80-10045

NASA CR-160366

JSC-16239

1979

NASA

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas 77058

Space and Life Sciences Directorate
Earth Observations Division

AS-BUILT DESIGN SPECIFICATION OF SOFTWARE FOR CLUSTER-BASED PROPORTION ESTIMATION

Job Order 76-662

(E80-10045) AS-BUILT DESIGN SPECIFICATION OF SOFTWARE FOR CLUSTER-BASED PROPORTION ESTIMATION (Lockheed Electronics Co.) 174 p HC A08/MF A01 CSCL 05B

N80-16395

Unclas

G3/43 00045

Prepared by

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

Contract NAS 9-15800

October 1979

LEC-13934

1. Report No. JSC-16239	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle As-Built Design Specification of Software for Cluster-Based Proportion Estimation		5. Report Date October 1979
		6. Performing Organization Code
7. Author(s) Nim-yau Chu Lockheed Electronics Company, Inc.		8. Performing Organization Report No. LEC-13934
9. Performing Organization Name and Address Lockheed Electronics Company, Inc. Systems and Services Division Houston, Texas 77058		10. Work Unit No.
		11. Contract or Grant No. NAS 9-15800
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 J. M. Sulester, Technical Monitor		13. Type of Report and Period Covered Design specification
		14. Sponsoring Agency Code

15. Supplementary Notes

Jame
5F 12

16. Abstract

Nine dot allocation and labeling programs, two conversion programs, and a utility subroutines package have been developed for use in testing, evaluating, and predicting the performance of Procedure 1 replacements. These dot allocation and labeling schemes can be tested in combination with cluster maps produced by such clustering algorithms as CLASSY, AMOEBA, and ISOCLS to determine if unsupervised clustering followed by such proportion estimation strategies will efficiently provide accurate proportion estimates.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Bayesian dot allocation Dot labeling Majority rule Proportion estimation Proportional dot allocation	Unclassified-Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	174	

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

JSC Form 1424 (Rev Nov 75)

NASA — JSC

AS-BUILT DESIGN SPECIFICATION OF
SOFTWARE FOR CLUSTER-BASED PROPORTION ESTIMATION

Job Order 76-662

Prepared by

Nim-yau Chu

Approved by

John I. Morrow, Supervisor
Scientific Applications Software Section

Lockheed Electronics Company, Inc.

Systems and Services Division

Houston, Texas

Contract NAS 9-15800

for

Earth Observations Division
Space and Life Sciences Directorate
National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas

October 1979 LEC-13934

CONTENTS

1. PURPOSE

The purpose of the software for cluster-based proportion estimation is to provide a standard tool for testing, evaluating, and predicting the performance of Procedure 1 replacements.

In the Supporting Research Branch of the Earth Observations Division there are two projects underway to investigate new techniques to classify Landsat data for crop inventory purposes. These projects are based on different philosophies concerning the way in which an efficient inventory might be conducted. One approach, which supports the current corn and soybeans project, is to determine if increasing the number of type 1 dots in Procedure 1 will decrease the variance of the classifier to the point that machine classification will provide a more accurate proportion estimate. The second approach, which supports a longer range effort to arrive at a replacement for Procedure 1, will study the performance of various clustering algorithms in combination with various stratified area estimations or cluster-labeling techniques. The attempt is to determine if unsupervised clustering followed by various proportion estimation strategies (such as cluster labeling or proportion allocation based on dots within clusters) will provide more efficiently proportion estimates having acceptable accuracy. The software described in this document is used to evaluate proposed replacements of Procedure 1 and thus supports the second approach.*

In Procedure 1, dots are labeled by an analyst-interpreter. In the tests performed by this software system, the computer labels the dots using ground truth. Thus, the variability resulting from the analysts is eliminated. In a normal Procedure 1 operation, dots are chosen from a grid. In these simulations, dots are chosen pseudorandomly from clusters. The clusters are formed

*This software was developed in the Test and Evaluation Group, P. J. Aucoin, Jr., Group Leader.

using one of three algorithms. Each pixel of a cluster is equally likely to be selected as a dot. On the basis of cluster size and label (or dot label), the proportion estimate of small grain for the area is then computed and compared with the ground truth proportion.

The implemented system consists of a total of nine different dot allocation and labeling schemes. One purpose of implementing these schemes is to test them in combination with cluster maps produced by various clustering algorithms. Currently, three cluster algorithms are being tested: CLASSY, AMOEBA, and ISOCLS. The objective is to answer the following questions:

a. Which dot allocation and labeling scheme is the most effective?

b. Which of the various clustering algorithms is most effective?

c. What are the adverse effects if fewer dots are used?

A way to answer these questions is to study the statistical properties of the proportion estimates. The system has a built-in function that allows a user to select a different pseudorandom sequence for dot allocation. By repeating a dot allocation and labeling scheme over a reasonably large number of different pseudorandom sequences, the following statistics may be obtained:

a. The bias, mean squared error (MSE), reduction in MSE, average, variance, and variance reduction of an estimate with respect to the ground truth proportion

b. The above statistics as functions of the number of dots allocated

c. The mean and variance of the number of dots allocated in cases where this number varies

Another purpose of this software is to provide a basic library of routines to be used to construct new dot allocation and labeling schemes for future testing.

To interpret and analyze the proportion estimates is beyond the scope of this document. The specification, however, does establish the baseline configuration of the software for cluster-based proportion estimation.

2. SCOPE

The system contains 11 independent programs (2 for conversions and 9 for dot allocation and labeling) and 1 utility subroutines package. These programs are written to be executed in an interactive environment on the PDP 11/45 image or support processor. However, batch versions are available. (A description of the hardware used is outside the scope of this document.) Table 2-1 lists the independent programs as they are named in their Fortran source files and in the executable task files.

2.1 CONVERSION PROGRAMS

Two conversion programs have been implemented: one for converting or stripping the available cluster map file, the other for the ground truth map file. The acceptable format of input files is Landsat Universal format, Fortran readable and sequentially accessible. The outputs are "stripped" files, Fortran readable and directly accessible. The stripped files are more efficient to access because they are only about half the size of their parent files and the direct access operation skips unnecessary disk inputs and outputs (I/O's). The name convention is as follows:

a. For cluster maps, the file element is replaced by "STP" (stripped). For example, cluster map file, 100519101.DT2 is stripped to 100519101.STP.

b. For ground truth map files, only the first four characters are used. The file element is replaced by "STP." For example, a ground truth file, 100577278.GT0 is converted to 1005.STP.

TABLE 2-1.– SOFTWARE FOR CLUSTER-BASED PROPORTION ESTIMATION

Fortran file	Task file	Comment
A10.FTN	A81.TSK	To "strip" down a ground truth map file
A11.FTN	A86.TSK	To "strip" down a cluster map file
A01.FTN	A82.TSK (A82B.TSK)	Proportional dot allocation
A08.FTN	A83.TSK	Proportional dot allocation, majority rule labeling
A09.FTN	A84.TSK	Bayesian dot allocation (uniform prior)
A12.FTN	A85.TSK*	Bayesian dot allocation (no prior)
A13.FTN	A87.TSK* (A87B.TSK)	Bayesian dot allocation (quadratic prior)
A16.FTN	A89.TSK*	Bayesian dot allocation (modified quadratic prior)
A18.FTN	A91.TSK*	Bayesian dot allocation (adaptive prior)
A15.FTN	A88.TSK	Bayesian dot allocation (uniform prior), majority rule labeling
A17.FTN	A90.TSK*	Non-Bayesian sequential dot allocation, majority rule labeling

*Not in original job order; added as requested.

2.2 DOT ALLOCATION AND LABELING PROGRAMS

There are nine independent programs representing nine different dot allocation and labeling schemes. These programs have a similar logic flow, accept the same input format, produce the same output format, and use the same utility subroutines package. A typical program consists of the following:

1. Interactive input of job parameters
2. Repetition capability to compute the statistics of a proportion estimate
3. Intermediate summary or grand summary or both
4. Resetting the pseudorandom number generator to any desired starting point
5. Optional detailed dot files
6. Optional status messages at the terminal

The available dot allocation and labeling schemes are listed in the Comment column of table 2-1.

2.3 UTILITY SUBROUTINES PACKAGE

A package consisting of utility subroutines for handling disk read, dot generation, and other basic operations is implemented. Table 2-2 lists the subroutines available. Reading the cluster map from its disk file is handled by subroutine CLMPCS (cluster map cluster sizes) and its other entry points CLMPXY (cluster map position – x,y) and CLMPLC (cluster map size – lines, columns). Reading the ground truth map from its disk file is handled by subroutine GTMPLB (ground truth map label). The selection of a dot and associated bookkeeping is done by subroutine GETDOT. The subroutine MR (majority rule) finds a majority label on a first-come-first-assigned basis. The subroutine LBLITP (label interpretation) interprets a given label and increments a counter if the label is a small grain.

TABLE 2-2.— CONTENT OF THE UTILITY SUBROUTINES PACKAGE

Subroutine	Entry	Usage	Comment
CLMPCS		Reads the cluster map, counts the total number of pixels, counts the number of clusters, counts how many pixels each cluster has, stores types of clusters. Internally, sets up a table: number of pixels as a function of line number and cluster number.	Must be called before calling CLMPXY
	CLMPXY	Returns the line and column number of a given dot in any cluster using the internal table set up previously.	
	CLMPLC	Sets the size of the cluster map.	Default to 117 lines by 195 columns
GETDOT		Chooses pseudorandomly with equal probability a pixel from a cluster and assigns it as a dot. A dot array containing previously generated dots is consulted in order to avoid double selection. Updates the dot array.	
	RANST (random start)	Sets the starting point of the pseudorandom sequence.	
GTMPLB		Returns the ground truth label of a pixel given its line and column position.	
MR		Returns a majority label for a group of labels.	
LBLITP		Interprets a given label as small grain or other. Increments a counter if the label is small grain.	

2.4 ORGANIZATION OF DOCUMENTATION

Each of the following sections (3-14) discusses one of the software components. The discussion begins with a description of the program, including its purpose and structure. This is followed by the program's linkage, interfaces, inputs, and outputs. A brief form of the algorithm is included, as is a flow chart. The section concludes with the listing of the program. Subprograms of the 10 programs having them are discussed in appendixes A-J.

2.5 APPLICABLE DOCUMENTS

An applicable document is the requirements, which were informally transmitted under TIRF 79-0009, March 1979.

3. A81: TO STRIP A GROUND TRUTH MAP

3.1 DESCRIPTION

This program converts the file format of a ground truth map to a form that permits faster data access. Several properties of the ground truth map are assumed: single channel, size 351 lines and 392 columns, written in Landsat Universal format, Fortran readable by sequential access.

The output is a stripped ground truth map file that is Fortran readable by direct access. It contains no header record, and only 117 data records. Each record contains only 196 bytes.

The conversion results in a reduction to one-third the number of lines and one-half the number of columns. The 6-to-1 pixel reduction is based on a first-come-first-assigned majority rule. The lack of a header record and trailing zero-filled bytes make the output file much smaller, about 8-percent the size of the input ground truth map file.

3.2 LINKAGE

A subroutine named MR6 is used to compute the majority label for every six pixels.

3.3 INTERFACE

N/A

3.4 INPUTS

The following should be entered at the terminal by the user (interactive) or by card image file (batch):

Card or line	Parameter	Format	Default input	Default value
1	Ground truth map file name	A13	None (However, if "STOP" is typed, program exits.)	

The following disk file is needed: the ground truth map file.

3.5 OUTPUTS

This program echoes the input specifications and outputs status messages.

A disk file (named with the first four characters of the input file name followed by .STP) is generated to contain the stripped ground truth map.

3.6 BRIEF ALGORITHM: TO STRIP A GROUND TRUTH MAP

1. Read ground truth map file name; if "STOP" is read, program stops.
2. Open the ground truth map file and the output file.
3. Repeat a through d for LINE = 1 to 117:
   a. Read three lines from ground truth map into a buffer. Sequential access.
   b. For K = 1 to 196, do (1) and (2):
      (1) Extract the pixels (i, j); i = 1, 2, 3;
      j = [(K - 1) * 2 + 1] to 2 * K.

(2) Call MR6 to compute the majority label for the six pixels.

c. Write on output file the 196 majority labels. Direct access.

d. If line count reaches a certain number, output a status message.

4. Write a job-ending message.

5. Close both input and output files.

6. Go back to 1.

3.7 LISTING

```
C------ PROGRAM NAME: A10.FTN (A81.TSK).
C------ CONVERTS GROUND TRUTH IMAGE (IN DISK FILE) OF
C------ SIZE 351 LINES BY 392 COLUMNS TO LACIE SEGMENT
C------ SIZE 117 LINES BY 196 COLUMNS.  MAJORITY RULE IS
C------ USED FOR EACH GROUP OF 6 SUBPIXELS.
C------ WRITTEN AND EDITTED BY NIM-YAU CHU.
C
      BYTE J1(3060),L1(2998),J2(540),L2(468),
     *     J3(540),L3(468),   J4(540),L4(468)
      BYTE IG(6),NSTOP
      BYTE NAME(15),NGT(10)
      EQUIVALENCE (J1(73),L1),(J2(73),L2),(J3(73),L3),(J4(73),L4)
      EQUIVALENCE (J4(71),LINEX)
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
      DATA NSTOP/'S'/
C
C
100   CONTINUE
      WRITE(8,201)
201   FORMAT(' PROGRAM:A10(A81.TSK). TO REDUCE GROUND TRUTH '/
     *     ' RESOLUTION. OUTPUT IN STRIPPED FORMAT'/
     *      ' TYPE GROUND TRUTH FILE NAME'/
     *      ' AAAAAAAAAAAAA')
      READ(7,202)(NAME(K),K=1,13)
202   FORMAT(13A1)
      IF(NAME(1).EQ.NSTOP) STOP
      WRITE(8,203)(NAME(K),K=1,13)
203   FORMAT(/' GROUND TRUTH FILE FROM ',13A1,/
     *        ' LABEL BY MAJORITY RULE...EXECUTION BEGINS...WAIT')
C
      OPEN(UNIT=2,NAME=NAME,TYPE='OLD',
     *     FORM='UNFORMATTED',READONLY,ERR=901)
      DO 241 K=1,4
241   NGT(K)=NAME(K)
      OPEN(UNIT=3,NAME=NGT,TYPE='NEW',FORM='UNFORMATTED',
     *     ACCESS='DIRECT',RECORDSIZE=49,MAXREC=117)
C
C
      READ(2)J1
      NLINE=117
      NCOL=196
      K3=2*NCOL+72
      K5=NLINE/6+1
C
      DO 231 L=1,NLINE
      READ(2)(J1(K),K=1,K3)
      READ(2)(J2(K),K=1,K3)
      READ(2)(J3(K),K=1,K3)
      DO 221 K=1,NCOL
      K1=(K-1)*2+1
      K2=K1+1
      IG(1)=L1(K1)
      IG(2)=L1(K2)
      IG(3)=L2(K1)
      IG(4)=L2(K2)
      IG(5)=L3(K1)
      IG(6)=L3(K2)
C     WRITE(8,99101)L,K,IG
99101 FORMAT(' L,K,IG=',15I4)
      CALL MR6(IG,L4(K))
221   CONTINUE
      WRITE(3'L)(L4(K),K=1,NCOL)
C     WRITE(8,99201)(J4(K),K=65,100)
99201 FORMAT(' OUPUT=',16I4)
      IF(MOD(L,K5).EQ.0) WRITE(8,225)L,(NGT(K),K=1,8)
225   FORMAT(I5,' LINES WRITTEN ON ', 8A1)
231   CONTINUE
```


ORIGINAL PAGE IS OF POOR QUALITY

```
C
      WRITE(8,351)NLINE,NCOL,(NGT(K),K=1,10)
351   FORMAT(' STRIPPED MAP OF SIZE',I4,' LINES BY',I4,' COLS',
     *        ' PRODUCED IN ',10A1,//,
     *        ' NEXT GROUND TRUTH MAP PLEASE. TO STOP, JUST TYPE STOP')
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      GOTO 100
C
901   WRITE(8,911)
911   FORMAT(' ERROR OCCURRED WHILE READING INPUT GROUND TRUTH FILE.')
991   STOP
      END
C
C
      SUBROUTINE MR6(IG,LABEL)
C------ JG (DIM 6) CONTAINS 6 G.T. SUBPIXELS
C------ ON RETURN, LABEL WILL CONTAIN THE MAJORITY RULE LABEL.
C
      BYTE IG(1),JG(6),LG(6),LABEL
C
C------ CLEAR BUFFER
      DO 201 I=1,6
201   LG(I)=0
      J1=1
      JG(1)=IG(J1)
      LG(J1)=1
C
C------ COUNT SUBPIXELS HAVEING SAME LABELS
      DO 251 I=2,6
      DO 221 J=1,J1
C     WRITE(8,99108)I,J,IG(I),JG(J),J1
99108 FORMAT(' I,J,IG,JG,J1=',10I4)
      IF(IG(I).EQ.JG(J)) GOTO 231
221   CONTINUE
      J1=J1+1
      JG(J1)=IG(I)
      LG(J1)=1
      GOTO 251
231   LG(J)=LG(J)+1
251   CONTINUE
C
C------ CHOOSE THE MAJORITY LABEL
      MAX=0
      DO 281 I=1,6
      IF(LG(I).LE.MAX) GOTO 281
      MAX=LG(I)
      J1=I
281   CONTINUE
      LABEL=JG(J1)
      RETURN
      END
```

4. A86: TO STRIP A CLUSTER MAP

4.1 DESCRIPTION

This program converts the file format of a cluster map to a form that permits faster data access. Several properties of the cluster map are assumed: single channel, size 117 lines and 196 columns, written in Landsat Universal format, Fortran readable by sequential access.

The output is a stripped cluster map file that is Fortran readable by direct access. It contains no header record, and only 117 data records. Each record contains only 196 bytes, corresponding to the 196 pixels of a line.

The lack of a header record and trailing zero-filled bytes make the output file much smaller, about half the size of the input cluster map file.

4.2 LINKAGE

N/A

4.3 INTERFACE

N/A

4.4 INPUTS

Fortran-formatted input of the following parameter is needed:

Card or line	Parameters	Format	Default input	Default value
1	Cluster map file name	A13	None (However, if "STOP" is typed, program exits.)	

The following disk file is needed: the cluster map file.

4.5 OUTPUTS

This program echoes the input specifications and outputs status messages.

A disk file (named with the first four characters of the input file name followed by .STP) is generated to contain the stripped cluster map.

4.6 BRIEF ALGORITHM: TO STRIP A CLUSTER MAP

1. Read cluster map file name; if "STOP" is read, program stops.
2. Open the cluster map file and the output file.
3. Repeat a through c for LINE = 1 to 117:
   a. Read from cluster map the appropriate number of bytes into a buffer. Sequential access.
   b. Write on output file the just read bytes. Direct access.
   c. If line count reaches a certain number, output a status message.
4. Write a job-ending message.
5. Close both input and output files.
6. Go back to 1.

4.7 LISTING

```
C----- PROGRAM A11.FTN (A86.TSK)
C----- TO STRIP A CLUSTER MAP IN UNIVERSAL FORMAT OF SIZE
C----- 117 LINES BY 196 COLS.
C----- OUTPUT IS A FORTRAN DIRECT-ACCESSIBLE FILE WITH NO HEADER
C----- RECORD NAME AS *.STP (I.E. SAME FILENAME BUT MUST HAVE
C----- FILE ELEMENT AS .STP ( STANDS FOR STRIPPED ).
C
      BYTE JX,NSTOP, NAME(17),NGT(17), JJ(400),LL(400)
      EQUIVALENCE (JJ(73),LL)
      DATA NGT(10)/'.'/,NGT(11)/'S'/,NGT(12)/'T'/,NGT(13)/'P'/
      DATA NSTOP/'S'/
C
100   CONTINUE
      WRITE(8,121)
121   FORMAT(' PROGRAM A11(A86.TSK). TO STRIP A CLUSTER MAP FILE.'
     *       /' INPUT CLUSTER MAP FILE NAME.'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
C
      IF(NAME(1).EQ.NSTOP) STOP
      DO 131 K=1,9
131   NGT(K)=NAME(K)
C
      OPEN(UNIT=2,NAME=NAME,READONLY,ACCESS='SEQUENTIAL',
     *     TYPE='OLD',FORM='UNFORMATTED')
      NLINE=117
      NCOL=196
      NWORD= (NCOL-1)/4 +1
      OPEN(UNIT=3,NAME=NGT,ACCESS='DIRECT',
     *    RECORDSIZE=NWORD,MAXREC=NLINE,
     *     TYPE='NEW',FORM='UNFORMATTED')
      READ(2)JX
      K1=NCOL+72
      K2= NLINE/6 +1
      DO 301 L=1,NLINE
      READ(2)(JJ(K),K=1,K1)
      WRITE(3'L)(LL(K),K=1,NCOL)
      IF(MOD(L,K2).EQ.0) WRITE(8,251)L,NGT
251   FORMAT(I5,' LINES WRITTEN ON ',17A1)
301   CONTINUE
C
      WRITE(8,351)NLINE,NCOL,NGT
351   FORMAT(' STRIPPED MAP OF SIZE',I4,' LINES BY',I4,' COLS',
     *       ' PRODUCED IN ',17A1,//,
     *     ' NEXT CLUSTER MAP PLEASE. TO STOP, JUST TYPE STOP')
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      GOTO 100
      END
```

5. A82: PROPORTIONAL DOT ALLOCATION

5.1 DESCRIPTION

Implemented in this piece of software is a scheme in which dot allocation is proportional and labeling is direct from ground truth.

Dots are selected from each cluster in a pseudorandom fashion with equal probability. The number of dots selected from cluster i is given by

$$n_i = n \frac{N_i}{N} \tag{5-1}$$

where n = total number of dots to be allocated to the entire scene

N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

The equation for the proportion estimate of small grain is

$$\hat{p}_{sg} = \sum_{i=1}^{m} \frac{x_i}{n_i} \frac{N_i}{N}$$

where m = total number of clusters

x_i = number of dots labeled as small grain

(notice $x_i \leq n_i$)

The algorithm for proportional dot allocation includes the following strategies:

a. If a cluster is too small to receive at least one dot, then this cluster is grouped with other small clusters at the end of the selection process. The number of dots for this mixture cluster is computed according to equation (1).

b. Since the right-hand side of equation (1) may be a fraction but the number of dots must be an integer, a rounding-off operation is applied to n_i.

c. Because all n_i are rounded off to the nearest integer, Σn_i may not be equal to n, the total number of dots to be allocated to the entire scene. To correct for this, dots will be added to or subtracted from the clusters, starting with the clusters that have the most dots.

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. An intermediate summary is printed.

Furthermore, the program permits entry of several values of n, the total number of dots to be allocated, at the job initialization stage. A grand summary is produced.

5.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAE, GETDOT, CLMPCS, GTMPLB, LBLITP
SAE (stratified area estimation)	GETDOT, CLMPCS, GTMPLB, LBLITP
GETDOT	RAN

5.3 INTERFACE

Interface with other routines is through the common block PRTFLG (print flag - 2 bytes), which is used to control the optional printing of dot files and other information.

5.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line	Parameters		Format	Default input	Default value
1	Cluster map file name		A13	None	
2	a.	Number of repetitions	I3	0 or blank	1
	b.	Starting point of first pseudorandom sequence	I5	0 or blank	10
	c.	Number of repetition printings	I3	0 or blank	5
3	a.	Number of "total number of dots"	I3	None	
	b.	1st total number of dots	I3	None	
	c.	2nd total number of dots	I3	None	
		⋮			
	r.	17th total number of dots	I3	None	
4	Number of status messages on terminal		I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

5.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report	Contents
Individual repetition	1. Processor header
	2. Ground truth small-grain proportion
	3. A table showing number of clusters, cluster name codes, cluster sizes, and number of dots assigned
	4. A dot file showing dots chosen, their x-y position, and their ground truth labels
Intermediate summary	1. Processor header
	2. A table showing the estimate and bias of each repetition
	3. Bias, MSE, reduction in MSE, average, variance, variance reduction
Grand summary	1. Processor header
	2. A table showing bias, MSE, reduction in MSE, average, variance, and variance reduction as functions of total number of dots

5.6 BRIEF ALGORITHM: FOR PROPORTIONAL DOT ALLOCATION AND DIRECT LABELING

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.

3. For each total number of dots specified, repeat a and b:

   a. Repeat (1) through (5) until all repetitions are finished:

      (1) Set starting point of the pseudorandom number generator.

      (2) Determine each cluster's dot allocation (proportional).

      (3) Select dots from the clusters.

      (4) Pick up small-grain labels.

      (5) Compute the proportion estimate.

   b. Compute bias, MSE, reduction in MSE, average, variance, and variance reduction, and print an intermediate summary.

4. Print a grand summary.

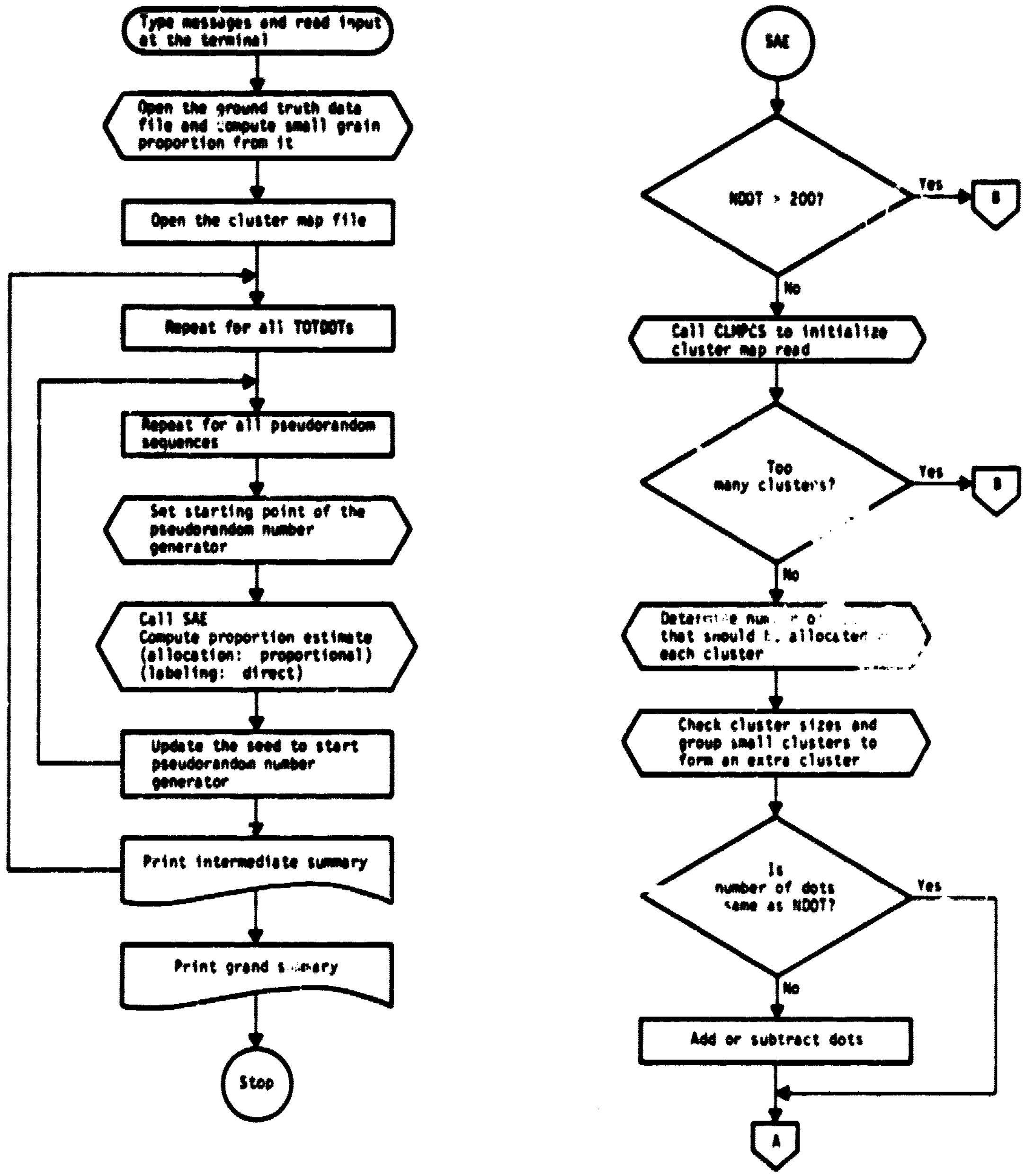

Figure 5-1.- Flow chart for proportional dot allocation.

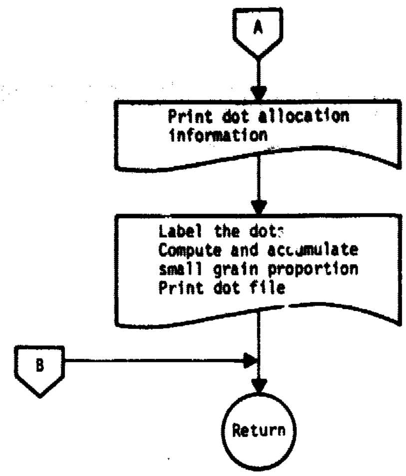

Figure 5-1.– Concluded.

