FORMAT(A2,I1)
        READ(1,20,END=3)TAPNO
20      FORMAT(3A2)
C
        CLOSE(UNIT=1,DISPOSE='DELETE')
        OPEN(UNIT=1,NAME='HEDREC.SIT;1',TYPE='ØLD',
     *  FORM='UNFORMATTED',READONLY,ERR=25)
        GØ TO 30
25      CONTINUE
        PRINT 4001
4001    FORMAT(' ERRØR ØN HEDREC.SIT;1 ØPEN ØR READ'/1H1)
        STOP       'ERR ØN HEDREC.SIT;1'
30      CONTINUE
        READ(1,END=25,ERR=25) ((SITE(J,K),J=1,SIZE),K=1,MAXDSK)
        CLOSE(UNIT=1,DISPOSE='SAVE')
        CALL FSTART(SERVR,DRIVE)
        EOF=0
        CALL BEGRPT(TAPNO)
        TOKFLG=0
40      CALL DPRIN(EOF,TOKFLG)
        IF(EOF .EQ. 1)GØ TO 200
        DO 50 K=1,MAXDSK
        DO 50 J=1,SIZE
        IF(SITE(J,K) .EQ. SEGNUM)GØ TØ 60
50      CONTINUE
        CALL TFILE(1,1)
        GØ TØ 40
60      FMNUM=K
        CALL FOLD(FMNUM)
        GO TO 40
200     CALL TEND(1)
        PRINT 400
400     FORMAT(1H1)
        STOP
        END
```

ORIGINAL PAGE IS
OF POOR QUALITY

3.4.2 IMUNLD SUBROUTINES

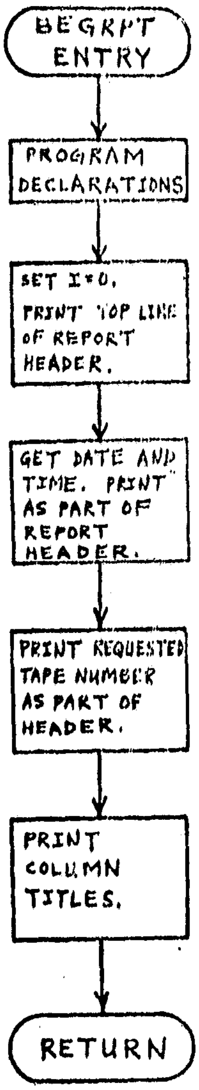
3.4.2.1 General

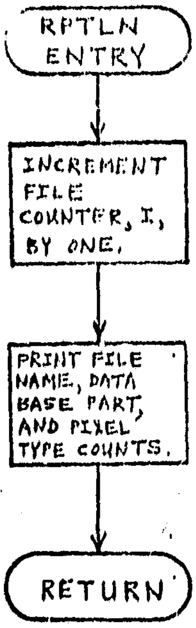
The special subroutines called by IMUNLD are BEGRPT, FBLD, and FSTART. FSTART calls special subroutines ETOA and SCREEN. BEGRPT has an additional entry point RPTLN, referenced by FBLD. FSTART has additional entry points HDRIN, referenced by IMUNLD, and DATOUT, referenced by FBLD. SCREEN has an additional entry point SNAG, referenced by FSTART.

3.4.2.2 Subroutine BEGRPT

This subroutine prints the header and each line of the output report.

3.4.2.2a Flowchart


BEGRPT ENTRY
PROGRAM DECLARATIONS
SET I=0. PRINT TOP LINE OF REPORT HEADER.
GET DATE AND TIME. PRINT AS PART OF REPORT HEADER.
PRINT REQUESTED TAPE NUMBER AS PART OF HEADER.
PRINT COLUMN TITLES.
RETURN


RPTLN ENTRY
INCREMENT FILE COUNTER, I, BY ONE.
PRINT FILE NAME, DATA BASE PART, AND PIXEL TYPE COUNTS.
RETURN

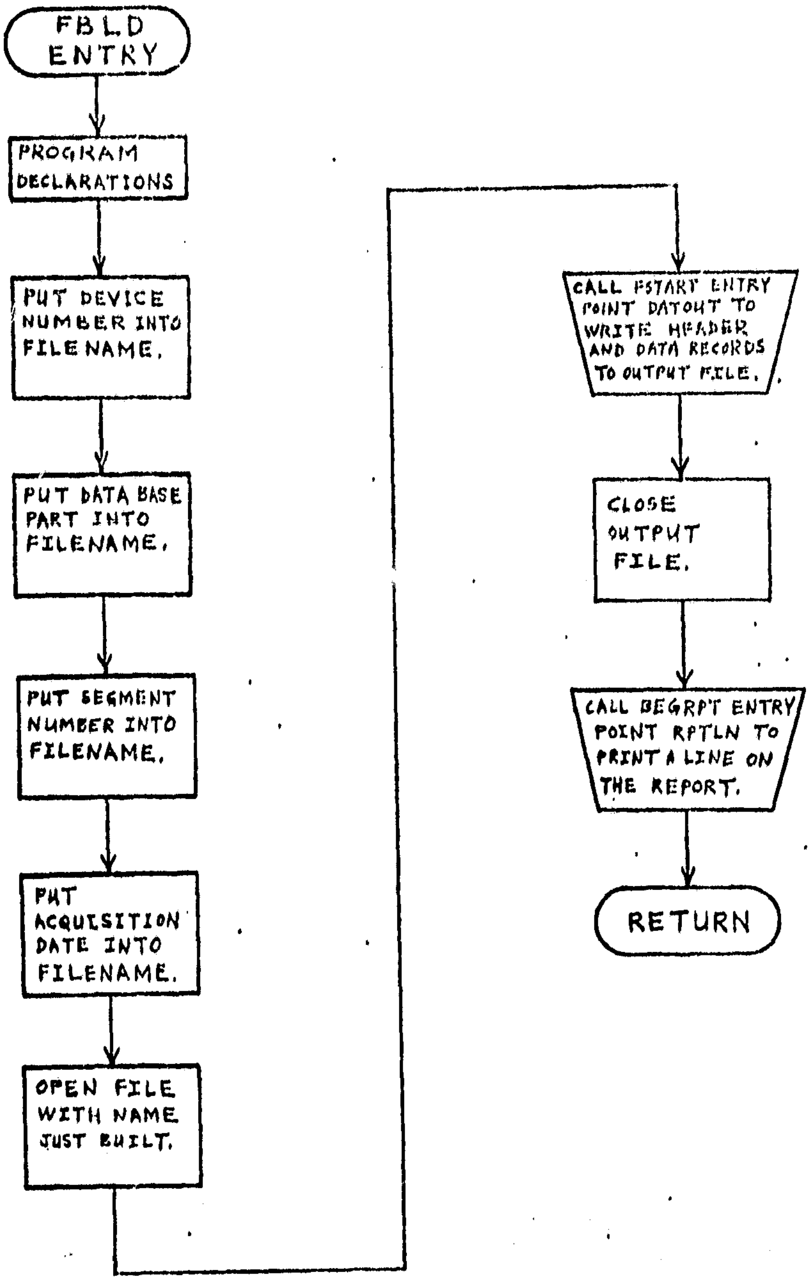
3.4.2.2b Listing

```
      SUBRØUTINE BEGRPT(TAPEN).
      IMPLICIT INTEGER(A-Z)
      DIMENSIØN TAPEN(3),PTC(5)
      LØGICAL*1 IDATE(9),ITIME(8),FILNAM(28),DBNUM
C
C
      I=0
      PRINT 20
20    FØRMAT(1H1,34X,'ACCURACY ASSESSMENT DATA TRANSACTIØN REPØRT')
      CALL DATE(IDATE)
      CALL TIME(ITIME)
      PRINT 30,IDATE,ITIME
30    FØRMAT(105X,'DATE ',9A1,/,105X,'TIME ',8A1)
      PRINT 40,TAPEN
40    FØRMAT(47X,'IMAGE DATA BASE UPDATE',/,47X,
     * 'MAG TAPE NUMBER ',3A2)
      PRINT 42
42    FØRMAT(1H0,13X,'FILE NAME',7X,'DATA      GØØD     GARBLED',
     *    6X,'CLØUD     SHADØW      WATER')
      PRINT 50
50    FØRMAT(30X,'BASE',5(5X,'PIXELS')/)
      RETURN
      ENTRY RPTLN(FILNAM,DBNUM,PTC)
      I=I+1
      PRINT 60,I,(FILNAM(J),J=12,24),DBNUM,PTC
60    FØRMAT(5X,I3,2H,),2X,13A1,7X,I1,5I11)
      RETURN
      END
```

3.4.2.3 Subroutine FBLD

This subroutine creates the proper file name, opens that file, transfers the data to it, closes it, and reports on it.

3.4.2.3a Flowchart


FBLD ENTRY
PROGRAM DECLARATIONS
PUT DEVICE NUMBER INTO FILENAME.
PUT DATA BASE PART INTO FILENAME.
PUT SEGMENT NUMBER INTO FILENAME.
PUT ACQUISITION DATE INTO FILENAME.
OPEN FILE WITH NAME JUST BUILT.
CALL FSTART ENTRY POINT DATOUT TO WRITE HEADER AND DATA RECORDS TO OUTPUT FILE.
CLOSE OUTPUT FILE.
CALL BEGRPT ENTRY POINT RPTLN TO PRINT A LINE ON THE REPORT.
RETURN

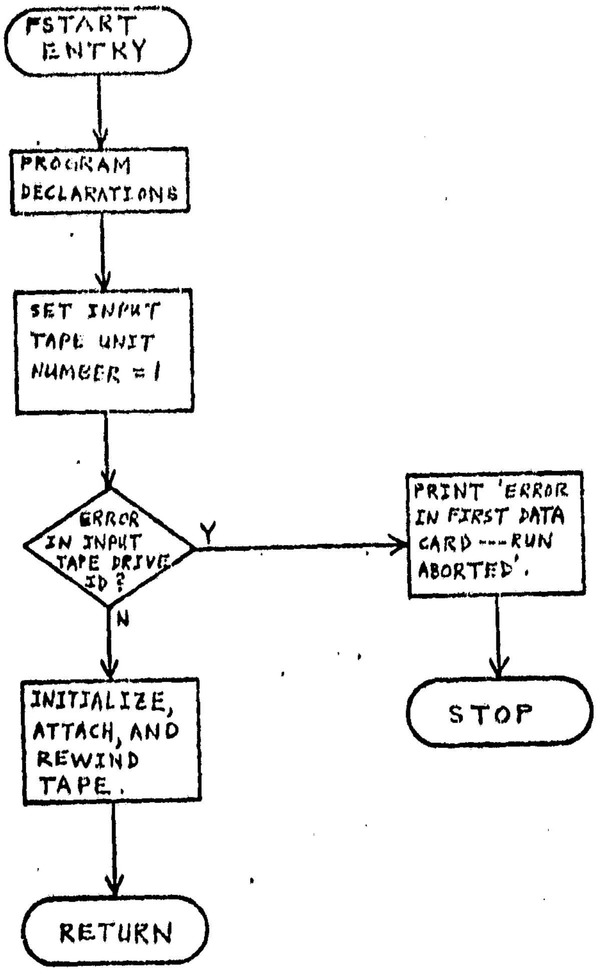
3.4.2.3b Listing

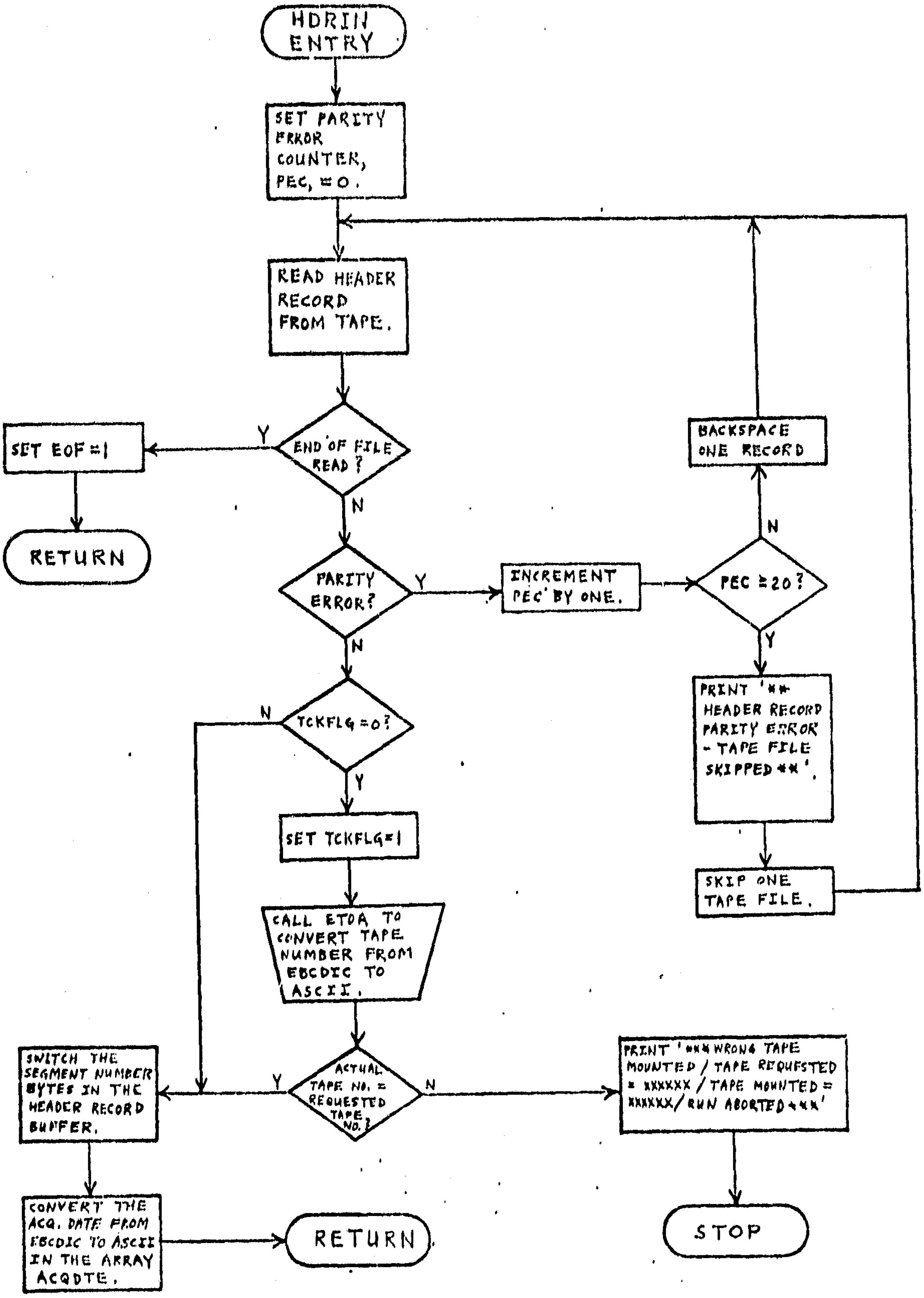
```
        SUBRØUTINE FBLD(DBNUM)
        IMPLICIT INTEGER(A-Z)
        LØGICAL*1 FILNAM(23),DBNUM,ACQDTE(6),DEVICE
        CØMMØN SEGNUM,ACQDTE,TAPE(3),DEVICE
        DIMENSIØN PTC(5),FIL(14)
        EQUIVALENCE (FIL,FILNAM)
        DATA FIL/2HDB,2H2:,2H[1,2H31,2H,1,2H]X,2HXX,2HXX,2HXX,
     *  2HXX,2H.I,2HM0,2H11,0/
C
C
        FILNAM(3)=DEVICE
        ENCØDE(1,50,FILNAM(10)) DBNUM
50      FØRMAT(I1)
        ENCØDE(4,51,FILNAM(12))SEGNUM
51      FØRMAT(I4)
        DØ 60 J=1,5
        FILNAM(J+15)=ACQDTE(J)
60      CØNTINUE
        DØ 71 J=12,15
        IF(FILNAM(J) .EQ. ' ') FILNAM(J)='0'
71      CØNTINUE
        ØPEN(UNIT=2,NAME=FILNAM,TYPE='UNKNØWN',FØRM='UNFØRMATTED')
        CALL DATØUT(PTC)
        CALL CLØSE(2)
        CALL RPTLN(FILNAM,DBNUM,PTC)
        RETURN
        END
```

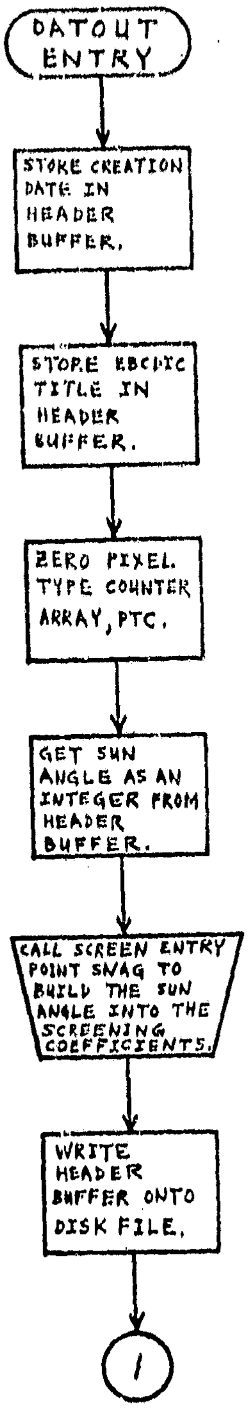
3.4.2.4 Subroutine FSTART

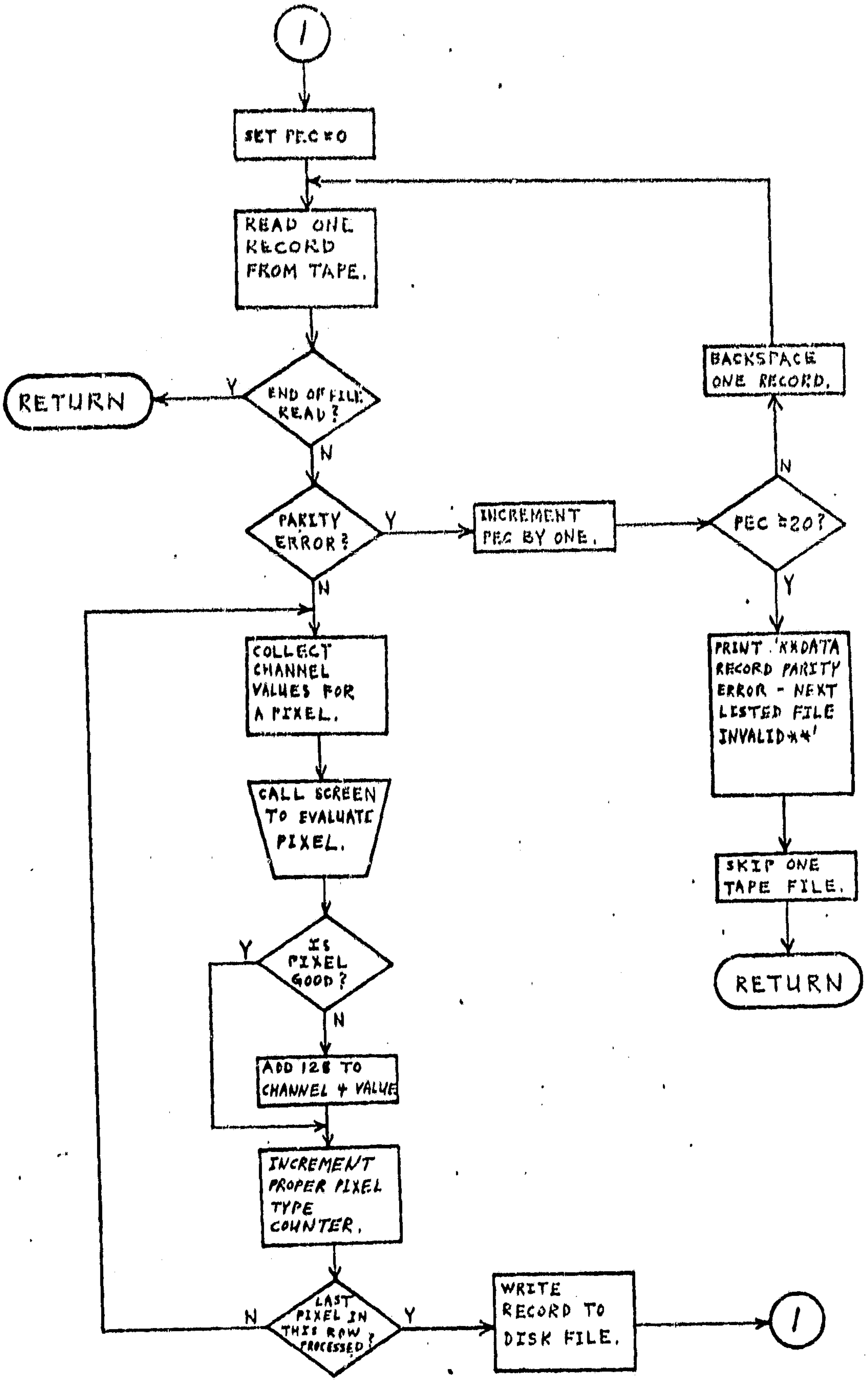
This subroutine sets up the tape drive, reads each file's header record, reads and screens the data from a particular file, and writes the output disk file.

3.4.2.4a Flowchart


FSTART ENTRY
PROGRAM DECLARATIONS
SET INPUT TAPE UNIT NUMBER = 1
ERROR IN INPUT TAPE DRIVE ID ?
Y
PRINT 'ERROR IN FIRST DATA CARD ---RUN ABORTED'.
STOP
N
INITIALIZE, ATTACH, AND REWIND TAPE.
RETURN


HDRIN ENTRY
SET PARITY ERROR COUNTER, PEC, =0.
READ HEADER RECORD FROM TAPE.
END OF FILE READ ?
Y
SET EOF=1
RETURN
N
PARITY ERROR?
Y
INCREMENT PEC BY ONE.
PEC ≥20 ?
N
BACKSPACE ONE RECORD
Y
PRINT '** HEADER RECORD PARITY ERROR - TAPE FILE SKIPPED **'.
SKIP ONE TAPE FILE.
N
TCKFLG=0?
N
Y
SET TCKFLG=1
CALL ETOA TO CONVERT TAPE NUMBER FROM EBCDIC TO ASCII.
ACTUAL TAPE NO. = REQUESTED TAPE NO.?
Y
SWITCH THE SEGMENT NUMBER BYTES IN THE HEADER RECORD BUFFER.
CONVERT THE ACQ. DATE FROM EBCDIC TO ASCII IN THE ARRAY ACQDTE.
RETURN
N
PRINT '*** WRONG TAPE MOUNTED / TAPE REQUESTED = XXXXXX / TAPE MOUNTED = XXXXXX/ RUN ABORTED ***'
STOP


DATOUT ENTRY
STORE CREATION DATE IN HEADER BUFFER.
STORE EBCDIC TITLE IN HEADER BUFFER.
ZERO PIXEL TYPE COUNTER ARRAY, PTC.
GET SUN ANGLE AS AN INTEGER FROM HEADER BUFFER.
CALL SCREEN ENTRY POINT SNAG TO BUILD THE SUN ANGLE INTO THE SCREENING COEFFICIENTS.
WRITE HEADER BUFFER ONTO DISK FILE.
1


1
SET PEC=0
READ ONE RECORD FROM TAPE.
END OF FILE READ?
Y
RETURN
N
PARITY ERROR?
Y
INCREMENT PEC BY ONE.
PEC ≥20?
N
BACKSPACE ONE RECORD.
Y
PRINT '**DATA RECORD PARITY ERROR - NEXT LISTED FILE INVALID**'
SKIP ONE TAPE FILE.
RETURN
N
COLLECT CHANNEL VALUES FOR A PIXEL.
CALL SCREEN TO EVALUATE PIXEL.
IS PIXEL GOOD?
Y
N
ADD 128 TO CHANNEL 4 VALUE
INCREMENT PROPER PIXEL TYPE COUNTER.
LAST PIXEL IN THIS ROW PROCESSED?
N
Y
WRITE RECORD TO DISK FILE.
1

3.4.2.4b Listing

```
          SUBROUTINE FSTART(SCRVB,DRIVE)
          IMPLICIT INTEGER(A-Z)
          PARAMETER [illegible]
          COMMON/STATUS/ S(2)
          LOGICAL*1 HEADBY(3060),ACQITE(6),TAPENE(6),TITL(32),SCRVB(4)
          LOGICAL*1 SCANCR(2),RTN(6),DEVICL,LTRAC
          COMMON SEGMT,ACQITE,TAPENM(3),DEVICE
          DIMENSION HEADER(1530),RECORD(1530),RTC(3),SCRV(4)
          EQUIVALENCE (HEADBY,HEADER,RECORD),(TAPEN ,TAPENE),(SCRV,SCRVB),
     *                (SCANCR,SCANER)
          DATA MT/'MT'/,XT/'XT'/,SCR /4*0/,LTRAC/
          DATA TITL/"356,"311,"305,"323,"304,"342,   3,"301,
     *              "343,"100,"342,"3  ,"331,"3  ,   2,"325,
     *              "305,"404,"100,"311,"374,"311,  7,"305,
     *              "100,"306,"311," 23,"305,"1 0,"1 0,"100/
C
C
          UNIT=1         !LOGICAL UNIT
C         MTORXT=0        ! 0=MT , 1=XT
C         DRIVE=1         !PHYSICAL UNIT
C
C
          IF(SCRV4-MT) 8,5,8
5         MTORXT=0
          GØ TØ 15
8         IF(SCRVB-XT)' 12,10,12
10        MTORXT=1
          GØ TØ 15
12        PRINT 14
14        FØRMAT(1H0,'ERRØR IN FIRST DATA CARD---RUN ABØRTED')
          STØP
15        IF(DRIVE .LT. 0 .ØR.DRIVE .GT. 1)GØ TØ 12
C
C
          CALL TINIT(UNIT,MTORXT,DRIVE)
          CALL TATCH(UNIT)
          CALL TRWD(1)
          RETURN
C
C
          ENTRY HDRIN(EØF,TCKFLG)
C
          REC=0
20        CALL TREAD(UNIT,HEADER,1530)
          IF(S(1) .EQ. "366) GØ TØ 90
          IF(S(1).NE."374)GØ TØ 205
          REC=REC+1
          IF(REC.GE.20)GØ TØ 202
          CALL TSKIP(UNIT,-1)
          GØ TØ 20
202       CØNTINUE
          PRINT 4001
4001      FØRMAT(' **HEADER RECØRD PARITY ERRØR - TAPE FILE SKIPPED**')
          CALL TFILE(UNIT,1)
          GØ TØ 20
205       CØNTINUE
          IF(TCKFLG.NE.0)GØ TØ 23
          TCKFLG=1
          CALL ETOA(HEADBY(33),RTN,6)
          DØ 24 I=1,6
          IF(TAPENE(I).NE.RTN(I))GØ TØ 23
```

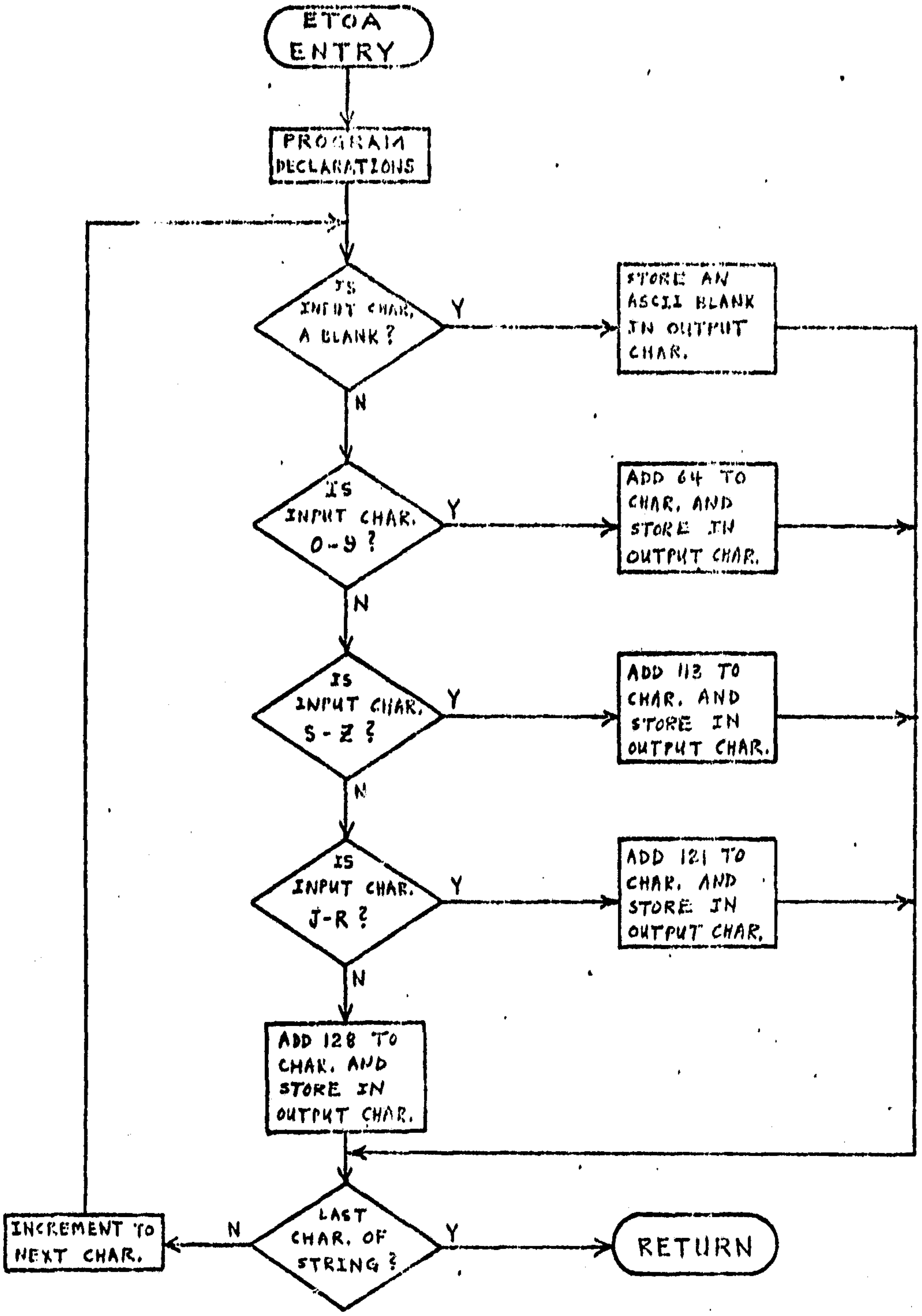
```
21      CØNTINUE
        GØ TØ 23
22      CØNTINUE
        PRINT 4000,TAPENØ,ITM
4000    FØRMAT('1*** WRØNG TAPE MØUNTED'/' TAPE REQUESTED = ',
     *   5A1/' TAPE MØUNTED = ',5A3/' RUN ABØRTED ***'/1H1)
        STØP    'WRØNG TAPE MØUNT'
23      CØNTINUE
        TEMP=HEADBY(67)
        HEADBY(67)=HEADBY(68)
        HEADBY(68)=TEMP
        SCRNUM=HEADER(34)
        DØ 30 I=1,5
        ACØDTE(I)=HEADBY(I+2248)+ETØAC  ! EBCDIC TØ ASCII CØNV.
30      CØNTINUE
        RETURN
        ENTRY DATØUT(PTC)
        CALL IDATE(MØ,DAY,YR)
        HEADBY(61)=DAY
        HEADBY(62)=MØ
        HEADBY(63)=YR
        DØ 35 I=1,32
        HEADBY(I)=TITL(I)
35      CØNTINUE
        DØ 36 I=1,5
        PTC(I)=0
36      CØNTINUE
        SNANGB(1)=HEADBY(2202)
        SNANGB(2)=HEADBY(2201)
        CALL SNAS(SUNANG)  ! PUT SUN ANGLE INTØ SCREENING RØUTINE.
        WRITE(2) HEADER
39      REC=0
40      CALL TREAD(UNIT,RECØRD,RLEN)
        IF(S(1).EQ."366)RETURN
        IF(S(1).NE."374)GØ TØ 45
        REC=REC+1
        IF(REC.GE.20)GØ TØ 42
        CALL TSKIP(UNIT,-1)
        GØ TØ 40
42      CØNTINUE
        PRINT 4002
4002    FØRMAT(' **DATA RECØRD PARITY ERRØR - NEXT LISTED FILE INVALID**')
        CALL TFILE(UNIT,1)
        RETURN
45      CØNTINUE
        DØ 70 I=1,196
        SCRVB(1)=HEADBY(I+72)
        SCRVB(3)=HEADBY(I+268)
        SCRVB(5)=HEADBY(I+464)
        SCRVB(7)=HEADBY(I+660)
        CALL SCREEN(SCRV,LPL)
        IF(LPL.EQ.1)GØ TØ 65
        SCRV(4)=SCRV(4)+128
        HEADBY(I+660)=SCRVB(7)
        IF(LPL.GT.10.AND.LPL.LT.14)GØ TØ 50
        IF(LPL.GT.20.AND.LPL.LT.27)GØ TØ 57
        IF(LPL.GT.30.AND.LPL.LT.35)GØ TØ 60
        PTC(2)=PTC(2)+1  ! GARBLED
        GØ TØ 70
50      CØNTINUE
        PTC(4)=PTC(4)+1  ! SHADØW
        GØ TØ 70
```

```
57      CONTINUE
        PTC(5)=PTC(5)+1   ! WATER
        GO TO 70
60      CONTINUE
        PTC(3)=PTC(3)+1   ! CLOUD
        GO TO 70
65      CONTINUE
        PTC(1)=PTC(1)+1   ! GRND
70      CONTINUE
        WRITE(2) (RECORD(I),I=1,RLEN)
        GO TO 39
C
90      EOF=1
        RETURN
        END
```

3.4.2.5 Subroutine ETOA

This subroutine converts an alphanumeric tape number from EBCDIC to ASCII characters.

3.4.2.5a Flowchart


ETOA ENTRY
PROGRAM DECLARATIONS
IS INPUT CHAR. A BLANK ?
Y
STORE AN ASCII BLANK IN OUTPUT CHAR.
N
IS INPUT CHAR. 0-9 ?
Y
ADD 64 TO CHAR. AND STORE IN OUTPUT CHAR.
N
IS INPUT CHAR. S-Z ?
Y
ADD 113 TO CHAR. AND STORE IN OUTPUT CHAR.
N
IS INPUT CHAR. J-R ?
Y
ADD 121 TO CHAR. AND STORE IN OUTPUT CHAR.
N
ADD 128 TO CHAR. AND STORE IN OUTPUT CHAR.
LAST CHAR. OF STRING ?
N
INCREMENT TO NEXT CHAR.
Y
RETURN

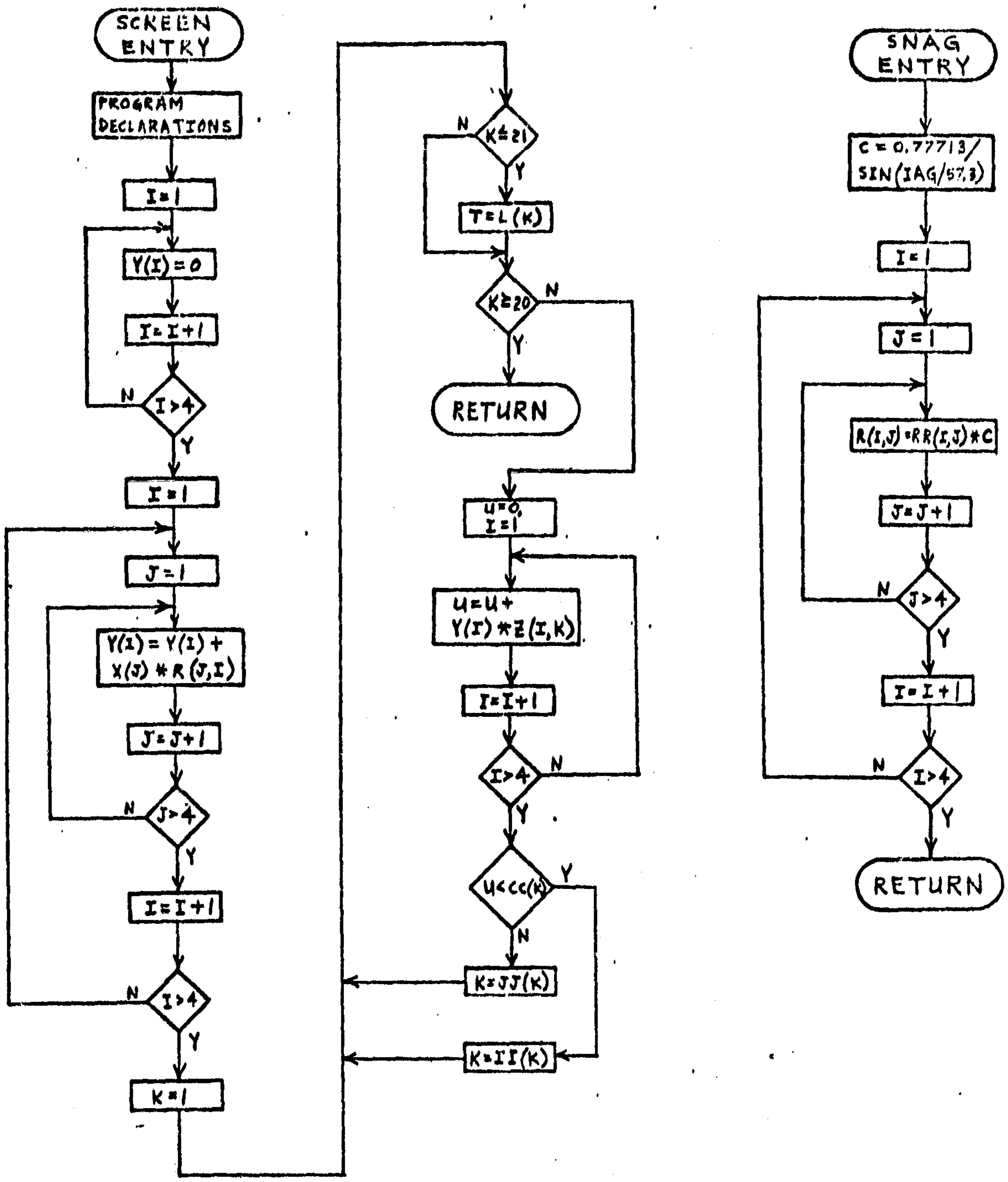
3.4.2.5b Listing

```
        SUBROUTINE [illegible]A(F,T,N)
C
C       THIS IS A SPECIAL ROUTINE TO ASCII [illegible] THAT ONLY
C       [illegible] BLANKS, DIGITS, AND UPPER CASE LETTERS.  IT IS
C       USED TO CONVERT THE INTERNAL TAPE NUMBER TO ASCII.
C
        LOGICAL*1 F(1),T(1),C1,C2,C3
        DATA C1,C2,C3/64,113,129/
        DO 80 I=1,N
        IF(F(I).NE."100)GO TO 10
        T(I)=' '
        GO TO 80
10      CONTINUE
        IF(F(I)+16)30,20,20
20      CONTINUE
        T(I)=F(I)+C1
        GO TO 80
30      CONTINUE
        IF(F(I)+53)50,40,40
40      CONTINUE
        T(I)=F(I)+C2
        GO TO 80
50      CONTINUE
        IF(F(I)+47)70,60,60
60      CONTINUE
        T(I)=F(I)+C3
        GO TO 80
70      CONTINUE
        T(I)=F(I)+C1+C1
80      CONTINUE  ! LOOP END
        RETURN
        END
```

3.4.2.6 Subroutine SCREEN

This subroutine, using the sun angle and the four channel values for a pixel, makes a series of tests and decides whether the pixel is good, garbled, cloud, shadow, or water.

3.4.2.6a Flowchart



3.4.2.6b Listing