5.7 LISTING

```
C----- PROGRAM A01 (A82.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING PROPORTIONAL DOT ALLOCATION
C----- LABELLING IS DIRECT BY GROUND TRUTH.
C
C
C
      BYTE NAME(15),NGT(13)
      INTEGER NDT(17)
      REAL PSG(200),PM(17),PB(17),RR(17)
C----- COMMON BLOCK IS A FLAG FOR PRINTING INDIVIDUAL RUNS
      COMMON /PRTFLG/JFLAG
C
C
C----- READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A01 (A82.TSK).'
     *       /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *       /' DOT ALLOCATION IS PROPORTIONAL.'
     *        /' LABELLING IS DIRECT BY GROUND TRUTH.'
     *       /' INPUT CLUSTLR MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
      DO 125 K=1,4
125   NGT(K)=NAME(K)
      NGT(5)='.'
      NGT(6)='S'
      NGT(7)='T'
      NGT(8)='P'
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *       /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *       /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *       /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT( ' YOU MAY SPECIFY MORE THAN ONE TOTAL DOT NO. (TOTDOT)'
     *       ,' FOR THE RUNS'/' HOW MANY TOTDOT S? INPUT THE TOTDOT S.'
     *       /, 18(1X,'III') )
      READ(7,142)NT,(NDT(K),K=1,NT)
142   FORMAT( 18(I3,1X) )
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL, HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED,NT,(NDT(K),K=1,NT)
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *        /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *        /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6,
     *       /' NO. OF TOTDOT S SPECIFIED = ',I2,
     *        '. THEY ARE:',/,2X,17I4 )
C
C----- COMPUTE PROPORTIONAL ESTIMATE FOR GROUND TRUTH
       NLINE=117
       NCOL=196
       OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *     ACCESS='DIRECT')
       DO 231 L=1,NLINE
       DO 221 K=1,NCOL
       CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
```

```
221     CONTINUE
231     CONTINUE
        P=FLOAT(IP)/NLINE/NCOL
C
C------ OPEN CLUSTER FILE
        OPEN(UNIT=1,NAME=NAME,TYPE='OLD',
      *     READONLY,ACCESS='DIRECT',FORM='UNFORMATTED')
        CALL CLMPLC(NLINE,NCOL)
C
C------ START ESTIMATION FOR EACH TOTDOT AND REPETITIONS
        JMES=0
        DO 361 K=1,NT
        JS=JSEED
        DO 331 J=1,JV
        JFLAG=0
        IF(MOD(J-1,JSKIP).EQ.0) JFLAG=1
        IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321     FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(4-25-79)',/,2X,
      * 'PROGRAM A01 (A82.TSK): ESTIMATION OF SMALL GRAINS PROPORTION'
      *       /,15X,'DOT ALLOCATION:PROPORTIONAL,'
      *       /,15X,'DOT  LABELLING:DIRECT BY GROUND TRUTH.'
      *      //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
      *       /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
        IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322     FORMAT(/,2X,' --- REPETITION RUN=',I3,
      *     ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
        CALL RANST(JSEED)
        CALL SAE(NDT(K),PSG(J))
        JMES=JMES+1
        IF(JMES.LE.NMES) WRITE(8,323)NDT(K),J,JSEED,PSG(J)
323     FORMAT(' TOTDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6,
      *     ' ESTIMATE=',F8.5)
        IF(JFLAG.EQ.1) WRITE(6,324)NDT(K),PSG(J)
324     FORMAT(/' FOR TOTAL DOT NO.=',I4,' PROPORTION ESTIMATE=',F8.5)
331     JSEED=JSEED+150
C
        WRITE(6,321)NAME,P
        WRITE(6,334)NDT(K)
334     FORMAT(/' --- SUMMARY OF REPETITION RUNS FOR TOTDOT NO.=',I4,
      *     ' ---',/,10X,
      *     'REPETITION  PSEDUO SEQ  SMALL GRAIN   BIAS',/,10X,
      *     '    RUN         SEED      ESTIMATE   WRT G.T.')
C------ COMPUTE BIAS AND M.S.E.
        PB(K)=0.
        PM(K)=0.
        DO 341 J=1,JV
        TEMP=PSG(J)-P
        PB(K)=PB(K)+TEMP
        PM(K)=PM(K)+TEMP**2
        WRITE(6,336)J,JS,PSG(J),TEMP
336     FORMAT(13X,I3,6X,I6,5X,F8.5,3X,F9.6)
341     JS=JS+150
        PB(K)=PB(K)/JV
        PM(K)=PM(K)/JV
        AVERG=PB(K)+P
        IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
        IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
C------ COMPUTE VARIANCE REDUCTION
        RR(K)=PM(K)/( P*(1.-P)/NDT(K) )
        REDVAR=VAR/( P*(1.-P)/NDT(K) )
        WRITE(6,357)PB(K),PM(K),RR(K),AVERG,VAR,REDVAR
357     FORMAT(/'     BIAS=',F10.6,'    M.S.E.=',F10.6,' REDUCTION=',F10.6,
      *      /' AVERAGE=',F10.6,' VARIANCE=',F10.6,' REDUCTION=',F10.6)
361     CONTINUE
C
C------ PRINT GRAND SUMMARY FOR THIS JOB
        WRITE(6,321)NAME,P
        WRITE(6,371)NT,JV
```

```
371    FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *         /,15X,' NO. OF DIFFERENT TOTDOT S =',I3,
     *        /,14X,' NO. OF REPETITION RUNS PER TOTDOT =',I3,
     *        //,1X,'TOTDOT    BIAS       M.S.E.   REDUCTION',
     *          '   AVERAGE  VARIANCE  VAR REDUCTION')
       DO 381 K=1,NT
       AVERG=PB(K)+P
       IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
       IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
       REDVAR=VAR/( P*(1.-P)/NDT(K) )
381    WRITE(6,382)NDT(K),PB(K),PM(K),RR(K),AVERG,VAR,REDVAR
382    FORMAT(2X,I3, 2X,F9.5, 4(F10.6), 3X,F10.6)
C
C
       WRITE(6,401)
401    FORMAT('1 ----- END OF THIS JOB ---')
C
       STOP
       END
C
C
       SUBROUTINE SAE(NDOT,PSG)
C----- STRATIFIED AREAL ESTIMATION USING PROPORTIONAL DOT ALLOCATION.
C----- GIVEN ANY TOTOAL NO. OF DOTS IN NDOT, PROGRAM WILL RETURN
C----- THE PROPORTIONAL ESTIMATE IN PSG. NOTICE THE FOLLOWING FEATURES:
C-----  (1) ALTHOUGH DOTS ARE ALLOCATED PROPORTIONAL TO CLUSTER SIZES,
C-----      ACTUAL NO. OF DOTS ARE ROUND OFF.  THOSE CLUSTERS GET LESS
C-----      THAN ONE DOT ARE GROUPED TOGETHER AND DOTS ARE REALLOCATED
C-----      TO THIS EXTRA CLUSTER.  AFTER THIS, DEDUCT FROM OR ADD DOTS
C-----      TO LARGEST CLUSTERS TO MAKE FINAL NO. OF DOTS ALLOCATED
C-----      EQUAL TO NO. OF NOTS DESIRED (NDOT).
C-----  (2) THE RANDOM NO. GENERATOR CAN BE SET TO ANY STARTING POINT
C-----      BY  CALL RANST(ISTART).  DOT GENERATION WILL BE USING
C-----      THE ISTART TH RANDOM NUMBER.
C-----      IF NO CALL TO RANST IS MADE, RANDOM GENERATOR ALWAYS STARTS
C-----      FROM J1=0 AND J2=0.
C
C----- WRITEN AND EDITED BY N.Y. CHU ON 4-2-79.
C
C
       INTEGER MM(51),NN(51),NDARY(200)
       INTEGER LL(51), JJ(10)
       BYTE LABEL
C----- COMMON BLOCK IS A FLAG FOR PRINTING INDIVIDUAL RUNS
       COMMON /PRTFLG/JFLAG
       DATA NSTART/0/
C
       IF(NDOT.GT.200) GOTO 911
C
C----- GET PIXEL COUNT FOR EACH CLUSTER
       IF(NSTART.NE.0) GOTO 121
       NSTART=1
       CALL CLMPCS(NPIXEL,M,MM,LL)
       IF(M.GE.51) GOTO 901
121    CONTINUE
C
C----- DETERMINE NO. OF DOTS FOR EACH CLUSTER
C
C----- BEGIN PROPORTIONAL DOT ALLOCATION
C
       K1=0
       DO 211 K=1,M
       NN(K)=FLOAT(NDOT)*MM(K)/NPIXEL+0.5
       K1=K1+NN(K)
C      WRITE(6,99101)K,NPIXEL,MM(K),NN(K)
99101  FORMAT(' K,NPIXEL=',2I4,' MM,NN=',2I4)
211    CONTINUE
```

```
C
C------ CHECK AND GROUP THOSE CLUSTERS GETTING NO DOT
      K2=0
      K3=0
      DO 221 K=1,M
      IF(NN(K).GT.0) GOTO 221
      K2=K2+MM(K)
      K3=K3+1
221   CONTINUE
      K3=M-K3
      MX=M
C------ CREATE AN EXTRA CLUSTER AND AS THE (M+1) TH CLUSTER
      IF(K2.EQ.0) GOTO 231
      K=NDOT*FLOAT(K2)/NPIXEL+0.5
      IF(K.EQ.0) GOTO 231
      MX=MX+1
      MM(MX)=K2
      NN(MX)=K
      K1=K1+K
231   CONTINUE
C
C
C------ CHECK IF NO. OF ASSIGNED DOTS EQUAL TO THE SPECIFIED
      IF(NDOT-K1)241,281,245
241   INC=-1
      GOTO 249
245   INC=1
249   CONTINUE
C
C------ ADD TO OR DEDUCT FROM LARGE CLUSTERS TO
C------ MAKE NO. OF DOTS ASSIGNED EQUAL TO THE SPECIFIED
      K1=IABS(NDOT-K1)
      L=0
250   IF(K1.EQ.0) GOTO 281
      L=L+1
      K1=K1-1
      ITEMP=0
      DO 261 K=1,M
      IF(L.EQ.1) GOTO 252
      K2=L-1
      DO 251 J=1,K2
      IF(K.EQ.JJ(J)) GOTO 261
251   CONTINUE
252   IF(NN(K).LE.ITEMP) GOTO 261
      ITEMP=NN(K)
      JJ(L)=K
261   CONTINUE
      NN(JJ(L))=NN(JJ(L))+INC
C     WRITE(6,99106)L,NDOT,JJ(L),NN(JJ(L))
99106 FORMAT(' 99106---  L,NDOT,JJ(L),NN=',5I4)
      IF(L.EQ.K3) L=0
      GOTO 250
281   CONTINUE
C
C----- WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)NDOT,NPIXEL,MX
282   FORMAT(//,5X,'            TOTAL NO. OF DOTS AVAILABLE =',I6,
     *          /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *          /,5X,'                  TOTAL NO. OF CLUSTERS =',I6,
     *          /,10X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS'
     *          /,10X,'   NO.      CODE  PIXELS   ASSIGNED')
      DO 283 K=1,MX
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K)
283   CONTINUE
284   FORMAT(13X,I3,5X,I4,3X,I6,4X,I3)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
```

```
285   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286   FORMAT(/'L THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *        ') ARE:', 3(/,16(I3,',')) )
288   CONTINUE
C
C
C------ END OF PROPORTIONAL DOT ALLOCIATION
C
C     WRITE(6,99005)(NN(K),K=1,MX)
99005 FORMAT(' NN=',15I4)
C
C------ FOR EACH CLUSTER, ASSIGN DOTS
      ND=1
      DO 291 K=1,MX
      K1=NN(K)
      IF(K1.EQ.0) GOTO 291
      DO 289 L=1,K1
289   CALL GETDOT(MM(K),L,NDARY(ND),K3)
99191 KK1=ND+K1-1
C     WRITE(6,99192)(NDARY(L),L=ND,KK1)
99192 FORMAT(' NDARY=',15I4)

291   ND=ND+K1
C
C------- FOR EACH CLUSTER, FIND LINE AND COL NO. OF EACH ASSIGNED DOTS
      IF(JFLAG.EQ.1) WRITE(6,295)
295   FORMAT(//'                     --- DOT FILE ---'
     *        /' CLUSTER     DOT NUMBER    POSITION   GROUND TRUTH LABEL'
     *        /'    NO.    WRT ITS CLUSTER LINE,COL         RAW, CODE')
      PSG=0.
      ND=1
      DO 331 K=1,MX
      K1=NN(K)
      IF(K1.EQ.0) GOTO 331
      ISG=0
      DO 321 L=1,K1
      K3=NDARY(ND)
C------ TO TEST FOR THE EXTRA CLUSTER
      IF(K.LE.M) GOTO 311
C------ ALG. TO GET CORRECT LABEL FOR THE EXTRA CLUSTER
      DO 305 J=1,M
      IF(NN(J).NE.0) GOTO 305
      IF(K3.LE.MM(J)) GOTO 307
      K3=K3-MM(J)
305   CONTINUE
307   CALL CLMPXY(J,K3,NDL,NDC,LL)
      CALL GTMPLB(NDL,NDC,LABEL)
      CALL LBLITP(LABEL,LB1,ISG)
      IF(JFLAG.EQ.1) WRITE(6,300)J,K3,NDL,NDC,LABEL,LB1
308   FORMAT(3X,I2,8X,I6,7X,I3,I4,5X,I4,I5)
      GOTO 321
311   CALL CLMPXY(K,K3,NDL,NDC,LL)
      CALL GTMPLB(NDL,NDC,LABEL)
      CALL LBLITP(LABEL,LB1,ISG)
      IF(JFLAG.EQ.1) WRITE(6,308)K,K3,NDL,NDC,LABEL,LB1
321   ND=ND+1
C------ COMPUTE S.G. ESTIMATE
      PSG=PSG+ FLOAT(ISG)/K1*MM(K)/NPIXEL
331   CONTINUE
      GOTO 990
C
901   WRITE(6,902)M
902   FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
911   WRITE(6,912)NDOT
914   FORMAT(' YOU SPECIFIED',I2,' TOO MANY DOTS(MAX=200)')
C
990   RETURN
      END
```

6. A83: PROPORTIONAL DOT ALLOCATION, MAJORITY RULE LABELING

6.1 DESCRIPTION

Implemented in this piece of software is a scheme in which dot allocation is proportional and labeling is by ground truth using majority rule.

Dots are selected from each cluster in a pseudorandom fashion with equal probability. The number of dots selected from a cluster i is given by

$$n_i = n \frac{N_i}{N} \qquad (6\text{-}1)$$

where n = total number of dots to be allocated to the entire scene

N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

For each cluster, all the dot labels are compared, then the cluster is labeled using majority rule.

The equation for the proportion estimate of small grain is

$$\hat{P}_{sg} = \sum_{\substack{\text{cluster } i \\ \text{labeled as} \\ \text{small grain}}} \frac{N_i}{N} \qquad (6\text{-}2)$$

The algorithm for proportional allocation includes the following strategies:

a. If a cluster is too small to receive at least one dot, then this cluster is grouped with other small clusters at the end of the selection process. The number of dots for this mixture cluster is computed according to equation (1).

b. Since the right-hand side of equation (1) may be a fraction but the number of dots must be an integer, a rounding-off operation is applied to n_i.

c. Because all n_i are rounded off to the nearest integer, Σn_i may not be equal to n, the total number of dots to be allocated to the entire scene. To correct for this, dots will be added to or subtracted from the clusters, starting with the clusters that have the most dots.

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. An intermediate summary is printed.

Furthermore, the program permits entry of several values of n, the total number of dots to be allocated, at the job initialization stage. A grand summary is produced.

6.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAEMR, GETDOT, CLMPCS, GTMPLB, LBLITP
SAEMR	GETDOT, CLMPCS, GTMPLB, LBLITP, MR
GETDOT	RAN

6.3 INTERFACE

Interface with other routines is through the common block PRTFLG (2 bytes), which is used to control the optional printing of dot files and other information.

6.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line		Parameters	Format	Default input	Default value
1	Cluster map file name		A13	None	
2	a.	Number of repetitions	I3	0 or blank	1
	b.	Starting point of first pseudorandom sequence	I5	0 or blank	10
	c.	Number of repetition printings	I3	0 or blank	5
3	a.	Number of "total number of dots"	I3	None	
	b.	1st total number of dots	I3	None	
	c.	2nd total number of dots	I3	None	
		⋮			
	r.	17th total number of dots	I3	None	
4	Number of status messages on terminal		I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

6.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report	Contents
Individual repetition	1. Processor header
	2. Ground truth small-grain proportion
	3. A table showing number of clusters, cluster name codes, cluster sizes, and number of dots assigned
	4. A dot file showing dots chosen, their x-y position, and their ground truth labels
	5. A table showing the majority labels for the clusters
Intermediate summary	1. Processor header
	2. A table showing the estimate and bias of each repetition
	3. Bias, MSE, reduction in MSE, average, variance, variance reduction
Grand summary	1. Processor header
	2. A table showing bias, MSE, reduction in MSE, average, variance, and variance reduction as functions of total number of dots

6.6 BRIEF ALGORITHM: FOR PROPORTIONAL DOT ALLOCATION AND MAJORITY RULE LABELING

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.
3. For each total number of dots specified, repeat a and b:
   a. Repeat (1) through (6) until all repetitions are finished:
      (1) Set starting point of the pseudorandom number generator.
      (2) Determine each cluster's dot allocation (proportional).
      (3) Select dots from the clusters.
      (4) Pick up small-grain labels.
      (5) Find the majority label for each cluster.
      (6) Compute the proportion estimate according to (5).
   b. Compute bias, MSE, reduction in MSE, average, variance, and variance reduction, and print an intermediate summary.
4. Print a grand summary.

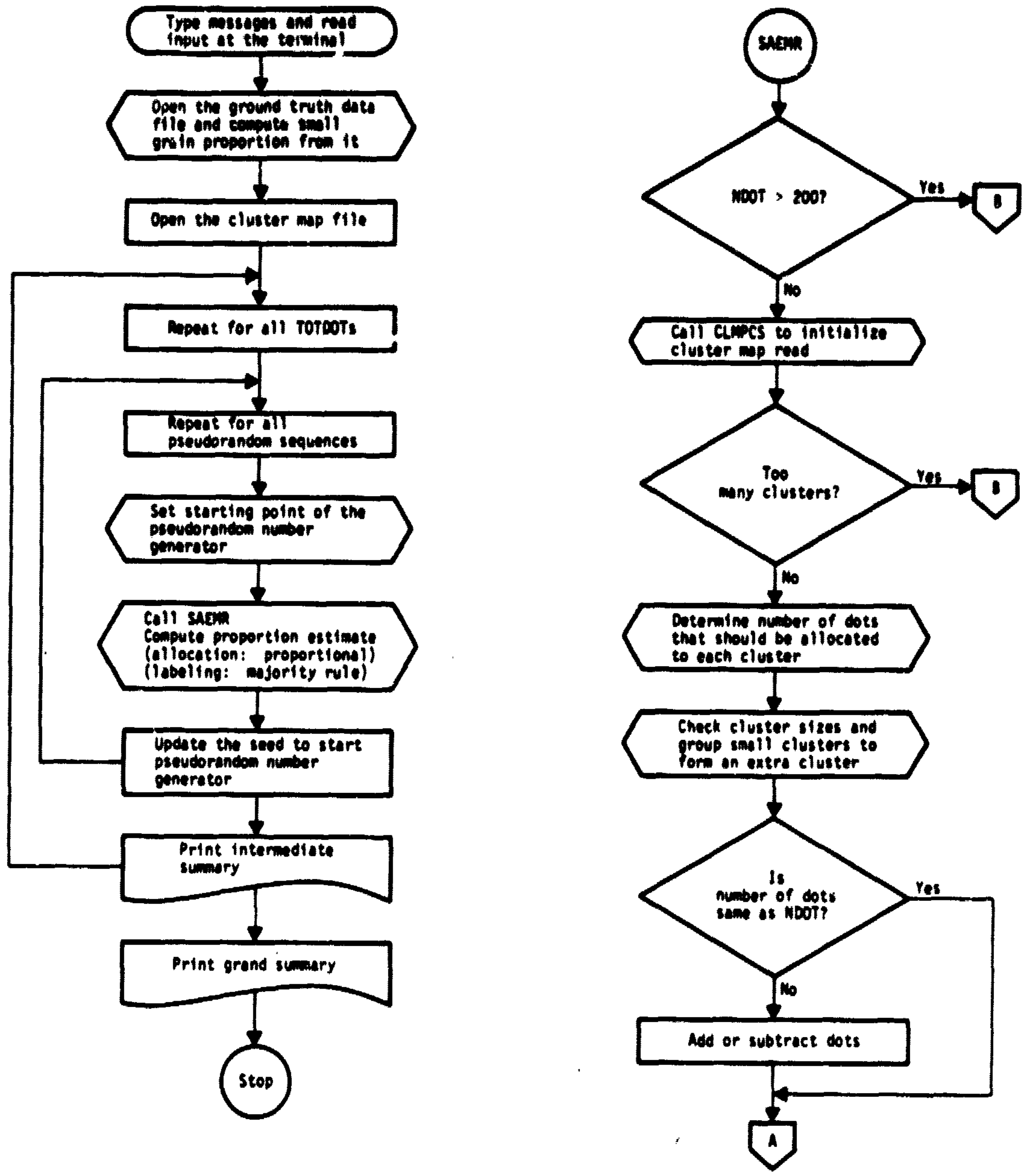

Figure 6-1.— Flow chart for proportional dot allocation using majority rule labeling.

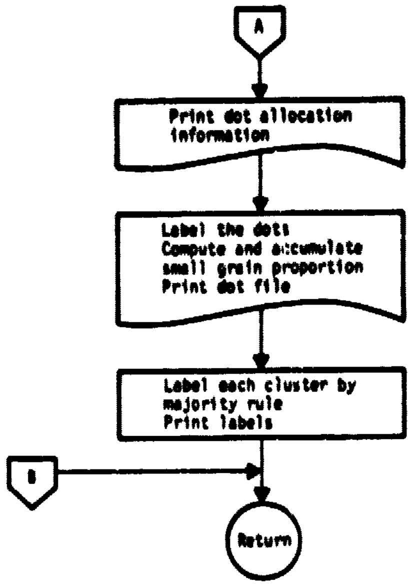

Figure 6-1.– Concluded.

6.7 LISTING

```
C
C----- PROGRAM A08(A83.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- DOT ALLOCATION IS PROPORTIONAL
C----- DOT LABELLING IS MAJORITY RULE BY GROUND TRUTH
C
C
C
      BYTE NAME(15),NGT(13)
      INTEGER NDT(17)
      REAL PSG(200),PM(17),PB(17),RR(17)
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C----- READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A08(A83.TSK).'
     *        /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *        /' DOT ALLOCATION IS PROPORTIONAL.'
     *         /' LABELLING IS MAJORITY RULE FROM GROUND TRUTH.'
     *        /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
      DO 125 K=1,4
125   NGT(K)=NAME(K)
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *        /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *        /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *        /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT( ' YOU MAY SPECIFY MORE THAN ONE TOTAL DOT NO. (TOTDOT)'
     *        ,' FOR THE RUNS'/' HOW MANY TOTDOT S? INPUT THE TOTDOT S.'
     *        /, 18(1X,'III') )
      READ(7,142)NT,(NDT(K),K=1,NT)
142   FORMAT( 18(I3,1X) )
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL. HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED,NT,(NDT(K),K=1,NT)
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *         /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *         /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6,
     *        /' NO. OF TOTDOT S SPECIFIED = ',I2,
     *         '. THEY ARE',/,2X,17I4 )
C
C----- COMPUTE PROPORTIONAL ESTIMATE FOR GROUND TRUTH
       NLINE=117
       NCOL=196
       OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *      ACCESS='DIRECT')
       DO 231 L=1,NLINE
       DO 221 K=1,NCOL
       CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
```

```
221    CONTINUE
231    CONTINUE
       P=FLOAT(IP)/NLINE/NCOL
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *    READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
C
C------ START ESTIMATION FOR EACH TOTDOT AND REPETITIONS
      JMES=0
      DO 361 K=1,NT
      JS=JSEED
      DO 331 J=1,JV
      JFLAG=0
      IF(MOD(J,JSKIP).EQ.1) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(4-26-79)'
     *         /,2X,'PROGRAM A08: ESTIMATION OF SMALL GRAINS PROPORTION'
     *         /,15X,'DOT ALLOCATION:PROPORTIONAL,'
     *         /,15X,'DOT  LABELLING:MAJORITY RULE BY GROUND TRUTH.'
     *        //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *         /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' --- REPETITION RUN=',I3,
     *         ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAEMR(NDT(K),PSG(J))
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)NDT(K),J,JSEED,PSG(J)
323   FORMAT(' TOTDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6,
     *         ' ESTIMATE=',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,324)NDT(K),PSG(J)
324   FORMAT(/' FOR TOTAL DOT NO.=',I4,' PROPORTION ESTIMATE=',F8.5)
331   JSEED=JSEED+150
C
      WRITE(6,321)NAME,P
      WRITE(6,334)NDT(K)
334   FORMAT(/' --- SUMMARY OF REPETITION RUNS FOR TOTDOT NO.=',I4,
     *         ' ---',/,10X,
     *         'REPETITION  PSEDUO SEQ  SMALL GRAIN   BIAS',/,10X,
     *         '    RUN         SEED       ESTIMATE   WRT G.T.')
C------ COMPUTE BIAS AND M.S.E.
      PB(K)=0.
      PM(K)=0.
      DO 341 J=1,JV
      TEMP=PSG(J)-P
      PB(K)=PB(K)+TEMP
      PM(K)=PM(K)+TEMP**2
      WRITE(6,336)J,JS,PSG(J),TEMP
336   FORMAT(13X,I3,6X,I6,5X,F8.5,3X,F9.6)
341   JS=JS+150
      PB(K)=PB(K)/JV
      PM(K)=PM(K)/JV
      AVERG=PB(K)+P
      IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
      IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
C------ COMPUTE VARIANCE REDUCTION
      RR(K)=PM(K)/( P*(1.-P)/NDT(K) )
      REDVAR=VAR/( P*(1.-P)/NDT(K) )
      WRITE(6,357)PB(K),PM(K),RR(K),AVERG,VAR,REDVAR
357   FORMAT(/'     BIAS=',F10.6,'    M.S.E.=',F10.6,' REDUCTION=',F10.6,
     *        /' AVERAGE=',F10.6,' VARIANCE=',F10.6,' REDUCTION=',F10.6)
361   CONTINUE
C
C------ PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)NT,JV
```

```
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *        /,15X,' NO. OF DIFFERENT TOTDOT S =',I3,
     *       /,14X,' NO. OF REPETITION RUNS PER TOTDOT =',I3,
     *       //,1X,'TOTDOT   BIAS      M.S.E.   REDUCTION',
     *            '   AVERAGE  VARIANCE  VAR REDUCTION')
      DO 381 K=1,NT
      AVERG=PB(K)+P
      IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
      IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
      REDVAR=VAR/( P*(1.-P)/NDT(K) )
381   WRITE(6,382)NDT(K),PB(K),PM(K),RR(K),AVERG,VAR,REDVAR
382   FORMAT(2X,I3, 2X,F9.5, 4(F10.6), 3X,F10.6)
C
C
      WRITE(6,401)
401   FORMAT('1 ----- END OF THIS JOB ---')
C
      STOP
      END
C
C
      SUBROUTINE SAEMR(NDOT,PSG)
C------ STRATIFIED AREAL ESTIMATION USING PROPORTIONAL DOT ALLOCATION,
C------ AND MAJORITY LABELLING BY GROUND TRUTH.
C------ GIVEN ANY TOTOAL NO. OF DOTS IN NDOT, PROGRAM WILL RETURN
C------ THE PROPORTIONAL ESTIMATE IN PSG. NOTICE THE FOLLOWING FEATURES:
C------  (1) ALTHOUGH DOTS ARE ALLOCATED PROPORTIONAL TO CLUSTER SIZES,
C------      ACTUAL NO. OF DOTS ARE ROUND OFF.  THOSE CLUSTERS GET LESS
C------      THAN ONE DOT ARE GROUPED TOGETHER AND DOTS ARE REALLOCATED
C------      TO THIS EXTRA CLUSTER.  AFTER THIS, DEDUCT FROM OR ADD DOTS
C------      TO LARGEST CLUSTERS TO MAKE FINAL NO. OF DOTS ALLOCATED
C------      EQUAL TO NO. OF NOTS DESIRED (NDOT).
C------  (2) THE RANDOM NO. GENERATOR CAN BE SET TO ANY STARTING POINT
C------      BY  CALL RANST(ISTART).  DOT GENERATION WILL BE USING
C------      THE ISTART TH RANDOM NUMBER.
C------      IF NO CALL TO RANST IS MADE, RANDOM GENERATOR ALWAYS STARTS
C------      FROM J1=0 AND J2=0.
C
C------ WRITEN AND EDITED BY N.Y. CHU ON 4-26-79.
C
C
      INTEGER MM(51),NN(51),NDARY(200),LL(51)
      INTEGER JJ(30),LG(60)
      BYTE LABEL,LBL(200),JG(60)
C------ COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/
C
      IF(NDOT.GT.200) GOTO 911
C
C------ GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C------ DETERMINE NO. OF DOTS FOR EACH CLUSTER
C
C------ BEGIN ROPORTIONAL DOT ALLOCATION
C
      K1=0
      DO 211 K=1,M
      NN(K)=FLOAT(NDOT)*MM(K)/NPIXEL+0.5
      K1=K1+NN(K)
211   CONTINUE
C
C------ CHECK AND GROUP THOSE CLUSTERS GETTING NO DOT
      K2=0
      K3=0
```

```
      DO 221 K=1,M
      IF(NN(K).GT.0) GOTO 221
      K2=K2+MM(K)
      K3=K3+1
221   CONTINUE
      K3=M-K3
      MX=M
C------ CREATE AN EXTRA CLUSTER AND AS THE (M+1) TH CLUSTER
      IF(K2.EQ.0) GOTO 231
      K=NDOT*FLOAT(K2)/NPIXEL+0.5
      IF(K.EQ.0) GOTO 231
      MX=MX+1
      MM(MX)=K2
      NN(MX)=K
      K1=K1+K
231   CONTINUE
C
C
C------ CHECK IF NO. OF ASSIGNED DOTS EQUAL TO THE SPECIFIED
      IF(NDOT-K1)241,281,245
241   INC=-1
      GOTO 249
245   INC=1
249   CONTINUE
C
C------ ADD TO OR DEUCT FROM LARGE CLUSTERS TO
C------ MAKE NO. OF DOTS ASSIGNED EQUAL TO THE SPECIFIED
      K1=IABS(NDOT-K1)
      L=0
250   IF(K1.EQ.0) GOTO 281
      L=L+1
      K1=K1-1
      ITEMP=0
      DO 261 K=1,M
      IF(L.EQ.1) GOTO 252
      K2=L-1
      DO 251 J=1,K2
      IF(K.EQ.JJ(J)) GOTO 261
251   CONTINUE
252   IF(NN(K).LE.ITEMP) GOTO 261
      ITEMP=NN(K)
      JJ(L)=K
261   CONTINUE
      NN(JJ(L))=NN(JJ(L))+INC
      IF(L.EQ.K3) L=0
      GOTO 250
281   CONTINUE
C
C------ WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)NDOT,NPIXEL,MX
282   FORMAT(//,5X,'            TOTAL NO. OF DOTS AVAILABLE =',I6,
     *         /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *         /,5X,'                   TOTAL NO. OF CLUSTERS =',I6,
     *         /,10X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS'
     *         /,10X,'   NO.     CODE  PIXELS   ASSIGNED')
      DO 283 K=1,MX
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K)
283   CONTINUE
284   FORMAT(13X,I2,5X,I4,3X,I6,4X,I3)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
285   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286   FORMAT(/'L THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *         ') ARE:', 3(/,16(I3,',')) )
288   CONTINUE
C
C
```

```
C----- END OF PROPORTIONAL DOT ALLOCIATION
C
C
C------ FOR EACH CLUSTER, ASSIGN DOTS
      ND=1
      DO 291 K=1,MX
      K1=NN(K)
      IF(K1.EQ.0) GOTO 291
      DO 289 L=1,K1
289   CALL GETDOT(MM(K),L,NDARY(ND),K3)

291   ND=ND+K1
C
C------- FOR EACH CLUSTER, FIND LINE AND COL NO. OF EACH ASSIGNED DOTS
      IF(JFLAG.EQ.1) WRITE(6,295)
295   FORMAT(//'                          --- DOT FILE ---'
     *       /' CLUSTER     DOT NUMBER    POSITION  GROUND TRUTH LABEL'
     *       /'   NO.    WRT ITS CLUSTER LINE,COL       RAW, CODE')
      PSG=0.
      ND=1
      DO 331 K=1,MX
      K1=NN(K)
      IF(K1.EQ.0) GOTO 331
      ISG=0
      K3=NDARY(ND)
C----- TO TEST FOR THE EXTRA CLUSTER
      IF(K.LE.M) GOTO 311
C------ ALG. TO GET CORRECT LBL(ND) FOR THE EXTRA CLUSTER
      DO 305 J=1,M
      IF(NN(J).NE.0) GOTO 305
      IF(K3.LE.MM(J)) GOTO 307
      K3=K3-MM(J)
305   CONTINUE
307   CALL CLMPXY(J,K3,NDL,NDC,LL)
      CALL GTMPLB(NDL,NDC,LBL(ND))
      CALL LBLITP(LBL(ND),LB1,ISG)
      IF(JFLAG.EQ.1) WRITE(6,308)J,K3,NDL,NDC,LBL(ND),LB1
308   FORMAT(3X,I2,8X,I6,7X,I3,I4,5X,I4,I5)
      GOTO 321
311   CALL CLMPXY(K,K3,NDL,NDC,LL)
      CALL GTMPLB(NDL,NDC,LBL(ND))
      CALL LBLITP(LBL(ND),LB1,ISG)
      IF(JFLAG.EQ.1) WRITE(6,308)K,K3,NDL,NDC,LBL(ND),LB1
321   ND=ND+1
331   CONTINUE
C
C----- COMPUTE MAJORITY LABEL FOR EACH CLUSTER
      IF(JFLAG.EQ.1) WRITE(6,421)
421   FORMAT(///' MAJORITY LABELLING RULE BY GROUND TRUTH:'/
     *       '     CLUSTER  NO. OF LABELS  CHOSEN LABEL')
      ND=1
      PSG=0.
      DO 471 K=1,MX
      IF(NN(K).EQ.0) GOTO 471
      CALL MR(LBL(ND),NN(K),LABEL,J1,JG,LG)
      ISG=0
      CALL LBLITP(LABEL,LB1,ISG)
      IF(ISG.EQ.1) PSG=PSG+FLOAT(MM(K))/NPIXEL
      IF(JFLAG.EQ.1) WRITE(6,451)K,J1,LABEL,LB1
451   FORMAT(5X,I3, 9X,I3, 8X,I4, 1X,I4)
      ND=ND+NN(K)
471   CONTINUE
      GOTO 990
C
901   WRITE(6,902)M
902   FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
911   WRITE(6,912)NDOT
912   FORMAT(' YOU SPECIFIED',I4,' TOO MANY DOTS(MAX=200)')
C
990   RETURN
      END
```

7. A84: BAYESIAN DOT ALLOCATION (UNIFORM PRIOR)

7.1 DESCRIPTION

Implemented in this piece of software is scheme in which dot allocation is sequentially Bayesian and labeling is direct from ground truth.