```
C     SUBROUTINE SCREEN(X,T) SCREENS DATA X AND RETURNS LABEL T
C
C    COMPILE WITH THE CØ150 SWITCH.
C
      SUBROUTINE SCREEN(X,T)
      REAL R(4,4),Z(4,19),RR(4,4),Y(4),CC(19)
      INTEGER II(19),JJ(19),L(21), (4),T
      DATA RR/.33234,.60316,.675 1,.26278,
    C        -.28317,-.66006,.57732,.38633,
    C        -.89952,.42530,.17572,-.04080,
    C        -.01594,.13168,-.45157,.83232/
      DATA Z/0,0,0,0,0,0,1,0,
    C    0,0,0,0,0,0,-1,0,
    C    -.09375,0,0,1,0,0,0,
    C    -.1875,0,0,-1,0,0,0,
    C    -0.1,-1,0,0,0,0,0,
    C   0.55556,1,0,0,0,0,0,
    C    -0.8333333,1,0,0,0,0,0,
    C    1,0,0,0,0,0,0,0,
    C    -.1,0,0,-1,0,0,0,
    C     1,0,0,0,0,0,0,0,
    C    -0.142857,0,0,-1,0,0,0,
    C    -1,0,0,0,0,0,0,0,
    C      -.0675,-1,0,0,0,0,0,
    C       0,0,0,0,0,0,-1,0,
    C      0,0,-1,0,0,0,-1,0,
    C     -.5,-1,0,-1,0,-.5,
    C       -.4,1,0,0,0,0,0,
    C    -0.4,1,0,-0.6,-0.6,
    C       -1,0,0.4,0,0,0,0/
      DATA  CC/15.,12.,-4.,14.,20.,156.,-8.,100.,7.5,69.,3.25,
    C -75.,0.5,-1.5,4.5,-10.,-12., ,-9.,-37.75/
      DATA  II/2,3,4,5,6,7,8,10,10,12,12,18,18,18,18,18,60,20,20/
      DATA  JJ/60,60,60,60,60,60,60,9,60,11,60,13,14,15,16,17,21,
    C   19,60/
      DATA L/47,46,45,44,43,42,41,34,33,32,31,26,25,24,23,22,21,
    *      13,12,1,11/
C       4* GARBLED; 34,33 CLOUD; 32,31 HAZE
C       2* WATER; 1* SHADOW, 11 OVER WATER; 1 GOOD.
C
       DØ 5 I=1,4
 5     Y(I)=0
       DØ 6 I=1,4
       DØ 6 J=1,4
 6     Y(I)=Y(I)+X(J)*R(J,I)
       K=1
  1    CONTINUE
       IF(K.LE.21)T=L(K)
       IF(K .GE. 20) RETURN
       U=0.
       DØ 7 I=1,4
  7    U=U+Y(I)*Z(I,K)
       IF(U .LT. CC(K)) GØ TØ 3
       K=JJ(K)
       GØ TØ 1
  3    CONTINUE
       K=II(K)
       GØ TØ 1
       ENTRY SNAG(IAG)
       C=0.77713/SIN(IAG/57.3)
       DØ 8 I=1,4
      DO 8 J=1,4                                                        SCR00570
    8 R(I,J)=RR(I,J)*C                                                  SCR00580
      RETURN                                                            SCR00590
      END                                                               SCR006
```

3.4.3 FIELDSTATS MAIN PROGRAM (FLDSTT)

3.4.3.1 Linkage

This program calls the special subroutines GAMGEN, SCREEN, GETCCR, FASRPT, FGPRPT, and FCCSRP in addition to system subroutines.

3.4.3.2 Interface

GAMGEN, SCREEN, GETCCR, FASRPT, FGPRPT, FCCSRP.

3.4.3.3 Input

A screened image disk file, a ground truth disk file, a crop code translations disk file, and a program parameters disk file.

3.4.3.4 Output

A dot pixel statistics disk file, a pure pixel statistics disk file, a mixed pixel statistics disk file, a subpixel statistics disk file, an acquisition summary report, an optional dot pixel statistics report, and an optional crop code summary report.

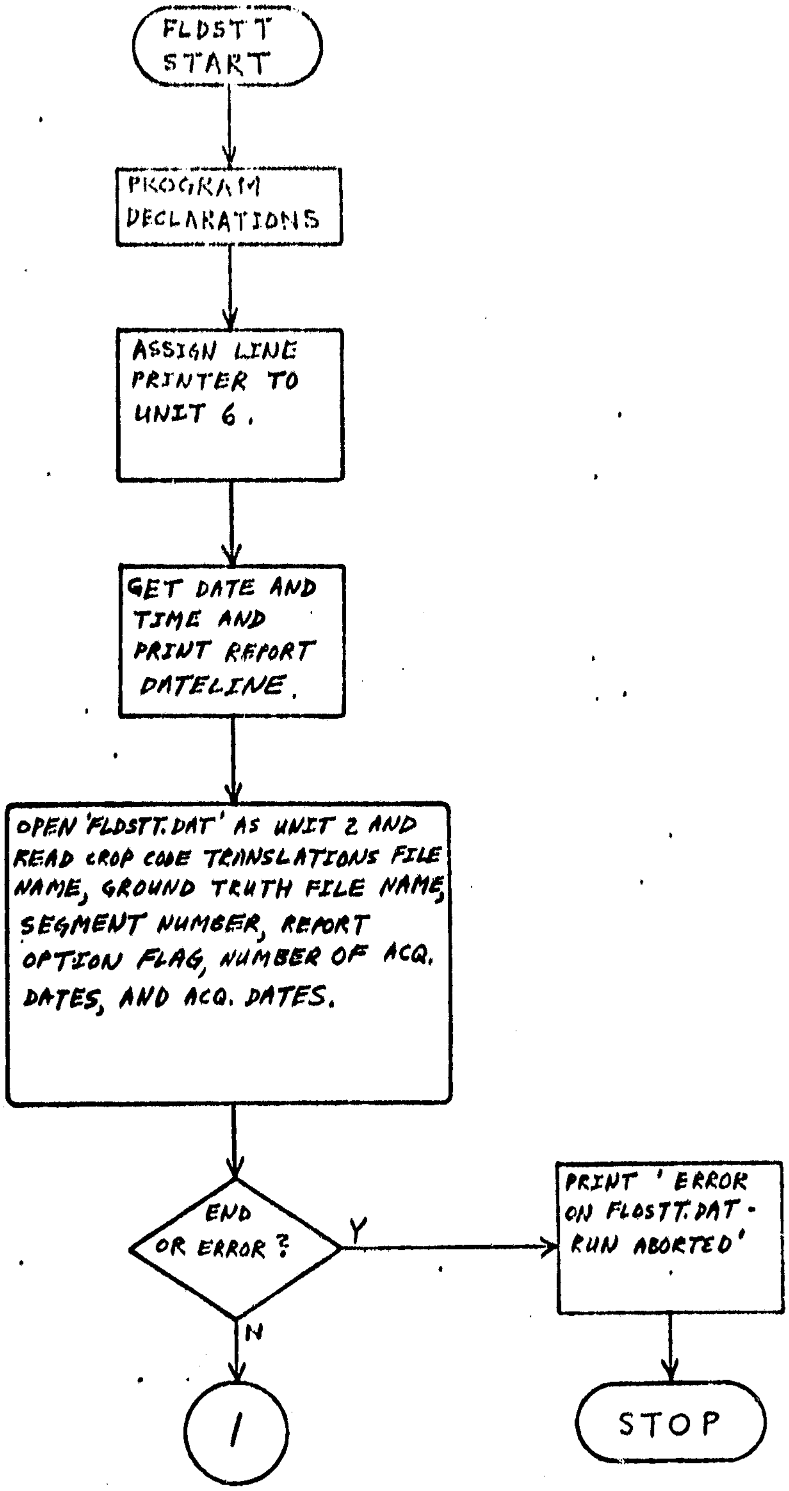
3.4.3.5 Storage

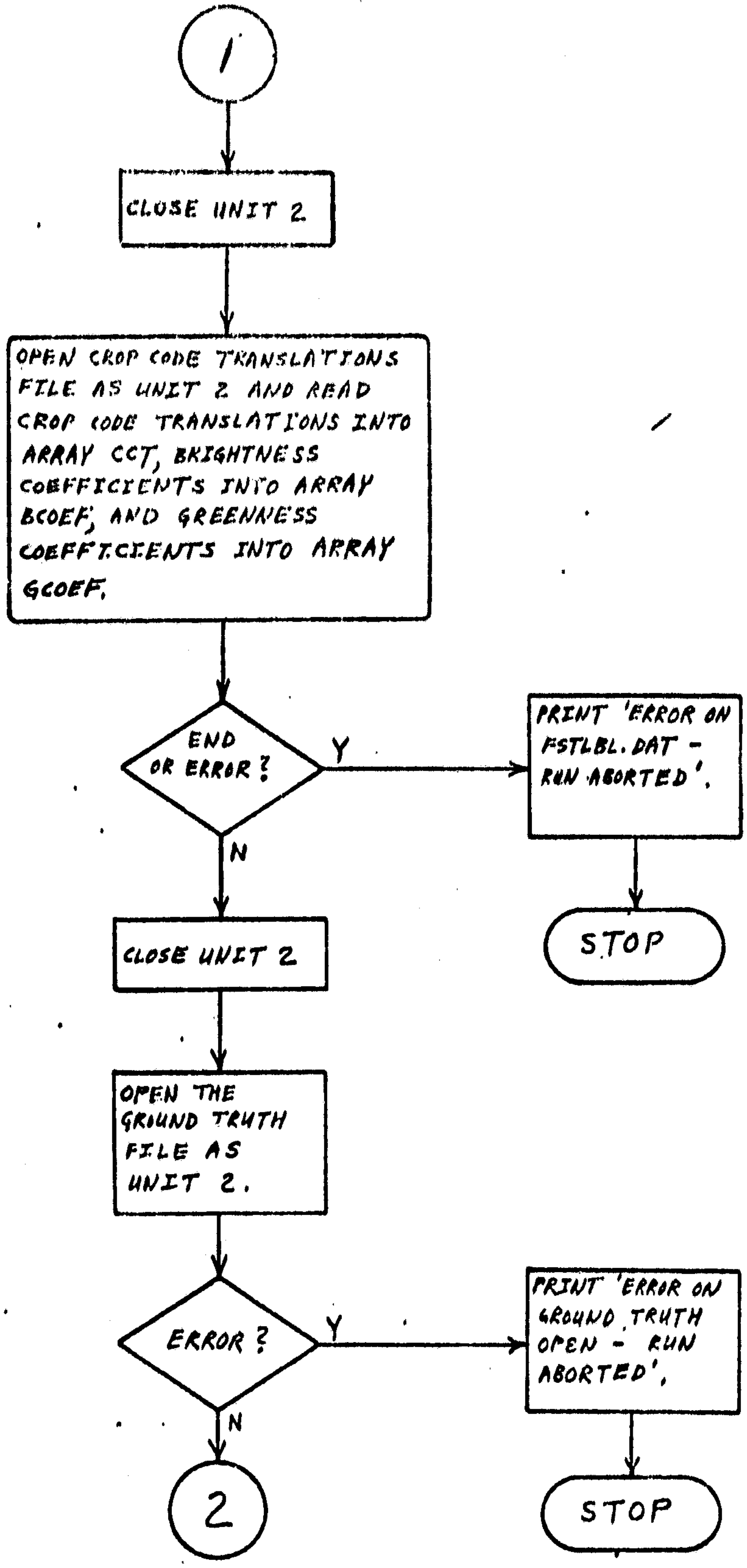
25,360 words.

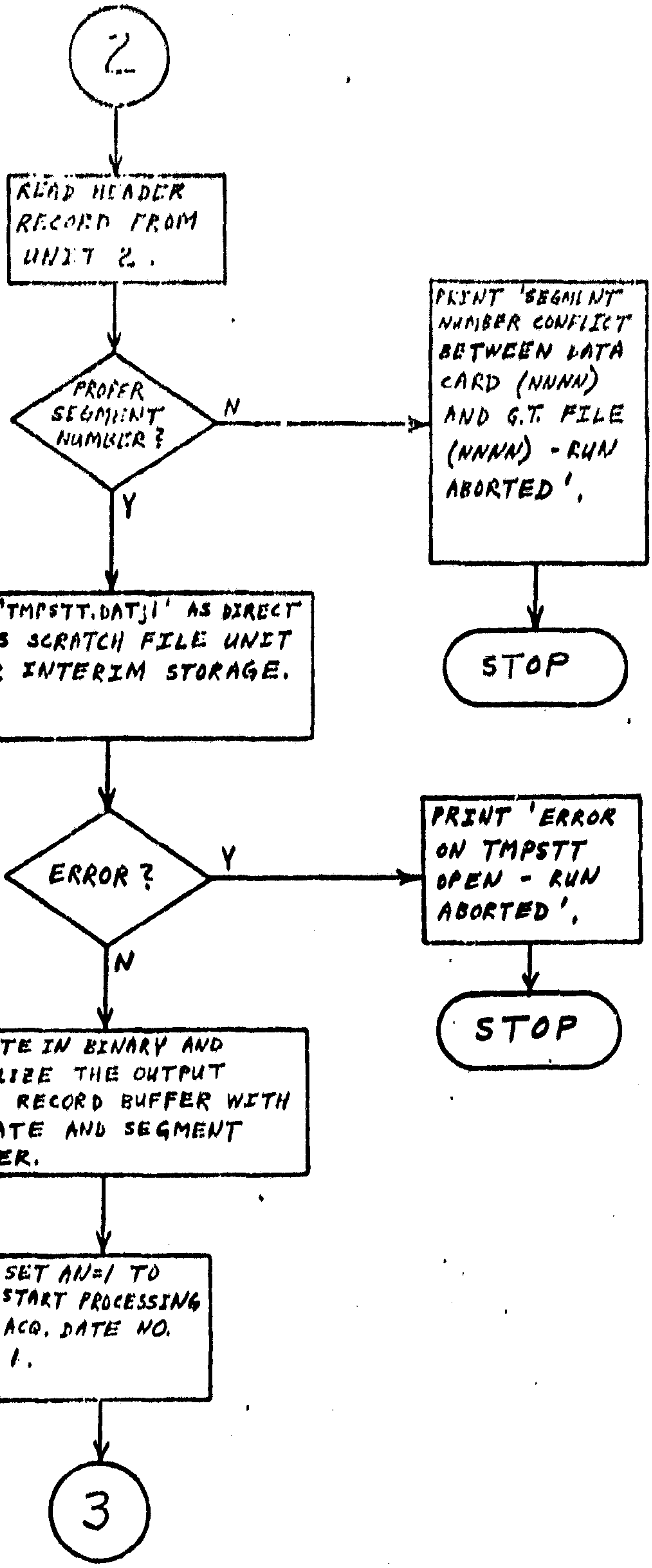
3.4.3.6 Description

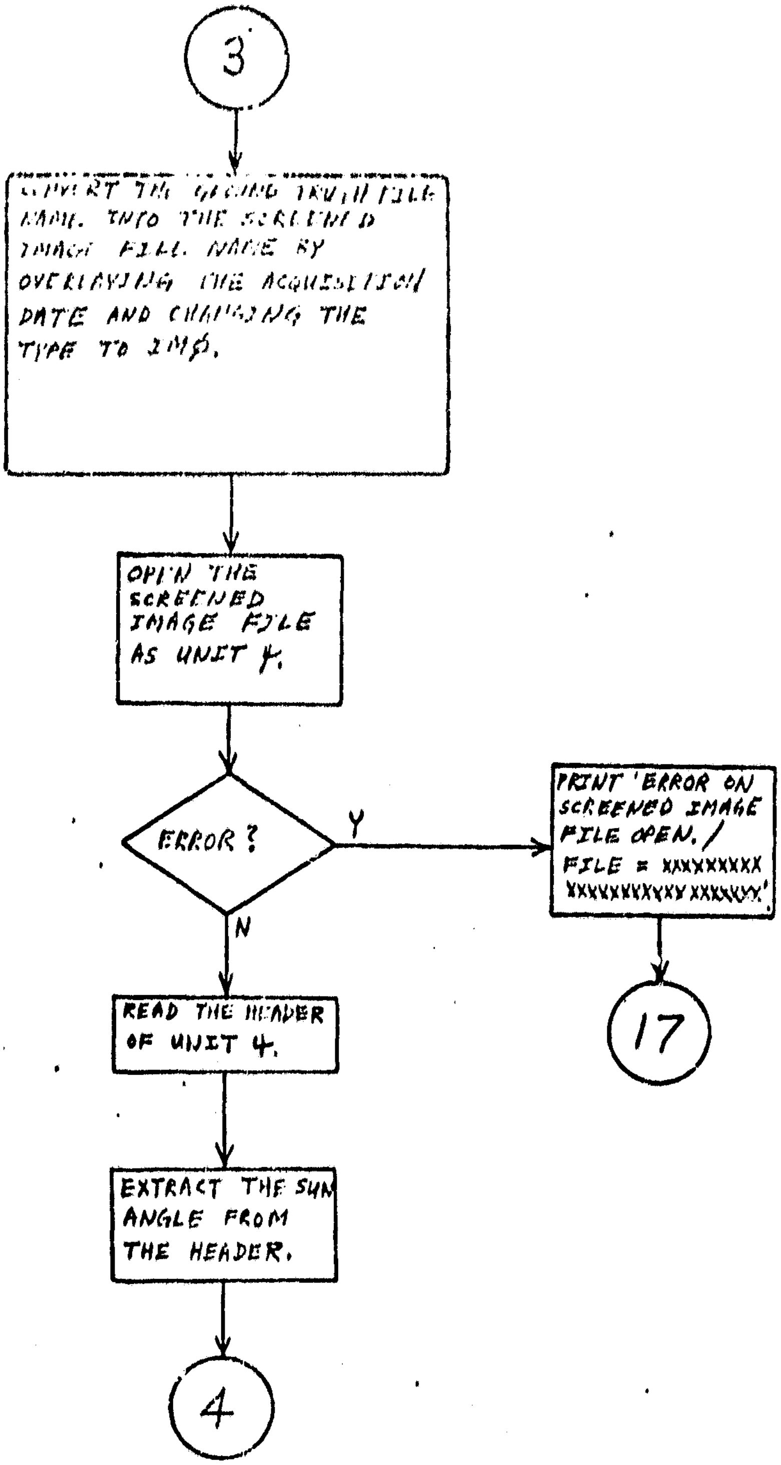
The FLDSTT program matches image unload data and ground truth data, calculates means, standard deviations, cross-channel correlations, and brightness and greenness histograms, stores these statistics on disk files, and produces reports on them.

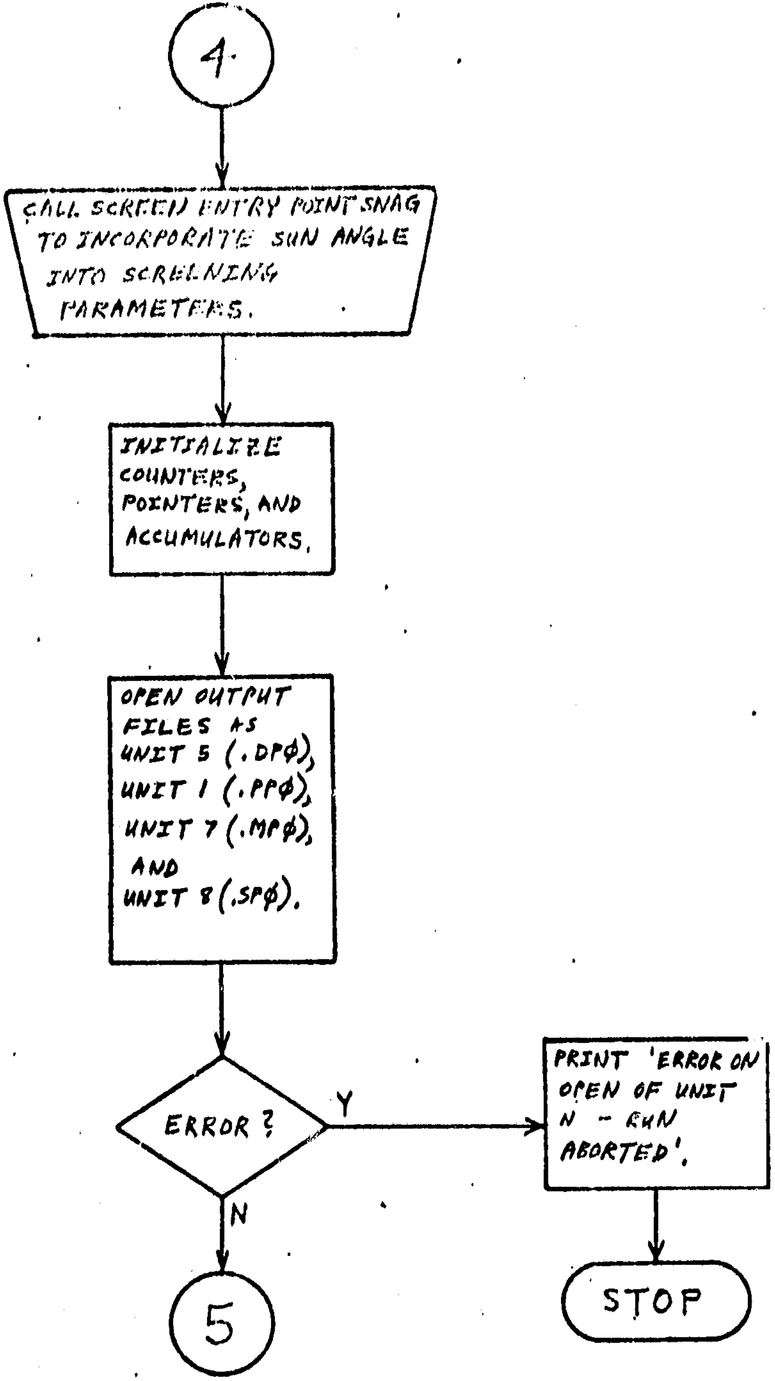
3.4.3.7 Flowchart

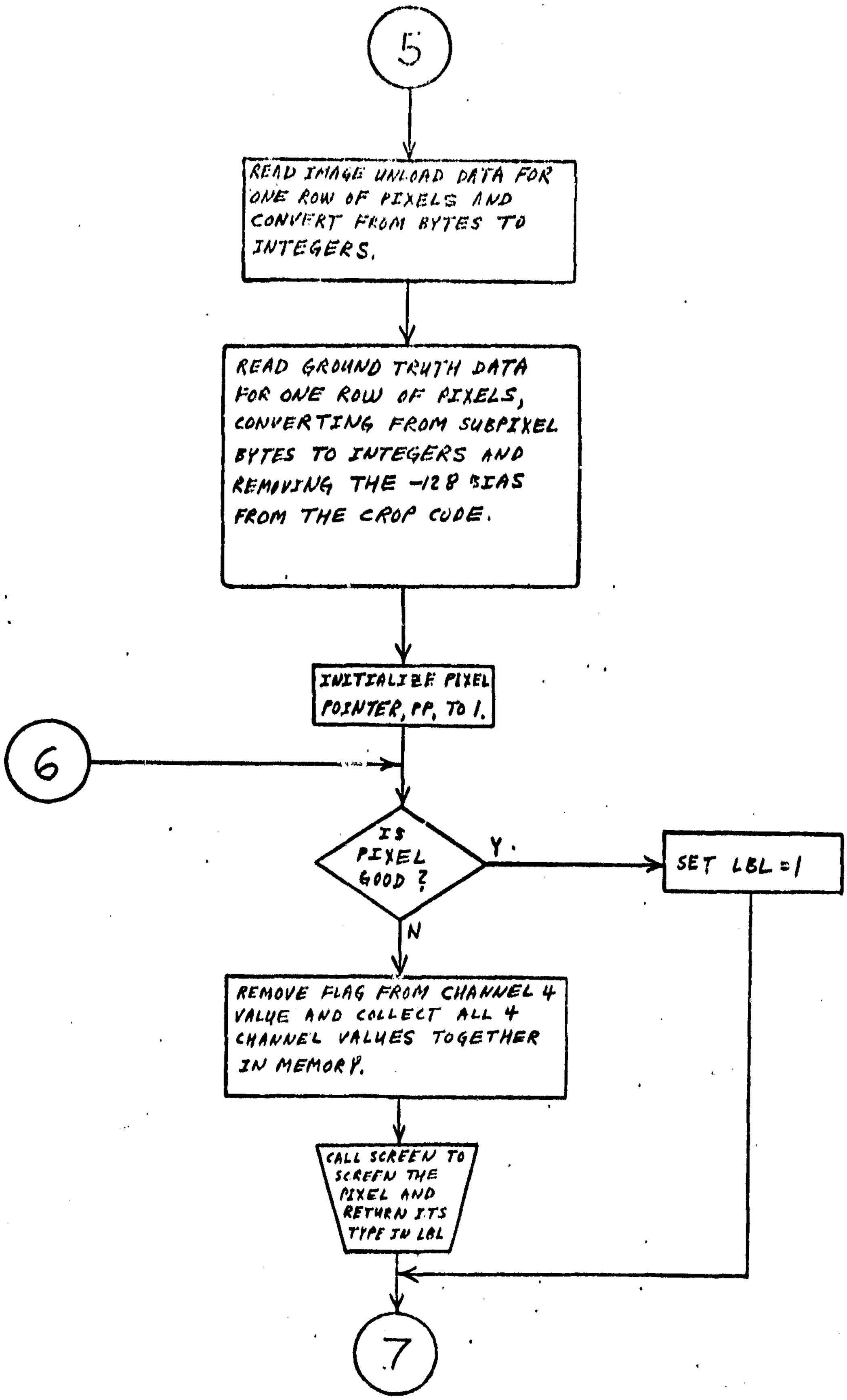

FLDSTT START
PROGRAM DECLARATIONS
ASSIGN LINE PRINTER TO UNIT 6.
GET DATE AND TIME AND PRINT REPORT DATELINE.
OPEN 'FLDSTT.DAT' AS UNIT 2 AND READ CROP CODE TRANSLATIONS FILE NAME, GROUND TRUTH FILE NAME, SEGMENT NUMBER, REPORT OPTION FLAG, NUMBER OF ACQ. DATES, AND ACQ. DATES.
END OR ERROR ?
Y
PRINT 'ERROR ON FLDSTT.DAT - RUN ABORTED'
STOP
N
1

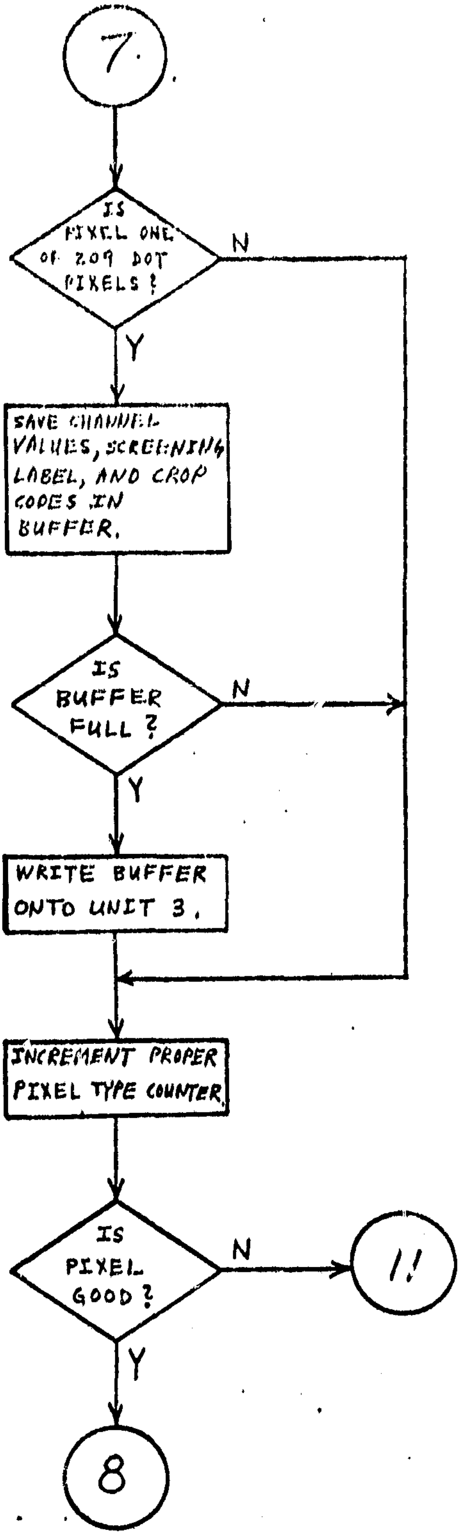

1
CLOSE UNIT 2
OPEN CROP CODE TRANSLATIONS FILE AS UNIT 2 AND READ CROP CODE TRANSLATIONS INTO ARRAY CCT, BRIGHTNESS COEFFICIENTS INTO ARRAY BCOEF, AND GREENNESS COEFFICIENTS INTO ARRAY GCOEF.
END OR ERROR ?
Y
PRINT 'ERROR ON FSTLBL.DAT - RUN ABORTED'.
STOP
N
CLOSE UNIT 2
OPEN THE GROUND TRUTH FILE AS UNIT 2.
ERROR ?
Y
PRINT 'ERROR ON GROUND TRUTH OPEN - RUN ABORTED'.
STOP
N
2

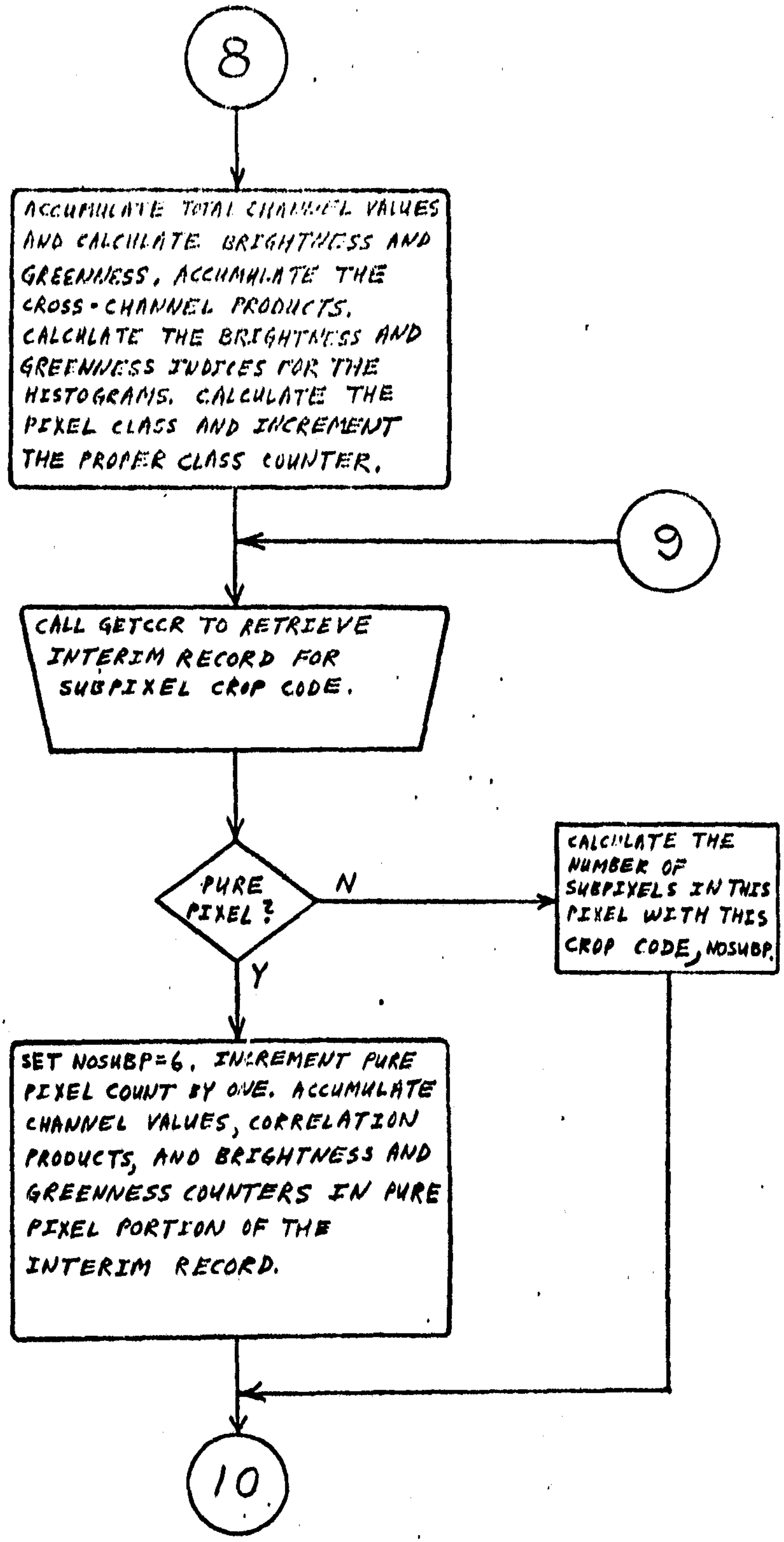

2
READ HEADER RECORD FROM UNIT 2.
PROPER SEGMENT NUMBER?
N
PRINT 'SEGMENT NUMBER CONFLICT BETWEEN DATA CARD (NNNN) AND G.T. FILE (NNNN) - RUN ABORTED'.
STOP
Y
OPEN 'TMPSTT.DAT;1' AS DIRECT ACCESS SCRATCH FILE UNIT 3 FOR INTERIM STORAGE.
ERROR?
Y
PRINT 'ERROR ON TMPSTT OPEN - RUN ABORTED'.
STOP
N
GET DATE IN BINARY AND INITIALIZE THE OUTPUT HEADER RECORD BUFFER WITH THE DATE AND SEGMENT NUMBER.
SET AN=1 TO START PROCESSING ACQ. DATE NO. 1.
3

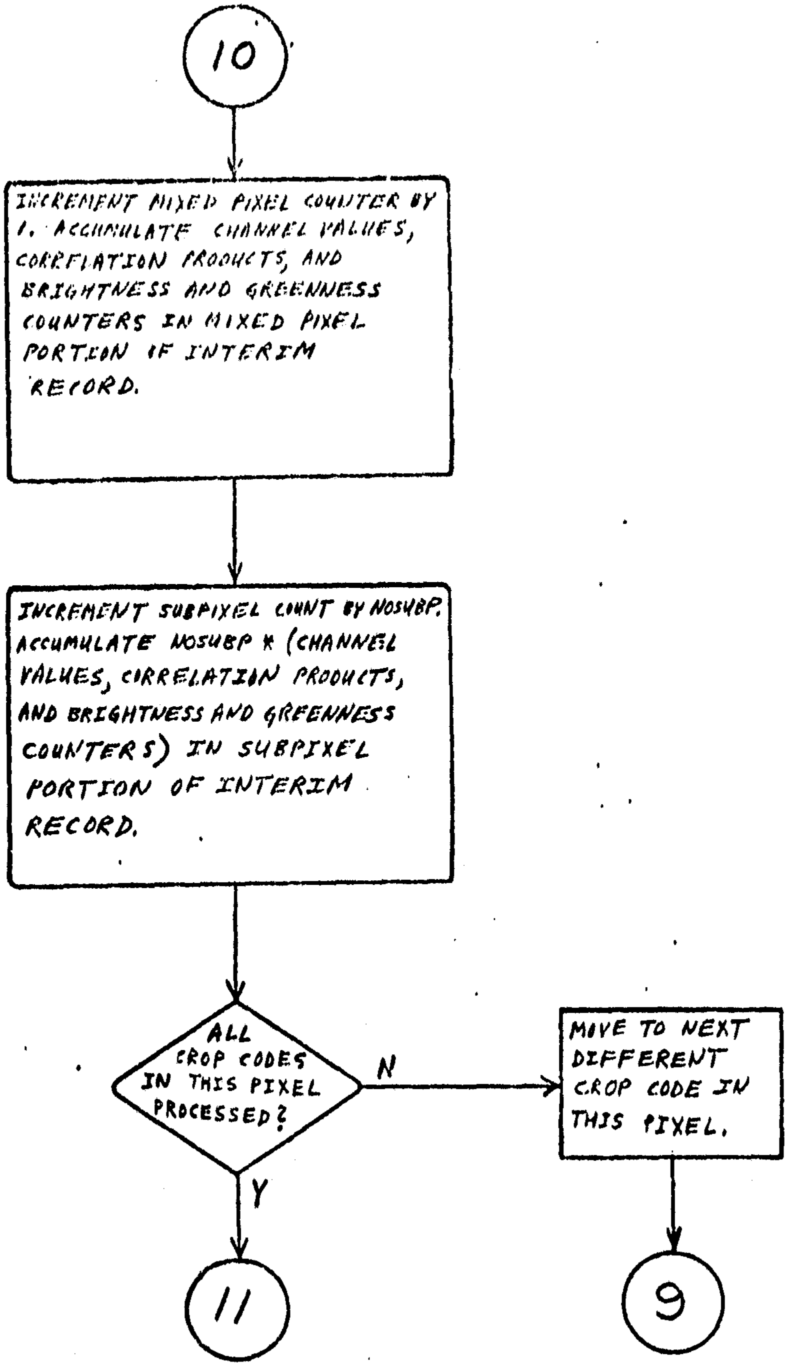

3
CONVERT THE GROUND TRUTH FILE NAME INTO THE SCREENED IMAGE FILE NAME BY OVERLAYING THE ACQUISITION DATE AND CHANGING THE TYPE TO IMØ.
OPEN THE SCREENED IMAGE FILE AS UNIT 4.
ERROR?
Y
PRINT 'ERROR ON SCREENED IMAGE FILE OPEN./ FILE = XXXXXXXXX XXXXXXXXXXX XXXXXXX'
17
N
READ THE HEADER OF UNIT 4.
EXTRACT THE SUN ANGLE FROM THE HEADER.
4

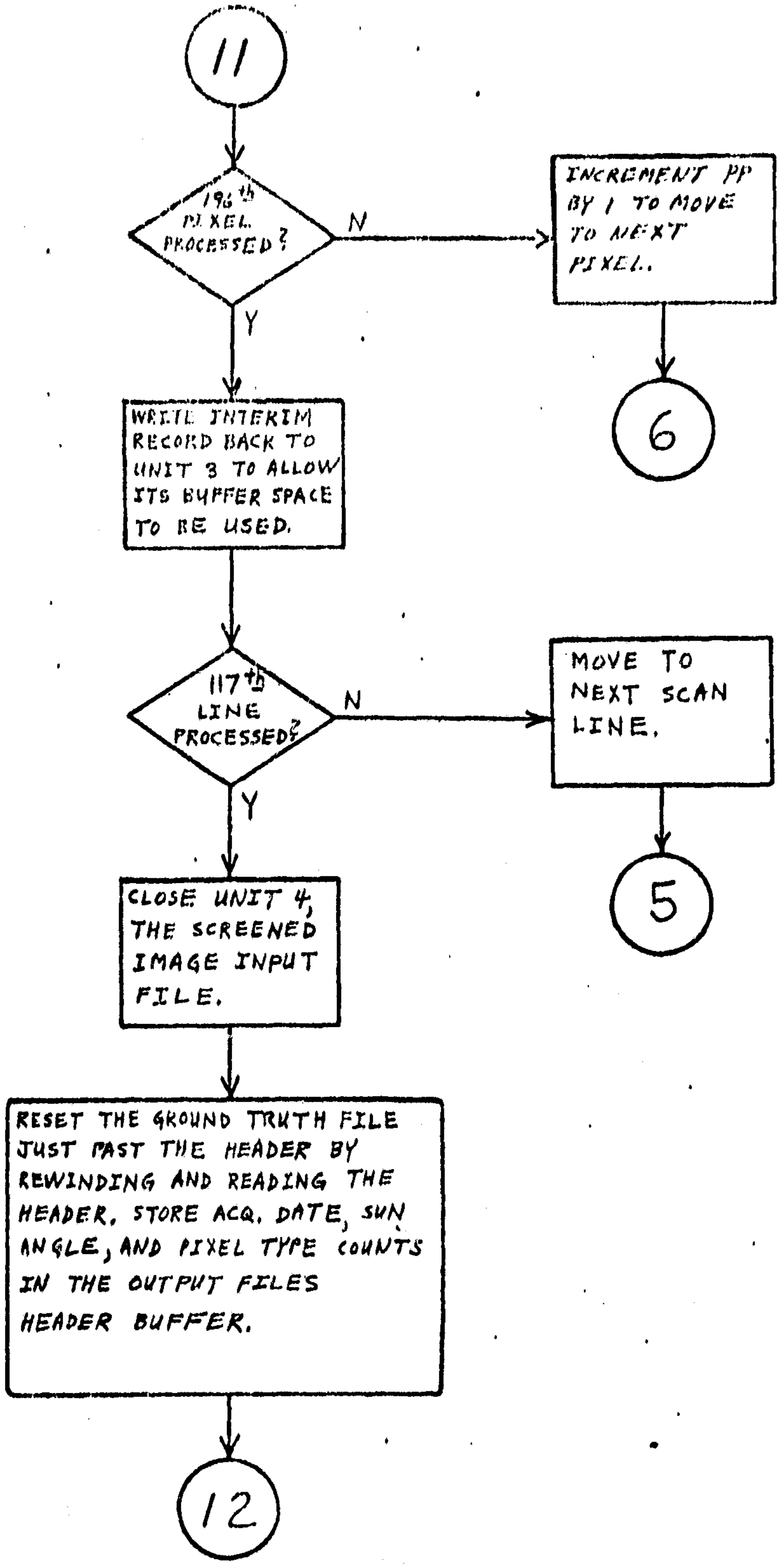

4
CALL SCREEN ENTRY POINT SNAG TO INCORPORATE SUN ANGLE INTO SCREENING PARAMETERS.
INITIALIZE COUNTERS, POINTERS, AND ACCUMULATORS.
OPEN OUTPUT FILES AS UNIT 5 (.DPØ), UNIT 1 (.PPØ), UNIT 7 (.MPØ), AND UNIT 8 (.SPØ).
ERROR ?
Y
PRINT 'ERROR ON OPEN OF UNIT N - RUN ABORTED'.
N
5
STOP

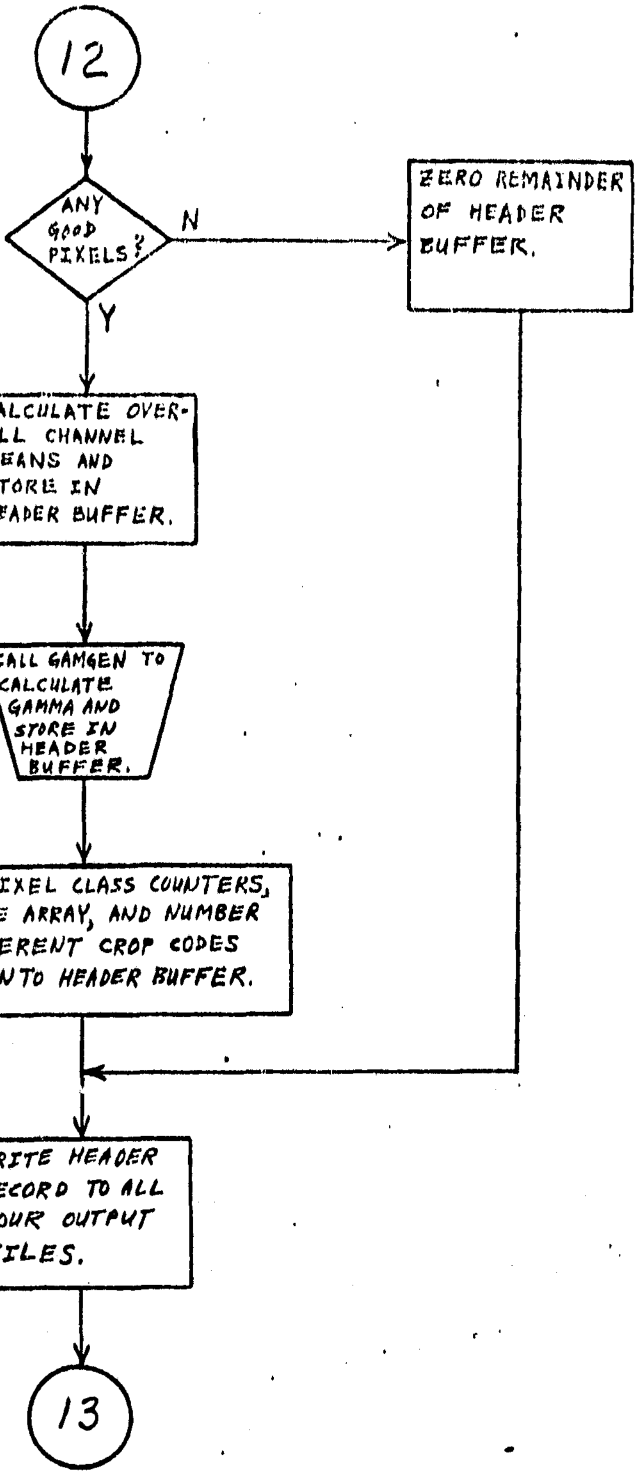

5
READ IMAGE UNLOAD DATA FOR ONE ROW OF PIXELS AND CONVERT FROM BYTES TO INTEGERS.
READ GROUND TRUTH DATA FOR ONE ROW OF PIXELS, CONVERTING FROM SUBPIXEL BYTES TO INTEGERS AND REMOVING THE -128 BIAS FROM THE CROP CODE.
INITIALIZE PIXEL POINTER, PP, TO 1.
6
IS PIXEL GOOD ?
Y.
SET LBL = 1
N
REMOVE FLAG FROM CHANNEL 4 VALUE AND COLLECT ALL 4 CHANNEL VALUES TOGETHER IN MEMORY.
CALL SCREEN TO SCREEN THE PIXEL AND RETURN ITS TYPE IN LBL
7

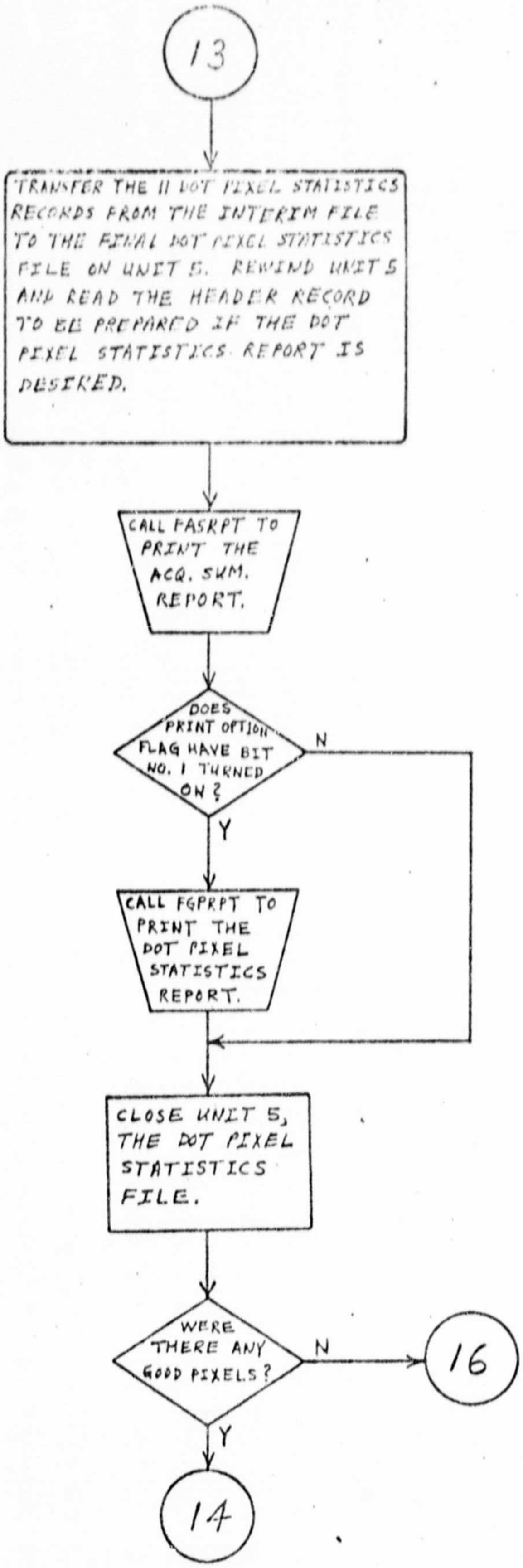

7.
IS PIXEL ONE OF 209 DOT PIXELS?
N
Y
SAVE CHANNEL VALUES, SCREENING LABEL, AND CROP CODES IN BUFFER.
IS BUFFER FULL?
N
Y
WRITE BUFFER ONTO UNIT 3.
INCREMENT PROPER PIXEL TYPE COUNTER.
IS PIXEL GOOD?
N
11
Y
8


8
ACCUMULATE TOTAL CHANNEL VALUES AND CALCULATE BRIGHTNESS AND GREENNESS, ACCUMULATE THE CROSS-CHANNEL PRODUCTS. CALCULATE THE BRIGHTNESS AND GREENNESS INDICES FOR THE HISTOGRAMS. CALCULATE THE PIXEL CLASS AND INCREMENT THE PROPER CLASS COUNTER.
9
CALL GETCCR TO RETRIEVE INTERIM RECORD FOR SUBPIXEL CROP CODE.
PURE PIXEL?
N
CALCULATE THE NUMBER OF SUBPIXELS IN THIS PIXEL WITH THIS CROP CODE, NOSUBP.
Y
SET NOSUBP=6. INCREMENT PURE PIXEL COUNT BY ONE. ACCUMULATE CHANNEL VALUES, CORRELATION PRODUCTS, AND BRIGHTNESS AND GREENNESS COUNTERS IN PURE PIXEL PORTION OF THE INTERIM RECORD.
10

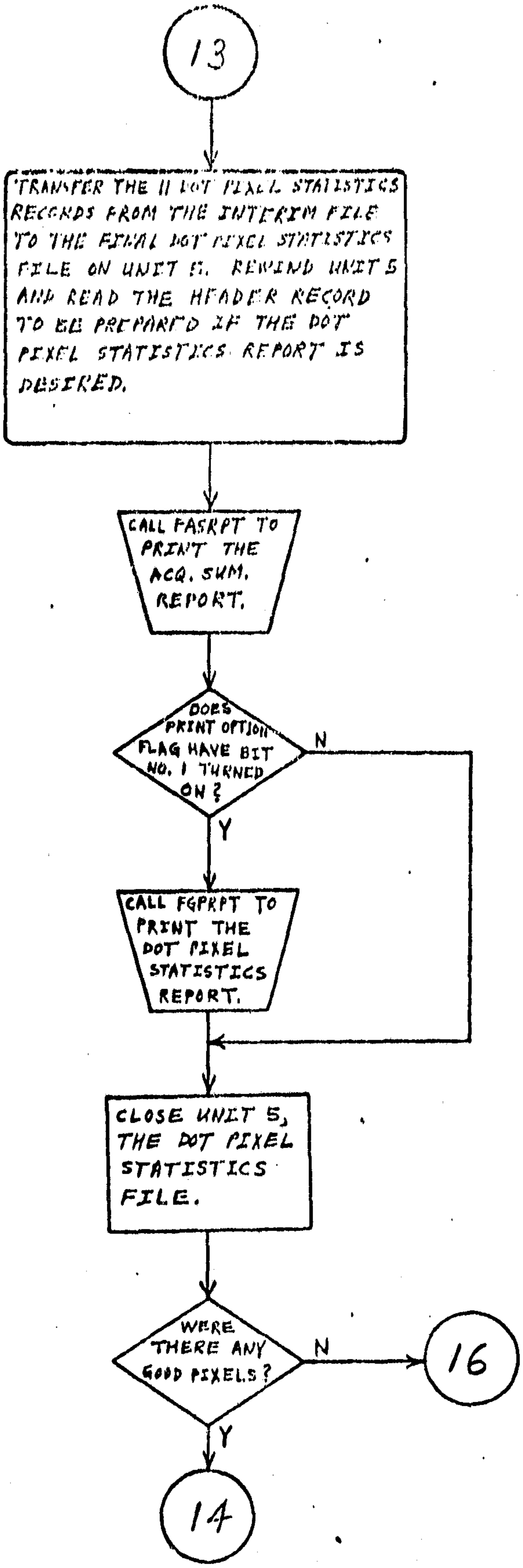

10
INCREMENT MIXED PIXEL COUNTER BY 1. ACCUMULATE CHANNEL VALUES, CORRELATION PRODUCTS, AND BRIGHTNESS AND GREENNESS COUNTERS IN MIXED PIXEL PORTION OF INTERIM RECORD.
INCREMENT SUBPIXEL COUNT BY NOSUBP. ACCUMULATE NOSUBP * (CHANNEL VALUES, CORRELATION PRODUCTS, AND BRIGHTNESS AND GREENNESS COUNTERS) IN SUBPIXEL PORTION OF INTERIM RECORD.
ALL CROP CODES IN THIS PIXEL PROCESSED?
N
MOVE TO NEXT DIFFERENT CROP CODE IN THIS PIXEL.
Y
11
9

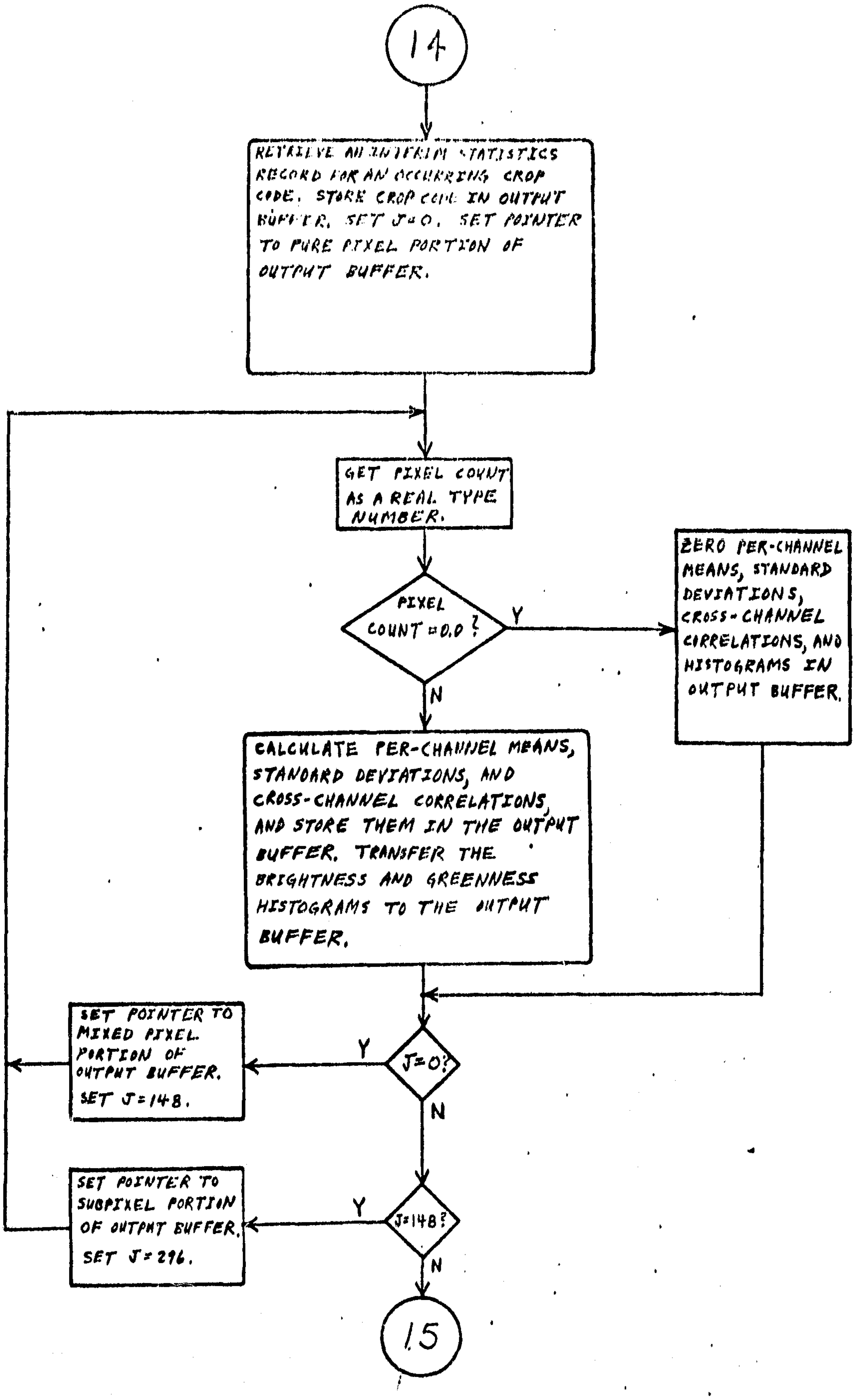

11
196th PIXEL PROCESSED?
N
INCREMENT PP BY 1 TO MOVE TO NEXT PIXEL.
6
Y
WRITE INTERIM RECORD BACK TO UNIT 3 TO ALLOW ITS BUFFER SPACE TO BE USED.
117th LINE PROCESSED?
N
MOVE TO NEXT SCAN LINE.
5
Y
CLOSE UNIT 4, THE SCREENED IMAGE INPUT FILE.
RESET THE GROUND TRUTH FILE JUST PAST THE HEADER BY REWINDING AND READING THE HEADER. STORE ACQ. DATE, SUN ANGLE, AND PIXEL TYPE COUNTS IN THE OUTPUT FILES HEADER BUFFER.
12

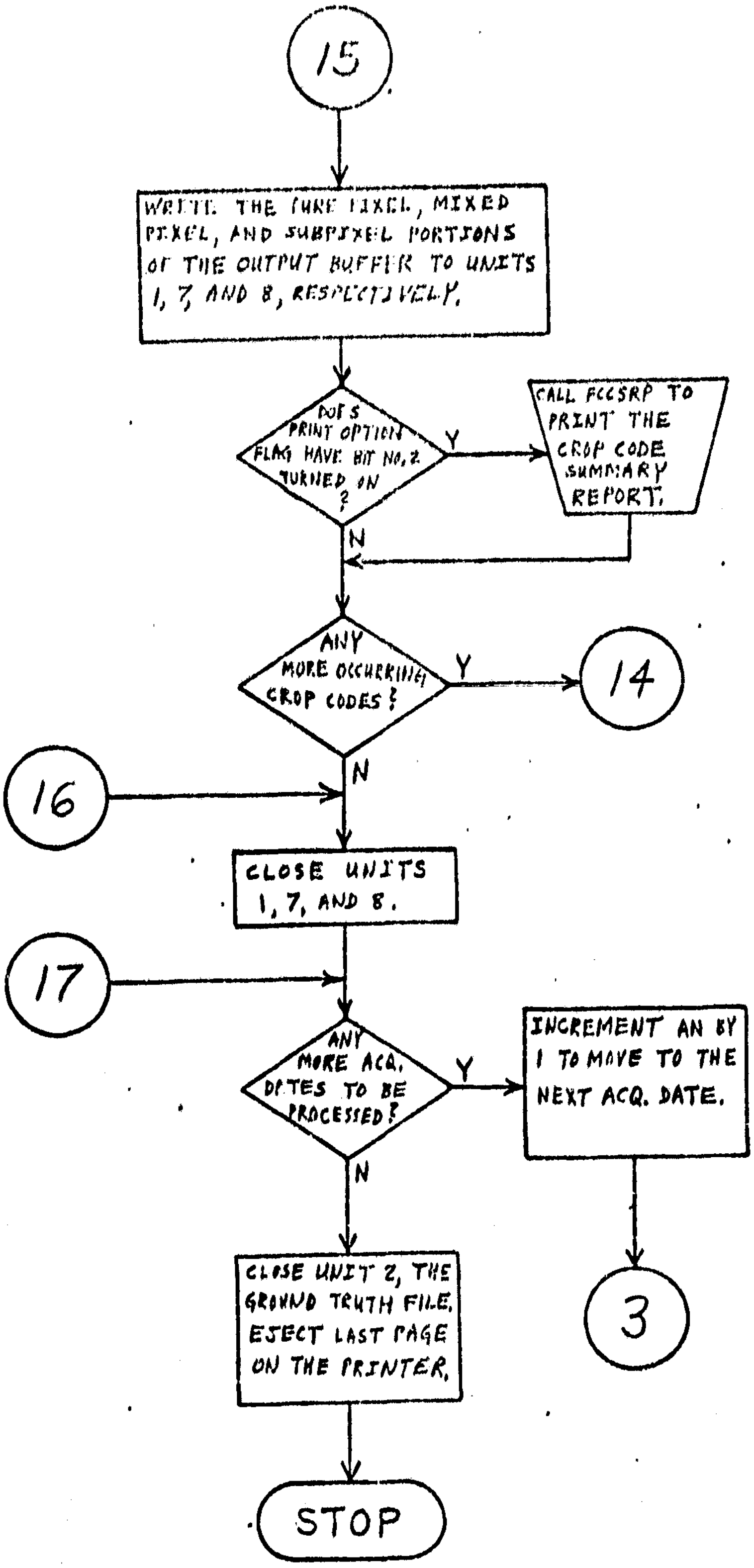

12
ANY GOOD PIXELS?
N
ZERO REMAINDER OF HEADER BUFFER.
Y
CALCULATE OVERALL CHANNEL MEANS AND STORE IN HEADER BUFFER.
CALL GAMGEN TO CALCULATE GAMMA AND STORE IN HEADER BUFFER.
STORE PIXEL CLASS COUNTERS, CROP CODE ARRAY, AND NUMBER OF DIFFERENT CROP CODES FOUND INTO HEADER BUFFER.
WRITE HEADER RECORD TO ALL FOUR OUTPUT FILES.
13



13
TRANSFER THE 11 DOT PIXEL STATISTICS RECORDS FROM THE INTERIM FILE TO THE FINAL DOT PIXEL STATISTICS FILE ON UNIT 5. REWIND UNIT 5 AND READ THE HEADER RECORD TO BE PREPARED IF THE DOT PIXEL STATISTICS REPORT IS DESIRED.
CALL FASRPT TO PRINT THE ACQ. SUM. REPORT.
DOES PRINT OPTION FLAG HAVE BIT NO. 1 TURNED ON?
N
Y
CALL FGPRPT TO PRINT THE DOT PIXEL STATISTICS REPORT.
CLOSE UNIT 5, THE DOT PIXEL STATISTICS FILE.
WERE THERE ANY GOOD PIXELS?
N
16
Y
14


13
TRANSFER THE 11 DOT PIXEL STATISTICS RECORDS FROM THE INTERIM FILE TO THE FINAL DOT PIXEL STATISTICS FILE ON UNIT 5. REWIND UNIT 5 AND READ THE HEADER RECORD TO BE PREPARED IF THE DOT PIXEL STATISTICS REPORT IS DESIRED.
CALL FASRPT TO PRINT THE ACQ. SUM. REPORT.
DOES PRINT OPTION FLAG HAVE BIT NO. 1 TURNED ON ?
N
Y
CALL FGPRPT TO PRINT THE DOT PIXEL STATISTICS REPORT.
CLOSE UNIT 5, THE DOT PIXEL STATISTICS FILE.
WERE THERE ANY GOOD PIXELS ?
N
16
Y
14


14
RETRIEVE AN INTERIM STATISTICS RECORD FOR AN OCCURRING CROP CODE. STORE CROP CODE IN OUTPUT BUFFER. SET J=0. SET POINTER TO PURE PIXEL PORTION OF OUTPUT BUFFER.
GET PIXEL COUNT AS A REAL TYPE NUMBER.
PIXEL COUNT = 0.0?
Y
ZERO PER-CHANNEL MEANS, STANDARD DEVIATIONS, CROSS-CHANNEL CORRELATIONS, AND HISTOGRAMS IN OUTPUT BUFFER.
N
CALCULATE PER-CHANNEL MEANS, STANDARD DEVIATIONS, AND CROSS-CHANNEL CORRELATIONS, AND STORE THEM IN THE OUTPUT BUFFER. TRANSFER THE BRIGHTNESS AND GREENNESS HISTOGRAMS TO THE OUTPUT BUFFER.
J=0?
Y
SET POINTER TO MIXED PIXEL PORTION OF OUTPUT BUFFER. SET J=148.
N
J=148?
Y
SET POINTER TO SUBPIXEL PORTION OF OUTPUT BUFFER. SET J=296.
N
15


15
WRITE THE PURE PIXEL, MIXED PIXEL, AND SUBPIXEL PORTIONS OF THE OUTPUT BUFFER TO UNITS 1, 7, AND 8, RESPECTIVELY.
DOES PRINT OPTION FLAG HAVE BIT NO. 2 TURNED ON ?
Y
CALL FCCSRP TO PRINT THE CROP CODE SUMMARY REPORT.
N
ANY MORE OCCURRING CROP CODES ?
Y
14
N
16
CLOSE UNITS 1, 7, AND 8.
17
ANY MORE ACQ. DATES TO BE PROCESSED ?
Y
INCREMENT AN BY 1 TO MOVE TO THE NEXT ACQ. DATE.
3
N
CLOSE UNIT 2, THE GROUND TRUTH FILE. EJECT LAST PAGE ON THE PRINTER.
STOP

3.4.3.8 Listing

```
      PROGRAM FLDSTT
C
C        THIS PROGRAM CALCULATES, BY CROP (I), THE PROCHANNEL
C        MEANS, THE PROCHANNEL STANDARD DEVIATIONS, THE CROSSCHANNEL
C        CORRELATIONS, A BRIGHTNESS HISTOGRAM, AND A GREENNESS HISTOGRAM
C        FOR A PARTICULAR LACIE SEGMENT USING A LIST FROM A GROUND TRUTH
C        DISK FILE AND A SUPERVISED LABEL DATA DISK FILE.  IT ALSO
C        PRODUCES STATISTICS ON THE 209 DOTS IN FILE TYPE .PTD AND
C        STATISTICS OF PURE, MIXED, AND SUB PIXELS IN FILES TYPES
C        .PPD, .MPD, AND .SPD.
C
C
C        PROGRAM DECLARATIONS.
C
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSION BUF(1530),CTFLAG(16),PRC(6),CHV(196,4),SPCC(196,6),
     #          DOTBUF(270),COSF(6),TSREC(3140),PPRCNT(151),PPGCNT(116),
     *          MPRCNT(151),MPGCNT(116),FSREC(1160),II(6),IJ(6),IK(6),
     *          SCOV(4),CBUF(270),IBUF(450),UNITNO(4),FHDR(360),
     *          PPSTT(296),MPSTT(296),SPSTT(574),ZEROES(333),SUBZ(65)
      DIMENSION FLPLOT(16)
      BYTE HDR(3060),CHVB(1568),CBUFB(540),TITL(32),
     *     TSRECB(2320),IUSEGB(2),INTB(2),SNANGB(2),PDATE(9),
     *     PTIME(8),TBUF  (901),FTYPE(2,4),CTFLAB(32)
      INTEGER*4 ACC(15),TCHA(4),CPA(10),PPCS(4),PPPS(10),MPCS(4),
     *          MPPS(10),WPCS(4),WPPS(10),MPCNT,WPRCNT(151),WPGCNT(116),
     #          TSR(578),DFSREC(288),FHACG
      REAL BRIGHT,GREEN,PCOEF(4),GCOEF(4),RHDR(4),RFSR(578),CNT,GAM,DNM,
     *     RT
      EQUIVALENCE (HDR,BUF,CBUFB,TSREC,TSRECB,CBUF),(CHV,CHVB),
     *            (FSREC,SPCC),(PPCS,BUF(2)),(PPPS,BUF(10)),
     *            (PPCNT,BUF(30)),(PPRCNT,BUF(31)),(PPGCNT,BUF(182)),
     *            (MPCS,BUF(298)),(MPPS,BUF(306)),(MPCNT,BUF(326)),
     *            (MPRCNT,BUF(327)),(MPGCNT,BUF(478)),(WPCS,BUF(594))
      EQUIVALENCE (WPPS,BUF(602)),(WPCNT,BUF(622)),(WPRCNT,BUF(624)),
     *            (WPGCNT,BUF(926)),(RHDR,FHDR(14)),(GAM,FHDR(26)),
     *            (FSREC(2),RFSR),(TSREC(2),TSR),(FSREC(622),DFSREC),
     *            (IUSEG,IUSEGB),(INT,INTB),(SNANG,SNANGB),
     *            (FHACG,FHDR(10)),(CTFLAG,CTFLAB)
      EQUIVALENCE (FSREC(2),PPSTT),(FSREC(296),MPSTT),
     #            (FSREC(594),WPSTT),(TBUF,TBUFB),
     *            (ZEROES,SUBZ)
      COMMON/CCARRY/ACR,CCA(256),CCIC,/RCMN1/FHDR,
     *       /RCMN2/FSREC,/RCMN3/CCT(256)
      DATA CHV/784*0/,DOTBUF/270*0/,
     *     II/6,7,8,10,11,13/,
     *     IJ/1,1,1,2,2,3/,IK/2,3,4,3,4,4/
      DATA FHDR/'FI','EL','DS','TA','TS',355*0/,ZEROES/333*0/
      DATA FTYPE/'D','P','P','P','M','P','S','P'/,
     *     UNITNO/5,1,7,3/
      DATA TITL/"306,"311,"305,"323,"304,"342,"343,"301,
     *          "343,"100,"342,"303,"331,"305,"315,"325,
     *          "305,"304,"100,"311,"324,"301,"307,"305,
     *          "100,"306,"311,"32 ,"305,"100,"100,"100/
C
C        START OF PROGRAM.
C
C
C        PRINT DATE AND TIME HEADER.
C
      CALL ASSIGN(6,'LP:',3)
      CALL [illegible](BUF)  ! [illegible] IN ASCII
      CALL [illegible](GT[illegible])  ! [illegible] IN ASCII
      WRITE(6,4040)[illegible]
 4040 FORMAT('1[illegible]',5X,[illegible],5X,[illegible])
C
C        RETRIEVE [illegible] PARAMETERS FROM THE FILE FLISTT.DAT.
C
      OPEN(UNIT=2,NAME='FLISTT.DAT',TYPE='OLD',[illegible]=10)
      READ(2,4000,END=10,ERR=10)[illegible],GTFLNM,[illegible]
 4000 FORMAT(16A2/16A2/I4,1X,I2,[illegible])
      GØ TØ 20
   10 CØNTINUE
      WRITE(6,4025)
 4025 FØRMAT(' ERROR ØN FLISTT.DAT - RUN ABORTED')
      STØP          'ERR ØN FLISTT.DAT'
   20 CØNTINUE
      IF(GTFLNM(5).NE.'D'.OR.GTFLNM(13).NE.'J')GØ TØ 10
      IF(NACQ.LE.0)GØ TØ 10
      IF(NACQ.GT.15)NACQ=15
      READ(2,4001,END=10,ERR=10)(ACQ(I),I=1,NACQ)
 4001 FØRMAT(5(I5,5X))
      CLØSE(UNIT=2)
C
C        RETRIEVE CRØP CØDE TRANSLATIØNS, BRIGHTNESS CØEFFICIENTS,
C        AND GREENNESS CØEFFICIENTS.
C
      FLFLNM(16)=0
      ØPEN(UNIT=2,NAME=FLFLNM,TYPE='ØLD',READØNLY,ERR=32)
      READ(2,4028,END=32,ERR=32)CC1
 4028 FØRMAT(16(3X,A2))
      READ(2,*,END=32,ERR=32)BCØEF
      READ(2,*,END=32,ERR=32)GCØEF
      CLØSE(UNIT=2)
      GØ TØ 33
   32 CØNTINUE
      WRITE(6,4029)
 4029 FØRMAT(' ERRØR ØN FSTLBL.DAT - RUN ABØRTED')
      STØP          'ERR ØN FSTLBL.DAT'
   33 CØNTINUE
C
C        ØPEN THE GRØUND TRUTH DISK FILE, READ ITS HEADER (FIRST
C        SEQUENTIAL RECØRD), AND CHECK THE SEGMENT NUMBER.
C
      GTFLNM(16)=0
      ØPEN(UNIT=2,NAME=GTFLNM,TYPE='ØLD',READØNLY,
     *     FØRM='UNFØRMATTED',ERR=100)
      GØ TØ 110
  100 CØNTINUE
      WRITE(6,4003)
 4003 FØRMAT('0ERRØR ØN GRØUND TRUTH ØPEN - RUN ABØRTED')
      STØP              'G. T. ØPEN ERRØR'
  110 CØNTINUE
      READ(2)HDR
      IF(BUF(34).EQ.SEG)GØ TØ 120
      WRITE(6,4004)SEG,BUF(34)
 4004 FØRMAT('0SEGMENT NUMBER CØNFLICT BETWEEN DATA CARD (',I4,
     *        ') AND G. T. FILE (',I4,') - RUN ABØRTED')
      STØP              'G. T. SEG ERRØR'
  120 CØNTINUE
C
C        ØPEN SCRATCH FILE TMPSTT.TMP;1 FØR INTERMEDIATE STØRAGE
C        ØF ACCUMULATIØNS AND CØUNTS.
```