In the Bayesian dot allocation scheme, dots are allocated sequentially to those clusters that may reduce the variance of the segment proportion estimate the most. In order to do this, we must first compute the expected change in variance, $\Delta\sigma^2$, for each cluster. The expected change in variance for cluster i is defined as

$$\Delta\sigma_i^2 = \left(\frac{N_i}{N}\right)^2 \left[\frac{(n_i + 5)^2(n_i^2 + 7n_i + 8)}{(n_i - 1)n_i(n_i + 2)(n_i + 3)^2}\right](x_i + 1)(n_i - x_i + 1) \tag{7-1}$$

where N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

n_i = number of dots previously allocated to cluster i

x_i = number of dots previously allocated to cluster i which are labeled as small grain

(notice $x_i \leq n_i$)

Next, a dot is allocated to the cluster whose $\Delta\sigma_i^2$ is the largest. Then, for this chosen cluster, n_i is updated to $n_i + 1$. That dot's label is read from the ground truth file. If the label is small grain, x_i is updated to $x_i + 1$. After this, the segment variance, σ^2, is computed as

$$\sigma^2 = \sum_{i=1}^{m} \left(\frac{N_i}{N}\right)^2 P_i(1 - P_i) \frac{1}{n_i - 1} \tag{7-2}$$

where m = total number of clusters

$$P_i = \frac{x_i + 1}{n_i + 2} \tag{7-3}$$

This σ^2 is printed and the above allocation scheme is repeated until the number of dots allocated reaches a value specified by the user.

The proportion estimate is computed as

$$\hat{P}_{sg} = \sum_{i=1}^{m} \frac{x_i}{n_i} \frac{N_i}{N} \tag{7-4}$$

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. A grand summary is printed.

7.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAEB1, GETDOT, CLMPCS, GTMPLB, LBLITP
SAEB1	GETDOT, CLMPCS, GTMPLB, LBLITP
GETDOT	RAN

7.3 INTERFACE

Interface with other routines is through the common block PRTFLG (2 bytes), which is used to control the optional printing of dot files and other information.

7.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line	Parameters	Format	Default input	Default value
1	Cluster map file name	A13	None	
2	a. Number of repetitions	I3	0 or blank	1
	b. Starting point of first pseudorandom sequence	I5	0 or blank	10
	c. Number of repetition printings	I3	0 or blank	5
3	a. Maximum number of dots that can be allocated	I3	None	
	b. Number of dots initially assigned to each cluster	I3	None	
4	Number of status messages on terminal	I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

7.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report		Contents
Individual repetition	1.	Processor header
	2.	Ground truth small-grain proportion
	3.	A dot file showing dots chosen, their x-y position, their ground truth labels, segment variance, and proportion estimate
	4.	A table showing number of clusters, cluster name codes, cluster sizes, and number of dots assigned
Grand summary	1.	Processor header
	2.	A table showing bias, MSE, reduction in MSE, average, variance, and variance reduction as functions of number of dots

7.6 BRIEF ALGORITHM: FOR BAYESIAN DOT ALLOCATION (UNIFORM PRIOR)

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.
3. Repeat a through g until all repetitions are finished:
   a. Set starting point of the pseudorandom number generator.
   b. Clear a dot counter.
   c. Increment the dot counter.

d. If the dot counter indicates that the current dot should be allocated as an initial dot, then go to the next step. Otherwise compute $\Delta\sigma_i^2$ for every cluster and choose the cluster with maximum $\Delta\sigma_i^2$.

e. Allocate a dot to the chosen cluster.

f. Pick up the small-grain labels, compute segment variance and proportion estimate, and print a line of the dot file.

g. If the dot count does not exceed specified value, go to c.

4. Compute bias, MSE, reduction in MSE, average, variance, and variance reduction, and print a grand summary.

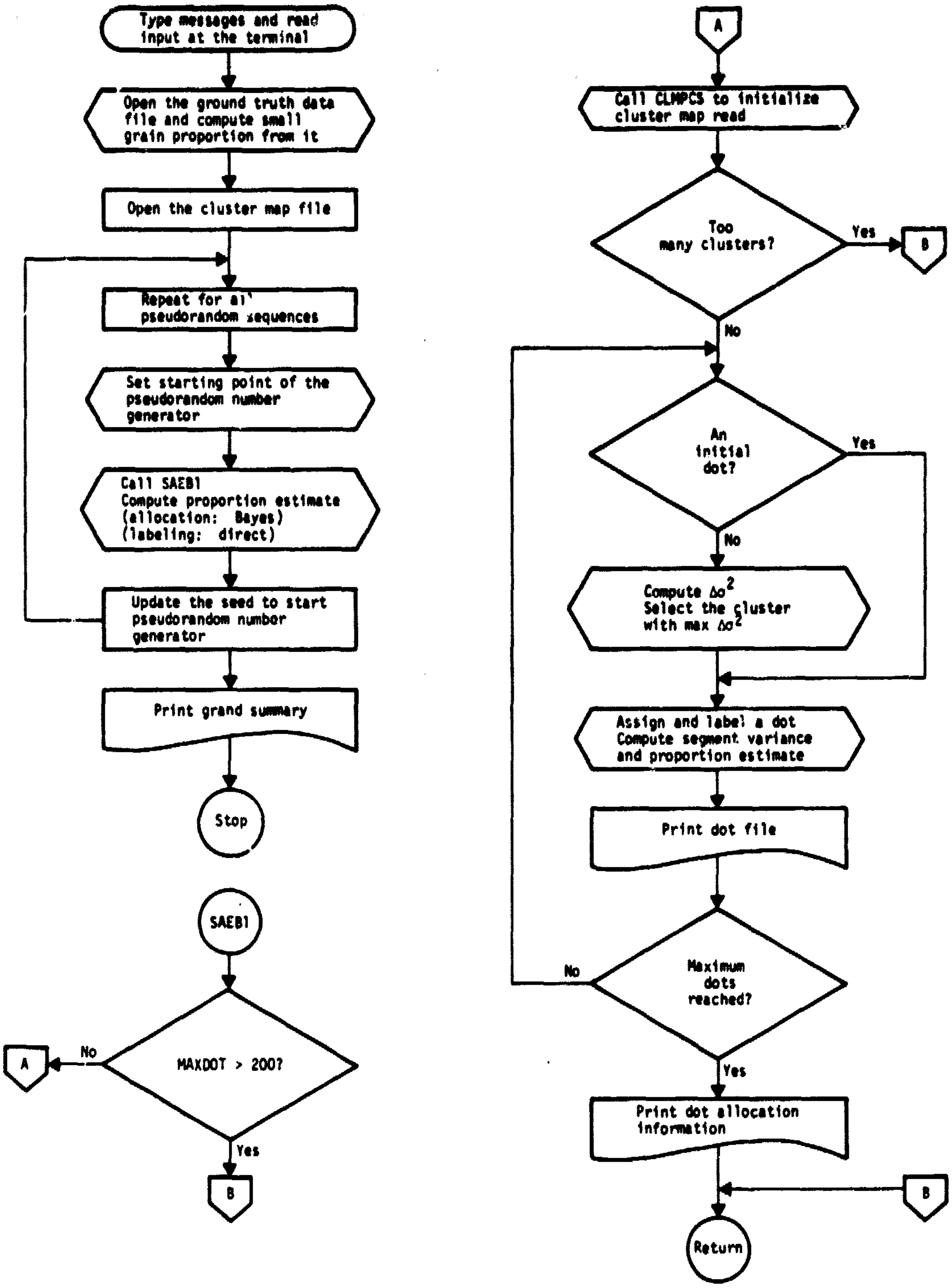

Figure 7-1.– Flow chart for Bayesian dot allocation (uniform prior).

7.7 LISTING

```
C
C
C----- PROGRAM A09: PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING BAYES (UNIFORM PRIOR)  DOT ALLOCATION
C----- LABELLING IS DIRECT BY GROUND TRUTH.
C
      BYTE NAME(15),NGT(13)
      REAL PSG(200),PM(200),PB(200)
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C----- READING INPUT FROM TERMINAL
C
      WRITE(8,121)
 121  FORMAT( ' PROGRAM: A09.'
     *       /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *       /' DOT ALLOCATION IS BAYES (UNIFORM PRIOR) .'
     *        /' LABELLING IS DIRECT BY GROUND TRUTH.'
     *       /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
 122  FORMAT(13A1)
C
      WRITE(8,131)
 131  FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *       /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *       /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *       /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
 132  FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
 141  FORMAT(' SPECIFY MAXIMUM NO. OF DOTS IN THE REPETITIONS RUNS,'/
     *       ' AND NO. OF DOTS INTIALLY ASSIGNED TO EACH CLUSTER.'/
     *       ' III III')
      READ(7,142)MAXDOT,NNIX
 142  FORMAT(I3,1X,I3)
      CALL INIT1(NNIX)
C
      WRITE(8,151)
 151  FORMAT(' STATUS MESSAGES ON TERMINAL, HOW MANY?'/' III')
      READ(7,152)NMES
 152  FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED,MAXDOT
 181  FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *        /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *        /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6,
     *        /' MAXIMUM NO. OF DOTS IN THE REPETITION RUNS=',I6)
C
C----- COMPUTE BAYES (UNIFORM PRIOR)  ESTIMATE FOR GROUND TRUTH
       NLINE=117
       NCOL=196
      DO 215 K=1,4
 215  NGT(K)=NAME(K)
       OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *      ACCESS='DIRECT')
       DO 231 L=1,NLINE
       DO 221 K=1,NCOL
       CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
 221   CONTINUE
 231   CONTINUE
       P=FLOAT(IP)/NLINE/NCOL
```

```
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *   READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
C
C------ START ESTIMATION FOR EACH TOTDOT AND REPETITIONS
      DO 311 K=1,MAXDOT
      PB(K)=0.
311   PM(K)=0.
      JMES=0
      JS=JSEED
      DO 361 J=1,JV
      JFLAG=0
      IF(MOD(J,JSKIP).EQ.1) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(5-9-79)'
     *       ,/,2X,'PROGRAM A09: ESTIMATION OF SMALL GRAINS PROPORTION'
     *        ,/,15X,'DOT ALLOCATION:BAYES (UNIFORM PRIOR) ,'
     *        ,/,15X,'DOT LABELLING:DIRECT BY GROUND TRUTH.'
     *       //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *         /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' --- REPETITION RUN=',I3,
     *          ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAEB1(MAXDOT,PSG)
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)MAXDOT,J,JSEED
323   FORMAT(' MAXDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6)
      JSEED=JSEED+150
C------ COMPUTE BIAS AND M.S.E.
      DO 341 K=1,MAXDOT
      TEMP=PSG(K)-P
      PB(K)=PB(K)+TEMP
341   PM(K)=PM(K)+TEMP**2
361   CONTINUE
C
C------ PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)MAXDOT,JV,JS
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *         /,5X,' MAX. NO. OF DOTS IN EACH REPETITION RUN=',I3,
     *        /,14X,'          NO. OF REPETITION RUNS =',I3,
     *        /,14X,' RANDOM DOTS START WITH SEED=',I6,
     *        //,1X,'  DOT     BIAS      M.S.E.  REDUCTION',
     *              '   AVERAGE   VARIANCE  VAR REDUCTION')
      DO 381 K=1,MAXDOT
      PB(K)=PB(K)/JV
      PM(K)=PM(K)/JV
      RR=PM(K)/( P*(1.-P)/K )
      AVERG=PB(K)+P
      IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
      IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
      REDVAR=VAR/( P*(1.-P)/K )
381   WRITE(6,382)K,PB(K),PM(K),RR,AVERG,VAR,REDVAR
382   FORMAT(2X,I3, 2X,F9.5, 4(F10.6), 3X,F10.6)
C
C
      WRITE(6,401)
401   FORMAT('1 ---- END OF THIS JOB ---')
C
      STOP
      END
C
C
C
      SUBROUTINE SAEB1(MAXDOT,PSG)
C------ STRATIFIED AREAL ESTIMATION USING BAYES  DOT ALLOCATION.
C------ FORMULA HERE ARE FOR BAYES WITH UNIFORM PRIOR.
C------ GIVEN ANY MAX NO. OF DOTS IN MAXDOT, PROGRAM WILL RETURN
C------ THE ESTIMATES ARE IN ARRAY PSG ON RETURN.
```

```
C
C----- WRITEN AND EDITED BY N.Y. CHU ON 5-9-79.
C
C
      INTEGER NDARY(200),MM(51),NN(51),NX(51),LL(51)
      REAL VV,PSG(1)
      BYTE LABEL
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/, MINSZ/5/
      DATA INIDOT/3/
C
      IF(MAXDOT.GT.200) GOTO 911
C
C----- GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C------ BEGIN BAYES  DOT ALLOCATION
C
      IF(JFLAG.EQ.1) WRITE(6,141)
141   FORMAT(//, 25X,'    --- DOT FILE ---'
     *       /'    DOT  CLUSTER INFORMATION  POSITION GROUND TRUTH'
     *       ,' SEGMENT PROPORTIONAL'
     *       /' NUMBER  NO. PIXEL DOTS S.G. LINE,COL   RAW,CODE'
     *       ,'    VARIANCE  ESTIMATE')
C
C------ MAKE SURE EACH CLUSTER HAS A MINIMUM NO. OF PIXELS
      K1=0
      NDS=0
      DO 151 K=1,M
      NX(K)=0
      NN(K)=0
      IF(MM(K).GE.MINSZ) NDS=NDS+INIDOT
      IF(MM(K).LT.MINSZ) K1=K1+MM(K)
      IF(MM(K).LT.MINSZ) NN(K)=-1
151   CONTINUE
      MX=M
      IF(K1.LT.MINSZ) GOTO 161
      MX=M+1
      MM(MX)=K1
      NDS=NDS+INIDOT
161   CONTINUE
C
C----- FROM HERE TO STATEMENT 261: LOOP TO ASSIGN 1 DOT AT A TIME
C
C----- IN ORDER TO ACHIEVE ASSIGNING 3 DOTS INITIALLY OR LESS
C--------TO EACH CLUSTER.
C------ THE DELTA VARIANCE COMPUTATION IS BYPASSED FOR THE FIRST
C----- NDS=INIDOT*MX DOTS
C
      KG1=0
      NDOT=0
191   NDOT=NDOT+1
      IF(NDOT.GT.NDS) GOTO 200
195   KG1=KG1+1
      IF(KG1.GT.MX) KG1=1
      K1=KG1
      IF(NN(KG1).GT.-1) GOTO 214
      GOTO 195
C
C
C------- FOR EACH CLUSTER, COMPUTE DELTA VARIANCE AND CHOOSE THE LARGEST
200   AMAX=0.
      DO 211 K=1,MX
      J=NN(K)
      IF(J.LE.1) GOTO 211
      VAR= ( FLOAT(MM(K))/NPIXEL*(J+5) )**2
```

ORIGINAL PAGE IS OF POOR QUALITY

```
     *            .   *(FLOAT(J)*J!7.*J+8.)*(NX(K)+1.)*(J-NX(K)+1.)
     *                /( (J-1.)*J*(J+2.)*(J+3.)**2 )
C     WRITE(6,99121)K,MM(K),J,NX(K),VAR
99121 FORMAT(' K,MM(K),J,NX(K),VAR=',I4,I8,2I5,E16.6)
      IF(VAR.LT.AMAX) GOTO 211
      K1=K
      AMAX=VAR
211   CONTINUE
C
C----- ASSIGN A DOT TO THE CLUSTER WITH MAX DELTA VARIANCE
      IF(NN(K1).GE.MM(K1)) WRITE(6,213)NDOT,K1,MM(K1)
      IF(NN(K1).GE.MM(K1)) GOTO 241
213   FORMAT(' ATTEMPT TO ASSIGN DOT=',I3,' WHILE NO PIXEL',
     *       ' IS AVAILABLE. CLUSTER=',I3,' TOT PIXEL=',I6)
C---- POSITION THE POINTER TO NDARY
214   ND=1
      DO 215 K=1,K1
      IF(NN(K).LE.0) GOTO 215
      ND=ND+NN(K)
215   CONTINUE
C---- MAKE ROOM FOR THE NEW DOT IN NDARY
      N1=-1
      DO 217 K=NDOT,ND,N1
217   NDARY(K+1)=NDARY(K)
      ND=ND-NN(K1)
      NN(K1)=NN(K1)+1
C----- ASSIGN A DOT
      CALL GETDOT(MM(K1),NN(K1),NDARY(ND),IDOT)
C     WRITE(6,99141)K1,ND,NN(K1),NX(K1),(NDARY(K),K=1,44)
99141 FORMAT(' K1,ND,NN,NX=',4I4, 4(/,2X,15I4) )
      IF(K1.LE.M) GOTO 227
      DO 225 K1=1,M
      IF(NN(K1).GT.0) GOTO 225
      IF(IDOT.LT.MM(K1)) GOTO 227
      IDOT=IDOT-MM(K1)
225   CONTINUE
227   CALL CLMPXY(K1,IDOT,NL,NC,LL)
      CALL GTMPLB(NL,NC,LABEL)
      CALL LBLITP(LABEL,LB1,NX(K1))
      PSG(NDOT)=0.
      DO 229 JX=1,MX
      IF(NN(JX).LE.0) GOTO 229
      PSG(NDOT)=PSG(NDOT)+ FLOAT(NX(JX))/NN(JX) *MM(JX)/NPIXEL
229   CONTINUE
C
C
C----- COMPUTE  SEGMENT VARIANCE
241   VU=0.
      DO 251 K=1,MX
      IF(NN(K).LE.1) GOTO 251
      P= (NX(K)+1.)/(NN(K)+2.)
      VU=VU+(FLOAT(MM(K))/NPIXEL )**2 *P*(1.-P)/( NN(K)-1. )
251   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,255)NDOT,K1,IDOT,NN(K1),NX(K1)
     *               ,NL,NC,LABEL,LB1,VU,PSG(NDOT)
255   FORMAT(2X,I3, 3X,I3,1X,I6,1X,I3,2X,I3, 2X,2I4, 2X,2I4, 2F11.6)
      IF(NDOT.LT.MAXDOT) GOTO 191
C
      DO 261 K=1,MX
      IF(NN(K).LE.-1) NN(K)=0
261   CONTINUE
C
C
C----- END OF BAYES DOT ALLOCATION
```

```
C
C
C------ WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)MAXDOT,NPIXEL,MX
282   FORMAT(//,5X,'              TOTAL NO. OD DOTS AVAILABLE =',I6,
     *          /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *          /,5X,'                TOTAL NO. OF CLUSTERS =',I6,
     *          /,5X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS  LABELLED'
     *          /,5X,'   NO.     CODE  PIXELS   ASSIGNED  SMALL GRAIN')
      DO 283 K=1,MX
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K),NX(K)
283   CONTINUE
284   FORMAT(8X,I2,5X,I4,3X,I6,4X,I3, 10X,I3)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
285   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286   FORMAT(/'  THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *        ') ARE:', 3(/,16(I3,',')) )
288   CONTINUE
C
C
C
C
      GOTO 990
C
C
901   WRITE(6,902)M
902   FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
911   WRITE(6,912)NDOT
912   FORMAT(' YOU SPECIFIED',I4,' TOO MANY DOTS(MAX=200)')
C
C
C---- ENTRY TO DEFINE INTIAL NO. OF DOTS TO BE ASSIGNED
      ENTRY INIT1(NNIT)
      INIDOT=NNIT
990   RETURN
      END
```

8. A85: BAYESIAN DOT ALLOCATION (NO PRIOR)

8.1 DESCRIPTION

Implemented in this piece of software is a scheme in which dot allocation is sequentially Bayesian (no prior) and labeling is direct from ground truth. This scheme uses the same algorithm as does A84 (Bayesian dot allocation, uniform prior) but has slightly different formulas for $\Delta\sigma_i^2$ and segment variance, as will be noted below.

In the Bayesian dot allocation scheme, dots are allocated sequentially to those clusters that may reduce the variance of the segment proportion estimate the most. In order to do this, we must first compute the expected change in variance, $\Delta\sigma^2$, for each cluster. The expected change in variance for cluster i is defined as

$$\Delta\sigma_i^2 = \left(\frac{N_i}{N}\right)^2 \frac{x_i(n_i - x_i)}{(n_i - 1)^2 n_i^2 (n_i + 1)^2} \tag{8-1}$$

where N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

n_i = number of dots previously allocated to cluster i

x_i = number of dots previously allocated to cluster i which are labeled as small grain

(notice $x_i \leq n_i$)

Next, a dot is allocated to the cluster whose $\Delta\sigma_i^2$ is the largest. Then, for this chosen cluster, n_i is updated to $n_i + 1$. That dot's label is read from the ground truth file. If the label is a small grain, x_i is updated to $x_i + 1$. After this, the segment variance, σ^2, is computed as

$$\sigma^2 = \sum_{i=1}^{m} \left(\frac{N_i}{N}\right)^2 P_i(1 - P_i) \frac{1}{n_i - 1} \qquad (8\text{-}2)$$

where m = total number of clusters

$$P_i = \frac{x_i}{n_i} \qquad (8\text{-}3)$$

This σ^2 is printed and the above allocation scheme is repeated until the number of dots allocated reaches a value specified by the user.

The proportion estimate is computed as

$$\hat{P}_{sg} = \sum_{i=1}^{m} \frac{x_i}{n_i} \frac{N_i}{N} \qquad (8\text{-}4)$$

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. A grand summary is printed.

8.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAEB2, GETDOT, CLMPCS, GTMPLB, LBLITP
SAEB2	GETDOT, CLMPCS, GTMPLB, LBLITP
GETDOT	RAN

8.3 INTERFACE

Interface with other routines is through the common block PRTFLG (2 bytes), which is used to control the optional printing of dot files and other information.

8.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line		Parameters	Format	Default input	Default value
1		Cluster map file name	A13	None	
2	a.	Number of repetitions	I3	0 or blank	1
	b.	Starting point of first pseudorandom sequence	I5	0 or blank	10
	c.	Number of repetition printings	I3	0 or blank	5
3	a.	Maximum number of dots that can be allocated	I3	None	
	b.	Number of dots initially assigned to each cluster	I3	None	
	4.	Number of status messages on terminal	I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

8.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report		Contents
Individual repetition	1.	Processor header
	2.	Ground truth small-grain proportion
	3.	A dot file showing dots chosen, their x-y position, their ground truth labels, segment variance, and proportion estimate
	4.	A table showing number of clusters, cluster name codes, cluster sizes, and number of dots assigned
Grand summary	1.	Processor header
	2.	A table showing bias, MSE, reduction in MSE, average, variance, and variance reduction as functions of number of dots

8.6 BRIEF ALGORITHM: FOR BAYESIAN DOT ALLOCATION (NO PRIOR)

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.
3. Repeat a through g until all repetitions are finished:
    a. Set starting point of the pseudorandom number generator.
    b. Clear a dot counter.

c. Increment the dot counter.

d. If the dot counter indicates that the current dot should be allocated as an initial dot, then go to the next step. Otherwise compute $\Delta\sigma_i^2$ for every cluster and choose the cluster with maximum $\Delta\sigma_i^2$.

e. Allocate a dot to the chosen cluster.

f. Pick up the small-grain labels, compute segment variance and proportion estimate, and print a line of the dot file.

g. If the dot count does not exceed specified value, go to c.

4. Compute bias, MSE, reduction in MSE, average, variance, and variance reduction, and print a grand summary.

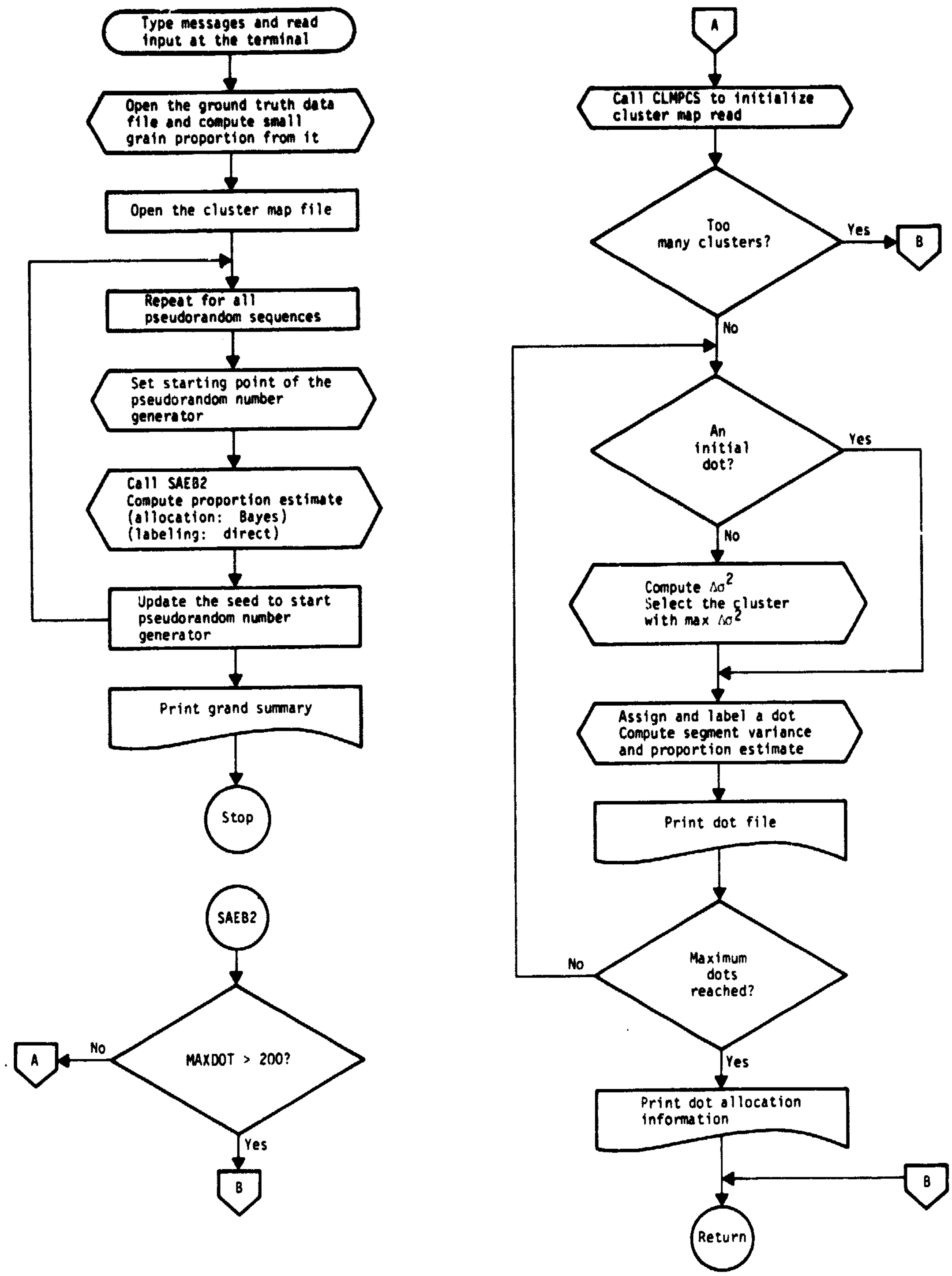

Figure 8-1.– Flow chart for Bayesian dot allocation (no prior).

8.7 LISTING

```
C
C
C------ PROGRAM A12(A85.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING BAYES (NO PRIOR)  DOT ALLOCATION
C------ LABELLING IS DIRECT BY GROUND TRUTH.
C
      BYTE NAME(15),NGT(13)
      REAL PSG(200),PM(200),PB(200)
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C----- READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A12(A85.TSK).'
     *       /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *       /' DOT ALLOCATION IS BAYES (NO PRIOR) .'
     *        /' LABELLING IS DIRECT BY GROUND TRUTH.'
     *       /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *       /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *       /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *       /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.EQ.0) JV=1
      IF(JSEED.EQ.0) JSEED=10
      IF(JPAGE.EQ.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT(' SPECIFY MAXIMUM NO. OF DOTS IN THE REPETITIONS RUNS,'/
     *       ' AND NO. OF DOTS INTIALLY ASSIGNED TO EACH CLUSTER.'/
     *       ' III III')
      READ(7,142)MAXDOT,NNIX
142   FORMAT(I3,1X,I3)
      CALL INIT2(NNIX)
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL, HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED,MAXDOT
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *        /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *        /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6,
     *        /' MAXIMUM NO. OF DOTS IN THE REPETITION RUNS=',I6)
C
C----- COMPUTE BAYES (NO PRIOR)  ESTIMATE FOR GROUND TRUTH
       NLINE=117
       NCOL=196
      DO 215 K=1,4
215   NGT(K)=NAME(K)
       OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *     ACCESS='DIRECT')
       DO 231 L=1,NLINE
       DO 221 K=1,NCOL
       CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
221    CONTINUE
231    CONTINUE
```

```
      P=FLOAT(IP)/NLINE/NCOL
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *   READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
C
C------ START ESTIMATION FOR EACH TOTDOT AND REPETITIONS
      DO 311 K=1,MAXDOT
      PB(K)=0.
311   PM(K)=0.
      JMES=0
      JS=JSEED
      DO 361 J=1,JU
      JFLAG=0
      IF(MOD(J,JSKIP).EQ.1) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(5-10-79)'
     *         /,2X,'PROGRAM A12: ESTIMATION OF SMALL GRAINS PROPORTION'
     *         /,15X,'DOT ALLOCATION:BAYES (NO PRIOR) ,'
     *         /,15X,'DOT LABELLING:DIRECT BY GROUND TRUTH.'
     *        //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *         /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' --- REPETITION RUN=',I3,
     *        ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAEB2(MAXDOT,PSG)
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)MAXDOT,J,JSEED
323   FORMAT(' MAXDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6)
      JSEED=JSEED+150
C------ COMPUTE BIAS AND M.S.E.
      DO 341 K=1,MAXDOT
      TEMP=PSG(K)-P
      PB(K)=PB(K)+TEMP
341   PM(K)=PM(K)+TEMP**2
361   CONTINUE
C
C------ PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)MAXDOT,JU,JS
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *        /,5X,' MAX. NO. OF DOTS IN EACH REPETITION RUN=',I3,
     *       /,14X,'            NO. OF REPETITION RUNS =',I3,
     *       /,14X,' RANDOM DOTS START WITH SEED=',I6,
     *      //,1X,'  DOT     BIAS       M.S.E.  REDUCTION',
     *       '  AVERAGE  VARIANCE  VAR REDUCTION')
      DO 381 K=1,MAXDOT
      PB(K)=PB(K)/JU
      PM(K)=PM(K)/JU
      RR=PM(K)/( P*(1.-P)/K )
      AVERG=PB(K)+P
      IF(JU.EQ.1) VAR= PM(K)-PB(K)**2
      IF(JU.GT.1) VAR= (PM(K)-PB(K)**2)*JU/(JU-1)
      REDUAR=VAR/( P*(1.-P)/K )
381   WRITE(6,382)K,PB(K),PM(K),RR,AVERG,VAR,REDUAR
382   FORMAT(2X,I3, 2X,F9.5, 4(F10.6), 3X,F10.6)
C
C
      WRITE(6,401)
401   FORMAT('1 ---- END OF THIS JOB ---')
C
      STOP
      END
C
C
C
      SUBROUTINE SAEB2(MAXDOT,PSG)
C------ STRATIFIED AREAL ESTIMATION USING BAYES  DOT ALLOCATION.
C------ FORMULA HERE ARE FOR BAYES WITH NO PRIOR.
C------ GIVEN ANY MAX NO. OF DOTS IN MAXDOT, PROGRAM WILL RETURN
```

```
C----- THE ESTIMATE IS IN ARRAY PSG ON RETURN.
C
C----- WRITEN AND EDITED BY N.Y. CHU ON 5-10-79.
C
C
      INTEGER NDARY(200),MM(51),NN(51),NX(51),LL(51)
      REAL VV,PSG(1)
      BYTE LABEL
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/, MINSZ/5/
      DATA INIDOT/3/
C
      IF(MAXDOT.GT.200) GOTO 911
C
C----- GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C----- BEGIN BAYES  DOT ALLOCATION
C
      IF(JFLAG.EQ.1) WRITE(6,141)
141   FORMAT(//, 25X,'    --- DOT FILE ---'
     *       /'    DOT   CLUSTER INFORMATION  POSITION GROUND TRUTH'
     *       ,' SEGMENT PROPORTIONAL'
     *       /' NUMBER  NO. PIXEL DOTS S.G. LINE,COL   RAW,CODE'
     *       ,'    VARIANCE  ESTIMATE')
C
C----- MAKE SURE EACH CLUSTER HAS A MINIMUM NO. OF PIXELS
      K1=0
      NDS=0
      DO 151 K=1,M
      NX(K)=0
      NN(K)=0
      IF(MM(K).GE.MINSZ) NDS=NDS+INIDOT
      IF(MM(K).LT.MINSZ) K1=K1+MM(K)
      IF(MM(K).LT.MINSZ) NN(K)=-1
151   CONTINUE
      MX=M
      IF(K1.LT.MINSZ) GOTO 161
      MX=M+1
      MM(MX)=K1
      NDS=NDS+INIDOT
161   CONTINUE
C
C----- FROM HERE TO STATEMENT 261: LOOP TO ASSIGN 1 DOT AT A TIME
C
C----- IN ORDER TO ACHIEVE ASSIGNING 3 DOTS INITIALLY OR LESS
C------ TO EACH CLUSTER,
C----- THE DELTA VARIANCE COMPUTATION IS BYPASSED FOR THE FIRST
C----- NDS=INIDOT*MX DOTS
C
      KG1=0
      NDOT=0
191   NDOT=NDOT+1
      IF(NDOT.GT.NDS) GOTO 200
195   KG1=KG1+1
      IF(KG1.GT.MX) KG1=1
      K1=KG1
      IF(NN(KG1).GT.-1) GOTO 214
      GOTO 195
C
C
```

~~8-9~~
60

```
C------ FOR EACH CLUSTER, COMPUTE DELTA VARIANCE AND CHOOSE THE LARGEST
200    AMAX=0.
       DO 211 K=1,MX
       J=NN(K)
       IF(J.LE.1) GOTO 211
       VAR= ( FLOAT(MM(K))/NPIXEL/(J-1.)/J/(J+1.) )**2
     *     *NX(K)*(J-NX(K))
C      WRITE(6,99121)K,MM(K),J,NX(K),VAR
99121 FORMAT(' K,MM(K),J,NX(K),VAR=',I4,I8,2I5,E16.6)
       IF(VAR.LT.AMAX) GOTO 211
       K1=K
       AMAX=VAR
211    CONTINUE
C
C----- ASSIGN A DOT TO THE CLUSTER WITH MAX DELTA VARIANCE
       IF(NN(K1).GE.MM(K1)) WRITE(6,213)NDOT,K1,MM(K1)
       IF(NN(K1).GE.MM(K1)) GOTO 241
213    FORMAT(' ATTEMPT TO ASSIGN DOT=',I3,' WHILE NO PIXEL',
     *        ' IS AVAILABLE. CLUSTER=',I3,' TOT PIXEL=',I6)
C---- POSITION THE POINTER TO NDARY
214    ND=1
       DO 215 K=1,K1
       IF(NN(K).LE.0) GOTO 215
       ND=ND+NN(K)
215    CONTINUE
C---- MAKE ROOM FOR THE NEW DOT IN NDARY
       N1=-1
       DO 217 K=NDOT,ND,N1
217    NDARY(K+1)=NDARY(K)
       ND=ND-NN(K1)
       NN(K1)=NN(K1)+1
C------ ASSIGN A DOT
       CALL GETDOT(MM(K1),NN(K1),NDARY(ND),IDOT)
C      WRITE(6,99141)K1,ND,NN(K1),NX(K1),(NDARY(K),K=1,44)
99141 FORMAT(' K1,ND,NN,NX=',4I4, 4(/,2X,15I4) )
       IF(K1.LE.M) GOTO 227
       DO 225 K1=1,M
       IF(NN(K1).GT.0) GOTO 225
       IF(IDOT.LT.MM(K1)) GOTO 227
       IDOT=IDOT-MM(K1)
225    CONTINUE
227    CALL CLMPXY(K1,IDOT,NL,NC,LL)
       CALL GTMPLB(NL,NC,LABEL)
       CALL LBLITP(LABEL,LB1,NX(K1))
       PSG(NDOT)=0.
       DO 229 JX=1,MX
       IF(NN(JX).LE.0) GOTO 229
       PSG(NDOT)=PSG(NDOT)+ FLOAT(NX(JX))/NN(JX) *MM(JX)/NPIXEL
229    CONTINUE
C
C
C----- COMPUTE AND STORE SEGMENT VARIANCE
241    VV=0.
       DO 251 K=1,MX
       IF(NN(K).LE.1) GOTO 251
       P= FLOAT(NX(K))/NN(K)
       VV=VV+(FLOAT(MM(K))/NPIXEL )**2 *P*(1.-P)/( NN(K)-1. )
251    CONTINUE
       IF(JFLAG.EQ.1) WRITE(6,255)NDOT,K1,IDOT,NN(K1),NX(K1)
     *          ,NL,NC,LABEL,LB1,VV,PSG(NDOT)
255    FORMAT(2X,I3, 3X,I3,1X,I6,1X,I3,2X,I3, 2X,2I4, 2X,2I4, 2F11.6)
       IF(NDOT.LT.MAXDOT) GOTO 191
C
       DO 261 K=1,MX
       IF(NN(K).LE.-1) NN(K)=0
261    CONTINUE
```

```
C
C
C----- END OF BAYES DOT ALLOCATION
C
C
C----- WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)MAXDOT,NPIXEL,MX
 282  FORMAT(//,5X,'          TOTAL NO. OD DOTS AVAILABLE =',I6,
     *         /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *         /,5X,'                 TOTAL NO. OF CLUSTERS =',I6,
     *         /,5X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS  LABELLED'
     *         /,5X,'   NO.     CODE  PIXELS   ASSIGNED  SMALL GRAIN')
      DO 283 K=1,MX
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K),NX(K)
 283  CONTINUE
 284  FORMAT(8X,I2,5X,I4,3X,I6,4X,I3, 10X,I3)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
 285  CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
 286  FORMAT(/'  THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *       ') ARE:', 3(/,16(I3,',')) )
 288  CONTINUE
C
C
C
C
C
      GOTO 990
C
C
 901  WRITE(6,902)M
 902  FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
 911  WRITE(6,912)NDOT
 912  FORMAT(' YOU SPECIFIED',I4,' TOO MANY DOTS(MAX=200)')
C
C
C---- ENTRY TO DEFINE INTIAL NO. OF DOTS TO BE ASSIGNED
      ENTRY INIT2(NNIT)
      INIDOT=NNIT
 990  RETURN
      END
```

9. A87: BAYESIAN DOT ALLOCATION (QUADRATIC PRIOR)

9.1 DESCRIPTION

Implemented in this piece of software is a scheme in which dot allocation is sequentially Bayesian (quadratic prior) and labeling is direct from ground truth. This scheme uses an algorithm similar to that used by A84 and A85 but has different formulas for $\Delta\sigma^2$ and segment variance. These two parameters are computed with the help of function TH1.

In the Bayesian dot allocation scheme, dots are allocated sequentially to those clusters that may reduce the variance of the segment proportion estimate the most. In order to do this, we must first compute the expected change in variance, $\Delta\sigma^2$, for each cluster. The expected change in variance for cluster i is defined as

$$\Delta\sigma_i^2 = \left(\frac{N_i}{N}\right)^2 \left[\frac{\hat{\theta}_{n_i,x_i}(1 - \hat{\theta}_{n_i,x_i})}{(n_i - 1)} - \frac{\hat{\theta}_{n_i,x_i}\hat{\theta}_{n_i+1,x_i+1}(1 - \hat{\theta}_{n_i+1,x_i+1})}{n_i} - \frac{(1 - \hat{\theta}_{n_i,x_i})\hat{\theta}_{n_i+1,x_i}(1 - \hat{\theta}_{n_i+1,x_i})}{n_i} \right] \tag{9-1}$$

where N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

$$\hat{\theta}_{n,x} = \frac{a(x+1)(x+2)(x+3) + b\,(x+1)(x+2)(n+4) + c\,(x+1)(n+3)(n+4)}{a(x+1)(x+2)(n+4) + b\,(x+1)(n+3)(n+4) + c\,(n+2)(n+3)(n+4)} \tag{9-2}$$

a = 6.31924, b = -8.19799, c = 2.99258

n_i = number of dots previously allocated to cluster i

x_i = number of dots previously allocated to cluster i which are labeled as small grain

(notice $x_i \le n_i$)

It should be noted that $\hat{\theta}_{n,x}$ is computed via a function called TH1.

Next, a dot is allocated to the cluster whose $\Delta\sigma_i^2$ is the largest. Then, for this chosen cluster, n_i is updated to $n_i + 1$. That dot's label is read from the ground truth file. If the label is a small grain, x_i is updated to $x_i + 1$. After this, the segment variance, σ^2, is computed as

$$\sigma^2 = \sum_{i=1}^{m} \left(\frac{N_i}{N}\right)^2 \frac{\hat{\theta}_{n_i,x_i}(1 - \hat{\theta}_{n_i,x_i})}{n_i - 1} \tag{9-3}$$

where m = total number of clusters

This σ^2 is printed and the above allocation scheme is repeated until the number of dots allocated reaches a value specified by the user.

The proportion estimate is computed as

$$\hat{P}_{sg} = \sum_{i=1}^{m} \frac{N_i}{N} \hat{\theta}_{n_i,x_i} \tag{9-4}$$

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. A grand summary is printed.

9.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in the section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAEB3, GETDO[illegible] CLMPCS, GTMPLB, LBLITP
SAEB3	GETDOT, CLMPCS, GTMPLB, LBLITP, TH1
GETDOT	RAN
TH1	None

9.3 INTERFACE

Interface with other routines is through the common block PRTFLG (2 bytes), which is used to control the optional printing of dot files and other information.

9.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line	Parameters	Format	Default input	Default value
1	Cluster map file name	A13	None	
2	a. Number of repetitions	I3	0 or blank	1
	b. Starting point of first pseudorandom sequence	I5	0 or blank	10
	c. Number of repetition printings	I3	0 or blank	5

3	a. Maximum number of dots that can be allocated	I3	None		
	b. Number of dots initially assigned to each cluster	I3	None		
4	Number of status messages on terminal	I3	0 or blank	NO MSG	

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

9.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report	Contents
Individual repetition	1. Processor header
	2. Ground truth small-grain proportion
	3. A dot file showing dots chosen, their x-y position, their ground truth labels, segment variance, and proportion estimate
	4. A table showing number of clusters, cluster name codes, cluster sizes, and number of dots assigned
Grand summary	1. Processor header
	2. A table showing bias, MSE, reduction in MSE, average, variance, and variance reduction as functions of number of dots

9.6 BRIEF ALGORITHM: FOR BAYESIAN DOT ALLOCATION (QUADRATIC PRIOR)

1. Type messages at terminal and read from it run specifications.

2. Compute ground truth small-grain proportion.

3. Repeat a through g until all repetitions are finished:

   a. Set starting point of the pseudorandom number generator.

   b. Clear a dot counter.

   c. Increment the dot counter.

   d. If the dot counter indicates that the current dot should be allocated as an initial dot, then go to the next step. Otherwise compute $\Delta\sigma_i^2$ for every cluster and choose the cluster with maximum $\Delta\sigma_i^2$.

   e. Allocate a dot to the chosen cluster.

   f. Pick up the small-grain labels, compute segment variance and proportion estimate, and print a line of the dot file.

   g. If the dot count does not exceed specified value, go to c.

4. Compute bias, MSE, reduction in MSE, average, variance, and variance reduction, and print a grand summary.

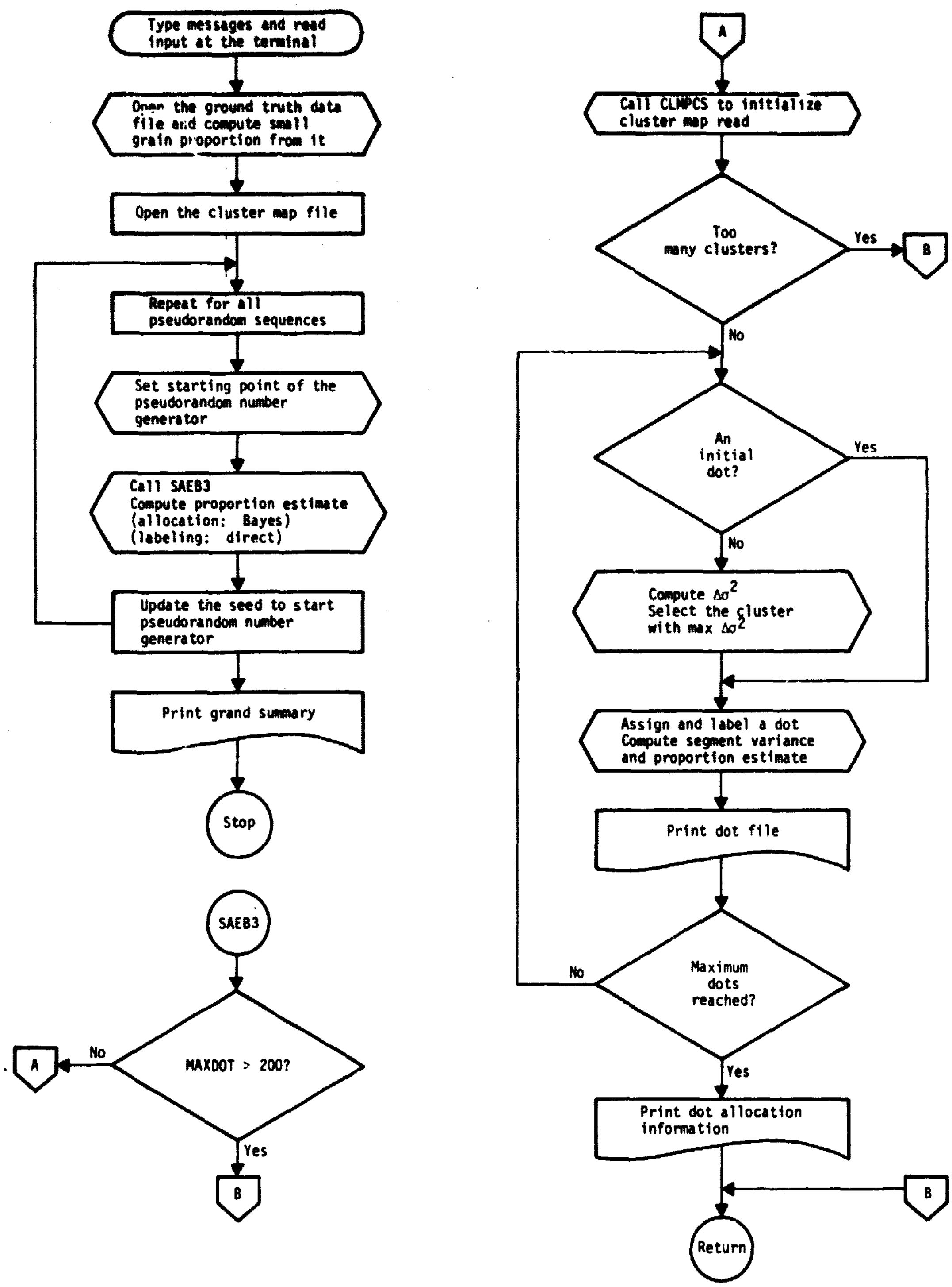

Figure 9-1.– Flow chart for Bayesian dot allocation (quadratic prior).

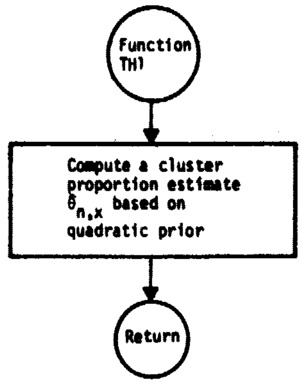

Figure 9-1.– Concluded.

9.7 LISTING

```
C
C
C----- PROGRAM A13 (A87.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING BAYES (QUADRATIC PRIOR) DOT ALLOCATION
C----- LABELLING IS DIRECT BY GROUND TRUTH.
C
      BYTE NAME(15),NGT(13)
      REAL PSG(200),PM(200),PB(200)
C------ COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C------ READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A13 (A87.TSK).'
     *       /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *       /' DOT ALLOCATION IS BAYES (QUADRATIC PRIOR).'
     *        /' LABELLING IS DIRECT BY GROUND TRUTH.'
     *       /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *       /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *       /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *       /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT(' SPECIFY MAXIMUM NO. OF DOTS IN THE REPETITIONS RUNS,'/
     *       ' AND NO. OF DOTS INTIALLY ASSIGNED TO EACH CLUSTER.'/
     *       ' III III')
      READ(7,142)MAXDOT,NNIX
142   FORMAT(I3,1X,I3)
      CALL INIT3(NNIX)
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL, HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED,MAXDOT
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *         /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *         /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6,
     *         /' MAXIMUM NO. OF DOTS IN THE REPETITION RUNS=',I6)
C
C------ COMPUTE BAYES (QUADRATIC PRIOR) ESTIMATE FOR GROUND TRUTH
       NLINE=117
       NCOL=196
      DO 215 K=1,4
215   NGT(K)=NAME(K)
       OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *      ACCESS='DIRECT')
       DO 231 L=1,NLINE
       DO 221 K=1,NCOL
       CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
221    CONTINUE
231    CONTINUE
```

```
      P=FLOAT(IP)/NLINE/NCOL
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *   READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
C
C------ START ESTIMATION FOR AN TOTDOT AND REPETITIONS
      DO 311 K=1,MAXDOT
      PB(K)=0.
311   PM(K)=0.
      JMES=0
      JS=JSEED
      DO 361 J=1,JV
      JFLAG=0
      IF(MOD(J-1,JSKIP).EQ.0) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(5-11-79)',/,2X,
     * 'PROGRAM A13 (A87.TSK): ESTIMATION OF SMALL GRAINS PROPORTION'
     *        /,15X,'DOT ALLOCATION:BAYES (QUADRATIC PRIOR),'
     *        /,15X,'DOT LABELLING:DIRECT BY GROUND TRUTH.'
     *       //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *        /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' ---- REPETITION RUN=',I3,
     *        ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAEB3(MAXDOT,PSG)
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)MAXDOT,J,JSEED
323   FORMAT(' MAXDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6)
      JSEED=JSEED+150
C----- COMPUTE BIAS AND M.S.E.
      DO 341 K=1,MAXDOT
      TEMP=PSG(K)-P
      PB(K)=PB(K)+TEMP
341   PM(K)=PM(K)+TEMP**2
361   CONTINUE
C
C------ PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)MAXDOT,JV,JS
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB --- ',
     *        /,5X,' MAX. NO. OF DOTS IN EACH REPETITION RUN=',I3,
     *       /,14X,'              NO. OF REPETITION RUNS =',I3,
     *       /,14X,' RANDOM DOTS START WITH SEED=',I6,
     *       //,1X,'  DOT     BIAS        M.S.E.   REDUCTION',
     *        '   AVERAGE  VARIANCE  VAR REDUCTION')
      DO 381 K=1,MAXDOT
      PB(K)=PB(K)/JV
      PM(K)=PM(K)/JV
      RR=PM(K)/( P*(1.-P)/K )
      AVERG=PB(K)+P
      IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
      IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
      REDVAR=VAR/( P*(1.-P)/K )
381   WRITE(6,382)K,PB(K),PM(K),RR,AVERG,VAR,REDVAR
382   FORMAT(2X,I3, 2X,F9.5, 4(F10.6), 3X,F10.6)
C
C
      WRITE(6,401)
401   FORMAT('1 ----- END OF THIS JOB ---')
C
      STOP
      END
C
C
C
```

```
      SUBROUTINE SAEB3(MAXDOT,PSG)
C----- STRATIFIED AREAL ESTIMATION USING BAYES DOT ALLOCATION.
C----- FORMULAS HERE ARE FOR BAYES WITH QUADRATIC PRIOR.
C----- GIVEN ANY MAX NO. OF DOTS IN MAXDOT, PROGRAM WILL RETURN
C----- THE PROPORTIONAL ESTIMATE IS IN ARRAY PSG.
C
C----- WRITEN AND EDITED BY N.Y. CHU ON 5-11-79.
C
C
      INTEGER NDARY(200),MM(51),NN(51),NX(51),LL(51)
      REAL VV,PSG(1)
      BYTE LABEL
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/, MINSZ/5/
      DATA INIDOT/3/
C
      IF(MAXDOT.GT.200) GOTO 911
C
C----- GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C----- BEGIN BAYES DOT ALLOCATION
C
      IF(JFLAG.EQ.1) WRITE(6,141)
141   FORMAT(//, 25X,'    --- DOT FILE ---'
     *          /'    DOT  CLUSTER INFORMATION  POSITION GROUND TRUTH'
     *          ,' SEGMENT PROPORTIONAL'
     *          /' NUMBER  NO. PIXEL DOTS S.G. LINE,COL   RAW,CODE'
     *          ,'    VARIANCE  ESTIMATE')
C
C----- MAKE SURE EACH CLUSTER HAS A MINIMUM NO. OF PIXELS
      K1=0
      NDS=0
      DO 151 K=1,M
      NX(K)=0
      NN(K)=0
      IF(MM(K).GE.MINSZ) NDS=NDS+INIDOT
      IF(MM(K).LT.MINSZ) K1=K1+MM(K)
      IF(MM(K).LT.MINSZ) NN(K)=-1
151   CONTINUE
      MX=M
      IF(K1.LT.MINSZ) GOTO 161
      MX=M+1
      MM(MX)=K1
      NDS=NDS+INIDOT
161   CONTINUE
C
C----- FROM HERE TO STATEMENT 261: LOOP TO ASSIGN 1 DOT AT A TIME
C
C----- IN ORDER TO ACHIEVE ASSIGNING 3 DOTS INITIALLY TO EACH CLUSTER
C----- THE DELTA VARIANCE COMPUTATION IS BYPASSED FOR THE FIRST
C----- NDS=INIDOT*MX DOTS
C
      KG1=0
      NDOT=0
191   NDOT=NDOT+1
      IF(NDOT.GT.NDS) GOTO 200
195   KG1=KG1+1
      IF(KG1.GT.MX) KG1=1
      K1=KG1
      IF(NN(KG1).GT.-1) GOTO 214
      GOTO 195
```

```
C
C
C------ FOR EACH CLUSTER, COMPUTE DELTA VARIANCE AND CHOOSE THE LARGEST
200    AMAX=0.
       DO 211 K=1,MX
       J=NN(K)
       JX=NX(K)
       IF(J.LE.1) GOTO 211
C
       VAR= ( FLOAT(MM(K))/NPIXEL )**2
      *      *( TH1(J,JX)*(1.-TH1(J,JX))/(J-1.)
      *  -TH1(J,JX)*TH1(J+1,JX+1)*(1.-TH1(J+1,JX+1))/J
      *  -(1.-TH1(J,JX))*TH1(J+1,JX)*(1.-TH1(J+1,JX))/J )
C
C      WRITE(6,99121)K,MM(K),J,NX(K),VAR
99121 FORMAT(' K,MM(K),J,NX(K),VAR=',I4,I8,2I5,E16.6)
       IF(VAR.LT.AMAX) GOTO 211
       K1=K
       AMAX=VAR
211    CONTINUE
C
C------ ASSIGN A DOT TO THE CLUSTER WITH MAX DELTA VARIANCE
       IF(NN(K1).GE.MM(K1)) WRITE(6,213)NDOT,K1,MM(K1)
       IF(NN(K1).GE.MM(K1)) GOTO 241
213    FORMAT(' ATTEMPT TO ASSIGN DOT=',I3,' WHILE NO PIXEL',
      *        ' IS AVAILABLE. CLUSTER=',I3,' TOT PIXEL=',I6)
C----- POSITION THE POINTER TO NDARY
214    ND=1
       DO 215 K=1,K1
       IF(NN(K).LE.0) GOTO 215
       ND=ND+NN(K)
215    CONTINUE
C----- MAKE ROOM FOR THE NEW DOT IN NDARY
       N1=-1
       DO 217 K=NDOT,ND,N1
217    NDARY(K+1)=NDARY(K)
       ND=ND-NN(K1)
       NN(K1)=NN(K1)+1
C------ ASSIGN A DOT
       CALL GETDOT(MM(K1),NN(K1),NDARY(ND),IDOT)
C      WRITE(6,99141)K1,ND,NN(K1),NX(K1),(NDARY(K),K=1,44)
99141 FORMAT(' K1,ND,NN,NX=',4I4, 4(/,2X,15I4) )
       IF(K1.LE.M) GOTO 227
       DO 225 K1=1,M
       IF(NN(K1).GT.0) GOTO 225
       IF(IDOT.LT.MM(K1)) GOTO 227
       IDOT=IDOT-MM(K1)
225    CONTINUE
227    CALL CLMPXY(K1,IDOT,NL,NC,LL)
       CALL GTMPLB(NL,NC,LABEL)
       CALL LBLITP(LABEL,LB1,NX(K1))
       PSG(NDOT)=0.
       DO 229 JX=1,MX
       IF(NN(JX).LE.0) GOTO 229
       PSG(NDOT)=PSG(NDOT)+ FLOAT(MM(JX))/NPIXEL*TH1(NN(JX),NX(JX))
229    CONTINUE
C
C
C------ COMPUTE  SEGMENT VARIANCE
241    VV=0.
       DO 251 K=1,MX
       IF(NN(K).LE.1) GOTO 251
       VV=VV+ (FLOAT(MM(K))/NPIXEL)**2 *TH1(NN(K),NX(K))
      *      *(1.-TH1(NN(K),NX(K)))/(NN(K)-1.)
251    CONTINUE
       IF(JFLAG.EQ.1) WRITE(6,255)NDOT,K1,IDOT,NN(K1),NX(K1)
      *              ,NL,NC,LABEL,LB1,VV,PSG(NDOT)
255    FORMAT(2X,I3, 3X,I3,1X,I6,1X,I3,2X,I3, 2X,2I4, 2X,2I4, 2F11.6)
       IF(NDOT.LT.MAXDOT) GOTO 191
C
       DO 261 K=1,MX
       IF(NN(K).LE.-1) NN(K)=0
```

```
261   CONTINUE
C
C
C----- END OF BAYES DOT ALLOCATION
C
C
C------ WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)MAXDOT,NPIXEL,MX
282   FORMAT(//,5X,'                TOTAL NO. OD DOTS AVAILABLE =',I6,
     *        /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *        /,5X,'                    TOTAL NO. OF CLUSTERS =',I6,
     *        /,5X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS  LABELLED'
     *        /,5X,'   NO.     CODE  PIXELS   ASSIGNED  SMALL GRAIN')
      DO 283 K=1,MX
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K),NX(K)
283   CONTINUE
284   FORMAT(8X,I2,5X,I4,3X,I6,4X,I3, 10X,I3)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
285   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286   FORMAT(/'  THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *        ') ARE:', 3(/,16(I3,',')) )
288   CONTINUE
C
C
C
C
      GOTO 990
C
C
901   WRITE(6,902)M
902   FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
911   WRITE(6,912)NDOT
912   FORMAT(' YOU SPECIFIED',I4,' TOO MANY DOTS(MAX=200)')
C
C
C----- ENTRY TO DEFINE INTIAL NO. OF DOTS TO BE ASSIGNED
      ENTRY INIT3(NNIT)
      INIDOT=NNIT
990   RETURN
      END
C
C
C
      FUNCTION TH1(N,NX)
C------ FUNCTION TO EVALUATE CLUSTER PROPORTIONAL ESTIMATE
C------ N=NO. OF DOTS ASSIGNED FOR A CLUSTER.
C------ NX= NO. OF DOTS LABELLED AS SMALL GRAIN
C------ BOTH N AND NX ARE SIMPLE INTEGER (2 BYTES IN PDP)
C------ TH1 (REAL NO.) CONTAINS THE ESTIMATE ON RETURN.
C
      INTEGER N,NX
      REAL TH1
C
      DATA A/6.0/, B/-7.877/, C/2.9385/
C
      X=NX
      TH1 =
     * ( A*(X+1.)*(X+2.)*(X+3.) + B*(X+1.)*(X+2.)*(N+4.)
     *      + C*(X+1.)*(N+3.)*(N+4.) ) /
     * ( A*(X+1.)*(X+2.)*(N+4.) + B*(X+1.)*(N+3.)*(N+4.)
     *      + C*(N+2.)*(N+3.)*(N+4.) )
      RETURN
      END
```

10. A89: BAYESIAN DOT ALLOCATION (MODIFIED QUADRATIC PRIOR)

10.1 DESCRIPTION

Implemented in this piece of software is a scheme in which dot allocation is sequentially Bayesian (modified quadratic prior) and labeling is direct from ground truth. This scheme is similar to those in A84, A85, and A87, but it uses different formulas for $\Delta\sigma^2$ and segment variance. The constant used in the function TH2 is also modified during execution.

In the Bayesian dot allocation scheme, dots are allocated sequentially to those clusters that may reduce the variance of the segment proportion estimate the most. In order to do this, we must first compute the expected change in variance, $\Delta\sigma^2$, for each cluster. The expected change in variance for cluster i is defined as

$$\Delta\sigma_i^2 = \left(\frac{N_i}{N}\right)^2 \left[\frac{\hat{\theta}_{n_i,x_i}(1 - \hat{\theta}_{n_i,x_i})}{(n_i - 1)} - \frac{\hat{\theta}_{n_i,x_i}\hat{\theta}_{n_i+1,x_i+1}(1 - \hat{\theta}_{n_i+1,x_i+1})}{n_i} - \frac{(1 - \hat{\theta}_{n_i,x_i})\hat{\theta}_{n_i+1,x_i}(1 - \hat{\theta}_{n_i+1,x_i})}{n_i}\right] \tag{10-1}$$

where N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

$$\hat{\theta}_{n,x} = \frac{a(x+1)(x+2)(x+3) + b\,(x+1)(x+2)(n+4) + c\,(x+1)(n+3)(n+4)}{a(x+1)(x+2)(n+4) + b\,(x+1)(n+3)(n+4) + c\,(n+2)(n+3)(n+4)} \tag{10-2}$$

For initial dots: $a = 6,\ b = -7.877,\ c = 2.9345$

After initial dots: $a = 6,\ b = 12(\hat{P}_{sg} - 1),\ c = 5 - 6\hat{P}_{sg}$ if $0.211 \le \hat{P}_{sg} \le 0.789$

$a = 6,\ b = 12(0.211 - 1),\ c = 5 - 6 \times 0.211$ if $0.211 > \hat{P}_{sg}$

$a = 6,\ b = 12(0.789 - 1),\ c = 5 - 6 \times 0.789$ if $0.789 < \hat{P}_{sg}$

n_i = number of dots previously allocated to cluster i

x_i = number of dots previously allocated to cluster i which are labeled as small grain

(notice $x_i \le n_i$)

It should be noted that $\hat{\theta}_{n,x}$ is computed via a function called TH2.

Next, a dot is allocated to the cluster whose $\Delta\sigma_i^{\,2}$ is the largest. Then, for this chosen cluster, n_i is updated to $n_i + 1$. That dot's label is read from the ground truth file. If the label is a small grain, x_i is updated to $x_i + 1$. After this, the segment variance, σ^2, is computed as

$$\sigma^2 = \sum_{i=1}^{m} \left(\frac{N_i}{N}\right)^2 \frac{\hat{\theta}_{n_i,x_i}(1 - \hat{\theta}_{n_i,x_i})}{n_i - 1} \qquad (10\text{-}3)$$

where m = total number of clusters

This σ^2 is printed and the above allocation scheme is repeated until the number of dots allocated reaches a value specified by the user.

The proportion estimate is computed as

$$\hat{P}_{sg} = \sum_{i=1}^{m} \frac{N_i}{N} \hat{\theta}_{n_i,x_i} \qquad (10\text{-}4)$$

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. A grand summary is printed.

10.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAEB4, GETDOT, CLMPCS, GTMPLB, LBLITP
SAEB4	GETDOT, CLMPCS, GTMPLB, LBLITP, TH2
GETDOT	RAN
TH2	None

10.3 INTERFACE

Interface with other routines is through the common block PRTFLG (2 bytes), which is used to control the optional printing of dot files and other information.

10.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line	Parameters	Format	Default input	Default value
1	Cluster map file name	A13	None	
2	a. Number of repetitions	I3	0 or blank	1

	b.	Starting point of first pseudorandom sequence	I5	0 or blank	10
	c.	Number of repetition printings	I3	0 or blank	5
3	a.	Maximum number of dots that can be allocated	I3	None	
	b.	Number of dots initially assigned to each cluster	I3	None	
4		Number of status messages on terminal	I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

10.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report	Contents
Individual repetition	1. Processor header
	2. Ground truth small-grain proportion
	3. A dot file showing dots chosen, their x-y position, their ground truth labels, segment variance, and proportion estimate
	4. A table showing number of clusters, cluster name codes, cluster sizes, and number of dots assigned

Grand summary

1. Processor header
2. A table showing bias, MSE, reduction in MSE, average, variance, and variance reduction as functions of number of dots

10.6 BRIEF ALGORITHM: FOR BAYESIAN DOT ALLOCATION (MODIFIED QUADRATIC PRIOR)

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.
3. Repeat a through h until all repetitions are finished:
   a. Set starting point of the pseudorandom number generator.
   b. Clear a dot counter.
   c. Increment the dot counter.
   d. If the dot counter indicates that the current dot should be allocated as an initial dot, then go to the next step. Otherwise compute $\Delta\sigma_i^2$ for every cluster and choose the cluster with maximum $\Delta\sigma_i^2$.
   e. Allocate a dot to the chosen cluster.
   f. Pick up the small-grain labels, compute segment variance and proportion estimate, and print a line of the dot file.
   g. If this is the end of initial dot assignment, compute the modified parameters a, b, c based on the current small-grain estimate, $\hat{P}_{sg}$. Call TH2PAR to set the modified parameters.
   h. If the dot count does not exceed specified value, go to c.
4. Compute bias, MSE, reduction in MSE, average, variance, and variance reduction, and print a grand summary.