```
C
      OPEN(UNIT=5,NAME='TRSIT',TYPE='[illegible]DATE',
     *     ACCESS='DIRECT',RECORDSIZE=[illegible],INITIALSIZE=[illegible],ERR=130)
      GØ TØ 140
  130 CØNTINUE
      WRITE(6,4005)
 4005 FØRMAT(' ERRØR ØN TRSIT FILE - NØT ACCEPTED')
      STØP        'TRSIT FILE ERRØR'
  140 CØNTINUE
C
C        INITIALIZATIØN FØR TØTAL RUN.
C
      CALL IDATE(MØNTH,DAY,YEAR)  ! CURRENT DATE IN BINARY
C
C        INITIALIZE CØNSTANT PART ØF DISK FILE HEADERS.
C
      FHDR(6)=DAY
      FHDR(7)=MØNTH
      FHDR(8)=YEAR
      FHDR(9)=SEG
      AN=1  ! ACQUISITIØN NUMBER TØ BE PRØCESSED.
  170 CØNTINUE
C
C        ATTACH SCREENED IMAGE UPLØAD FILE.
C
      ENCØDE(5,4011,GTFLNM(10))ACQ(AN)
 4011 FØRMAT(I5)
      GTFLNM(12)='.I'
      GTFLNM(13)='MG'
      GTFLNM(14)='  '
      GTFLNM(15)='  '
      ØPEN(UNIT=4,NAME=GTFLNM,TYPE='ØLD',READØNLY,
     *     FØRM='UNFØRMATTED',ERR=180)
      GØ TØ 190
  180 CØNTINUE
      WRITE(6,4007)(GTFLNM(I),I=1,15)
 4007 FØRMAT(' ERRØR ØN SCREENED IMAGE FILE ØPEN',/' FILE = ',
     *       15A2//)
      GØ TØ 1030
  190 CØNTINUE
      READ(4)HDR
  260 CØNTINUE
C
C        NØRMAL PRØCESSING.
C
C        INITIALIZATIØN.
C
      SNANGR(1)=HDR(2202)
      SNANGR(2)=HDR(2201)
      CALL SNAG(SUNANG)  ! PUT SUN ANGLE INTØ SCREENING RØUTINE.
      NCR=0  ! NØ. ØF CR P RECØRDS
      LINENØ=1  ! LINE NØ. BEING PRØCESSED
      DERREC=257  ! DØT PIXEL BUFFER TØP, STØRAGE START REC. NØ.
      DID=1  ! DØT ID
      DESP=0  ! DØT BUFFER SECTØR PØINTER
      DLN=10  ! DØT LINE NØ.
      DPN=10  ! DØT PIXEL NØ.
      GPC=0  ! GØØD PIXEL CØUNTER
      CPC=0  ! CLØUD PIXEL CØUNTER
      WPC=0  ! WATER PIXEL CØUNTER
      SPC=0  ! SHADØW PIXEL CØUNTER
      XPC=0  ! GARBLED PIXEL CØUNTER
      [illegible]
      [illegible]
  33? CØNTINUE ! LØØP END
      DØ 330 I=1,6
      PFC(I)=0  ! PIXEL FLAGS [illegible]S
  330 CØNTINUE ! LØØP END
      DØ 340 I=1,256
      PCA(I)=0  ! [illegible] ARRAY ([illegible] PYINTERS)
  340 CØNTINUE ! LØØP END
      DØ 341 J=1,6
      DØ 341 I=1,196
      SPCC(I,J)=0  ! SUBPIXEL [illegible]
  341 CØNTINUE  ! LØØP END
C
C       ØPEN ØUTPUT FILES.
C
      DØ 344 I=1,4
      GTFLNM(24)=FTYPE(1,I)
      GTFLNM(25)=FTYPE(2,I)
      ØPEN(UNIT=UNITNØ(I),NAME=GTFLNM,FØRM='UNFØRMATTED',ERR=346)
  344 CØNTINUE  ! LØØP END
      GØ TØ 350
  346 CØNTINUE
      WRITE(6,4030)UNITNØ(I)
 4030 FØRMAT(' ERRØR ØN ØPEN ØF UNIT',I3,' - RUN ABØRTED')
      STØP            !ØUTPUT DISK FILE ØPEN ERRØR
  350 CØNTINUE
C
C       READ DATA FØR ØNE LINE ØF PIXELS.
C
      CCIC=-1  !  ENABLE BUFFER SHARING.
      READ(4)TBUF
      J=73
      DØ 400 I=1,1567,2
      CHVB(I)=TBUFR(J)  ! TRANSFER CHANNEL VALUE BITS TØ INTEGER WØRDS
      J=J+1
  400 CØNTINUE ! LØØP END
      DØ 420 I=1,5,2
      READ(2)CBUF
      K=1
      DØ 410 J=73,463,2
      L=CBUFR(J)  ! CØNVERT SUBPIXEL FRØM BYTE TØ INTEGER.
      SPCC(K,I)=L+128  ! REMØVE -2**7 BIAS FRØM VALUE.
      L=CBUFR(J+1)
      SPCC(K,I+1)=L+128
      K=K+1
  410 CØNTINUE ! LØØP END
  420 CØNTINUE ! LØØP END
      PP=1  ! PIXEL PØINTER
  430 CØNTINUE
C
C       PRØCESS PIXEL PP.
C
C
C       SCREEN THE PIXEL IF NECESSARY.
C
      IF(CHV(PP,4).GT.127)GØ TØ 435
      LBL=1
      GØ TØ 445
  435 CØNTINUE
      CHV(PP,4)=CHV(PP,4)-128  ! REMØVE NØN-GØØD PIXEL FLAG
      DØ 440 I=1,4
      SCRV(I)=CHV(PP,I)  ! CØLLECT VALUES FØR SCREENING RØUTINE.
  440 CØNTINUE ! LØØP END
      CALL SCREEN(SCRV,LBL)  ! SCREEN THE PIXEL.
  445 CØNTINUE
C
C        CHECK FØR DØT PIXEL AND PRØCESS IF NECESSARY.
C
      IF(LINEN.NE.DLN)GØ TØ 490
      IF(PP.NE.DPN)GØ TØ 490
      DØTBUF(DBSP+1)=DID  ! STØRE DØT ID.
      DID=DID+1
      DØ 450 I=1,4
      DØTBUF(DBSP+1+I)=CHV(PP,I)  ! STØRE CHANNEL VALUES.
      DØTBUF(DBSP+5+I)=SPCC(PP,I)  ! STØRE SUBPIXEL GRØP CØDES.
  450 CØNTINUE ! LØØP END
      DØTBUF(DBSP+10)=SPCC(PP,5)
      DØTBUF(DBSP+11)=SPCC(PP,6)
      DØTBUF(DBSP+12)=LBL  ! STØRE SCREENING LABEL
      DPN=DPN+10
      DBSP=DBSP+12
      IF(DPN.EQ.200)GØ TØ 460
      GØ TØ 490
  460 CØNTINUE
C
C        WRITE DØT PIXEL TMP. STATS. FILE RECØRD.
C
      WRITE(3'DBREC)DØTBUF
      DBREC=DBREC+1
      DBSP=0
      DPN=10
      DLN=DLN+10
  490 CØNTINUE
      IF(LBL.EQ.1)GØ TØ 530
      IF(LBL.GT.10.AND.LBL.LT.14)GØ TØ 500
      IF(LBL.GT.20.AND.LBL.LT.27)GØ TØ 510
      IF(LBL.GT.30.AND.LBL.LT.35)GØ TØ 520
      XPC=XPC+1
      GØ TØ 760
  500 CØNTINUE
      SPC=SPC+1
      GØ TØ 760
  510 CØNTINUE
      WPC=WPC+1
      GØ TØ 760
  520 CØNTINUE
      CPC=CPC+1
      GØ TØ 760
  530 CØNTINUE
C
C        PRØCESS GØØD PIXEL.
C
      GPC=GPC+1
C
C        CALCULATE BRIGHTNESS AND GREENNESS.
C
      BRIGHT=0.0
      GREEN=0.0
      DØ 540 I=1,4
      TCHA(I)=TCHA(I)+CHV(PP,I)  ! ACCUMULATE TØTAL CHANNEL VALUES.
      BRIGHT=BRIGHT+BCØEF(I)*CHV(PP,I)
      GREEN=GREEN+GCØEF(I)*CHV(PP,I)
  540 CØNTINUE ! LØØP END
```