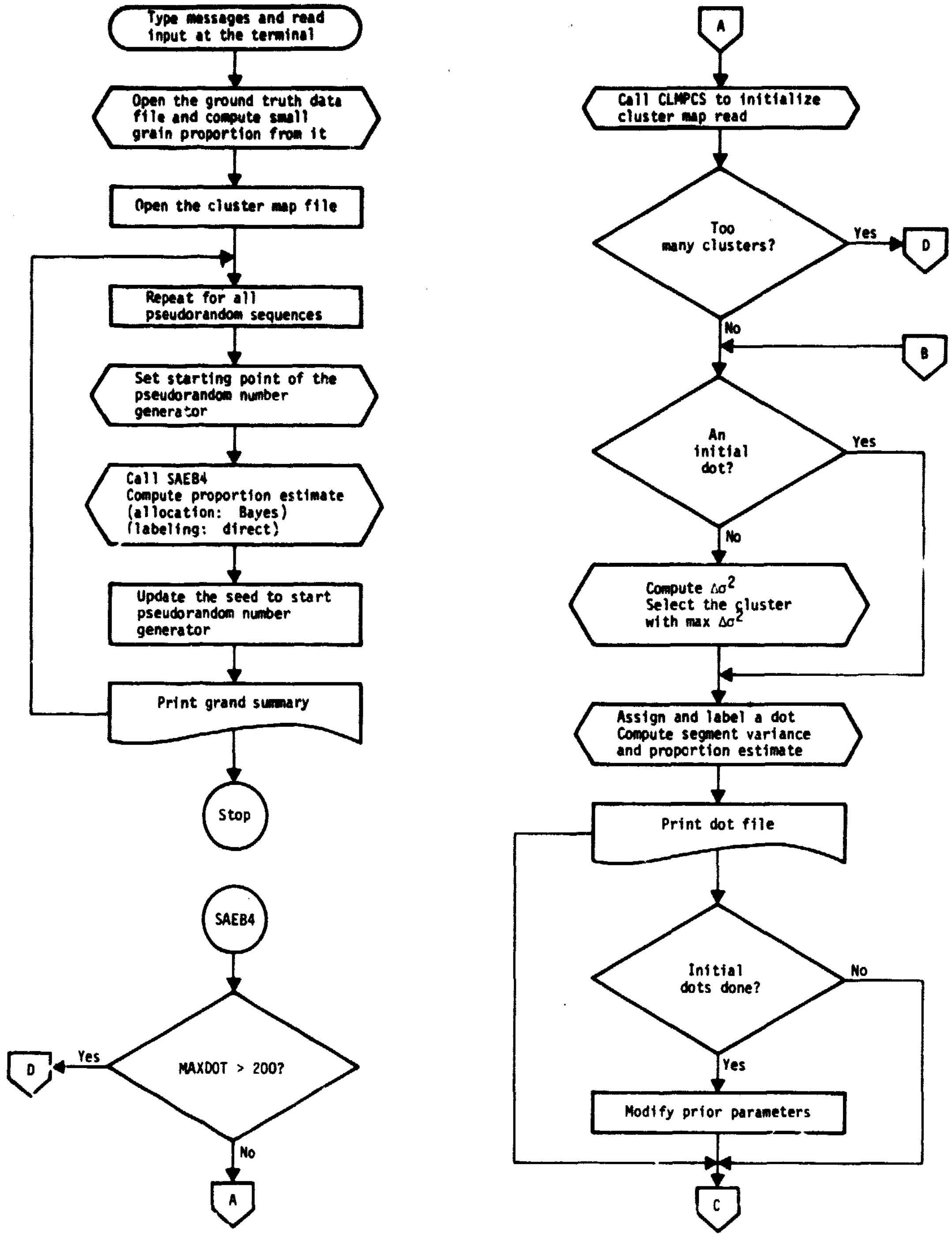

Figure 10-1.– Flow chart for Bayesian dot allocation (modified quadratic prior).

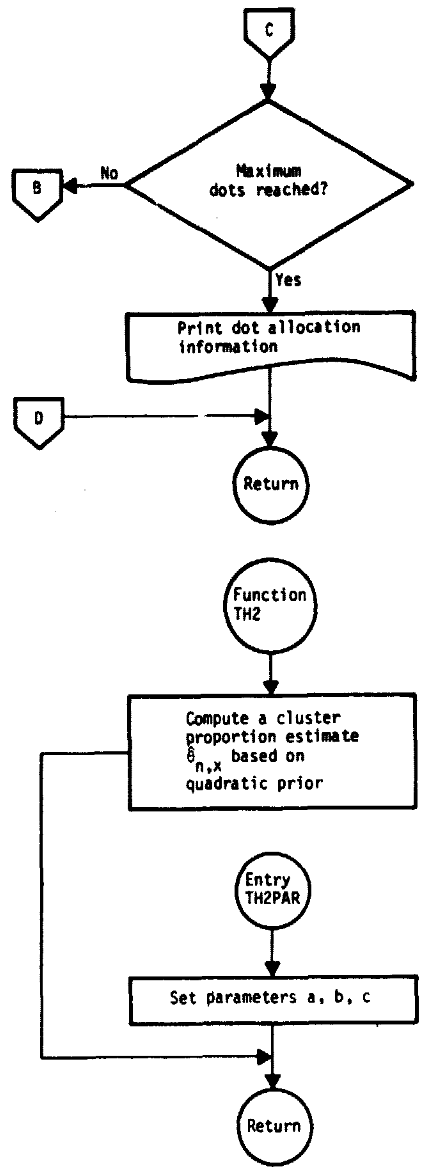

Figure 10-1.– Concluded.

10.7 LISTING

```
C----- PROGRAM A16 (A89.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING BAYES (MODIFIED QUAD PRIOR) DOT ALLOCATION
C----- LABELLING IS DIRECT BY GROUND TRUTH.
C
      BYTE NAME(15),NGT(13)
      REAL PSG(200),PM(200),PB(200)
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C------ READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A16 (A89.TSK).'
     *       /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *       /' DOT ALLOCATION IS BAYES (MODIFIED QUAD PRIOR).'
     *        /' LABELLING IS DIRECT BY GROUND TRUTH.'
     *       /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *       /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *       /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *       /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT(' SPECIFY MAXIMUM NO. OF DOTS IN THE REPETITIONS RUNS.'/
     *       ' AND NO. OF DOTS INTIALLY ASSIGNED TO EACH CLUSTER.'/
     *       ' III III')
      READ(7,142)MAXDOT,NNIX
142   FORMAT(I3,1X,I3)
      CALL INITS(NNIX)
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL. HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED,MAXDOT
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *        /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *        /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6,
     *        /' MAXIMUM NO. OF DOTS IN THE REPETITION RUNS=',I6)
C
C------ COMPUTE BAYES (MODIFIED QUAD PRIOR) ESTIMATE FOR GROUND TRUTH
       NLINE=117
       NCOL=196
      DO 215 K=1,4
215   NGT(K)=NAME(K)
       OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *      ACCESS='DIRECT')
       DO 231 L=1,NLINE
       DO 221 K=1,NCOL
       CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
221    CONTINUE
231    CONTINUE
       P=FLOAT(IP)/NLINE/NCOL
```

```
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *   READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
C
C------ START ESTIMATION FOR AN TO1DOT AND REPETITIONS
      DO 311 K=1,MAXDOT
      PB(K)=0.
311   PM(K)=0.
      JMES=0
      JS=JSEED
      DO 361 J=1,JV
      JFLAG=0
      IF(MOD(J-1,JSKIP).EQ.0) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(5-25-79)'/,2X,
     * 'PROGRAM A16 (A89.TSK): ESTIMATION OF SMALL GRAINS PROPORTION'
     *        /,15X,'DOT ALLOCATION:BAYES (MODIFIED QUAD PRIOR),'
     *        /,15X,'DOT LABELLING:DIRECT BY GROUND TRUTH.'
     *       //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *        /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' --- REPETITION RUN=',I3,
     *        ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAEB4(MAXDOT,PSG)
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)MAXDOT,J,JSEED
323   FORMAT(' MAXDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6)
      JSEED=JSEED+150
C------ COMPUTE BIAS AND M.S.E.
      DO 341 K=1,MAXDOT
      TEMP=PSG(K)-P
      PB(K)=PB(K)+TEMP
341   PM(K)=PM(K)+TEMP**2
361   CONTINUE
C
C------ PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)MAXDOT,JV,JS
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *       /,5X,' MAX. NO. OF DOTS IN EACH REPETITION RUN=',I3,
     *      /,14X,'             NO. OF REPETITION RUNS =',I3,
     *      /,14X,' RANDOM DOTS START WITH SEED=',I6,
     *      //,1X,'  DOT     BIAS        M.S.E.  REDUCTION',
     *            '  AVERAGE  VARIANCE  VAR REDUCTION')
      DO 381 K=1,MAXDOT
      PB(K)=PB(K)/JV
      PM(K)=PM(K)/JV
      RR=PM(K)/( P*(1.-P)/K )
      AVERG=PB(K)+P
      IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
      IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
      REDVAR=VAR/( P*(1.-P)/K )
381   WRITE(6,382)K,PB(K),PM(K),RR,AVERG,VAR,REDVAR
382   FORMAT(2X,I3, 2X,F9.5, 4(F10.6), 3X,F10.6)
C
C
      WRITE(6,401)
401   FORMAT('1 ---- END OF THIS JOB ----')
C
      STOP
      END
C
C
C
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      SUBROUTINE SAEB4(MAXDOT,PSG)
C----- STRATIFIED AREAL ESTIMATION USING BAYES DOT ALLOCATION.
C----- FORMULAS HERE ARE FOR BAYES WITH MODIFIED QUAD PRIOR.
C----- GIVEN ANY MAX NO. OF DOTS IN MAXDOT, PROGRAM WILL RETURN
C----- THE PROPORTIONAL ESTIMATE IS IN ARRAY PSG.
C
C----- WRITEN AND EDITED BY N.Y. CHU ON 5-25-79.
C
C
      INTEGER NDARY(200),MM(51),NN(51),NX(51),LL(51)
      REAL VV,PSG(1)
      BYTE LABEL
C------ COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/, MINSZ/5/
      DATA INIDOT/3/
C
      IF(MAXDOT.GT.200) GOTO 911
C
C------ GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C------ BEGIN BAYES DOT ALLOCATION
C
      IF(JFLAG.EQ.1) WRITE(6,141)
141   FORMAT(//, 25X,'   --- DOT FILE ---'
     *         /'   DOT  CLUSTER INFORMATION  POSITION GROUND TRUTH'
     *         ,' SEGMENT PROPORTIONAL'
     *         /' NUMBER  NO. PIXEL DOTS S.G. LINE,COL   RAW,CODE'
     *         ,'   VARIANCE  ESTIMATE')
C
C------ MAKE SURE EACH CLUSTER HAS A MINIMUM NO. OF PIXELS
      K1=0
      NDS=0
      DO 151 K=1,M
      NX(K)=0
      NN(K)=0
      IF(MM(K).GE.MINSZ) NDS=NDS+INIDOT
      IF(MM(K).LT.MINSZ) K1=K1+MM(K)
      IF(MM(K).LT.MINSZ) NN(K)=-1
151   CONTINUE
      MX=M
      IF(K1.LT.MINSZ) GOTO 161
      MX=M+1
      MM(MX)=K1
      NDS=NDS+INIDOT
161   CONTINUE
C
C------ FROM HERE TO STATEMENT 261: LOOP TO ASSIGN 1 DOT AT A TIME
C
C------ IN ORDER TO ACHIEVE ASSIGNING 3 DOTS INITIALLY TO EACH CLUSTER
C------ THE DELTA VARIANCE COMPUTATION IS BYPASSED FOR THE FIRST
C------ NDS=INIDOT*MX DOTS
C
      KG1=0
      NDOT=0
191   NDOT=NDOT+1
      IF(NDOT.GT.NDS) GOTO 200
195   KG1=KG1+1
      IF(KG1.GT.MX) KG1=1
      K1=KG1
      IF(NN(KG1).GT.-1) GOTO 214
      GOTO 195
C
C
```

```
C------ FOR EACH CLUSTER, COMPUTE DELTA VARIANCE AND CHOOSE THE LARGEST
200   AMAX=0.
      DO 211 K=1,MX
      J=NN(K)
      JX=NX(K)
      IF(J.LE.1) GOTO 211
C
      VAR= ( FLOAT(MM(K))/NPIXEL )**2
     *     *( TH2(J,JX)*(1.-TH2(J,JX))/(J-1.)
     *  -TH2(J,JX)*TH2(J+1,JX+1)*(1.-TH2(J+1,JX+1))/J
     *  -(1.-TH2(J,JX))*TH2(J+1,JX)*(1.-TH2(J+1,JX))/J )
C
C     WRITE(6,99121)K,MM(K),J,NX(K),VAR
99121 FORMAT(' K,MM(K),J,NX(K),VAR=',I4,I8,2I5,E16.6)
      IF(VAR.LT.AMAX) GOTO 211
      K1=K
      AMAX=VAR
211   CONTINUE
C
C------ ASSIGN A DOT TO THE CLUSTER WITH MAX DELTA VARIANCE
      IF(NN(K1).GE.MM(K1)) WRITE(6,213)NDOT,K1,MM(K1)
      IF(NN(K1).GE.MM(K1)) GOTO 241
213   FORMAT(' ATTEMPT TO ASSIGN DOT=',I3,' WHILE NO PIXEL',
     *       ' IS AVAILABLE. CLUSTER=',I3,' TOT PIXEL=',I6)
C---- POSITION THE POINTER TO NDARY
214   ND=1
      DO 215 K=1,K1
      IF(NN(K).LE.0) GOTO 215
      ND=ND+NN(K)
215   CONTINUE
C---- MAKE ROOM FOR THE NEW DOT IN NDARY
      N1=-1
      DO 217 K=NDOT,ND,N1
217   NDARY(K+1)=NDARY(K)
      ND=ND-NN(K1)
      NN(K1)=NN(K1)+1
C------ ASSIGN A DOT
      CALL GETDOT(MM(K1),NN(K1),NDARY(ND),IDOT)
C     WRITE(6,99141)K1,ND,NN(K1),NX(K1),(NDARY(K),K=1,44)
99141 FORMAT(' K1,ND,NN,NX=',4I4, 4(/,2X,15I4) )
      IF(K1.LE.M) GOTO 227
      DO 225 K1=1,M
      IF(NN(K1).GT.0) GOTO 225
      IF(IDOT.LT.MM(K1)) GOTO 227
      IDOT=IDOT-MM(K1)
225   CONTINUE
227   CALL CLMPXY(K1,IDOT,NL,NC,LL)
      CALL GTMPLB(NL,NC,LABEL)
      CALL LBLITP(LABEL,LB1,NX(K1))
      PSG(NDOT)=0.
      DO 229 JX=1,MX
      IF(NN(JX).LE.0) GOTO 229
      PSG(NDOT)=PSG(NDOT)+ FLOAT(MM(JX))/NPIXEL*TH2(NN(JX),NX(JX))
229   CONTINUE
C
C
C------ COMPUTE  SEGMENT VARIANCE
241   VV=0.
      DO 251 K=1,MX
      IF(NN(K).LE.1) GOTO 251
      VV=VV+ (FLOAT(MM(K))/NPIXEL)**2 *TH2(NN(K),NX(K))
     *       *(1.-TH2(NN(K),NX(K)))/(NN(K)-1.)
251   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,255)NDOT,K1,IDOT,NN(K1),NX(K1)
     *           ,NL,NC,LABEL,LB1,VV,PSG(NDOT)
255   FORMAT(2X,I3, 3X,I3,1X,I6,1X,I3,2X,I3, 2X,2I4, 2X,2I4, 2F11.6)
C
C------ MODIFY QUADRATIC PRIOR PARAMETERS UPON
C------ FINISHING ASSIGNMENT OF INITIAL DOTS
C
```

```
      IF(NDOT.NE.NDS) GOTO 260
      A=6.
      PX=PSG(NDOT)
      IF (0.211.GT.PX) PX=0.211
      IF (PX.GT.0.789) PX=0.789
      B = 12.*(PX-1.)
      C = 5.-6.*PX
      CALL THEPAR(A,B,C)
      IF(JFLAG.EQ.1) WRITE(6,257)NDOT,PX,A,B,C
257   FORMAT(' AFTER ASSIGNING',I4,' DOTS AND ESTIMATE =',F10.5,/,
     * ' RESETTING PRIOR PARAMETERS: A=',F9.5,' B=',F9.5,' C=',F9.5)
260   CONTINUE
C
C----- END OF MODIFYING PRIOR PARAMETERS
C
      IF(NDOT.LT.MAXDOT) GOTO 191
C
      DO 261 K=1,MX
      IF(NN(K).LE.-1) NN(K)=0
261   CONTINUE
C
C
C----- END OF BAYES DOT ALLOCATION
C
C
C------ WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)MAXDOT,NPIXEL,MX
282   FORMAT(//,5X,'            TOTAL NO. OF DOTS AVAILABLE =',I6,
     *         /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *         /,5X,'                   TOTAL NO. OF CLUSTERS =',I6,
     *         /,5X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS  LABELLED'
     *         /,5X,'   NO.      CODE  PIXELS   ASSIGNED  SMALL GRAIN')
      DO 283 K=1,MX
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K),NX(K)
283   CONTINUE
284   FORMAT(8X,I2,5X,I4,3X,I6,4X,I3, 10X,I3)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
285   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286   FORMAT(/'  THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *         ') ARE:', 3(/,16(I3,',')) )
288   CONTINUE
C
C
C
C
      GOTO 990
C
C
901   WRITE(6,902)M
902   FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
911   WRITE(6,912)NDOT
912   FORMAT(' YOU SPECIFIED',I4,' TOO MANY DOTS(MAX=200)')
C
C---- ENTRY TO DEFINE INTIAL NO. OF DOTS TO BE ASSIGNED
      ENTRY INITS(NNIT)
      INIDOT=NNIT
990   RETURN
      END
```

```
C
C
C
      FUNCTION TH2(N,NX)
C----- FUNCTION TO EVALUATE CLUSTER PROPORTIONAL ESTIMATE
C------ N=NO. OF DOTS ASSIGNED FOR A CLUSTER,
C------ NX= NO. OF DOTS LABELLED AS SMALL GRAIN
C------ BOTH N AND NX ARE SIMPLE INTEGER (2 BYTES IN PDP)
C------ TH2 (REAL NO.) CONTAINS THE ESTIMATE ON RETURN.
C
      INTEGER N,NX
      REAL TH2
C
C     DATA A/6.31924/, B/-8.19799/, C/2.99258/
      DATA A/6./,B/-7.877/,C/2.9345/
C
      X=NX
      TH2 =
     * ( A*(X+1.)*(X+2.)*(X+3.) + B*(X+1.)*(X+2.)*(N+4.)
     *      + C*(X+1.)*(N+3.)*(N+4.) ) /
     * ( A*(X+1.)*(X+2.)*(N+4.) + B*(X+1.)*(N+3.)*(N+4.)
     *      + C*(N+2.)*(N+3.)*(N+4.) )
      RETURN
C
C----- ENTRY TO ALLOW CHANGING OF PARAMETERS
C
      ENTRY TH2PAR(AX,BX,CX)
      A=AX
      B=BX
      C=CX
      RETURN
C
      END
```

11. A91: BAYESIAN DOT ALLOCATION (ADAPTIVE PRIOR)

11.1 DESCRIPTION

Implemented in this piece of software is a scheme in which dot allocation is sequentially Bayesian (adaptive prior) and labeling is direct from ground truth. This scheme is similar to those in A84, A85, A87, and A89, except that here estimates based on two priors are computed for the initial dots, then one prior is chosen and all estimates are based on the selected prior. Such a scheme improves the estimate if the small-grain proportion is less than 0.21.

In the Bayesian dot allocation scheme, dots are allocated sequentially to those clusters that may reduce the variance of the segment proportion estimate the most. In order to do this, we must first compute the expected change in variance, $\Delta\sigma^2$, for each cluster. The expected change in variance for cluster i is defined as

$$\Delta\sigma_i^2 = \left(\frac{N_i}{N}\right)^2 \left[\frac{\hat{\theta}_{n_i,x_i}(1 - \hat{\theta}_{n_i,x_i})}{n_i - 1} - \frac{\hat{\theta}_{n_i,x_i}\hat{\theta}_{n_i+1,x_i+1}(1 - \hat{\theta}_{n_i+1,x_i+1})}{n_i} - \frac{(1 - \hat{\theta}_{n_i,x_i})\,\hat{\theta}_{n_i+1,x_i}(1 - \hat{\theta}_{n_i+1,x_i})}{n_i}\right] \tag{11-1}$$

where N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

n_i = number of dots previously allocated to cluster i

x_i = number of dots previously allocated to cluster i which are labeled as small grain

(notice $x_i \leq n_i$)

For quadratic prior, $\hat{\theta}_{n,x}$ is defined as

$$\hat{\theta}_{n,x} = \frac{a(x+1)(x+2)(x+3) + b(x+1)(x+2)(n+4) + c(x+1)(n+3)(n+4)}{a(x+1)(x+2)(n+4) + b(x+1)(n+3)(n+4) + c(n+2)(n+3)(n+4)} \tag{11-2}$$

For initial dots: $a = 6,\ b = -7.877,\ c = 2.9345$

After initial dots: $a = 6,\ b = 12(\hat{P}_{sg} - 1),\ c = 5 - 6\hat{P}_{sg}$ if $0.211 \le \hat{P}_{sg} < 0.789$

$a = 6,\ b = 12(0.211 - 1),\ c = 5 - 6 \times 0.211$ if $0.211 > \hat{P}_{sg}$

$a = 6,\ b = 12(0.789 - 1),\ c = 5 - 6 \times 0.789$ if $0.789 < \hat{P}_{sg}$

where $\hat{P}_{sg}$ = estimate of small-grain proportion after assigning initial dots

For exponential prior, $\hat{\theta}_{n,x}$ is defined as

$$\hat{\theta}_{n,x} = \frac{\frac{x+1-\alpha}{n+2-\alpha} - \frac{x+1}{n+2}\left(1 - \frac{\alpha}{n+1}\right)\left(1 - \frac{\alpha}{n}\right)\cdots\left(1 - \frac{\alpha}{x+2}\right)\left(1 - \frac{\alpha}{x+1}\right)}{1 - \left(1 - \frac{\alpha}{n+1}\right)\left(1 - \frac{\alpha}{n}\right)\cdots\left(1 - \frac{\alpha}{x+2}\right)\left(1 - \frac{\alpha}{x+1}\right)} \tag{11-3}$$

$$\text{where } \alpha = \frac{1 - 4\hat{P}_{sg}}{1 - 2\hat{P}_{sg}} \tag{11-4}$$

$\hat{P}_{sg}$ = estimate of small-grain proportion after assigning initial dots

For the initial dots assignment, estimates of small-grain proportion are computed on the basis of both priors. After all initial dots are assigned, the estimate based on the exponential prior is used to determine which prior is going to be used. If it is larger than 0.21, then the quadratic prior is selected; otherwise the exponential prior is selected.

Next, a dot is allocated to the cluster whose $\Delta\sigma_i^2$ is the largest. Then, for this chosen cluster, n_i is updated to $n_i + 1$. That dot's label is read from the ground truth file. If the label is a small grain, x_i is updated to $x_i + 1$. After this, the segment variance, σ^2, will be computed as

$$\sigma^2 = \sum_{i=1}^{m} \left(\frac{N_i}{N}\right)^2 \frac{\hat{\theta}_{n_i,x_i}(1 - \hat{\theta}_{n_i,x_i})}{n_i - 1} \qquad (11\text{-}5)$$

where m = total number of clusters

This σ^2 is printed and the above allocation scheme is repeated until the number of dots allocated reaches a value specified by the user.

The proportion estimate based on the quadratic prior is computed as

$$\hat{P}_{sg} = \sum_{i=1}^{m} \frac{N_i}{N} \hat{\theta}_{n_i,x_i} \qquad (11\text{-}6)$$

The proportional estimate based on the exponential prior is

$$\hat{P}_{sg} = \sum_{i=1}^{m} \frac{x_i}{n_i} \frac{N_i}{N}$$

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. A grand summary is printed.

11.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator, and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAEB5, GETDOT, CLMPCS, GTMPLB, LBLITP
SAEB5	GETDOT, CLMPCS, GTMPLB, LBLITP, TH3
GETDOT	RAN
TH3	None

11.3 INTERFACE

Interface with other routines is through the common block PRTFLG (2 bytes), which is used to control the optional printing of dot files and other information.

11.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line		Parameters	Format	Default input	Default value
1		Cluster map file name	A13	None	
2	a.	Number of repetitions	I3	0 or blank	1
	b.	Starting point of first pseudorandom sequence	I5	0 or blank	10
	c.	Number of repetition printings	I3	0 or blank	5
3	a.	Maximum number of dots that can be allocated	I3	None	

	b. Number of dots initially assigned to each cluster	I3	None	
4	Number of status messages on terminal	I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

11.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report	Contents
Individual repetition	1. Processor header
	2. Ground truth small-grain proportion
	3. A dot file showing dots chosen, their x-y position, their ground truth labels, segment variance, and proportion estimates based on both quadratic and exponential priors
	4. A table showing number of clusters, cluster name codes, cluster sizes, and number of dots assigned
Grand summary	1. Processor header
	2. A table showing bias, MSE, reduction in MSE, average, variance, and variance reduction of the estimate based on the selected prior as functions of number of dots

11.6 BRIEF ALGORITHM: FOR BAYESIAN DOT ALLOCATION (ADAPTIVE PRIOR)

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.
3. Repeat a through h until all repetitions are finished:
   a. Set starting point of the pseudorandom number generator.
   b. Clear a dot counter.
   c. Increment the dot counter.
   d. If the dot counter indicates that the current dot should be allocated as an initial dot, compute estimates based on both priors, then go to the next step. Otherwise compute $\Delta\sigma_i^2$ for every cluster and choose the cluster with maximum $\Delta\sigma_i^2$.
   e. Allocate a dot to the chosen cluster.
   f. Pick up the small-grain labels, compute segment variance and proportion estimate, and print a line of the dot file.
   g. If this is the end of initial dot assignment, do the following:
      (1) Determine if current exponential prior estimate exceeds 0.21, then call SETPRI to select to compute exponential prior estimate and call SETEP to set the value of α.
      (2) Otherwise, call SETPRI to select to compute quadratic prior estimate, then call TH3PAR to modify the prior parameters.
   h. If the dot count does not exceed specified value, go to c.
4. Compute bias, MSE, reduction in MSE, average, variance, variance reduction, and print a grand summary.

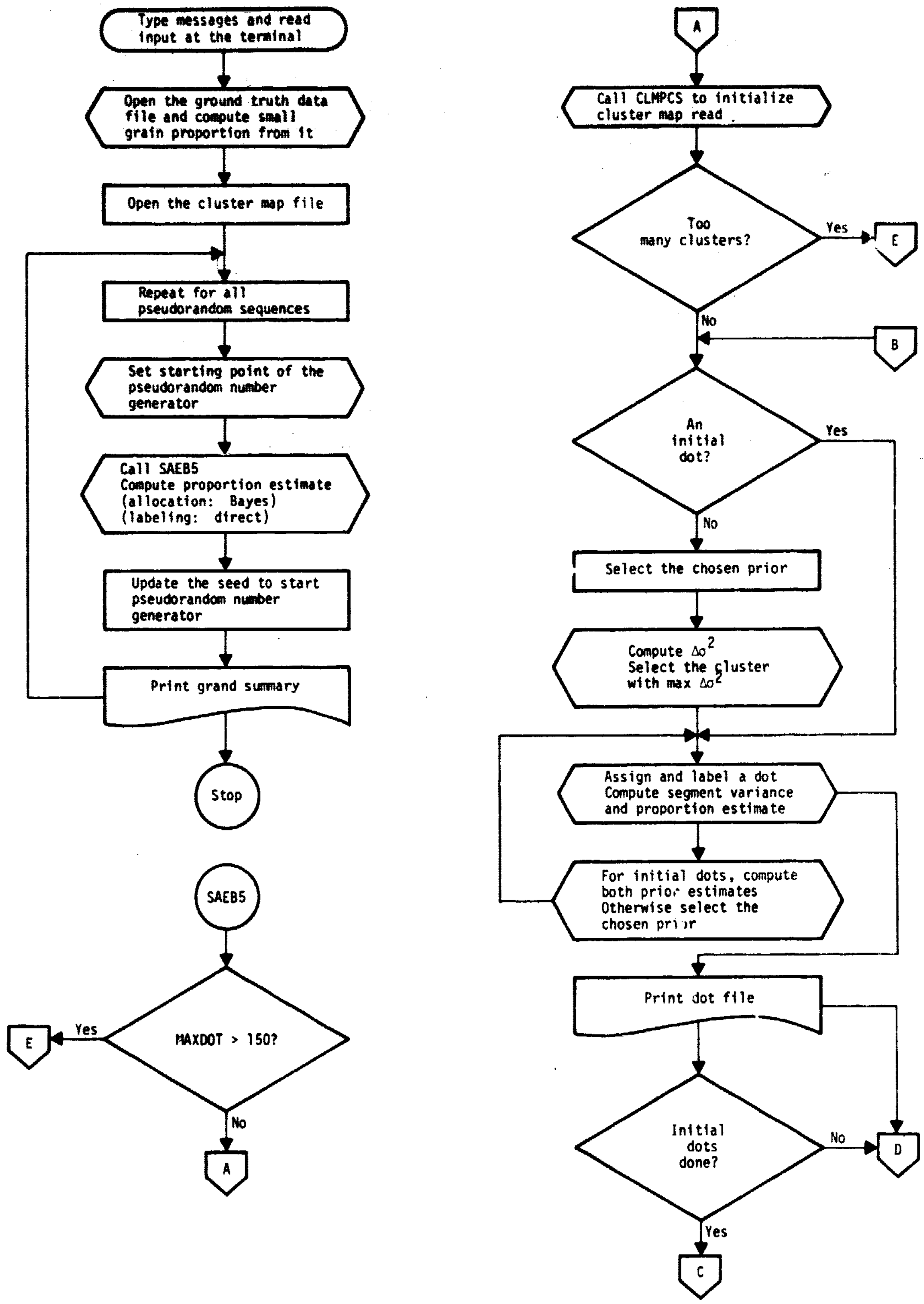

Figure 11-1.– Flow chart for Bayesian dot allocation (adaptive prior).

[redacted]

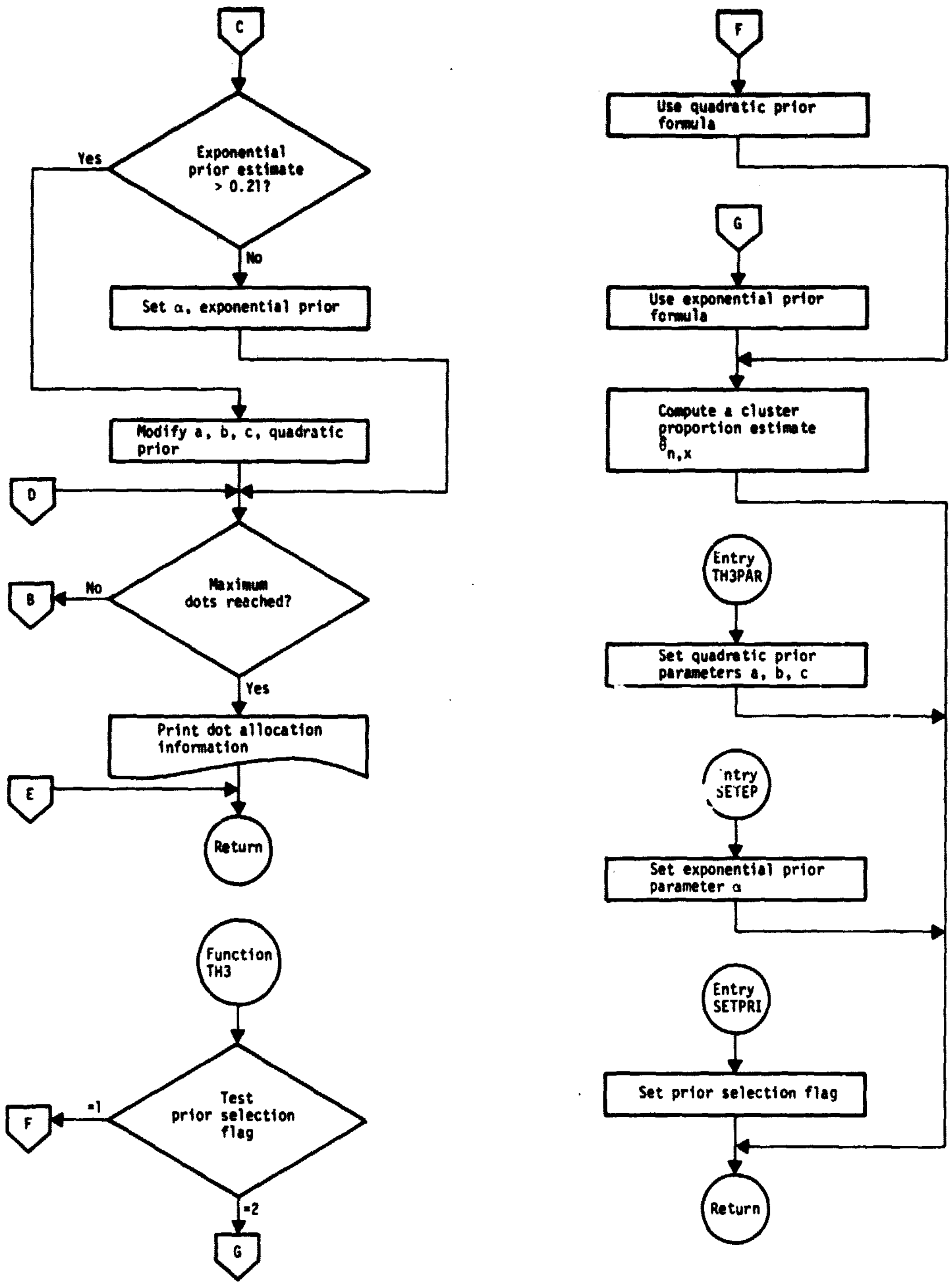

Figure 11-1.– Concluded.