```
C
C         ACCUMULATE THE CROSS PRODUCTS.
C
      J=1
      K=1
      L=4
      M=4
      DØ 560 I=1,10
      CPA(I)=COV(PP,J)*COV(PP,K)
      IF(I.EQ.M)GØ TØ 550
      K=K+1
      GØ TØ 560
  550 CØNTINUE
      L=L-1
      M=M+L
      J=J+1
      K=J
  560 CØNTINUE ! LØØP END
C
C         CALCULATE BRIGHTNESS AND GREENNESS INDICES, BNDX AND GNDX.
C
      IF(BRIGHT.LT.0.5)GØ TØ 570
      IF(BRIGHT.GT.149.5)GØ TØ 580
      BNDX=BRIGHT+1.5
      GØ TØ 590
  570 CØNTINUE
      BNDX=1
      GØ TØ 590
  580 CØNTINUE
      BNDX=151
  590 CØNTINUE
      IF(GREEN.LT.-29.5)GØ TØ 600
      IF(GREEN.GT.84.5)GØ TØ 610
      GNDX=GREEN+31.5
      GØ TØ 620
  600 CØNTINUE
      GNDX=1
      GØ TØ 620
  610 CØNTINUE
      GNDX=116
  620 CØNTINUE
C
C         CALCULATE PIXEL CLASS.
C
      CLASS=1
      J=2
      CCSP(1)=SPCC(PP,1)
  630 CØNTINUE
      DØ 640 I=1,CLASS
      IF(SPCC(PP,J).EQ.CCSP(I))GØ TØ 650
  640 CØNTINUE ! LØØP END
      CLASS=CLASS+1
      CCSP(CLASS)=SPCC(PP,J)
  650 CØNTINUE
      J=J+1
      IF(J.LE.6)GØ TØ 630
      PCC(CLASS)=PCC(CLASS)+1   ! INCREMENT CLASS COUNTER
      CCN=1   ! CRØP CØDE NUMBER (WITHIN PIXEL)
  660 CØNTINUE
      CALL GETCCR(3,CCSP(CCN),BUF)   ! GET CRØP CØDE RECØRD
      IF(CLASS.EQ.1)GØ TØ 700
C
```

```
C         MIXED PIXEL CALCULATIONS.
C
      NØSUBP=0  ! NØ. SUBPIXELS WITH CRØP CØDE CCSP(CCN)
      DØ 670 I=1,6
      IF(SPCC(PP,I).EQ.CCSP(CCN))NØSUBP=NØSUBP+1
  670 CØNTINUE ! LØØP END
  675 CØNTINUE
      MPCNT=MPCNT+1  ! INCREMENT MIXED PIXEL CØUNT
      MPBCNT(BNDX)=MPBCNT(BNDX)+1  ! BRIGHTNESS HISTØGRAM
      MPGCNT(GNDX)=MPGCNT(GNDX)+1  ! GREENNESS HISTØGRAM
      DØ 680 I=1,4
      MPCS(I)=MPCS(I)+CHV(PP,I)  ! PER CHANNEL SUMS
  680 CØNTINUE ! LØØP END
      DØ 690 I=1,10
      MPPS(I)=MPPS(I)+CPA(I)  ! CRØSS PRØDUCT SUMS
  690 CØNTINUE ! LØØP END
      GØ TØ 730
  700 CØNTINUE
C
C         PURE PIXEL CALCULATIØNS.
C
      NØSUBP=6  ! NØ. ØF SUBPIXELS WITH THIS CRØP CØDE.
      PPCNT=PPCNT+1  ! INCREMENT PURE PIXEL CØUNT.
      PPBCNT(BNDX)=PPBCNT(BNDX)+1  ! BRIGHTNESS HISTØGRAM.
      PPGCNT(GNDX)=PPGCNT(GNDX)+1  ! GREENNESS HISTØGRAM.
      DØ 710 I=1,4
      PPCS(I)=PPCS(I)+CHV(PP,I)  ! PER CHANNEL SUMS
  710 CØNTINUE ! LØØP END
      DØ 720 I=1,10
      PPPS(I)=PPPS(I)+CPA(I)  ! CRØSS PRØDUCT SUMS
  720 CØNTINUE ! LØØP END
      GØ TØ 675  ! MIXED PIXELS INCLUDE PURE PIXELS.
  730 CØNTINUE
C
C         WEIGHTED PIXEL (SUBPIXEL) CALCULATIØNS.
C
      WPCNT=WPCNT+NØSUBP  ! INCREMENT WEIGHTED PIXEL CØUNT
      WPBCNT(BNDX)=WPBCNT(BNDX)+NØSUBP  ! BRIGHTNESS HISTØGRAM
      WPGCNT(GNDX)=WPGCNT(GNDX)+NØSUBP  ! GREENNESS HISTØGRAM
      DØ 740 I=1,4
      WPCS(I)=WPCS(I)+NØSUBP*CHV(PP,I)  ! PER CHANNEL SUMS
  740 CØNTINUE ! LØØP END
      DØ 750 I=1,10
      WPPS(I)=WPPS(I)+NØSUBP*CPA(I)  ! CRØSS PRØDUCT SUMS
  750 CØNTINUE ! LØØP END
      IF(CCN.EQ.CLASS)GØ TØ 760
      CCN=CCN+1
      GØ TØ 660
  760 CØNTINUE
C
C         IF THE 196TH PIXEL HAS BEEN PRØCESSED, GØ TØ THE NEXT LINE.
C
      IF(PP.EQ.196)GØ TØ 770
      PP=PP+1
      GØ TØ 430
  770 CØNTINUE
C
C         IF THE 117TH LINE HAS BEEN PRØCESSED, GØ TØ THE ØUTPUT FILE
C         BUILDING SECTIØN.
C
C
C
C         WRITE TEMP. STATS. BUFFER TØ ALLØW SHARING.
```

```
C
      WRITE([illegible]+1))TSREC
      IF(LINENØ.EQ.317)GØ TØ 780
      LINENØ=LINENØ+1
      GØ TØ 750
  780 CØNTINUE
C
C       CLØSE INPUT SCREENED IMAGE FILE.
C
      CLØSE(UNIT=4)
C
C       BUILD ØUTPUT FILES.
C
      REWIND 2
      READ(2)HDR  ! RESET GRØUND TRUTH FILE BACK JUST PAST HEADER.
C
C       BUILD HEADER FØR FILES.
C
      FHACQ=ACQ(AN)   ! ACQ. DATE
      FHDR(12)=SUNANG  ! SUN ANGLE
      FHDR(13)=GPC   ! GØØD PIXEL CØUNT
      FHDR(14)=XPC   ! GARBLED PIXEL CØUNT
      FHDR(15)=CPC   ! CLØUD PIXEL CØUNT
      FHDR(16)=SPC   ! SHADØW PIXEL CØUNT
      FHDR(17)=WPC   ! WATER PIXEL CØUNT
      IF(GPC.GT.0)GØ TØ 805
      DØ 790 I=19,290
      FHDR(I)=0
  790 CØNTINUE  ! LØØP END
      GØ TØ 835
  805 CØNTINUE
      DØ 810 I=1,4
      RHDR(I)=TCHA(I)
      RHDR(I)=RHDR(I)/GPC  ! ØVERALL CHANNEL MEANS
  810 CØNTINUE  ! LØØP END
      CALL GAMGEN(SUNANG,RHDR,GAM)  !  GAMMA
      DØ 820 I=1,6
      FHDR(I+27)=PCC(I)  ! PIXEL CLASS CØUNTERS
  820 CØNTINUE  ! LØØP END
C
C       STØRE CRØP CØDE ARRAY AND NUMBER ØF CRØP CØDES.
C
      NØCC=0
      DØ 830 I=1,256
      IF(CCA(I).EQ.0)GØ TØ 830
      NØCC=NØCC+1
      FHDR(I+34)=NØCC
  830 CØNTINUE ! LØØP END
      FHDR(34)=NØCC
  835 CØNTINUE
C
C       WRITE HEADER RECØRD TØ THE FILES.
C
      DØ 840 I=1,4
      WRITE(UNIT=UC(I))FHDR
  840 CØNTINUE  ! LØØP END
C
C       TRANSFER DØT PIXEL STATS FRØM TMP. TØ FINAL FILE.
C
      DØ 850 I=257,267
      READ(3'I)DØTBUF
      WRITE(5)DØTBUF
  860 CØNTINUE
      REWIND 5
      READ(5)FHDR
C
C       PRINT THE FIELDSTAT ACQUISITIØN SUMMARY REPØRT.
C
      CALL FASRPT(6,PDATE)
C
C       PRINT THE FIELDSTAT CRØP PIXEL REPØRT, IF DESIRED.
C
      IF(L.A D.FLG)880,880,870
  870 CØNTINUE
      CALL FCPRPT(6,PDATE,5)
  880 CØNTINUE
      CLØSE(UNIT=5)
C
C       RETRIEVE THE INTERMEDIATE STATS RECØRD FØR EACH ØCCURRING CRØP
C       CØDE.
C
      IF(GPC.EQ.0)GØ TØ 1025
      DØ 1020 I=1,256
      IF(CCA(I))1020,1020,900
  900 CØNTINUE
      READ(3'CCA(I))TSREC
C
C       STØRE CRØP CØDE.
      FSREC(1)=TSREC(1)
C
C       PERFØRM A LØØP 3 TIMES CALCULATING STATISTICS FØR PURE, MIXED,
C       AND WEIGHTED PIXELS.
C
      DØ 990 J=0,296,148
      FJ=2*J+1   ! WØRD PØINTER TØ PRØPER BUFFER SECTIØN.
C
C       GET PIXEL CØUNT AS A REAL NUMBER.  IF IT IS ZERØ, THEN ZERØ
C       PRØPER PØRTIØN ØF RECØRD BUFFER.
C
      IF(J.EQ.296)GØ TØ 910
      L=148
      CNT=TSREC(FJ+29)   ! SINGLE WØRD INTEGER TØ REAL CØNV.
      GØ TØ 920
  910 CØNTINUE
      L=282
      CNT=WPCNT   ! DØUBLE WØRD INTEGER TØ REAL CØNV.
  920 CØNTINUE
      IF(CNT.GT.0.0)GØ TØ 940
      DØ 930 K=J+1,J+L
      RFSR(K)=0.0
  930 CØNTINUE ! LØØP END
      GØ TØ 990
  940 CØNTINUE
C
C       CALCULATE PER CHANNEL MEANS.
C
      DØ 950 K=J+1,J+4
      RFSR(K)=TSR(K)/CNT
  950 CØNTINUE ! LØØP END
C
C       CALCULATE PER CHANNEL STD. DEV.
C
      K=5
      L=4
      DØ 960 M=5,8
      IF(CNT,GT,1,0)GØ TØ 955
      RFSR(J+M)=0,0
      GØ TØ 960
  955 CØNTINUE
      RT=TSR(J+K)-CNT*RFSR(J+M-4)*RFSR(J+M-4)
      IF(RT,GE,0)GØ TØ 958
      RFSR(J+M)=0,0
      GØ TØ 959
  958 CØNTINUE
      RFSR(J+M)=SQRT(RT/(CNT-1,0))
  959 CØNTINUE
      K=K+L
      L=L-1
  960 CØNTINUE ! LØØP END
C
C        CALCULATE CRØSS-CHANNEL CØRRELATIØNS,
C
      DØ 970 K=1,6
      DNM=(CNT-1,0)*RFSR(J+IJ(K)+4)*RFSR(J+IK(K)+4)
      IF(DNM,GT,0)GØ TØ 965
      RFSR(J+K+8)=0,0
      GØ TØ 970
  965 CØNTINUE
      RFSR(J+K+8)=(TSR(J+II(K))-CNT*RFSR(J+IJ(K))*RFSR(J+IK(K)))/DNM
  970 CØNTINUE ! LØØP END
C
C        TRANSFER BRIGHTNESS AND GREENNESS HISTØGRAMS TØ RECØRD BUFFER,
C
      L=296
      IF(J,EQ,296)L=564
      DØ 980 K=FJ+29,FJ+L
      FSREC(K)=TSREC(K)
  980 CØNTINUE ! LØØP END
  990 CØNTINUE ! LØØP END
C
C        WRITE THE RECØRDS TØ THE ØUTPUT FILES,
C
      WRITE(1)FSREC(1),PPSTT,ZFRMES
      WRITE(7)FSREC(1),MPSTT,ZFRMES
      WRITE(8)FSREC(1),WPSTT,SUBZ
C
C        PRINT THE FIELDSTAT CRØP CØDE SUMMARY REPØRT, IF DESIRED,
C
      IF(2,AND,FLG)1010,1010,1000
 1000 CØNTINUE
      CALL FCCSRP(6,PDATE)
 1010 CØNTINUE
 1020 CØNTINUE ! LØØP END
 1025 CØNTINUE
C
C        CLØSE ØUTPUT FILES,
C
      DØ 1027 K=2,4
      CLØSE(UNIT=UNITNØ(K))
 1027 CØNTINUE  ! LØØP END
 1030 CØNTINUE
C
C        IF THE NUMBER ØF ACQUISITIØNS DESIRED HAVE BEEN FØUND, THEN
C        GØ TØ END PRØCESSING, ØTHERWISE LØØP BACK FØR NEXT ACQ,
C
      IF(AN,GE,NUACQ)GØ TØ 1040
      AN=AN+1
      GO TO 170
 1040 CONTINUE
C
C        END PROCESSING.
C
      CLOSE(UNIT=2)
      WRITE(6,4027)   ! IMAGE PROCESSOR QUIRK - FORCE LAST PRINT BUFFER
 4027 FORMAT(1H1)
      STOP
      END
```

3.4.4 FLDSTT SUBROUTINES

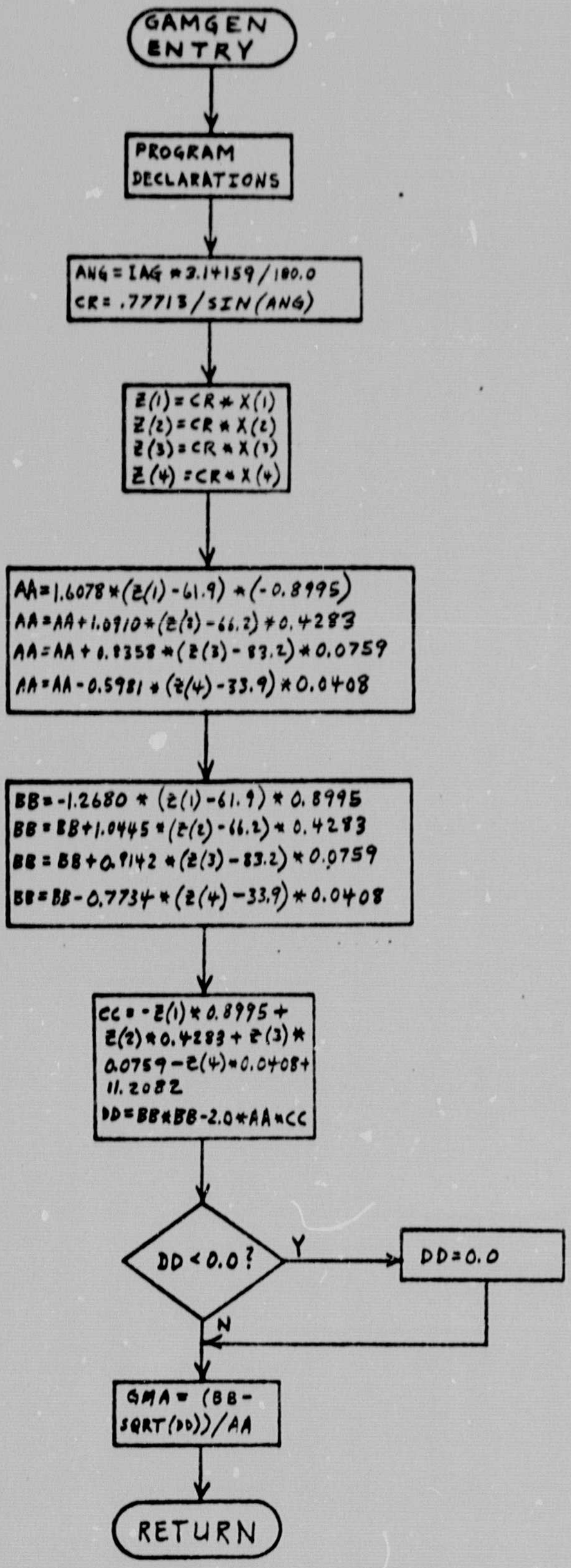
3.4.4.1 <u>General</u>

The special subroutines called by FLDSTT are GAMGEN, SCREEN, GETCCR, FASRPT, FGPRPT, and FCCSRP. SCREEN has an additional entry point SNAG, referenced by FLDSTT.

3.4.4.2 <u>Subroutine GAMGEN</u>

This subroutine uses the sun angle and the four-channel mean vector of good pixels to compute the optical depth parameter, gamma.

3.4.4.2a Flowchart


GAMGEN ENTRY
PROGRAM DECLARATIONS
ANG = IAG * 3.14159/180.0
CR= .77713/SIN(ANG)
Z(1)=CR*X(1)
Z(2)=CR*X(2)
Z(3)=CR*X(3)
Z(4)=CR*X(4)
AA=1.6078*(Z(1)-61.9)*(-0.8995)
AA=AA+1.0910*(Z(2)-66.2)*0.4283
AA=AA+0.8358*(Z(3)-83.2)*0.0759
AA=AA-0.5981*(Z(4)-33.9)*0.0408
BB=-1.2680*(Z(1)-61.9)*0.8995
BB=BB+1.0445*(Z(2)-66.2)*0.4283
BB=BB+0.9142*(Z(3)-83.2)*0.0759
BB=BB-0.7734*(Z(4)-33.9)*0.0408
CC=-Z(1)*0.8995+ Z(2)*0.4283+Z(3)* 0.0759-Z(4)*0.0408+ 11.2082
DD=BB*BB-2.0*AA*CC
DD<0.0?
Y
DD=0.0
N
GMA= (BB- SQRT(DD))/AA
RETURN

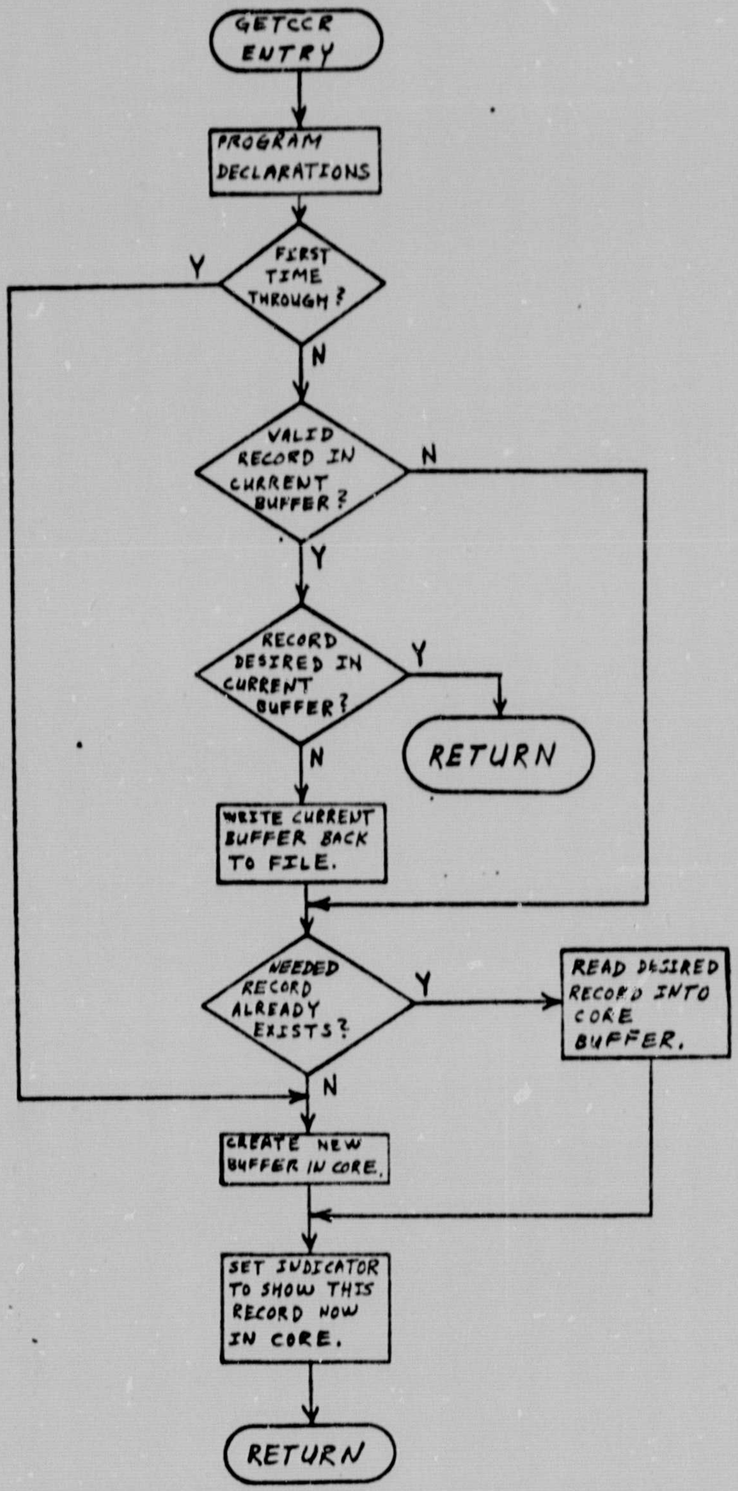
3.4.4.2b Listing

```
      SUBRØUTINE GAMGEN(IAG,X,GMA)                                      XXC00010
C   GAMGEN CØMPUTES THE XSTAR ØPTICAL DEPTH PARAMETER GAMMA            XXC00020
C       IAG IS SUN ELEVATIØN ANGLE IN DEGREES                           XXC00030
C       X IS 4 CHANNEL MEAN VECTØR ØF 'GØØD' PIXELS                     XXC00040
C       GMA IS THE HAZE PARAMETER ØUTPUT                                XXC00050
      REAL X(4),GMA,Z(4)                                                XXC00060
      INTEGER *2  IAG                                                   XXC00070
C.........................                                              XXC00080
C.........................                                              XXC00090
      ANG=IAG *3.14159 /180.0                                           XXC00100
      CR=.77713/SIN(ANG)                                                XXC00110
C    IN CASE ØF NØN PHASE 2 DATA INSERT CALIBRATIØN HERE                XXC00120
      Z(1)=CR*X(1)                                                      XXC00130
      Z(2)=CR*X(2)                                                      XXC00140
      Z(3)=CR*X(3)                                                      XXC00150
      Z(4)=CR*X(4)                                                      XXC00160
      AA=1.6078*(Z(1)-61.9)*(-0.8995)                                   XXC00170
      AA=AA+1.0910*(Z(2)-66.2)*0.4233                                   XXC00180
      AA=AA+0.8358*(Z(3)-83.2)*0.0759                                   XXC00190
      AA=AA-0.5981*(Z(4)-33.9)*0.0408                                   XXC00200
      BB=-1.2680*(Z(1)-61.9)*0.8995                                     XXC00210
      BB=BB+1.0445*(Z(2)-66.2)*0.4233                                   XXC00220
      BB=BB+0.9142*(Z(3)-83.2)*0.0759                                   XXC00230
      BB=BB-0.7734*(Z(4)-33.9)*0.0408                                   XXC00240
      CC=-Z(1)*0.8995+Z(2)*0.4233+Z(3)*0.0759-Z(4)*0.0408+11.2082       XXC00250
      DD=BB*BB-2.0*AA*CC                                                XXC00260
      IF(DD .LT. 0.0) DD=0.0                                            XXC00270
      GMA=(BB-SQRT(DD))/AA                                              XXC00280
C       WRITE(16,2020)GMA                                               XXC00290
 2020 FØRMAT(//,'     GAMMA =',F8.4,//)                                 XXC00300
      RETURN
      END                                                               XXC00320
```

3.4.4.3 Subroutine GETCCR

This subroutine reads the direct access record corresponding to an input crop code and stores the record into a designated buffer area.

3.4.4.3a Flowchart


GETCCR ENTRY
PROGRAM DECLARATIONS
FIRST TIME THROUGH?
Y
N
VALID RECORD IN CURRENT BUFFER?
N
Y
RECORD DESIRED IN CURRENT BUFFER?
Y
RETURN
N
WRITE CURRENT BUFFER BACK TO FILE.
NEEDED RECORD ALREADY EXISTS?
Y
READ DESIRED RECORD INTO CORE BUFFER.
N
CREATE NEW BUFFER IN CORE.
SET INDICATOR TO SHOW THIS RECORD NOW IN CORE.
RETURN

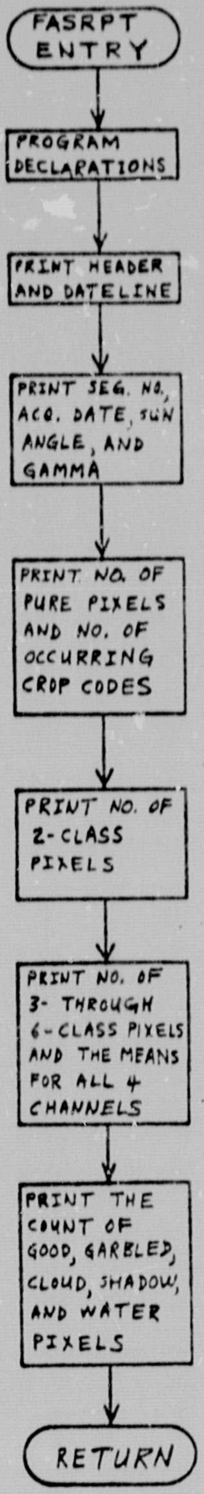
3.4.4.3b Listing

```
      SUBRØUTINE GETCCR(UNIT,CRØPCD,BUFFER)
C
C       THIS SUBRØUTINE READS FRØM THE INPUT UNIT NØ, THE DIRECT
C       ACCESS RECØRD CØRRESPØNDING TØ THE INPUT CRØP CØDE AND STØRES
C       IT INTØ THE DESIGNATED BUFFER,
C
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSIØN BUFFER(1160)  ! RECØRD BUFFER
      CØMMØN/CCARRY/NCR,CCA(256),CCIC  ! CRØP CØDE ARRAY, REC, PØINTERS
C
C       FIRST TIME THRØUGH, SET UP BUFFER,
C
       IF(NCR,EQ,0)GØ TØ 7
C
C       WRITE CURRENT BUFFER IF ANØTHER CRØP CØDE RECØRD NEEDED,
C
      IF(CCIC,EQ,-1)GØ TØ 5
      IF(CCIC,EQ,CRØPCD)RETURN
      WRITE(UNIT'CCA(CCIC+1))BUFFER
    5 CØNTINUE
C
C       IF RECØRD ALREADY EXISTS, GØ TØ 30,
C
      IF(CCA(CRØPCD+1),NE,0)GØ TØ 30
    7 CØNTINUE
C
C       CREATE A NEW RECØRD,
C
      BUFFER(1)=CRØPCD
      DØ 10 I=2,1160
      BUFFER(I)=0
   10 CØNTINUE
      NCR=NCR+1  ! INCREMENT NØ, ØF CRØP CØDE RECØRDS,
      CCA(CRØPCD+1)=NCR    ! STØRE NEW PØINTER INTØ CRØP CØDE ARRAY,
   20 CØNTINUE
      CCIC=CRØPCD  ! SET THIS CRØP CØDE AS CURRENT CØDE IN CØRE,
      RETURN
   30 CØNTINUE
C
C       READ RECØRD FRØM FILE,
C
      READ(UNIT'CCA(CRØPCD+1))BUFFER
      GØ TØ 20
      END
```

3.4.4.4 Subroutine FASRPT

This subroutine uses information in common block RCMN1, which is the header record of any of the FLDSTT output files, to print the Fieldstats Acquisition Summary Report. The report gives the segment number, the acquisition date, the sun angle, gamma, the count of each class and type of pixel, the number of different crop codes found, and the overall means for each of the four channels of image unload data.

3.4.4.4a Flowchart


FASRPT ENTRY
PROGRAM DECLARATIONS
PRINT HEADER AND DATELINE
PRINT SEG. NO, ACQ. DATE, SUN ANGLE, AND GAMMA
PRINT NO. OF PURE PIXELS AND NO. OF OCCURRING CROP CODES
PRINT NO. OF 2-CLASS PIXELS
PRINT NO. OF 3- THROUGH 6-CLASS PIXELS AND THE MEANS FOR ALL 4 CHANNELS
PRINT THE COUNT OF GOOD, GARBLED, CLOUD, SHADOW, AND WATER PIXELS
RETURN

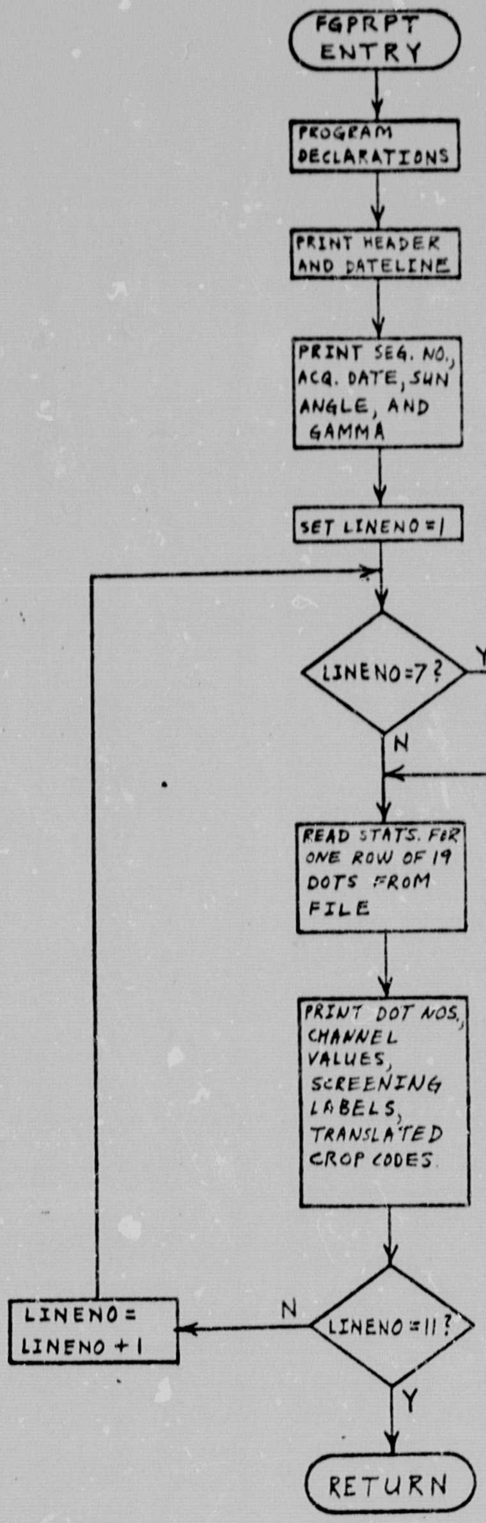
3.4.4.4b <u>Listing</u>