11.7 LISTING

```
C----- PROGRAM A18 (A91.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING BAYES (ADAPTIVE PRIOR) DOT ALLOCATION
C----- LABELLING IS DIRECT BY GROUND TRUTH.
C
C----- ADAPTIVE PRIOR MEANS
C----- WHEN ESTIMATE IS GREATER TH3N 0.21, USE QUADRATIC PRIOR
C----- OTHERWISE USE EXPONENTIAL PRIOR
C
C
      BYTE NAME(15),NGT(13)
      REAL PSG(150),PM(150),PB(150)
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C----- READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A18 (A91.TSK).'
     *         /' PROPORTION ESTIMATION OF SMALL GRAIN'
     *         /' DOT ALLOCATION IS BAYES (ADAPTIVE PRIOR).'
     *          /' LABELLING IS DIRECT BY GROUND TRUTH.'
     *         /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *         /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *         /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *         /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT(' SPECIFY MAXIMUM NO. OF DOTS IN THE REPETITIONS RUNS,'/
     *        ' AND NO. OF DOTS INTIALLY ASSIGNED TO EACH CLUSTER.'/
     *        ' III III')
      READ(7,142)MAXDOT,NNIX
142   FORMAT(I3,1X,I3)
      CALL INITS(NNIX)
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL, HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED,MAXDOT
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *          /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *          /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6,
     *          /' MAXIMUM NO. OF DOTS IN THE REPETITION RUNS=',I6)
C
C----- COMPUTE BAYES (ADAPTIVE PRIOR) ESTIMATE FOR GROUND TRUTH
       NLINE=117
       NCOL=196
      DO 215 K=1,4
215   NGT(K)=NAME(K)
       OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *      ACCESS='DIRECT')
       DO 231 L=1,NLINE
       DO 221 K=1,NCOL
       CALL GTMPLB(L,K,LABEL)
```

```
      CALL LBLITP(LABEL,LB1,IP)
221    CONTINUE
231    CONTINUE
       P=FLOAT(IP)/NLINE/NCOL
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *   READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
C
C----- START ESTIMATION FOR AN TOTDOT AND REPETITIONS
      DO 311 K=1,MAXDOT
      PB(K)=0.
311   PM(K)=0.
      JMES=0
      JS=JSEED
      DO 361 J=1,JV
      JFLAG=0
      IF(MOD(J-1,JSKIP).EQ.0) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(6-19-79)'/,2X,
     * 'PROGRAM A18 (A91.TSK): ESTIMATION OF SMALL GRAINS PROPORTION'
     *       /,15X,'DOT ALLOCATION:BAYES (ADAPTIVE PRIOR),'
     *       /,15X,'DOT LABELLING:DIRECT BY GROUND TRUTH.'
     *      //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *       /,10X,' PROPORTION ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' --- REPETITION RUN=',I3,
     *       ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAEB5(MAXDOT,PSG)
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)MAXDOT,J,JSEED,PSG(MAXDOT)
323   FORMAT(' MAXDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6,
     *       ' LAST ESTIMATE=',F10.6)
      JSEED=JSEED+150
C------ COMPUTE BIAS AND M.S.E.
      DO 341 K=1,MAXDOT
      TEMP=PSG(K)-P
      PB(K)=PB(K)+TEMP
341   PM(K)=PM(K)+TEMP**2
361   CONTINUE
C
C------ PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)MAXDOT,JV,JS
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *       /,5X,' MAX. NO. OF DOTS IN EACH REPETITION RUN=',I3,
     *      /,14X,'           NO. OF REPETITION RUNS =',I3,
     *      /,14X,' RANDOM DOTS START WITH SEED=',I6,
     *      //,1X,'  DOT     BIAS        M.S.E.   REDUCTION',
     *           '  AVERAGE  VARIANCE  VAR REDUCTION')
      DO 381 K=1,MAXDOT
      PB(K)=PB(K)/JV
      PM(K)=PM(K)/JV
      RR=PM(K)/( P*(1.-P)/K )
      AVERG=PB(K)+P
      IF(JV.EQ.1) VAR= PM(K)-PB(K)**2
      IF(JV.GT.1) VAR= (PM(K)-PB(K)**2)*JV/(JV-1)
      REDVAR=VAR/( P*(1.-P)/K )
381   WRITE(6,382)K,PB(K),PM(K),RR,AVERG,VAR,REDVAR
382   FORMAT(2X,I3, 2X,F9.5, 4(F10.6), 3X,F10.6)
C
C
      WRITE(6,401)
401   FORMAT('1 ----- END OF THIS JOB ---')
C
      STOP
      END
```

```
C
C
C
      SUBROUTINE SAEBS(MAXDOT,PSG)
C----- STRATIFIED AREAL ESTIMATION USING BAYES DOT ALLOCATION.
C----- FORMULAS HERE ARE FOR BAYES WITH ADAPTIVE PRIOR.
C----- GIVEN ANY MAX NO. OF DOTS IN MAXDOT, PROGRAM WILL RETURN
C----- THE PROPORTION ESTIMATE IS IN ARRAY PSG.
C
C----- WRITEN AND EDITED BY N.Y. CHU ON 6-19-79.
C
C
      INTEGER NDARY(150),MM(51),NN(51),NX(51),LL(51)
      REAL VU,PSG(1)
      BYTE LABEL
      INTEGER*4 NMPR
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/, MINSZ/5/
      DATA INIDOT/3/
      DATA JFP/1/,JINIT/0/
C
      IF(MAXDOT.GT.150) GOTO 911
C
C----- GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C----- BEGIN BAYES DOT ALLOCATION
C
      IF(JFLAG.EQ.1) WRITE(6,141)
141   FORMAT(//, 25X,'   --- DOT FILE ---'
     *       /'    DOT  CLUSTER INFORMATION  POSITION GROUND TRUTH'
     *       ,' SEGMENT PROPORTION ESTIMATE'
     *       /' NUMBER  NO. PIXEL DOTS S.G. LINE,COL   RAW,CODE'
     *       ,'   VARIANCE   QUAD        EXP')
C
C----- MAKE SURE EACH CLUSTER HAS A MINIMUM NO. OF PIXELS
      K1=0
      NDS=0
      DO 151 K=1,M
      NX(K)=0
      NN(K)=0
      IF(MM(K).GE.MINSZ) NDS=NDS+INIDOT
      IF(MM(K).LT.MINSZ) K1=K1+MM(K)
      IF(MM(K).LT.MINSZ) NN(K)=-1
151   CONTINUE
      MX=M
      IF(K1.LT.MINSZ) GOTO 161
      MX=M+1
      MM(MX)=K1
      NDS=NDS+INIDOT
161   CONTINUE
C
      IF(JFP.EQ.1) NMPR='QUAD'
      IF(JFP.EQ.2) NMPR='EXP '
      IF(JFLAG.EQ.1) WRITE(6,171)NDS,NMPR
171   FORMAT(' INITIAL ASSIGNMENT:',I4,' DOTS.',
     *       ' SEQ VAR IS BASED ON ',A4,' PRIOR.')
C
C----- FROM HERE TO STATEMENT 261: LOOP TO ASSIGN 1 DOT AT A TIME
C
C----- IN ORDER TO ACHIEVE ASSIGNING 3 DOTS OR LESS INITIALLY TO EACH CLUSTER
C----- THE DELTA VARIANCE COMPUTATION IS BYPASSED FOR THE FIRST
C----- NDS=INIDOT*MX DOTS
C
```

```
      KG1=0
      NDOT=0
191   NDOT=NDOT+1
      IF(NDOT.GT.NDS) GOTO 200
195   KG1=KG1+1
      IF(KG1.GT.MX) KG1=1
      K1=KG1
      IF(NN(KG1).GT.-1) GOTO 214
      GOTO 195
C
C
C------ FOR EACH CLUSTER, COMPUTE DELTA VARIANCE AND CHOOSE THE LARGEST
200   CONTINUE
C------ SET THE PRIOR TO BE USED
      CALL SETPRI(JFP)
      AMAX=0.
      DO 211 K=1,MX
      J=NN(K)
      JX=NX(K)
      IF(J.LE.1) GOTO 211
C
      VAR= ( FLOAT(MM(K))/NPIXEL )**2
     *     *( TH3(J,JX)*(1.-TH3(J,JX))/(J-1.)
     *  -TH3(J,JX)*TH3(J+1,JX+1)*(1.-TH3(J+1,JX+1))/J
     *  -(1.-TH3(J,JX))*TH3(J+1,JX)*(1.-TH3(J+1,JX))/J )
C
C     WRITE(6,99121)K,MM(K),J,NX(K),VAR
99121 FORMAT(' K,MM(K),J,NX(K),VAR=',I4,I8,2I5,E16.6)
      IF(VAR.LT.AMAX) GOTO 211
      K1=K
      AMAX=VAR
211   CONTINUE
C
C----- ASSIGN A DOT TO THE CLUSTER WITH MAX DELTA VARIANCE
      IF(NN(K1).GE.MM(K1)) WRITE(6,213)NDOT,K1,MM(K1)
      IF(NN(K1).GE.MM(K1)) GOTO 241
213   FORMAT(' ATTEMPT TO ASSIGN DOT=',I3,' WHILE NO PIXEL',
     *       ' IS AVAILABLE. CLUSTER=',I3,' TOT PIXEL=',I6)
C---- POSITION THE POINTER TO NDARY
214   ND=1
      DO 215 K=1,K1
      IF(NN(K).LE.0) GOTO 215
      ND=ND+NN(K)
215   CONTINUE
C---- MAKE ROOM FOR THE NEW DOT IN NDARY
      N1=-1
      DO 217 K=NDOT,ND,N1
217   NDARY(K+1)=NDARY(K)
      ND=ND-NN(K1)
      NN(K1)=NN(K1)+1
C------ ASSIGN A DOT
      CALL GETDOT(MM(K1),NN(K1),NDARY(ND),IDOT)
C     WRITE(6,99141)K1,ND,NN(K1),NX(K1),(NDARY(K),K=1,44)
99141 FORMAT(' K1,ND,NN,NX=',4I4, 4(/,2X,15I4) )
      IF(K1.LE.M) GOTO 227
      DO 225 K1=1,M
      IF(NN(K1).GT.0) GOTO 225
      IF(IDOT.LT.MM(K1)) GOTO 227
      IDOT=IDOT-MM(K1)
225   CONTINUE
227   CALL CLMPXY(K1,IDOT,NL,NC,LL)
      CALL GTMPLB(NL,NC,LABEL)
      CALL LBLITP(LABEL,LB1,NX(K1))
```

```
C
C----- COMPUTE PROP EST DUE TO QUADRATIC PRIOR
C----- COMPUTE PROP EST DUE TO EXPONENTIAL PRIOR
      P1=0.
      P2=0.
      DO 231 JX=1,MX
      IF(NN(JX).LE.0) GOTO 231
C------ AMAX,K1,N1 ARE TEMPORY VARIABLES
      AMAX= FLOAT(MM(JX))/NPIXEL
      N1=NN(JX)
      J=NX(JX)
C
C----- FIRST, ACCUMULATE REST WITH QUAD PRIOR
      IF(NDOT.LE.NDS) CALL SETPRI(1)
      IF(NDOT.GT.NDS) CALL SETPRI(JFP)
      P1= P1 + AMAX*TH3(N1,J)
C
C------ NEXT, ACCUMULATE EST WITH EXP PRIOR
      IF(NDOT.GT.NDS) GOTO 231
      P2= P2 + AMAX*J/N1
231   CONTINUE
C
C------- STORE THE APPROPIATE PROP EST IN THEIR CORRECT PLACE
      IF(NDOT.GT.NDS) GOTO 233
      PSG( (NDOT-1)*2+1 )=P1
      PSG( NDOT*2 ) =P2
      GOTO 235
233   PSG(NDOT)=P1
      IF(JFP.EQ.1) P2=0.
      IF(JFP.EQ.2) P2=P1
      IF(JFP.EQ.2) P1=0.
235   CONTINUE
C
C
C------ COMPUTE  SEGMENT VARIANCE
241   UU=0.
      CALL SETPRI(JFP)
      DO 251 K=1,MX
      IF(NN(K).LE.1) GOTO 251
      UU=UU+ (FLOAT(MM(K))/NPIXEL)**2 *TH3(NN(K),NX(K))
     *        *(1.-TH3(NN(K),NX(K)))/(NN(K)-1.)
251   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,255)NDOT,K1,IDOT,NN(K1),NX(K1)
     *        ,NL,NC,LABEL,LB1,UU,P1,P2
255   FORMAT(2X,I3, 3X,I3,1X,I6,1X,I3,2X,I3, 2X,2I4, 2X,I4,I5, 3F10.6)
C
C------ DETERMINE WHICH PRIOR TO USE
      IF(NDOT.NE.NDS) GOTO 275
C
C------ CHOOSE PRIOR ACCORDING TO VERY FIRST REPETITION OF THE RUN
      IF((JINIT.NE.0).AND.(JFP.EQ.1)) GOTO 256
      IF((JINIT.NE.0).AND.(JFP.EQ.2)) GOTO 259
      JINIT=1
C------ CHOOSE PRIOR ACCORDING TO ESTIMATE FOR THE FIRST REPETITION
C------ USE THE EXP PRIOR ESTIMATE TO DECIDE
      PX=PSG( NDOT*2 )
      IF(PX.LE.0.21) GOTO 259
C
C------ QUAD PRIOR IS CHOSEN, NEED TO MODIFY QUAD PRIOR PARAMETERS
256   PX=PSG( (NDOT-1)*2+1 )
      JFP=1
      NMPR='QUAD'
C
```

```
      A=6.
      IF (0.211.GT.PX) PX=0.211
      IF (PX.GT.0.789) PX=0.789
      B = 12.*(PX-1.)
      C = 5.-6.*PX
      CALL TH3PAR(A,B,C)
      IF(JFLAG.EQ.1) WRITE(6,257)A,B,C
257   FORMAT(' RESETTING',
     * ' QUAD PRIOR PARAMETERS: A=',F9.5,' B= ,F9.5,' C=',F9.5)
C
      GOTO 265
C
C----- EXP PRIOR IS CHOSEN, NEED TO SET EXP PRIOR PARAMETERS
259   PX=PSG(NDOT*2)
      IF(PX.LE.0.01 ) PX=0.01
      JFP=2
      NMPR='EXP '
      IF(PX.LE.0.249) GOTO 262
      IF(JFLAG.EQ.1) WRITE(6,261)PX
261   FORMAT(' ESTIMATE= ',F9.6,' TOO HIGH, BEING RESET TO AVOID',
     *        ' NEGATIVE ALPHA.')
      PX=0.249
262   CALL SETEP( (1.-4.*PX)/(1.-2.*PX) )
C
C----- STORE THE CHOSEN PRIOR ESTIMATES, THESE WILL BE RETURNED
265   DO 267 K=1,NDS
267   PSG(K)=PSG( (K-1)*2+JFP )
C
      IF(JFLAG.EQ.1) WRITE(6,271)PX,NMPR,NMPR
271   FORMAT(' ESTIMATE=',F10.6,'. NOW STORE THE ',A4,' PRIOR'
     *        ' ESTIMATES...'/' SEQ VAR AND DELTA VAR ARE',
     *        ' BEING COMPUTED WITH ',A4,' PRIOR.')
C
275   CONTINUE
C
C
      IF(NDOT.LT.MAXDOT) GOTO 191
C
277   DO 278 K=1,MX
      IF(NN(K).LE.-1) NN(K)=0
278   CONTINUE
C
C
C----- END OF BAYES DOT ALLOCATION
C
C
C
C----- WRITE CLUSTER INFORMATION
280   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,282)MAXDOT,NPIXEL,MX
282   FORMAT(//,5X,'              TOTAL NO. OF DOTS AVAILABLE =',I6,
     *         /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *         /,5X,'                  TOTAL NO. OF CLUSTERS =',I6,
     *         /,5X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS  LABELLED'
     *         /,5X,'   NO.     CODE  PIXELS   ASSIGNED  SMALL GRAIN')
      DO 283 K=1,MX
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K),NX(K)
283   CONTINUE
284   FORMAT(8X,I2,5X,I4,3X,I6,4X,I3, 10X,I3)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
```

```
285    CONTINUE
       IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286    FORMAT(/'  THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
      *        ') ARE:', 3(/,16(I3,',')) )
288    CONTINUE
C
C
C
C
       GOTO 990
C
C
901    WRITE(6,902)M
902    FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
       GOTO 990
911    WRITE(6,912)NDOT
912    FORMAT(' YOU SPECIFIED',I4,' TOO MANY DOTS(MAX=150)')
C
C
C----- ENTRY TO DEFINE INTIAL NO. OF DOTS TO BE ASSIGNED
       ENTRY INITS(NNIT)
       INIDOT=NNIT
990    RETURN
       END
C
C
C
       FUNCTION TH3(N,NX)
C------ FUNCTION TO EVALUATE CLUSTER PROPORTION ESTIMATE
C------- FOR THE CASE (1) QUADRATIC PRIOR IS USED
C--------             (2) EXPONENTIAL PRIOR IS USED
C------ N=NO. OF DOTS ASSIGNED FOR A CLUSTER.
C------ NX= NO. OF DOTS LABELLED AS SMALL GRAIN
C------ BOTH N AND NX ARE SIMPLE INTEGER (2 BYTES IN PDP)
C------ TH3 (REAL NO.) CONTAINS THE ESTIMATE ON RETURN.
C
       INTEGER N,NX
       REAL TH3
C
       DATA A/6./,B/-7.877/,C/2.9345/
C
C---- CHOOSE THE DESIRED PRIOR
       IF(JFP.EQ.2) GOTO 200
C
C----- THE FOLLOWING IS FOR QUADRATIC PRIOR
120    X=NX
       TH3 =
      * ( A*(X+1.)*(X+2.)*(X+3.) + B*(X+1.)*(X+2.)*(N+4.)
      *     + C*(X+1.)*(N+3.)*(N+4.) ) /
      * ( A*(X+1.)*(X+2.)*(N+4.) + B*(X+1.)*(N+3.)*(N+4.)
      *     + C*(N+2.)*(N+3.)*(N+4.) )
C
       RETURN
C
C
C------- THE FOLLOWING IS FOR EXPONENTIAL PRIOR
200    KX=NX+1
C
       P1=(1.-AL/KX)
211    IF(KX.GT.N) GOTO 221
       KX=KX+1
       P1=P1*(1.-AL/KX)
       GOTO 211
C
221    TH3= ( (NX+1.-AL)/(N+2.-AL) - (NX+1.)/(N+2.)*P1 )/ ( 1. - P1 )
C
       RETURN
```

```
C
C----- ENTRY TO ALLOW CHANGING OF PARAMETERS
C
      ENTRY TH3PAR(AX,BX,CX)
      A=AX
      B=BX
      C=CX
      RETURN
C
C------ ENTRY TO DEFINE ALPHA USED BY EXP PRIOR
C
      ENTRY SETEP(ALPHA)
      AL=ALPHA
      RETURN
C
C------ ENTRY TO DEFINE WHICH PRIOR TO CHOOSE
C
      ENTRY SETPRI(JFPP)
      JFP=JFPP
      RETURN
C
      END
```

12. A88: BAYESIAN DOT ALLOCATION, MAJORITY RULE LABELING

12.1 DESCRIPTION

This dot allocation and labeling scheme is different from most other schemes in one important respect: the allocation procedure is fused with the labeling procedure.

For each cluster, dots are allocated continuously until the cluster is labeled. The rule is Bayesian majority rule, stated as follows:

a. Assign 2 dots; if their labels are identical, then give the cluster that label and stop.

b. Assign 3 more dots; if the 5 labels yield a 1-4 split, then label the cluster by the majority and stop.

c. Assign 2 more dots; if the 7 labels yield a 2-5 split, then label the cluster by the majority and stop.

d. Assign 3 more dots; if the 10 labels yield a 3-7 split, then label the cluster by the majority and stop.

e. Assign 3 more dots and label the cluster by the majority.

It can be seen from the above that the number of dots assigned to the cluster may vary, depending on where the allocation and labeling is stopped. Also, it should be noted that the labels above are either small grain or other.

The equation for the proportion estimate is

$$\hat{P}_{sg} = \sum_{\substack{\text{cluster } i \\ \text{labeled as} \\ \text{small grain}}} \frac{N_i}{N} \tag{12-1}$$

where N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom requences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. A grand summary is printed.

12.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAEBMR, GETDOT, CLMPCS, GTMPLB, LBLITP
SAEBMR	CLMPCS, BMR1
GETDOT	RAN
BMR1 (Bayesian majority rule 1)	BMRSUP
BMRSUP (Bayesian majority rule support)	GETDOT, CLMPCS, GTMPLB, LBLITP

12.3 INTERFACE

Interface with other routines is through the common block PRTFLG (2 bytes), which is used to control the optional printing of dot files and other information.

12.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line	Parameters		Format	Default input	Default value
1	Cluster map file name		A13	None	
2	a.	Number of repetitions	I3	0 or blank	1
	b.	Starting point of first pseudorandom sequence	I5	0 or blank	10
	c.	Number of repetition printings	I3	0 or blank	5
3	Number of status messages on terminal		I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

12.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report	Contents
Individual repetition	1. Processor header
	2. Ground truth small-grain proportion
	3. A dot file showing dots chosen, their x-y position, and their ground truth labels
	4. A table showing number of clusters, cluster name codes, cluster sizes, number of dots assigned, and the split

Grand summary	1.	Processor header
	2.	A table showing the estimate and bias of each repetition
	3.	Bias, MSE, reduction in MSE, average estimate, variance, variance reduction, average and standard deviation of the total number of dots per repetition

12.6 BRIEF ALGORITHM: FOR BAYESIAN DOT ALLOCATION, MAJORITY RULE LABELING

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.
3. Repeat a and b until all repetitions are finished:
   a. Set starting point of the pseudorandom number generator.
   b. For each cluster, allocate and label dots by Bayesian majority rule.
4. Compute bias, MSE, reduction in MSE, average estimate, variance, variance reduction, average and standard deviation of the total number of dots per repetition, and print a grand summary.

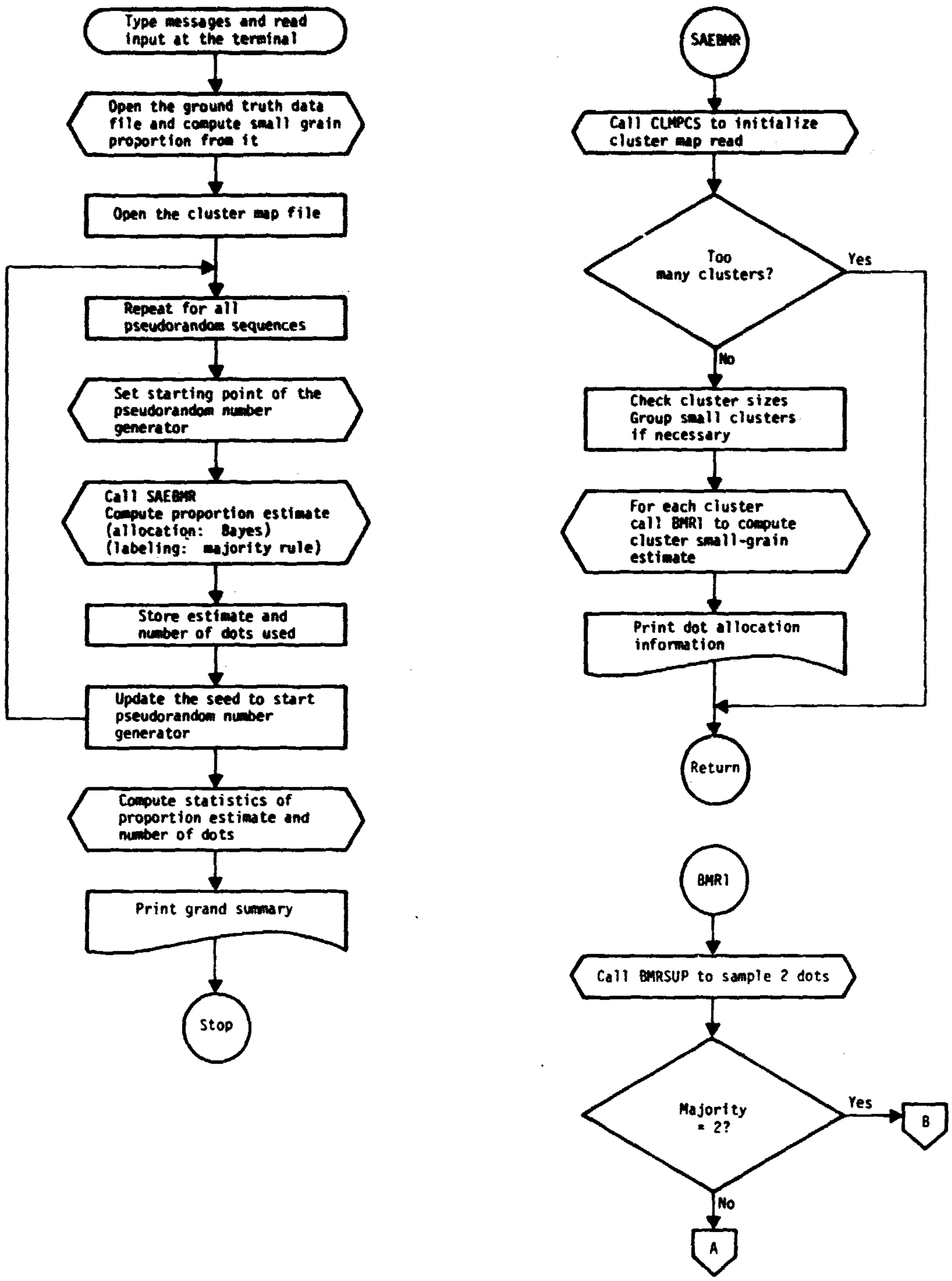

Figure 12-1.– Flow chart for Bayesian majority rule labeling.