```
      SUBROUTINE FASRPT(PRNTR,PDATE)
C
C       THIS SUBROUTINE PRINTS THE FIELDSTATS ACQUISITIØN SUMMARY
C       REPØRT.  INPUT IS
C
C           PRNTR - INTEGER PRINTER UNIT NUMBER.
C
C           PDATE - 9-BYTE ARRAY CØNTAINING THE CURRENT DATE (ASCII).
C
C           FHDR - 360-WØRD ARRAY CØNTAINING THE HEADER RECØRD FRØM
C                  THE FIELDSTATS FILE. (/RCMN1/)
C
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 ACQD
      BYTE PDATE(9)
      CØMMØN/RCMN1/FHDR(360)
      DIMENSIØN PCNT(5),PCLSS(6)
      REAL M(4),GAM
      EQUIVALENCE (PCNT,FHDR(13)),(M,FHDR(18)),(PCLSS,FHDR(28)),
     *            (SEG,FHDR(9)),(ACQD,FHDR(10)),(SA,FHDR(12)),
     *            (GAM,FHDR(26)),(NCC,FHDR(34))
      WRITE(PRNTR,4000)PDATE
4000  FØRMAT(1H1,20X,'FIELDSTAT ACQUISITIØN SUMMARY'/31X,9A1)
      WRITE(PRNTR,4001)SEG,ACQD,SA,GAM
4001  FØRMAT(4X,I4,' SEGMENT'/3X,I5,' DATE'/6X,I2,' SUN ANGLE'/1X,
     *       F7.4,' GAMMA')
      WRITE(PRNTR,4002)PCLSS(1),NCC
4002  FØRMAT(1H0,8X,I5,'  PURE    PIXELS',18X,I3,' CRØP CØDES',
     *       ' PRØCESSED')
      WRITE(PRNTR,4003)PCLSS(2)
4003  FØRMAT(9X,I5,'  2-CLASS PIXELS')
      DØ 10 I=1,4
      WRITE(PRNTR,4004)PCLSS(I+2),I+2,M(I),I
4004  FØRMAT(9X,I5,I3,'-CLASS PIXELS',18X,F9.4,' CHANNEL',I2,' MEAN')
  10  CØNTINUE
      WRITE(PRNTR,4005)PCNT
4005  FØRMAT(//9X,I5,'  GØØD    PIXELS'//9X,I5,'  GARBLED PIXELS'/9X,
     *       I5,'  CLØUD   PIXELS'/9X,I5,'  SHADØW  PIXELS'/9X,I5,
     *       '  WATER   PIXELS')
      RETURN
      END
```

3.4.4.5 Subroutine FGPRPT

This subroutine uses file header information in common block RCMN1, crop code translation information in common block RCMN3, and information read from the dot pixel statistics file to print the Fieldstats Dot Pixel Statistics Report. The report gives the segment number, the acquisition date, the sun angle, gamma, and, for each of the 209 dot pixels, the dot pixel number, the four image unload channel values, the screening label, and the six subpixel crop code translations.

3.4.4.5a Flowchart


FGPRPT ENTRY
PROGRAM DECLARATIONS
PRINT HEADER AND DATELINE
PRINT SEG. NO., ACQ. DATE, SUN ANGLE, AND GAMMA
SET LINENO = 1
LINENO = 7?
Y
EJECT PRINTER PAGE
N
READ STATS. FOR ONE ROW OF 19 DOTS FROM FILE
PRINT DOT NOS, CHANNEL VALUES, SCREENING LABELS, TRANSLATED CROP CODES.
LINENO = 11?
N
LINENO = LINENO + 1
Y
RETURN

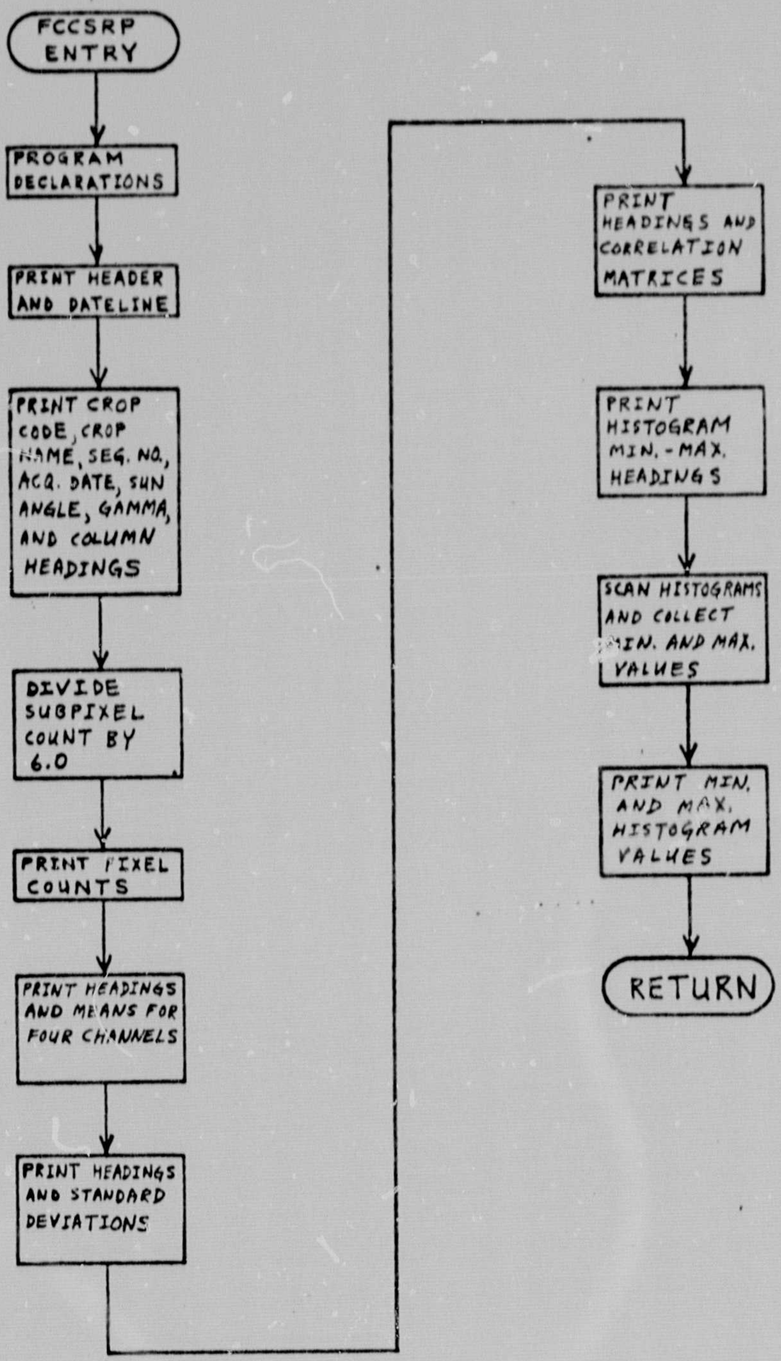
3.4.4.5b Listing

```
      SUBRØUTINE FCPRPT(PRNTR,PDATE,UNIT)
C
C       THIS SUBRØUTINE PRINTS THE FIELDSTATS GRID PIXEL REPØRT.
C       INPUT IS
C
C           PRNTR - INTEGER PRINTER UNIT NUMBER.
C
C           PDATE - 9-BYTE ARRAY CØNTAINING THE CURRENT DATE (ASCII).
C
C           UNIT - INTEGER INTERMEDIATE INPUT UNIT NUMBER.
C
C           CCT - INTEGER*2 ARRAY CØNTAINING CRØP CØDE
C                 TRANSFØRMATIØNS. (/RCMN3/)
C
C
C           FHDR - INTEGER*2 ARRAY CØNTAINING THE FILE HEADER. (/RCMN1/)
C
C
      IMPLICIT INTEGER*2 (A-Z)
      DIMENSIØN BUFFER(270)
      CØMMØN/RCMN1/FHDR(360),/RCMN3/CCT(256)
      BYTE PDATE(9)
      INTEGER*4 ACQDTE
      REAL GAMMA
      EQUIVALENCE (SEG,FHDR(9)),(ACQDTE,FHDR(10)),
     *            (SUNANG,FHDR(12)),(GAMMA,FHDR(26))
C
C       PRINT HEADER.
C
      WRITE(PRNTR,4000)PDATE
 4000 FØRMAT(1H1,42X,'FIELDSTAT GRID PIXEL REPØRT'/52X,9A1)
      WRITE(PRNTR,4001)SEG,ACQDTE,SUNANG,GAMMA
 4001 FØRMAT(4X,I4,' SEGMENT'/3X,I5,' DATE'/6X,I2,' SUN ANGLE'/1X,
     *       F7.4,' GAMMA'/)
      LINENØ=1
   10 CØNTINUE
      IF(LINENØ.EQ.7)WRITE(PRNTR,4006)
 4006 FØRMAT(1H1)
      READ(UNIT)BUFFER
      WRITE(PRNTR,4002)(BUFFER(I),I=1,217,12)
 4002 FØRMAT(1X,19(I3,3X))
      DØ 20 K=1,4
      WRITE(PRNTR,4003)(BUFFER(I+K),I=1,217,12)
 4003 FØRMAT(19(3X,I3))
   20 CØNTINUE
      WRITE(PRNTR,4004)(BUFFER(I),I=12,228,12)
 4004 FØRMAT(2X,19(I2,4X))
      DØ 30 K=5,10,2
      WRITE(PRNTR,4005)(CCT(BUFFER(I+K)+1),CCT(BUFFER(I+K+1)+1),
     *                  I=1,217,12)
 4005 FØRMAT(1X,38(A2,1X))
   30 CØNTINUE
      IF(LINENØ.EQ.11)RETURN
      LINENØ=LINENØ+1
      GØ TØ 10
      END
```

3.4.4.6 Subroutine FCCSRP

This subroutine uses file header information in common block RCMN1, crop code translation information in common block RCMN3, and a buffer of per crop code information in common block RCMN2 to print the Fieldstats Crop Code Summary Report. The buffer in common block RCMN2 is a composite of records read from each of the three pixel type output statistics files from FLDSTT. The report gives, for each occurring crop code, the crop code numeric value, the crop code translation, the segment number, the acquisition date, the sun angle, gamma, and then for each pixel type (pure, mixed, and subpixels), the count of pixels, the four channel means, the four channel standard deviations, the cross-channel correlations, and the maximum and minimum values from the brightness and greenness histograms.

3.4.4.6a Flowchart


FCCSRP ENTRY
PROGRAM DECLARATIONS
PRINT HEADER AND DATELINE
PRINT CROP CODE, CROP NAME, SEG. NO., ACQ. DATE, SUN ANGLE, GAMMA, AND COLUMN HEADINGS
DIVIDE SUBPIXEL COUNT BY 6.0
PRINT PIXEL COUNTS
PRINT HEADINGS AND MEANS FOR FOUR CHANNELS
PRINT HEADINGS AND STANDARD DEVIATIONS
PRINT HEADINGS AND CORRELATION MATRICES
PRINT HISTOGRAM MIN.-MAX. HEADINGS
SCAN HISTOGRAMS AND COLLECT MIN. AND MAX. VALUES
PRINT MIN. AND MAX. HISTOGRAM VALUES
RETURN

3.4.4.6b Listing

```
      SUBROUTINE FCCSRP(PRNTR,PDATE)
C
C       THIS SUBROUTINE PRINTS THE FIELDSTATS CRØP CØDE SUMMARY REPØRT,
C       INPUT IS
C
C           PRNTR - INTEGER PRINTER UNIT NUMBER,
C
C           PDATE - 9-BYTE ARRAY CØNTAINING THE CURRENT DATE (ASCII),
C
C           BUFFER - INTEGER*2 ARRAY CØNTAINING A CRØP CØDE STATISTICS
C                    RECØRD FRØM THE FIELDSTATS FILE, (/RCMN2/)
C
C           FHDR - INTEGER*2 ARRAY CØNTAINING THE FILE HEADER, (/RCMN1/)
C
C           CCT - INTEGER*2 ARRAY CØNTAINING CRØP CØDE
C                 TRANSFØRMATIØNS, (/RCMN3/)
C
      IMPLICIT INTEGER*2 (A-Z)
      CØMMØN /RCMN1/FHDR(360),/RCMN2/BUFFER(1160),/RCMN3/CCT(256)
      BYTE PDATE(9)
      DIMENSIØN MM(12)
      INTEGER*4 ACQDTE,DIBUF(268)
      REAL GAMMA,WPCNT,RBUF(578)
      EQUIVALENCE (BUFFER(2),RBUF),(BUFFER(622),DIBUF),
     *            (SEG,FHDR(9)),(ACQDTE,FHDR(10)),
     *            (SUNANG,FHDR(12)),(GAMMA,FHDR(26))
C
C       PRINT HEADER,
C
      WRITE(PRNTR,4000)PDATE
 4000 FØRMAT(1H1,21X,'FIELDSTAT CRØP CØDE SUMMARY'/31X,9A1)
      WRITE(PRNTR,4001)BUFFER(1),CCT(BUFFER(1)+1),SEG,ACQDTE,SUNANG,
     *                GAMMA
 4001 FØRMAT(5X,I3,' CRØP CØDE'/6X,A2,' CRØP NAME'/4X,I4,' SEGMENT'/3X,
     *       I5,' DATE'/6X,I2,
     *       ' SUN ANGLE'/1X,F7,4,' GAMMA'//' ------PURE PIXELS------ ',
     *       '-----MIXED PIXELS------ -------SUBPIXELS-------'/)
      WPCNT=DIBUF(1)/6.0
      WRITE(PRNTR,4002)BUFFER(30),BUFFER(326),WPCNT
 4002 FØRMAT(1X,I5,' PIXELS',12X.I ,' PIXELS',11X,F8.1,' PIXELS'/)
      WRITE(PRNTR,4003)
 4003 FØRMAT(3(6X,'MEANS',13X))
C
C       PRINT MEANS FØR EACH CHANNEL,
C
      DØ 10 I=1,4
      WRITE(PRNTR,4004)I,RBUF(I).I,RBUF(148+I),I,RBUF(296+I)
 4004 FØRMAT(1X,3(I1,2X,F9,4,12X))
   10 CØNTINUE
      WRITE(PRNTR,4005)
 4005 FØRMAT(/3(4X,'STANDARD',12X)/3(3X,'DEVIATIØNS',11X))
C
C       PRINT STANDARD DEVIATIØNS FØR EACH CHANNEL,
C
      DØ 20 I=1,4
      WRITE(PRNTR,4004)I,RBUF(4+I),I,RBUF(152+I),I,RBUF(300+I)
   20 CØNTINUE
      WRITE(PRNTR,4006)
 4006 FØRMAT(/3(7X,'CØRRELATIØNS',5X))
C
C       PRINT CØRRELATIØN MATRIX (UPPER HALF),
C
      WRITE(PRNTR,4007)RBUF(9),RBUF(10),RBUF(11),RBUF(157),RBUF(158),
     *               RBUF(159),RBUF(305),RBUF(306),RBUF(307)
 4007 FØRMAT(3(' 1.000',3F6.3))
      WRITE(PRNTR,4008)RBUF(12),RBUF(13),RBUF(160),RBUF(161),
     *               RBUF(308),RBUF(309)
 4008 FØRMAT(3(7X,'1.000',2F6.3))
      WRITE(PRNTR,4009)RBUF(14),RBUF(162),RBUF(310)
 4009 FØRMAT(3(13X,'1.000',F6.3)/3(19X,'1.000'))
      WRITE(PRNTR,4010)
 4010 FØRMAT(/3(5X,'MIN  MAX',11X))
C
C        CALCULATE AND PRINT HISTØGRAM MINIMUMS AND MAXIMUMS.
C
      L=1   ! MIN/MAX ARRAY MM PØINTER.
      J=31   ! PØINTER TØ HISTØGRAM ARRAY.
      K=0   ! HISTØGRAM ØFFSET BELØW ZERØ.
      N=150   ! NØ. ØF VALUES (-1) IN HISTØGRAM ARRAY.
   30 CØNTINUE
C
C        INITIALIZE MIN AND MAX VALUES AND LØØP INDEX I.
C
      MM(L)=0   ! MIN.
      MM(L+1)=0   ! MAX.
      I=J
   40 CØNTINUE
      IF(BUFFER(I).GT.0)GØ TØ 50
      I=I+1
      IF(I.GT.J+N)GØ TØ 80
      GØ TØ 40
   50 CØNTINUE
      MM(L)=I-J-K
   60 CØNTINUE
      MM(L+1)=I-J-K
   70 CØNTINUE
      I=I+1
      IF(I.GT.J+N)GØ TØ 80
      IF(BUFFER(I).GT.0)GØ TØ 60
      GØ TØ 70
   80 CØNTINUE
      IF(L.NE.1)GØ TØ 90
      L=7
      J=182
      K=30
      N=115
      GØ TØ 30
   90 CØNTINUE
      IF(L.NE.7)GØ TØ 100
      L=3
      J=327
      K=0
      N=150
      GØ TØ 30
  100 CØNTINUE
      IF(L.NE.3)GØ TØ 110
      L=9
      J=478
      K=30
      N=115
      GØ TØ 30
  110 CØNTINUE
C
```

```
C        REPEAT PRØCEDURE EXCEPT DØUBLE WØRD INTEGERS
C        ARE USED FØR THE WEIGHTED  PIXEL HISTØGRAMS.
C
      L=5
      J=2
      K=0
      N=150
  120 CØNTINUE
      MM(L)=0
      MM(L+1)=0
      I=J
  130 CØNTINUE
      IF(DIBUF(I).GT.0)GØ TØ 140
      I=I+1
      IF(I.GT.J+N)GØ TØ 170
      GØ TØ 130
  140 CØNTINUE
      MM(L)=I-J-K
  150 CØNTINUE
      MM(L+1)=I-J-K
  160 CØNTINUE
      I=I+1
      IF(I.GT.J+N)GØ TØ 170
      IF(DIBUF(I).GT.0)GØ TØ 150
      GØ TØ 160
  170 CØNTINUE
      IF(L.NE.5)GØ TØ 180
      L=11
      J=153
      K=30
      N=115
      GØ TØ 120
  180 CØNTINUE
      WRITE(PRNTR,4011)MM
 4011 FØRMAT(3(' BR ',I4.1X,I4,11X)/3(' GR ',I4,1X,I4,11X))
      RETURN
      END
```

4. OPERATIONS

4.1 OPERATOR'S GUIDE

4.1.1 IMUNLD PROGRAM

When the operator receives an IMUNLD batch run, he should perform the following steps:

1. Log onto the system under UIC[5,5].

2. Check system time and date for correct entry.

3. Mount the Image Unload tape (specified on the Run Request Form, Figure 4-1) without a write ring as a foreign tape.

4. Mount the AA RP04 disk (specified on the Run Request Form) on DB2:.

5. Check the IMUNLD card deck to ensure that the second card after the $CREATE IMUNLD.DAT card contains the tape drive ID that matches the drive on which the tape was mounted.

6. Load the card reader with the IMUNLD card deck and start the batch processor.

7. At the end of the run dismount the tape and disk, and return the card deck, Run Request Form (signed off), Image Unload tape, and printer listing to the user, and the disk to its locker.

4.1.2 FLDSTT PROGRAM

When the operator receives a FLDSTT batch run, he should perform the following steps:

1. Log onto the system under UIC[5,5].

2. Check system time and date for correct entry.

3. Mount the AA RP04 disk (specified on the Run Request Form) on DB2:.

AEROJET ( ) BENDIX ( ) G E 100 ( ) ANALOG ( ) DIGITAL GRAPHICS ( )	NAME	REQUEST/DPAR	SEQ. #	
	ORGANIZATION/PHONE	DATA		DATE

INPUT INFORMATION

TAPE IDENTIFICATION	MISSION	FLIGHT	SITE	TYPE
START	OTHER INFORMATION			
STOP				
START				
STOP				

OUTPUT INFORMATION

PRODUCT	FORMAT	FILM B&W/COLOR	TYPE
PROGRAM	VISICORDER	PRINTOUT	SPECIAL

REQUESTER REMARKS/JUSTIFICATION

OPERATOR REMARKS

FIGURE 4 - 1

RUN REQUEST FORM

4. Load the card reader with the FLDSTT card deck and start the batch processor.

5. At the end of the run, dismount the disk and return it to its locker, and return the card deck, Run Request Form (signed off), and printer listing to the user.

4.2 USER's GUIDE

4.2.1 BATCH PROCEDURES

4.2.1.1 IMUNLD Program

To run the IMUNLD program in batch, the user should perform the following steps:

1. Prepare the following deck of cards:

```
$JOB/NAME=IMUNLD/LIMIT=999/ACCOUNT=110 6/MCR
$CREATE IMUNLD.DAT
     Data Set 1
$EOD
$MCR REM RSXBAT
$RUN IMUNLD.TSK
$EOJ
```

Where Data Set 1 contains the three cards described below:
Card 1 - contains a 2 in column one to indicate that DB2: is the drive for the AA disk.

Card 2 - contains the input tape drive device ID in the first three columns, for example, MTØ or MT1. This card will be adjusted by the operator if he needs to use a different drive than the one specified.

Card 3 - Contains the tape number in the first six columns. This number will be checked against the number stored internally on the tape

2. Complete the NAME, ORGANIZATION/PHONE, DATE, TAPE IDENTIFICATION, OTHER INFORMATION (specify what AA RP04 disk to use) boxes, and check the PRINTOUT box on the Run Request Form (Figure 4-1).

3. Submit the Image Unload tape, the card deck, and the Run Request Form to Operations for running on the Image Processor.

4. After the program has been run, retrieve the tape, card deck, and printout, and sign the bottom of the Run Request Form to indicate receipt of the output.

4.2.1.2 FLDSTT Program

To run the FLDSTT program in batch, the user should perform the following steps:

1. Prepare the following deck of cards:

```
$JOB/NAME=FLDSTT/LIMIT=999/ACCOUNT=110 6/MCR
$MCR PIP FLDSTT.DAT;*/DE
$CREATE FLDSTT.DAT
     Data Set 2
$EOD
$MCR REM RSXBAT
$RUN FLDSTT.TSK
$EOJ
```

Where Data Set 2 contains the four to six cards described below:

Card 1 - contains, in the first thirty columns, the file ID where the crop code translations and brightness and greenness coefficients are stored. The file name is FSTLBL.DAT, but this card should include the specific device and user file directory to enable location of the desired file, for example, DB2:[131,ØØ1] FSTLBL.DAT.

Card 2 - contains, in the first thirty columns, the file ID where ground truth data are stored. After this file is assigned, this character string is modified and used to create the output files, therefore it must be very exact in format. For example, the file ID DB2:[131,ØØ1]161177353.GTØ is acceptable (the acquisition date must be in columns eighteen through twenty-two).

Card 3 - contains three fields:

Field 1 - columns one through four - contains the segment number to be processed. This is checked against the number stored internally on the ground truth file.

Field 2 - columns six and seven - contains the output report option flag. This is a two-digit decimal number specifying which output reports are desired. The number is the sum of the numbers assigned to the various reports. The Acquisition Summary report is always produced, whereas the Dot Pixel Statistics report (assigned number 1) and the Crop Code Summary report (assigned number 2) are optional.

Field 3 - columns nine and ten - contains a two-digit decimal number, greater than zero and less than sixteen, which specifies how many acquisition dates are to be processed during this run of the FLDSTT program. Note that each acquisition date processed will take one hour of elasped time at a minimum.

Cards 4,5, and 6 - contain fifteen fields which are the acquisition dates to be processed, represented in standard YYDDD format. On each card there are five five-character fields, starting in column one and skipping five spaces between each field. Cards 5 and 6 do not need to be present if they are not needed.

2. Complete the NAME, ORGANIZATION/PHONE, DATE, OTHER INFORMATION (specify what AA RP04 disk to use) boxes, and check the PRINTOUT box on the Run Request Form.

3. Submit the card deck and Run Request Form to Operations for running on the Image Processor.

4. After the program has been run, retrieve the card deck and printout, and sign the bottom of the Run Request Form to indicate receipt of the output.

4.2.2 ONLINE PROCEDURES

4.2.2.1 IMUNLD Program

To run the IMUNLD program from an interactive terminal, the user should sign on to UIC[110,6], have the proper tape and disk mounted, create the file IMUNLD.DAT containing Data Set 1 (described in 4.2.1.1), and execute IMUNLD.TSK.

4.2.2.2 FLDSTT Program

To run the FLDSTT program from an interactive terminal, the user should sign on to UIC[110,6], have the proper disk mounted, edit the data in Data Set 2 (described in 4.2.1.2) in file FLDSTT.DAT and execute FLDSTT.TSK.