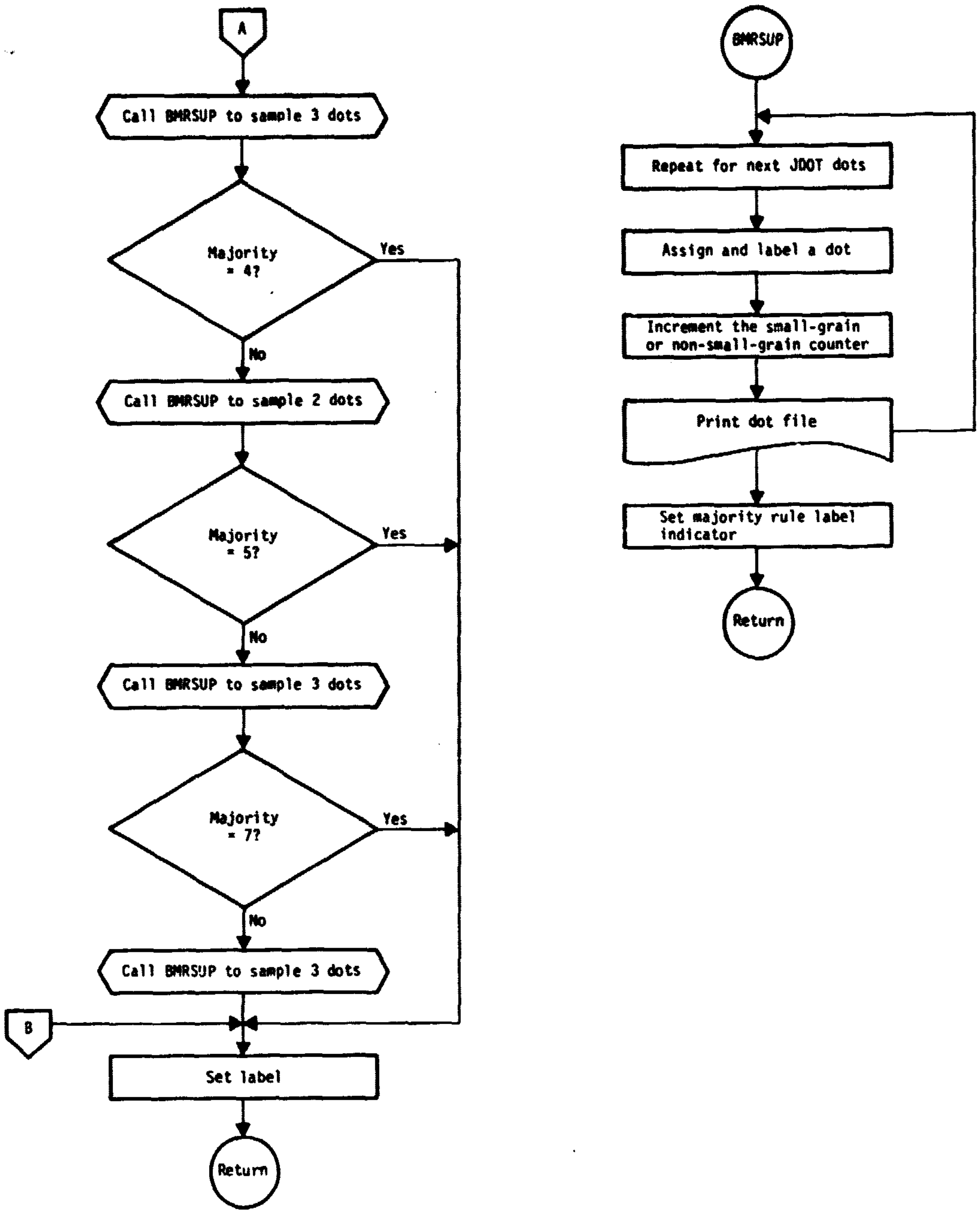

Figure 12-1.– Concluded.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

12.7 LISTING

```
C----- PROGRAM A15 (A88.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING BAYES MAJ RULE (UNIFORM PRIOR)  DOT ALLOCATION
C------ LABELLING IS BAYES MAJORITY RULE.
C
      BYTE NAME(15),NGT(13)
      REAL PSG(350)
      INTEGER NDOT(350)
C------ COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C----- READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A15 (A88.TSK).'
     *        /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *        /' DOT ALLOCATION IS BAYES MAJ RULE (UNIFORM PRIOR) .'
     *         /' LABELLING IS BAYES MAJORITY RULE.'
     *        /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *        /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *        /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *        /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT(' --- NOTICE THAT NO. OF DOTS ASSGINED IN EACH',
     *    ' REPETITION MAY VARY ---')
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL, HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *        /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *        /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6)
C
C------- COMPUTE PROPORTIONAL ESTIMATE FOR GROUND TRUTH
      NLINE=117
      NCOL=196
      DO 215 K=1,4
215   NGT(K)=NAME(K)
      OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *     ACCESS='DIRECT')
      DO 231 L=1,NLINE
      DO 221 K=1,NCOL
      CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
221   CONTINUE
231   CONTINUE
      P=FLOAT(IP)/NLINE/NCOL
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *    READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
```

```
C
C----- START ESTIMATION FOR EACH TOTDOT AND REPETITIONS
      JMES=0
      JS=JSEED
      DO 361 J=1,JV
      JFLAG=0
      IF(MOD(J-1,JSKIP).EQ.0) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(5-17-79)',/,2X,
     * 'PROGRAM A15 (A88.TSK): ESTIMATION OF SMALL GRAINS PROPORTION'
     *   /,15X,'DOT ALLOCATION:BAYES MAJ RULE (UNIFORM PRIOR) ,'
     *        /,15X,'DOT LABELLING:BAYES MAJORITY RULE.'
     *       //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *        /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' --- REPETITION RUN=',I3,
     *          ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAEBMR(NDOT(J),PSG(J))
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)MAXDOT,J,JSEED
323   FORMAT(' MAXDOT=',I3,' REPETITION RUN=',I3,' SEED=',I6)
      JSEED=JSEED+150
361   CONTINUE
C
C----- PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)JV,JS
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *       /,14X,'                NO. OF REPETITION RUNS =',I3,
     *       /,14X,' RANDOM DOTS START WITH SEED=',I6,/
     *   ,10X,'REPETITION  PSEDUO SEQ   DOT   SMALL GRAIN   BIAS',/,
     *     10X,'    RUN          SEED     ASSIGNED  ESTIMATE   WRT G.T.')
C----- COMPUTE BIAS AND M.S.E.
      PB=0.
      PM=0.
      AM=0.
      VM=0.
      DO 381 J=1,JV
      TEMP=PSG(J)-P
      PB=PB+TEMP
      PM=PM+TEMP**2
      AM=AM+NDOT(J)
      VM=VM+NDOT(J)**2
      WRITE(6,375)J,JS,NDOT(J),PSG(J),TEMP
375   FORMAT(13X,I3,6X,I6, 6X,I4, 5X,F8.5,3X,F9.6)
381   JS=JS+150
      PB=PB/JV
      PM=PM/JV
      AM=AM/JV
      VM=VM/JV
      IF(JV.EQ.1) VM=VM - AM**2
      IF(JV.GT.1) VM= (VM-AM**2)*JV/(JV-1)
      VM=SQRT(VM)
      AVERG=PB+P
      IF(JV.EQ.1) VAR= PM-PB**2
      IF(JV.GT.1) VAR= (PM-PB**2)*JV/(JV-1)
C------ COMPUTE VARIANCE REDUCTION
      RR=PM/( P*(1.-P)/AM )
      REDVAR=VAR/( P*(1.-P)/AM )
      WRITE(6,385)PB,PM,RR,AVERG,VAR,REDVAR, AM,VM
385   FORMAT(/'     BIAS=',F10.6,'    M.S.E.=',F10.6,' REDUCTION=',F10.6,
     *        /' AVERAGE=',F10.6,' VARIANCE=',F10.6,' REDUCTION=',F10.6,
     *         /' AVE DOT=',F10.3,' DOT S.D.=',F10.3)
C
```

```
C
      WRITE(6,401)
401   FORMAT('1 ----- END OF THIS JOB ---')
C
      STOP
      END
C
C
      SUBROUTINE SAEBMR(NDOT,PSG)
C------ STRATIFIED AREAL ESTIMATION USING BAYES MAJORITY RULE  DOT ALLOCATION.
C------ FORMULA HERE ARE FOR BAYES MAJORITY RULE WITH UNIFORM PRIOR.
C------ ON RETURN, THE PROPORTIONAL ESTIMATE IS IN PSG, AND THE
C------ TOTAL NO. OF DOTS ASSIGNED IS IN NDOT.
C
C------ WRITEN AND EDITED BY N.Y. CHU ON 5-17-79.
C
C
      INTEGER NDARY(350),MM(51),NN(51),LL(51)
      INTEGER NNS(2,51)
      INTEGER*4 NSG
C------ COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/, MINSZ/13/
C
C------ GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C------ MAKE SURE EACH CLUSTER HAS A MINIMUM NO. OF PIXELS
      K1=0
      DO 151 K=1,M
      NN(K)=0
      IF(MM(K).LT.MINSZ) K1=K1+MM(K)
      IF(MM(K).LT.MINSZ) NN(K)=-1
151   CONTINUE
      MX=M
      IF(K1.LT.MINSZ) GOTO 161
      MX=M+1
      MM(MX)=K1
161   CONTINUE
C
C
C------ BEGIN BAYES MAJORITY RULE  DOT ALLOCATION
C
C
      IF(JFLAG.EQ.1) WRITE(6,211)
211   FORMAT(//'                         --- DOT FILE ---'
     *       /' CLUSTER    DOT NUMBER    POSITION    GROUND TRUTH LABEL'
     *    /'    NO.    WRT ITS CLUSTER LINE,COL  RAW,CODE SMALL GRAIN')
C
C---- ASSIGN DOTS AND COMPUTE ESTIMATE
      NDOT=0
      PSG=0.
      ND=1
      DO 231 K=1,MX
      IF(NN(K).LE.-1) GOTO 231
      CALL BMR1(ISG,NNS(1,K), NDARY(ND),K,MM,NN,LL,M)
      PSG=PSG+FLOAT(ISG)*MM(K)/NPIXEL
      NDOT=NDOT+NN(K)
      ND=ND+NN(K)
231   CONTINUE
C
```

```
C
C----- END OF BAYES MAJORITY RULE DOT ALLOCATION
C
C----- WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)NDOT,NPIXEL,MX
282   FORMAT(//,5X,'              TOTAL NO. OF DOTS  ASSIGNED =',I6,
     *       /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *       /,5X,'                  TOTAL NO. OF CLUSTERS =',I6,
     *       /,10X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS'
     *       /,10X,'   NO.     CODE  PIXELS   ASSIGNED   SPLIT')
      DO 283 K=1,MX
      IF(NN(K).LE.-1) NN(K)=0
      IF(NNS(1,K).GE.NNS(2,K)) NSG='S.G.'
      IF(NNS(1,K).LT.NNS(2,K)) NSG='    '
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K),NNS(1,K),NNS(2,K)
     *            ,NSG
283   CONTINUE
284   FORMAT(13X,I2,5X,I4,3X,I6,4X,I3, 7X,I2,'-',I2, 1X,A4)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
285   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286   FORMAT(/'L THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *       ') ARE:', 3(/,16(I3,',')) )
288   CONTINUE
C
      IF(JFLAG.EQ.1) WRITE(6,351)NDOT,PSG
351   FORMAT(' TOTAL DOTS ASSIGNED=',I4,' PROPORTIONAL',
     *       ' ESTIMATE=',F10.6)
C
      GOTO 990
C
901   WRITE(6,902)M
902   FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
C
C
990   RETURN
      END
C
C
C
      SUBROUTINE DMR1(ISG,NS, NDARY,K1,MM,NN,LL,M)
C----- BAYES MAJORITY RULE (UNIFORM PRIOR)
C----- TOGETHER WITH SUB. BMRSUP, THIS ROUTINE ASSIGN
C----- DOTS ACCORDING A BAYES MAJORITY RULE.
C----- ON RETURN, NN(K1) CONTAINS THE NO. OF DOTS ASSIGNED DURING THIS CALL
C----- ISG=1, ON RETURN, MEANS SMALL GRAIN LABEL IS ASSIGNED TO
C----- THE CURRENT (K1) CLUSTER, IF ISG=0, NOT SMALL GRAIN
C----- NDARY= DOT ARRAY FOR THE CURRENT CLUSTER
C----- MM= ARRAY CONTAINING CLUSTER SIZES
C----- NN= ARRAY CONTAINING CLUSTERS' FUTURE DOT OF ASSIGNED DOTS
C----- LL= ARRAY CONTAINING CLUSTERS' CLASS NAME,
C-----      JUST FOR USE IN SUB. CLMPXY
C
      INTEGER NDARY(1),MM(1),NN(1),LL(1)
      INTEGER NS(2)
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
C
      NS(1)=0
      NS(2)=0
```

```
C
C----- BAYES MAJ RULE WITH UNIFORM PRIOR
       CALL BMRSUP(2,NS,MR, NDARY,K1,MM,NN,LL,M)
       IF(NS(MR).EQ.2) GOTO 401
       CALL BMRSUP(3,NS,MR, NDARY,K1,MM,NN,LL,M)
       IF(NS(MR).EQ.4) GOTO 401
       CALL BMRSUP(2,NS,MR, NDARY,K1,MM,NN,LL,M)
       IF(NS(MR).EQ.5) GOTO 401
       CALL BMRSUP(3,NS,MR, NDARY,K1,MM,NN,LL,M)
       IF(NS(MR).EQ.7) GOTO 401
       CALL BMRSUP(3,NS,MR, NDARY,K1,MM,NN,LL,M)
C
401    IF(MR.EQ.1) ISG=1
       IF(MR.EQ.2) ISG=0
       RETURN
       END
C
       SUBROUTINE BMRSUP(JDOT,NS,MR, NDARY,KOLD,MM,NN,LL,M)
C----- TOGETHER WITH SUBROUTINE BMR1, THIS ROUTINE
C----- ASSIGN DOTS TO A CURRENT CLUSTER (SPECIFIED IN K1), AND
C----- DETERMINE THE SPLIT BETWEEN SMALL GRAIN AND NON SMALL GRAIN
C----- LABELS.
C----- JDOT= NO. OF DOTS DESIRED TO BE ASSIGNED IN THIS CALL
C----- NS= (DIM 2) ARRAY CONTAINING THE SPLIT
C-----      FIRST MEMBER REFERS TO SMALL GRAIN, SECOND MEMEBER NON S.G.
C           NOTICE FOR NS, NEW DOTS ARE ACCUMULATED TO PREVIOUS VALUES.
C----- MR= 1 OR 2, IF =1, MEANS S.G. IS THE MAJORITY, OTHERWISE NON S.G.
C----- NDARY= ARRAY NECESSARY FOR DOT GENERATION
C
       INTEGER NS(2),NDARY(1),MM(1),NN(1),LL(1)
       BYTE LABEL
       INTEGER*4 NSG
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
       COMMON /PRTFLG/JFLAG
C
C------- ASSIGN JDOT DOTS
       DO 391 J=1,JDOT
       K1=KOLD
       NN(K1)=NN(K1)+1
       CALL GETDOT(MM(K1),NN(K1),NDARY,IDOT)
C      WRITE(6,99141)K1,NN(K1),(NDARY(K),K=1,15)
99141  FORMAT(' K1,NN=',2I4, 4(/,2X,15I4) )
       IF(K1.LE.M) GOTO 227
       DO 225 K1=1,M
       IF(NN(K1).GE.0) GOTO 225
       IF(IDOT.LT.MM(K1)) GOTO 227
       IDOT=IDOT-MM(K1)
225    CONTINUE
227    CALL CLMPXY(K1,IDOT,NL,NC,LL)
       CALL GTMPLB(NL,NC,LABEL)
       ISG=0
       CALL LBLITP(LABEL,LB1,ISG)
       IF(ISG.EQ.1) NSG='YES'
       IF(ISG.EQ.0) NSG=' NO'
       IF(ISG.EQ.1) NS(1)=NS(1)+1
       IF(ISG.EQ.0) NS(2)=NS(2)+1
       IF(JFLAG.EQ.1) WRITE(6,371)K1,IDOT,NL,NC,LABEL,LB1,NSG
371    FORMAT(3X,I2,8X,I6,6X,2I4, 1X,I4,I5, 4X,A4)
391    CONTINUE
C
       IF(NS(1).GE.NS(2)) MR=1
       IF(NS(1).LT.NS(2)) MR=2
C      WRITE(6,99301)JDOT,NS,MR
99301  FORMAT(' JDOT=',I3,' NS=',2I4,' MR=',I3)
C
901    RETURN
       END
```

13. A90: NON-BAYESIAN SEQUENTIAL DOT ALLOCATION, MAJORITY RULE LABELING

13.1 DESCRIPTION

This dot allocation and labeling scheme is different from most other schemes in one important respect: the allocation procedure is fused with the labeling procedure.

For each cluster, dots are allocated continuously until the cluster is labeled or until the number of dots assigned exceeds a preselected value (currently, it is 35). The rule is a non-Bayesian sequential rule, stated as follows:

a. Allocate INIDOT dots to the cluster. INIDOT was specified by the user during the program INPUT session.

b. Allocate one dot, and compute the test interval (a, b):

$$\left.\begin{aligned} a &= \frac{x_i}{n_i} - 1.534\,\hat{\sigma}_i \\ b &= \frac{x_i}{n_i} + 1.534\,\hat{\sigma}_i \\ \hat{\sigma}_i &= \sqrt{\frac{x_i(n_i - x_i)}{n_i^2(n_i - 1)}} \end{aligned}\right\} \qquad (13\text{-}1)$$

where n_i = number of dots allocated to cluster i

x_i = number of dots labeled as small grain for cluster i

(notice $x_i \leq n_i$)

c. If $0.5 \in (a, b)$ and n_i does not exceed a preselected value (35), repeat step b.

d. Label the cluster by majority rule.

It can be seen from the above that the number of dots assigned to the cluster may vary, depending on where the allocation and labeling is stopped. Also, it should be noted that the labels above are either small grain or other.

The equation for the proportion estimate is

$$\hat{P}_{sg} = \sum_{\substack{\text{cluster } i \\ \text{labeled as} \\ \text{small grain}}} \frac{N_i}{N} \qquad (13\text{-}2)$$

where N_i = number of pixels in cluster i

N = total number of pixels in the entire scene

This scheme, like other allocation and labeling schemes, may be repeated with different pseudorandom sequences. The main program resets the random number generator to a different starting point for each repetition.

All computation of statistics such as mean and variance is performed in the main program. A grand summary is printed.

13.2 LINKAGES

The PDP 11/45 Fortran library (for functions and subroutines such as SQRT), the PDP 11/45 system library (for routines such as the pseudorandom number generator), and the utility package (described in section 14) are required.

Main program or subprogram	Subroutine or function required
MAIN	SAENBS, GETDOT, CLMPCS, GTMPLB, LBLITP
SAENBS	CLMPCS, NBSMR
GETDOT	RAN

NBSMR (non-Bayesian sequential majority rule)	GETDOT, CLMPCS, GTMPLB, LBLITP

13.3 INTERFACE

Interface with other routines is through the common block PRTGLG (2 bytes), which is used to control the optional printing of dot files and other information.

13.4 INPUTS

Fortran-formatted input of the following parameters is needed:

Card or line	Parameters		Format	Default input	Default value
1	Cluster map file name		A13	None	
2	a.	Number of repetitions	I3	0 or blank	1
	b.	Starting point of first pseudorandom sequence	I5	0 or blank	10
	c.	Number of repetition printings	I3	0 or blank	5
3	Number of initial dots		I3	None	
4	Number of status messages on terminal		I3	0 or blank	NO MSG

Two disk files are needed, one containing the stripped cluster map, the other the stripped ground truth map. These files should have been created by conversion programs A81 and A86.

13.5 OUTPUTS

In addition to the normal echo of input specifications and status messages on the terminal, the following will be output on a line printer.

Report	Contents
Individual repetition	1. Processor header
	2. Ground truth small-grain proportion
	3. A dot file showing dots chosen, their x-y position, their ground truth labels, and the test interval
	4. A table showing number of clusters, cluster name codes, cluster sizes, number of dots assigned, and the split
Grand summary	1. Processor header
	2. A table showing the estimate and bias of each repetition
	3. Bias, MSE, reduction in MSE, average estimate, variance, variance reduction, average and standard deviation of the total number of dots per repetition

13.6 BRIEF ALGORITHM: FOR NON-BAYESIAN SEQUENTIAL DOT ALLOCATION, MAJORITY RULE LABELING

1. Type messages at terminal and read from it run specifications.
2. Compute ground truth small-grain proportion.
3. Repeat a and b until all repetitions are finished:
    a. Set starting point of the pseudorandom number generator.
    b. For each cluster, allocate and label dots by non-Bayesian sequential majority rule.

4. Compute bias, MSE, reduction in MSE, average estimate, variance, variance reduction, average and standard deviation of the total number of dots per repetition, and print a grand summary.

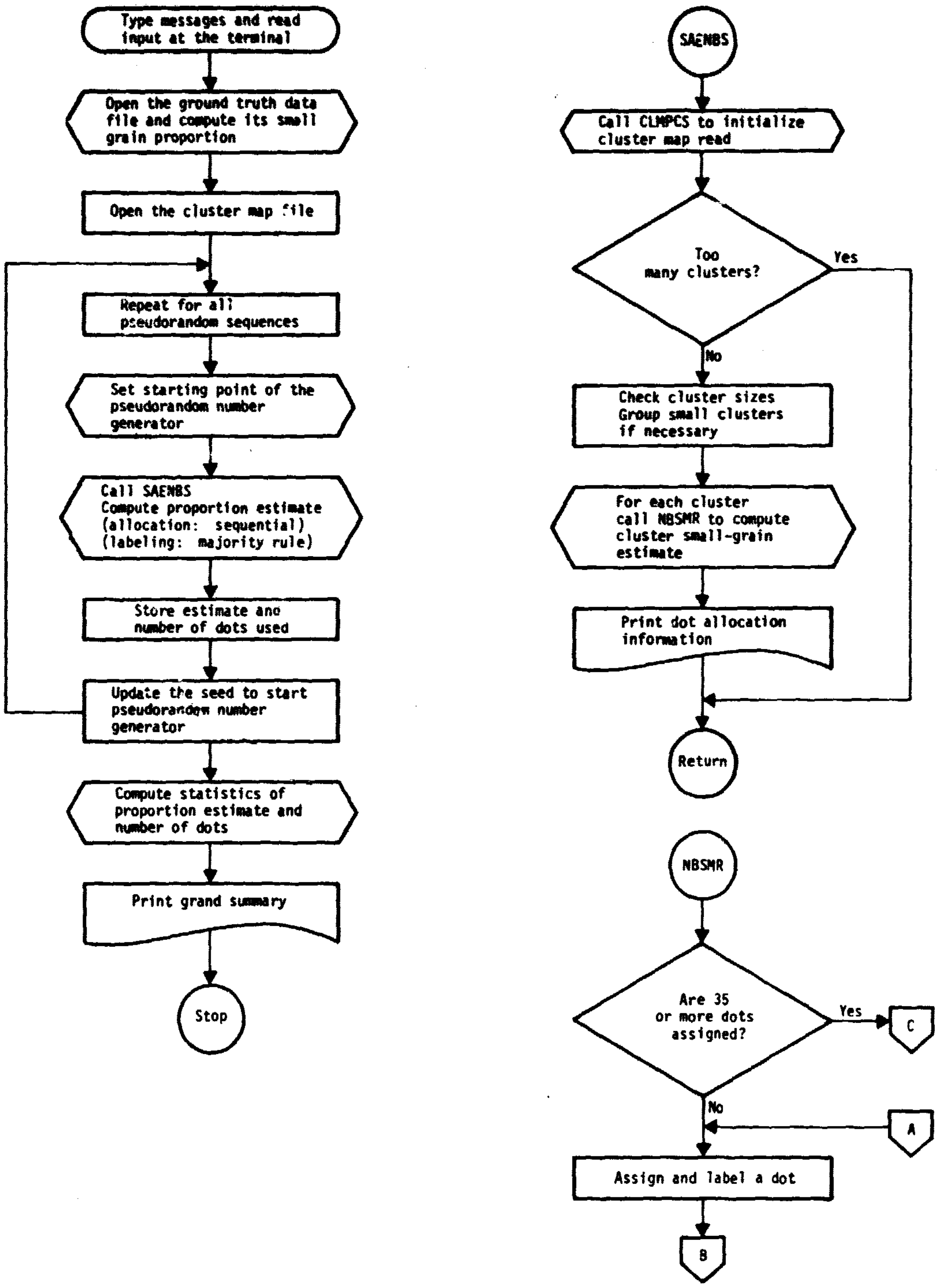

Figure 13-1.– Flow chart for non-Bayesian sequential majority rule labeling.

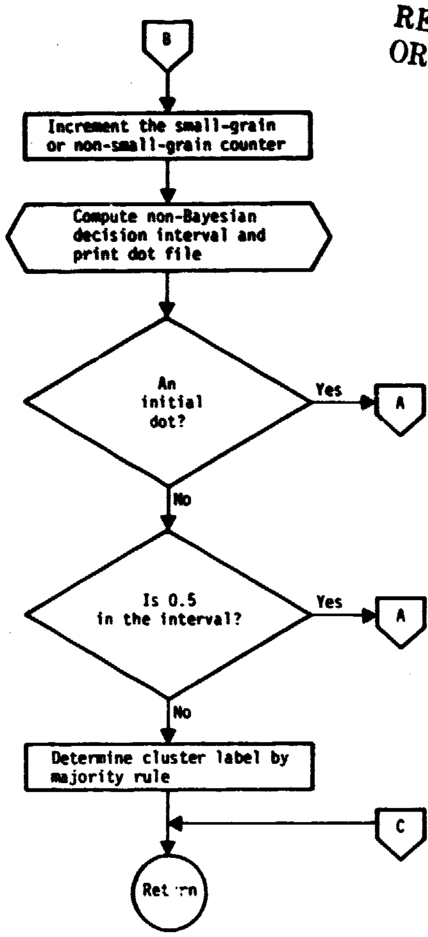

Figure 13-1.— Concluded.

13.7 LISTING

```
C----- PROGRAM A17 (A90.TSK): PROPORTION ESTIMATE OF SMALL GRAIN
C----- USING NON-BAYESIAN SEQUENTIAL  DOT ALLOCATION
C----- LABELLING IS NON-BAYESIAN SEQ MAJORITY RULE.
C
      BYTE NAME(15),NGT(13)
      REAL PSG(435)
      INTEGER NDOT(435)
C------ COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NGT(5)/'.'/,NGT(6)/'S'/,NGT(7)/'T'/,NGT(8)/'P'/
C
C
C------ READING INPUT FROM TERMINAL
C
      WRITE(8,121)
121   FORMAT( ' PROGRAM: A17 (A90.TSK).'
     *        /' PROPORTIONAL ESTIMATION OF SMALL GRAIN'
     *        /' DOT ALLOCATION IS NON-BAYESIAN SEQUENTIAL .'
     *         /' LABELLING IS NON-BAYESIAN SEQ MAJORITY RULE.'
     *        /' INPUT CLUSTER MAP FILENAME'/' AAAAAAAAAAAAA')
      READ(7,122)(NAME(K),K=1,13)
122   FORMAT(13A1)
C
      WRITE(8,131)
131   FORMAT( ' AN ASSIGNMENT OF PSEUDORANDOM DOTS IS A REPETITION RUN'
     *        /' HOW MANY REPETITION RUNS? SEED FOR FIRST PSEUDO SEQ?'
     *        /' HOW MANY PRINTINGS OF INDIVIDUAL RUNS?'
     *        /' III IIIII III')
      READ(7,132)JV,JSEED,JPAGE
132   FORMAT(I3, 1X,I5, 1X,I3)
      IF(JV.LE.0) JV=1
      IF(JSEED.LE.0) JSEED=10
      IF(JPAGE.LE.0) JPAGE=5
      JSKIP=(JV-1)/JPAGE+1
C
      WRITE(8,141)
141   FORMAT(' --- NOTICE THAT NO. OF DOTS ASSGINED IN EACH',
     *     ' REPETITION MAY VARY ---'/
     *     ' SPECIFY NO. OF DOTS INITIALLY ASSIGNED TO EACH CLUSTER.'/
     *     ' III')
      READ(7,142)NNIX
142   FORMAT(I3)
      IF(NNIX.GE.1) CALL INIT(NNIX)
C
      WRITE(8,151)
151   FORMAT(' STATUS MESSAGES ON TERMINAL, HOW MANY?'/' III')
      READ(7,152)NMES
152   FORMAT(I3)
C
      WRITE(8,181)(NAME(K),K=1,13),JV,JSEED
181   FORMAT(//' CLUSTER FILE GIVEN = ',13A1,
     *         /' NO. OF REPETITION RUNS FOR EACH TOTDOT =',I3,
     *         /' THE FIRST REPETITION RUN STARTS WITH SEED = ',I6)
C
C------ COMPUTE PROPORTIONAL ESTIMATE FOR GROUND TRUTH
      NLINE=117
      NCOL=196
      DO 215 K=1,4
215   NGT(K)=NAME(K)
      OPEN(UNIT=2,NAME=NGT,TYPE='OLD',READONLY,FORM='UNFORMATTED',
     *     ACCESS='DIRECT')
      DO 231 L=1,NLINE
      DO 221 K=1,NCOL
      CALL GTMPLB(L,K,LABEL)
      CALL LBLITP(LABEL,LB1,IP)
221   CONTINUE
231   CONTINUE
      P=FLOAT(IP)/NLINE/NCOL
```

```
C
      OPEN(UNIT=1,NAME=NAME,TYPE='OLD',FORM='UNFORMATTED',
     *   READONLY,ACCESS='DIRECT')
      CALL CLMPLC(NLINE,NCOL)
C
C----- START ESTIMATION FOR EACH TOTDOT AND REPETITIONS
      JMES=0
      JS=JSEED
      DO 361 J=1,JV
      JFLAG=0
      IF(MOD(J-1,JSKIP).EQ.0) JFLAG=1
      IF(JFLAG.EQ.1) WRITE(6,321)NAME,P
321   FORMAT('1',10X,'ACCURACY ACCESSEMENT SOFTWARE(5-29-79)',/,2X,
     * 'PROGRAM A17 (A90.TSK): ESTIMATION OF SMALL GRAINS PROPORTION'
     *   /,15X,'DOT ALLOCATION:NON-BAYESIAN SEQUENTIAL  ,'
     *        /,15X,'DOT LABELLING:NON-BAYESIAN SEQ MAJORITY RULE.'
     *      //,10X,' INPUT CLUSTER MAP IS FILE ',15A1,
     *        /,10X,' PROPORTIONAL ESTIMATE FOR GROUND TRUTH IS',F8.5)
      IF(JFLAG.EQ.1) WRITE(6,322)J,JSEED
322   FORMAT(/,2X,' --- REPETITION RUN=',I3,
     *       ' RANDOM DOT SEQUENCE STARTS WITH',I8,' ---')
      CALL RANST(JSEED)
      CALL SAENBS(NDOT(J),PSG(J))
      JMES=JMES+1
      IF(JMES.LE.NMES) WRITE(8,323)J,JSEED,NDOT(J),PSG(J)
323   FORMAT(' REPETITION RUN=',I3,' SEED=',I6,' NDOT=',I3,
     *       ' ESTIMATE=',F9.5)
      JSEED=JSEED+150
361   CONTINUE
C
C----- PRINT GRAND SUMMARY FOR THIS JOB
      WRITE(6,321)NAME,P
      WRITE(6,371)JV,JS
371   FORMAT(/,14X,' --- GRAND SUMMARY OF THIS JOB ---',
     *       /,14X,'             NO. OF REPETITION RUNS =',I3,
     *       /,14X,' RANDOM DOTS START WITH SEED=',I6,/
     *   ,10X,'REPETITION  PSEDUO SEQ   DOT   SMALL GRAIN   BIAS',/,
     *    10X,'    RUN          SEED     ASSIGNED  ESTIMATE   WRT G.T.')
C----- COMPUTE BIAS AND M.S.E.
      PB=0.
      PM=0.
      AM=0.
      VM=0.
      DO 381 J=1,JV
      TEMP=PSG(J)-P
      PB=PB+TEMP
      PM=PM+TEMP**2
      AM=AM+NDOT(J)
      VM=VM+FLOAT(NDOT(J))**2
      WRITE(6,375)J,JS,NDOT(J),PSG(J),TEMP
375   FORMAT(13X,I3,6X,I6,5X,I4,  5X,F8.5,3X,F9.6)
381   JS=JS+150
      PB=PB/JV
      PM=PM/JV
      AM=AM/JV
      VM=VM/JV
      IF(JV.EQ.1) VM=VM-AM**2
      IF(JV.GT.1) VM= (VM-AM**2)*JV/(JV-1)
      VM=SQRT(VM)
      AVERG=PB+P
      IF(JV.EQ.1) VAR= PM-PB**2
      IF(JV.GT.1) VAR= (PM-PB**2)*JV/(JV-1)
C------ COMPUTE VARIANCE REDUCTION
      RR=PM/( P*(1.-P)/AM )
      REDVAR=VAR/( P*(1.-P)/AM )
      WRITE(6,385)PB,PM,RR,AVERG,VAR,REDVAR, AM,VM
385   FORMAT(/'      BIAS=',F10.6,'    M.S.E.=',F10.6,' REDUCTION=',F10.6,
     *        /' AVERAGE=',F10.6,' VARIANCE=',F10.6,' REDUCTION=',F10.6,
     *        /' AVE DOT=',F10.3,' DOT S.D.=',F10.3)
```

```
C
C
      WRITE(6,401)
401   FORMAT('1 ---- END OF THIS JOB ---')
C
      STOP
      END
C
C
      SUBROUTINE SAENBS(NDOT,PSG)
C----- STRATIFIED AREAL ESTIMATION USING NON-BAYESIAN
C----- SEQUENTIAL DOT ALLOCATION WITH MAJORITY RULE LABELLING.
C----- ON RETURN, THE PROPORTIONAL ESTIMATE IS IN PSG, AND THE
C----- TOTAL NO. OF DOTS ASSIGNED IS IN NDOT.
C
C----- WRITEN AND EDITED BY N.Y. CHU ON 5-29-79.
C
C
      INTEGER NDARY(435),MM(51),NN(51),NX(51),LL(51)
      INTEGER NNS(2,51)
      INTEGER*4 NSG
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA NSTART/0/, MINSZ/35/
C
C----- GET PIXEL COUNT FOR EACH CLUSTER
      IF(NSTART.NE.0) GOTO 121
      NSTART=1
      CALL CLMPCS(NPIXEL,M,MM,LL)
      IF(M.GE.51) GOTO 901
121   CONTINUE
C
C----- MAKE SURE EACH CLUSTER HAS A MINIMUM NO. OF PIXELS
      K1=0
      DO 151 K=1,M
      NN(K)=0
      NX(K)=0
      IF(MM(K).LT.MINSZ) K1=K1+MM(K)
      IF(MM(K).LT.MINSZ) NN(K)=-1
151   CONTINUE
      MX=M
      IF(K1.LT.MINSZ) GOTO 161
      MX=M+1
      MM(MX)=K1
161   CONTINUE
C
C
C------ BEGIN NON-BAYESIAN SEQ MAJORITY RULE  DOT ALLOCATION
C
C
      IF(JFLAG.EQ.1) WRITE(6,211)
211   FORMAT(//'                          --- DOT FILE ---'
     *         /' CLUSTER    DOT NUMBER   POSITION GROUND TRUTH   '
     *          ,'  TEST INTERVAL'
     *      /'   NO.    WRT ITS CLUSTER LINE,COL   RAW,CODE'
     *       '          LOWER     UPPER')
C
C---- ASSIGN DOTS AND COMPUTE ESTIMATE
      NDOT=0
      PSG=0.
      ND=1
      DO 231 K=1,MX
      IF(NN(K).LE.-1) GOTO 231
      IF(NDOT.GT.(-35+435)) GOTO 251
      CALL NBSMR(ISG,NNS(1,K), NDARY(ND),K,MM,NN,LL,NX,M)
      PSG=PSG+FLOAT(ISG)*MM(K)/NPIXEL
      NDOT=NDOT+NN(K)
      ND=ND+NN(K)
231   CONTINUE
      GOTO 255
```

```
C
C----- TERMINATE DOT ASSIGNING IF ARRAY OVERFLOW IS ANTICIPATED
251   IF(JFLAG.EQ.1) WRITE(6,253)NDOT
253   FORMAT(' DOTS ASSIGNED:',I4,' MAY CAUSE ARRAY OVERFLOW,',
     *       ' ALLOCATION TERMINATED. ')
255   CONTINUE
C
C----- END OF NON-BAYESIAN SEQ MAJORITY RULE DOT ALLOCATION
C
C----- WRITE CLUSTER INFORMATION
      IF(JFLAG.EQ.1) WRITE(6,282)NDOT,NPIXEL,MX
282   FORMAT(//,5X,'               TOTAL NO. OF DOTS  ASSIGNED =',I6,
     *        /,5X,' TOTAL NO. OF PIXEL IN THE CLUSTER MAP =',I6,
     *        /,5X,'                  TOTAL NO. OF CLUSTERS =',I6,
     *        /,10X,' CLUSTER CLUSTER NO. OF  NO. OF DOTS'
     *        /,10X,'   NO.     CODE  PIXELS  ASSIGNED  SPLIT')
      DO 283 K=1,MX
      IF(NN(K).LE.-1) NN(K)=0
      IF(NNS(1,K).GE.NNS(2,K)) NSG='S.G.'
      IF(NNS(1,K).LT.NNS(2,K)) NSG='    '
      IF(JFLAG.EQ.1) WRITE(6,284)K,LL(K),MM(K),NN(K),NNS(1,K),NNS(2,K)
     *              ,NSG
283   CONTINUE
284   FORMAT(13X,I2,5X,I4,3X,I6,4X,I3, 7X,I2,'-',I2, 1X,A4)
      IF(MX.EQ.M) GOTO 288
      KX=0
      DO 285 K=1,M
      IF(NN(K).NE.0) GOTO 285
      KX=KX+1
      NDARY(KX)=K
285   CONTINUE
      IF(JFLAG.EQ.1) WRITE(6,286)MX,(NDARY(K),K=1,KX)
286   FORMAT(/'L THOSE SMALLER CLUSTERS GROUPED (AS CLUSTER',I3,
     *       ') ARE:', 3(/,16(I3,',')) )
288   CONTINUE
C
      IF(JFLAG.EQ.1) WRITE(6,351)NDOT,PSG
351   FORMAT(' TOTAL DOTS ASSIGNED=',I4,' PROPORTIONAL',
     *       ' ESTIMATE=',F10.6)
C
      GOTO 990
C
901   WRITE(6,902)M
902   FORMAT(' YOU HAVE',I3,' TOO MANY CLUSTERS(MAX=50)')
      GOTO 990
C
C
990   RETURN
      END
C
C
C
      SUBROUTINE NBSMR(ISG,NS, NDARY,K1,MM,NN,LL,NX,M)
C------ NON-BAYESIAN SEQ MAJORITY RULE
C------ THIS ROUTINE ASSIGN A DOT OR MORE TO
C------ CLUSTER K1 ACCORDING TO A NON-BAYESIAN SEQ MAJORITY RULE.
C------ ON RETURN, NN(K1) CONTAINS THE NO. OF DOTS ASSIGNED
C------- DURING THIS CALL.
C------ ISG=1, ON RETURN, MEANS SMALL GRAIN LABEL IS ASSIGNED TO
C------ THE CURRENT (K1) CLUSTER, IF ISG=0, NOT SMALL GRAIN
C------ NDARY= DOT ARRAY FOR THE CURRENT CLUSTER
C------ MM= ARRAY CONTAINING CLUSTER SIZES
C------ NN= ARRAY CONTAINING CLUSTERS' NO. OF ASSIGNED DOTS
C------ NX= ARRAY CONTAINING CLUSTERS' NO. OF SMALL GRAIN DOTS
C------ LL= ARRAY CONTAINING CLUSTERS' CLASS NAME,
C------      JUST FOR USE IN SUB. CLMPXY
C------  M= NO. OF CLUSTER (NOT COUNTING THE GROUPED CLUSTER, IF EXISTS)
```

```
C
      INTEGER NDARY(1),MM(1),NN(1),NX(1),LL(1)
      INTEGER NS(2)
      INTEGER*4 NSG
C----- COMMON BLOCK IS A FLAG FOR PRINT ENABLE/DISABLE
      COMMON /PRTFLG/JFLAG
      DATA INIDOT/1/
C
C
C----- ASSIGN A DOT
      NS(1)=0
      NS(2)=0
      KOLD=K1
      KCOUNT=0
121   IF((KCOUNT.EQ.35).AND.(JFLAG.EQ.1)) WRITE(6,122)K1
      IF(KCOUNT.EQ.35) GOTO 901
122   FORMAT(' FOR CLUSTER',I3,' DOT ASSIGNMENT TERMINATED',
     *      '(ONLY 35 DOTS ALLOWED).')
      KCOUNT=KCOUNT+1
      NN(K1)=NN(K1)+1
      CALL GETDOT(MM(K1),NN(K1),NDARY,IDOT)
C     WRITE(6,99141)K1,NN(K1),(NDARY(K),K=1,15)
99141 FORMAT(' K1,NN=',2I4, 4(/,2X,15I4) )
      IF(K1.LE.M) GOTO 227
      DO 225 K1=1,M
      IF(NN(K1).GE.0) GOTO 225
      IF(IDOT.LT.MM(K1)) GOTO 227
      IDOT=IDOT-MM(K1)
225   CONTINUE
227   CALL CLMPXY(K1,IDOT,NL,NC,LL)
      CALL GTMPLB(NL,NC,LABEL)
      K1=KOLD
      ISG=0
      CALL LBLITP(LABEL,LB1,ISG)
      IF(ISG.EQ.1) NX(K1)=NX(K1)+1
      IF(ISG.EQ.1) NSG='S.G.'
      IF(ISG.EQ.0) NSG='    '
      IF(ISG.EQ.1) NS(1)=NS(1)+1
      IF(ISG.EQ.0) NS(2)=NS(2)+1
C
C------ NON-BAYESIAN SEQ CLUSTER LABELLING
      TX=NX(K1)
      TN=NN(K1)
      IF(NN(K1).GT.1) SHEAD=SQRT( TX*(TN-TX)/( TN*TN*(TN-1.) ) )
      TL=TX/TN-1.534*SHEAD
      TU=TX/TN+1.534*SHEAD
      IF(NN(K1).EQ.1) TL=0.
      IF(NN(K1).EQ.1) TU=0.
      IF(JFLAG.EQ.1) WRITE(6,371)K1,IDOT,NL,NC,LABEL,LB1,NSG,TL,TU
371   FORMAT(3X,I2,8X,I6,6X,2I4, 1X,I4,I5, 1X,A4, 2F9.4)
C------ CONT'NUE DOT ASSIGNMENT UNTIL ALL INITIAL DOTS ARE DONE
      IF(NN(K1).LT.INIDOT) GOTO 121
C------ CONTINUE DOT ASSIGNMENT UNTIL THE CRITERIA IS NOT SATISFIED
      IF( (0.5.GT.TL).AND.(0.5.LT.TU) ) GOTO 121
C---- NOW TERMINATE DOT ASSIGNMENT AND LABEL BY MAJORITY RULE
      IF(NS(1).GE.NS(2)) ISG=1
      IF(NS(1).LT.NS(2)) ISG=0
C
901   RETURN
C
C------ ENTRY TO SET INITIAL NO. OF DOTS ASSIGNED
      ENTRY INIT(NNIT)
      INIDOT=NNIT
      RETURN
C---- END OF THIS ENTRY
C
      END
```

14. A06: THE UTILITY PACKAGE

The utility package consists of the subroutines CLMPCS, GETDOT, GTMPLB, MR, and LBLITP. Subroutine CLMPCS and its entries CLMPXY and CLMPLC read the cluster map from its disk file. Subroutine GETDOT chooses pixels from clusters and assigns them as dots. Subroutine GTMPLB reads the ground truth map from its disk file. Subroutine MR finds a majority label for a group of labels. Subroutine LBLITP interprets a given label as small grain or other and increments a counter if the label is small grain. Each subroutine is discussed in one of the subsections following. The discussion includes the purpose and entry points of the subroutine, its linkages and interfaces, its inputs and outputs, the storage requirements, and a description of the operation of the subroutine. The listing for the entire utility subroutines package is given as the last subsection.

14.1 CLMPCS

The subroutine CLMPCS and its entries read the cluster map. This subroutine has three entry points:

CLMPCS – to read the cluster map and return cluster sizes and numbers

CLMPXY – to return the x,y position of the jth pixel of the ith cluster

CLMPLC – to set the size of the input cluster map

14.1.1 LINKAGES

This routine does not call any other subprogram.

14.1.2 INTERFACES

The CLMPCS subroutine interfaces with other routines through the calling arguments.

14.1.3 INPUTS

Input to the subroutine consists of the cluster map, named as *.STP, the file output by the A86 processor.

Calling sequence: CALL CLMPCS(NPIXEL,M,MM,LL)

Parameter	Type	Dimension	In/Out	Definition
NPIXEL	I*2	1	Out	Total number of pixels in this map
M	I*2	1	Out	Total number of clusters
MM	I*2	Variable	Out	Array containing sizes of all clusters
LL	I*2	Variable	Out	Array containing the clusters' numbers (the first cluster is 1, second 2, etc.)

Entry calling sequence: CALL CLMPXY(I,J,NL,NC,LL)

Parameter	Type	Dimension	In/Out	Definition
I	I*2	1	In	The cluster ith in consideration
J	I*2	1	In	The jth pixel of the ith cluster
NL	I*2	1	Out	Line number of that pixel
NC	I*2	1	Out	Column number of that pixel
LL	I*2	Variable	In	Array containing the clusters' numbers

Entry calling sequence: CALL CLMPLC(LL1,LC1)

Parameter	Type	Dimension	In/Out	Definition
LL1	I*2	1	In	Total line number of the cluster map

LCl	I*2	1	In	Total column number of the cluster map

14.1.4 OUTPUTS

The results are returned for use by the calling routine.

14.1.5 STORAGE REQUIREMENTS

This subroutine requires a fair amount of storage.

14.1.6 DESCRIPTION

When CLMPCS is called, it reads, line by line, the entire cluster map and determines the number of clusters and their sizes. Next, it sets up an internal table that indicates the number of pixels of cluster i in line ℓ, for all clusters i and lines ℓ. This table will later be referenced to determine the x,y position of a pixel of a cluster.

When CLMPXY is called, it searches through the internal table that was set up when CLMPCS was called. The table immediately gives the line number of pixel j of cluster i. Next CLMPXY reads in the line from the cluster map, and, by pixel-to-pixel comparison, it gives the column number of that pixel.

When CLMPLC is called, the cluster map size is set to that specified by the calling arguments, even when the actual cluster map is larger than that indicated by those arguments.

14.2 GETDOT

The subroutine GETDOT selects pseudorandomly with uniform probability a pixel from a group of pixels (cluster). This subroutine has two entry points:

GETDOT — to get a dot assigned

RANST — to skip some pseudorandom numbers so that the pseudorandom sequence will be different

14.2.1 LINKAGES

This routine calls the RAN subprogram.

14.2.2 INTERFACES

The GETDOT subroutine interfaces with other routines through the calling arguments.

14.2.3 INPUTS

Calling sequence: CALL GETDOT(NP,L,NDARY,IX)

Parameter	Type	Dimension	In/Out	Definition
NP	I*2	1	In	Total number of dots in the cluster of interest
L	I*2	1	In	Lth dot is requested (i.e., L - 1 dots exist in the dot array)
NDARY	I*2	Variable	In/Out	Dot array (all L - 1 dots are in ascending order)
IX	I*2	1	Out	The selected pixel (now called dot)

Entry calling sequence: CALL RANST(ISTART)

Parameter	Type	Dimension	In/Out	Definition
ISTART	I*2	1	In	Number of pseudorandom numbers to be skipped

14.2.4 OUTPUTS

The results are returned for use by the calling routine.

14.2.5 STORAGE REQUIREMENTS

This subroutine requires small storage.

14.2.6 DESCRIPTION

Depending on L, the subprogram GETDOT computes the number of free pixels available; it is (NP - L + 1). Using the system pseudorandom number generator (RAN), a number between 0 and 1 is obtained and then scaled to between 1 and (NP - L + 1). If there are no previously selected dots (i.e., L = 1), then the scaled number is the pixel number of the selected dot. However, if L > 1, then the subroutine needs to adjust the scaled number by referring to dots in the dot array. Since the dots are present in ascending order, the insertion of the new dot can be done in an orderly fashion, starting with the first dot in the dot array.

14.3 GTMPLB

The subroutine GTMPLB returns the label of a pixel (given its x,y position) by reading the ground truth map.

14.3.1 LINKAGES

This routine does not call any other subprogram.

14.3.2 INTERFACES

The GTMPLB subroutine interfaces with other routines through the calling arguments.

14.3.3 INPUTS

Input to the subroutine consists of the ground truth map, named as *.STP, the file output by the A81 processor.

Calling sequence: CALL GTMPLB(NL,NC,LABEL)

Parameter	Type	Dimension	In/Out	Definition
NL	I*2	1	In	Line number of the pixel
NC	I*2	1	In	Column number of the pixel
LABEL	Byte	1	Out	The label of the pixel

14.3.4 OUTPUTS

The results are returned for use by the calling routine.

14.3.5 STORAGE REQUIREMENTS

This subroutine requires small storage.

14.3.6 DESCRIPTION

Since the ground truth map stripped by processor A81 is directly accessible by Fortran, all that is needed to read in a line is just a simple READ statement. After a line has been read into a buffer, the label of the pixel can easily be picked up.

14.4 MR

The subroutine MR returns a majority rule label for a group of labels.

14.4.1 LINKAGES

This routine does not call any other subprogram.

14.4.2 INTERFACES

The MR subroutine interfaces with other routines through the calling arguments.

14.4.3 INPUTS

Calling sequence: CALL MR(IG,NG,LABEL,J1,JG,LG)

Parameter	Type	Dimension	In/Out	Definition
IG	Byte	Variable	In	Array containing the group of labels
NG	I*2	1	In	Dimension of IG
LABEL	Byte	1	Out	The found majority rule label
J1	I*2	1	Out	Number of different labels in IG
JG	Byte	Variable	Out	Array storing those different labels (dimensioned as J1)
LG	I*2	Variable	Out	Numbers of the labels in JG (dimensioned as J1)

14.4.4 OUTPUTS

The results are returned for use by the calling routine.

14.4.5 STORAGE REQUIREMENTS

This subroutine requires small storage.

14.4.6 DESCRIPTION

The subroutine MR scans the input array IG in ascending order, stores the different labels in JG, counts the number of each label and stores these counts in LG, and selects the label with the greatest count. If two labels have that count, the one appearing first will be selected.

14.5 LBLITP

The subroutine LBLITP interprets a raw ground truth label and increments a counter if the label is a small grain.

14.5.1 LINKAGES

This routine does not call any other subprogram.

14.5.2 INTERFACES

The LBLITP subroutine interfaces with other routines through the calling arguments.

14.5.3 INPUTS

Calling sequence: CALL LBLITP(LABEL,LB1,IP)

Parameter	Type	Dimension	In/Out	Definition
LABEL	Byte	1	In	The input raw ground truth label
LB1	I*2	1	Out	The crop code of the label
IP	I*2	1	In/Out	The counter to be incremented if the label is a small grain

14.5.4 OUTPUTS

The results are returned for use by the calling routine.

14.5.5 STORAGE REQUIREMENTS

This subroutine requires small storage.

14.5.6 DESCRIPTION

The crop code of the label is computed as LB1 = LABEL + 128. The counter IP is incremented if any one of the following is true:

a. $1 \leq LB1 \leq 62$

b. $99 \leq LB1 \leq 104$

c. $LB1 = 109$

d. $124 \leq LB1 \leq 129$

e. $LB1 = 134$

14.6 LISTING

```
C------ IN PROGRAM A06.FTN: THE UTILITY PACKAGE
C------ THIS CONTAINS ALL ESSENTIAL SUBROUTINES FOR AA SOFTWARE
C------ PREPARED DURING SPRING, 1979. FOR R.T.E USUAGE.
C
C------ DIRECT-ACCESS (I.E. RANDOM ACCESS) TO CLUSTER MAP
C------ AND GROUND TRUTH MAP IS EMPLOYED.
C------ THEY SHOULD BE THE STRIPPED VERSIONS CREATED BY
C------ A DIFFERENT TASK. N CHU 5-3-79
C
C
C
      SUBROUTINE CLMPCS(NPIXEL,M,MM,LL)
C----- ROUTINES TO READ CLUSTER MAP
C
C       ENTRY:
C         CLMPCS ---- READ CLUSTER MAP AND RETURN CLUSTER SIZES AND NO.
C         CLMPXY ---- GET JTH PIXEL OF I TH CLUSTER, RETURN X,Y POSITION
C         CLMPLC ---- SET SIZE OF CLUSTER MAP, LINE & COL DIMENSION
C
C
      LOGICAL*1 JL(540),JD(468),J1
      INTEGER*2 J2,II(256),III1(256,51)
      INTEGER MM(1),LL(1)
      EQUIVALENCE (JL(73),JD),(J1,J2)
      DATA LINEAV/0/, NLINE/117/,NCOL/196/
C
C----- ENTERING CLMPCS: FIND CLUSTER SIZES AND NO.
C
C
C----- CLEAR THE ARRAYS
      DO 141 K=1,256
141   II(K)=0
      DO 161 L=1,NLINE
      DO 151 K=1,51
151   III1(L,K)=0
161   CONTINUE
C
C------ CHECK FOR CLUSTERS
      K1=NCOL+72
      DO 221 L=1,NLINE
      READ(1'L)(JD(K),K=1,NCOL)
      DO 211 K=1,NCOL
      J1=JD(K)
C     WRITE(6,99002)J2
99002 FORMAT(' J2=',I6)
211   II(J2)=II(J2)+1
C     WRITE(6,99005)L,(JD(K),K=1,NCOL)
99005 FORMAT(' LINE=',I3,/,10(' ',16I4,/) )
221   CONTINUE
C
C     WRITE(6,99101)II
99101 FORMAT(' II=',13I5)
C
C----- COMPACT CLUSTER SIZES IN ARRAY MM
      M=0
      NPIXEL=0
      DO 291 K=1,256
      IF(II(K).EQ.0) GOTO 291
      M=M+1
      MM(M)=II(K)
      LL(M)=K
      II(K)=M
C     WRITE(6,99104)K,M,LL(M)
99104 FORMAT(' 99104----   K,M,LL=',5I5)
      NPIXEL=NPIXEL+MM(M)
291   CONTINUE
L     WRITE(6,99121)(MM(K),K=1,M)
99121 FORMAT(' MM=',12I5)
```

```
C     WRITE(6,99131)(LL(K),K=1,M)
99131 FORMAT(' LL=',12I5)
C
C----- STORE PIXELNO FOR EACH CLUSTER ON EACH LINE
      DO 321 L=1,NLINE
      READ(1'L)(JD(K),K=1,NCOL)
C     WRITE(6,99135)(JD(K),K=1,NCOL)
99135 FORMAT(' AF=',16I4)
      DO 311 K=1,NCOL
      J1=JD(K)
      MX=II(J2)
      II1(L,MX)=II1(L,MX)+1
311   CONTINUE
      IF(L.EQ.1) GOTO 321
      DO 315 K=1,M
315   II1(L,K)=II1(L,K)+II1((L-1),K)
C     WRITE(6,99141)(II1(L,K),K=1,M)
99141 FORMAT(' II1=',12I5)
321   CONTINUE
C
      GOTO 990
C
C
C------- END OF CLMPCS
C
C
C------ ENTERING CLMPXY: FIND LINE AND COL NO. OF PIXEL J IN CLUSTER I
C
      ENTRY CLMPXY(I,J,NL,NC,LL)
C
C
C----- LOOK THROUGH TABLE II1(L,K) TO FIND LINE NO.
      DO 411 NL=1,NLINE
      IF(II1(NL,I).GE.J) GOTO 451
411   CONTINUE
451   CONTINUE
C
C----- DIRECT-ACCESS TO THE CLUSTER MAP IN *.STP FILE
      IF(NL.EQ.LINEAV) GOTO 461
      LINEAV=NL
      READ(1'NL)(JD(K),K=1,NCOL)
461   CONTINUE
C
C----- TEST THE LINE BY COUNTING TO THE COLUMN NO.
481   CONTINUE
      IF(NL.EQ.1) JX=J
      IF(NL.NE.1) JX=J-II1(NL-1,I)
      MX=LL(I)
C     WRITE(6,99741)JX,MX
99741 FORMAT(' 99741****   JX,MX=',5I5)
      DO 491 NC=1,NCOL
      IF(JD(NC).EQ.MX) JX=JX-1
      IF(JX.EQ.0) GOTO 495
491   CONTINUE
495   CONTINUE
C     WRITE(6,99751)I,J,NL,NC
99751 FORMAT(' 99751***   I,J,NL,NC=',6I5)
      GOTO 990
C
C----- ENTERING CLMPLC: TO SET SIZE OF CLUSTER MAP IN CONSIDERATION
C
      ENTRY CLMPLC(LL1,LC1)
      NLINE=LL1
      NCOL=LC1
      GOTO 990
C
C----- END OF CLMPLC
C
C
990   RETURN
      END
```

```
C
C
C
      SUBROUTINE GETDOT(NP,L,NDARY,IX)
C------- FROM A POOL OF (NP-L+1) PIXELS, THIS ROUTINE
C------- PROVIDES A DOT BASED ON UNIFORM PROBABILITY DISTRIBUTION.
C------- THE DOT WILL BE CORRECTLY INSERTED AMONG PREVIOUS GENERATED
C------- DOTS.
C------ NP= TOTAL NO. OF PIXEL IN THE CLUSTER OF INTEREST
C------ L= L TH DOT IS REQUESTED
C------ NDARY IS THE ARRAY CONTAINING ALL (L-1) DOTS (ASCENDING ORDER)
C------ IX= THE PRESENT ASSIGNED DOT'S PIXEL NO. W.R.T. NP PIXELS
C
      INTEGER NDARY(1)
      DATA J1/0/,J2/0/
C
C------ GENERATE RANDOMLY THE DOT'S PIXEL NUMBER
C------ IX= THE NO W.R.T. (NP-L+1) PIXELS
      X=RAN(J1,J2)
      IX=X*(NP-L+1)+1
C     WRITE(6,99201)L,X,IX
99201 FORMAT(' IN GETDOT, L,X,IX=',I4,F7.3,I4)
C
C------- CHECK IF NO DOT PREVIOUSLY ASSIGNED
      IF(L.GT.1) GOTO 131
      NDARY(L)=IX
      GOTO 901
C
C------ FIND THE DOT'S CORRECT PIXEL NO W.R.T. TO NP PIXELS
131   K1=L-1
      DO 161 K=1,K1
      IF(IX-NDARY(K))181,151,151
151   IX=IX+1
C     WRITE(6,99301)K1,K2,IX,NDARY(K)
99301 FORMAT(' K1,K2,IX,NDARY(K)=',6I4)
161   CONTINUE
      K=L
C
C------- INSERT THE NEW DOT INTO THE ARRAY
181   IF((K1+1).LE.K) GOTO 191
      NDARY(K1+1)=NDARY(K1)
      K1=K1-1
C     WRITE(6,99401)K1,NDARY(K1),K
99401 FORMAT(' 99401, K1,NDARY(K1),K=',4I4)
      GOTO 181
191   NDARY(K)=IX
      GOTO 901
C
C
      ENTRY RANST(ISTART)
C------ SETTING STARTING OF RANDOM NUMBER GENERATOR
C------ INSTART MUST BE GREATER THAN ONE
C
      J1=0
      J2=0
      DO 401 K=1,ISTART
401   X=RAN(J1,J2)
901   RETURN
      END
C
C
      SUBROUTINE GTMPLB(NL,NC,LABEL)
C------ TO FIND THE LABEL ON THE COMPRESSED GROUND TRUTH (IN *.STP)
C------ THAT LOCATES AT NL LINE AND NC COL.
C
C
      BYTE J1(360),JX,LABEL
      DATA LINEAV/0/,NCOL/196/
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C----- DIRECT-ACCESS TO GROUND TRUTH DISK FILE
      IF(NL.EQ.LINEAV) GOTO 201
      LINEAV=NL
      READ(2'NL)(J1(K),K=1,NCOL)
201   LABEL=J1(NC)
990   RETURN
      END
C
C
      SUBROUTINE MR(IG,NG,LABEL,J1,JG,LG)
C----- IG, WHEN ENTERS, CONTAIN NG LABELS
C----- ON RETURN, LABEL WILL CONTAIN THE FIRST LABEL
C----- BY MAJORITY RULE
C----- ALSO, ON RETURN, J1 WILL CONTAIN NO. OF DIFFERENT LABELS FOUND
C----- JG CONTAIN THOSE LABELS, AND LG CONTAIN THE NO. OF THESE LABELS
C
      BYTE IG(1),JG(1),LABEL
      INTEGER LG(1)
C
      J1=1
      JG(1)=IG(J1)
      LG(J1)=1
      LABEL=IG(J1)
      IF(NG.LE.1) GOTO 901
C
C----- COUNT SUBPIXELS HAVEING SAME LABELS
      DO 251 I=2,NG
      DO 221 J=1,J1
C     WRITE(6,99108)I,J,IG(I),JG(J),J1
99108 FORMAT(' I,J,IG,JG,J1=',10I4)
      IF(IG(I).EQ.JG(J)) GOTO 231
221   CONTINUE
      J1=J1+1
      JG(J1)=IG(I)
      LG(J1)=1
      GOTO 251
231   LG(J)=LG(J)+1
251   CONTINUE
C
C----- CHOOSE THE MAJORITY LABEL
      MAX=0
      DO 281 I=1,J1
      IF(LG(I).LE.MAX) GOTO 281
      MAX=LG(I)
      J=I
281   CONTINUE
      LABEL=JG(J)
901   RETURN
      END
C
C
       SUBROUTINE LBLITP(LABEL,LB1,IP)
C------ INTERPRETATE A RAW LABEL (ONE BYTE LONG) AND RETURN
C------ CODE IN LB1 (INTEGER). IP IS INCREMENTED BY ONE IF
C------ THE LABEL SHOWS SMALL GRAIN.
C
      BYTE LABEL
      LB1=LABEL
      LB1=LB1+128
      IF((LB1.GE.1).AND.(LB1.LE.62)) GOTO 401
      IF((LB1.GE.99).AND.(LB1.LE.104)) GOTO 401
      IF(LB1.EQ.109) GOTO 401
      IF((LB1.GE.124).AND.(LB1.LE.129)) GOTO 401
      IF(LB1.EQ.134) GOTO 401
      GOTO 901
401   IP=IP+1
901   RETURN
      END
```

15. EXAMPLE OF OPERATING PROCEDURE

All programs are written for an interactive environment but execution in batch is also possible. When a program begins execution, guide messages will appear on the terminal and inputs are expected to be keyed in. Most messages contain a format specification: "A" indicates alphanumeric characters are expected, "I" an integer expected, and "F" a floating point number. The format field is indicated by the length of a chain of characters on the terminal. Those keyed-in characters not underneath the format field will be ignored. To run any program, just type, after MCR>, "RUN (the program's name)" and hit the ESC key.

Suppose an analyst desires to test a cluster map 100519101.DT2 against the Bayesian dot allocation scheme with adaptive prior. Suppose the ground truth map is initially in 100577278.GT0. The analyst needs to execute three jobs.

1. To strip the cluster map

```
MCR>RUN A86$
PROGRAM A11(A86.TSK). TO STRIP A CLUSTER MAP FILE.
INPUT CLUSTER MAP FILE NAME.
AAAAAAAAAAAAA
100519101.DT2
  20 LINES WRITTEN ON 100519101.STP
  40 LINES WRITTEN ON 100519101.STP
  60 LINES WRITTEN ON 100519101.STP
  80 LINES WRITTEN ON 100519101.STP
 100 LINES WRITTEN ON 100519101.STP
STRIPPED MAP OF SIZE 117 LINES BY 196 COLS PRODUCED IN 100519101.STP

NEXT CLUSTER MAP PLEASE. TO STOP, JUST TYPE STOP
PROGRAM A11(A86.TSK). TO STRIP A CLUSTER MAP FILE.
INPUT CLUSTER MAP FILE NAME.
AAAAAAAAAAAAA
STOP
A86  --  STOP
```

2. **To strip a ground truth map**

```
MCR>RUN A81$
PROGRAM:A10(A81.TSK). TO REDUCE GROUND TRUTH
RESOLUTION. OUTPUT IN STRIPPED FORMAT
TYPE GROUND TRUTH FILE NAME
AAAAAAAAAAAAA
100577278.GT0

GROUND TRUTH FILE FROM 100577278.GT0
LABEL BY MAJORITY RULE...EXECUTION BEGINS...WHIT
  20 LINES WRITTEN ON 1005.STP
  40 LINES WRITTEN ON 1005.STP
  60 LINES WRITTEN ON 1005.STP
  80 LINES WRITTEN ON 1005.STP
 100 LINES WRITTEN ON 1005.STP
STRIPPED MAP OF SIZE 117 LINES BY 196 COLS PRODUCED IN 1005.STP

NEXT GROUND TRUTH MAP PLEASE. TO STOP, JUST TYPE STOP
PROGRAM:A10(A81.TSK). TO REDUCE GROUND TRUTH
RESOLUTION. OUTPUT IN STRIPPED FORMAT
TYPE GROUND TRUTH FILE NAME
AAAAAAAAAAAAA
STOP
A81   --   STOP
```

3. To run the Bayesian dot allocation

There are two choices:

a. Interactive – just type, after MCR>, "RUN A01.TSK$(ESC key)," and then answer the questions.

b. Batch – submit a batch job with the following cards.

```
$JOB/NAME = SOMEONE/MCR/LIMIT = 360/ACCOUNT = 131 1
$CREATE FOR007.DAT
Data cards
$EOD
$MCR RUN A81.TSK
$EOJ
```

All program sources (Fortran), object files, and task files can be accessed from UIC = [131,1] on the Image Processor (PDP 11/45) in JSC Building 17.

APPENDIX A

SUBPROGRAMS OF A81: TO STRIP A GROUND TRUTH MAP

A.1 MR6

The subprogram MR6 determines a majority rule label for a group of six labels.

A.1.1 LINKAGES

This routine does not call any other subprogram. It is called by the MAIN program, A81.

A.1.2 INTERFACES

The MR6 subprogram interfaces with other routines through the calling arguments.

A.1.3 INPUTS

Calling sequence: CALL MR6(IG,LABEL)

Parameter	Type	Dimension	In/Out	Definition
IG	Byte	6	In	Array containing the six labels
LABEL	Byte	1	Out	The chosen majority rule label

A.1.4 OUTPUTS

The results are returned for use by the calling routine.

A.1.5 STORAGE REQUIREMENTS

This subprogram requires little storage.

A.1.6 DESCRIPTION

The subprogram MR6 scans the input array IG in ascending order, counts the number of identical labels, and selects the label with the greatest count. If two labels have that count, the one appearing first will be selected.

A.1.7 LISTING

The subprogram listing is provided in section 3.7.

APPENDIX B

SUBPROGRAMS OF A82: PROPORTIONAL DOT ALLOCATION

B.1 SAE

The subprogram SAE computes a proportion estimate using proportional dot allocation.

B.1.1 LINKAGES

This routine calls the CLMPCS, GETDOT, GTMPLB, and LBLITP subprograms. It is called by the MAIN program, A82.

B.1.2 INTERFACES

The SAE subprogram interfaces with other routines through common block PRTFLG.

B.1.3 INPUTS

Calling sequence: CALL SAE(NDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
NDOT	I*2	1	In	Total number of dots to be allocated
PSG	R*4	1	Out	The proportion estimate for small grain

B.1.4 OUTPUTS

The results are returned for use by the calling routine.

B.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

B.1.6 DESCRIPTION

For a description of subprogram SAE, see section 5.1.

B.1.7 FLOW CHART

The subprogram flow chart is provided in section 5.6.

B.1.8 LISTING

The subprogram listing is provided in section 5.7.

APPENDIX C

SUBPROGRAMS OF A83: PROPORTIONAL DOT ALLOCATION, MAJORITY RULE LABELING

C.1 SAEMR

The subprogram SAEMR computes a proportion estimate using proportional dot allocation and majority rule labeling.

C.1.1 LINKAGES

This routine calls the GETDOT, CLMPCS, GTMPLB, LBLITP, and MR subprograms. It is called by the MAIN program, A83.

C.1.2 INTERFACES

The SAEMR subprogram interfaces with other routines through common block PRTFLG.

C.1.3 INPUTS

Calling sequence: CALL SAEMR(NDOT,PSG).

Parameter	Type	Dimension	In/Out	Definition
NDOT	I*2	1	In	Total number of dots to be allocated
PSG	R*4	1	Out	The proportion estimate for small grain

C.1.4 OUTPUTS

The results are returned for use by the calling routine.

C.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

C.1.6 DESCRIPTION

For a description of subprogram SAEMR, see section 6.1.

C.1.7 FLOW CHART

The subprogram flow chart is provided in section 6.6.

C.1.8 LISTING

The subprogram listing is provided in section 6.7.

APPENDIX D

SUBPROGRAMS OF A84: BAYESIAN DOT ALLOCATION (UNIFORM PRIOR)

D.1 SAEB1

The subprogram SAEB1 computes a proportion estimate using Bayesian dot allocation (uniform prior). This subprogram has two entry points:

SAEB1 – to compute an estimate

INIT1 – to set the number of initial dots

D.1.1 LINKAGES

This routine calls the CLMPCS, GETDOT, CLMPXY, GTMPLB, and LBLITP subprograms. It is called by the MAIN program, A84.

D.1.2 INTERFACES

The SAEB1 subprogram interfaces with other routines through common block PRTFLG.

D.1.3 INPUTS

Calling sequence: CALL SAEB1(MAXDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
MAXDOT	I*2	1	In	Maximum number of dots to be assigned
PSG	R*2	Variable	Out	The proportion estimate for small grain

Entry calling sequence: CALL INIT1(NNIT)

Parameter	Type	Dimension	In/Out	Definition
NNIT	I*2	1	In	Number of initial dots to be assigned to each cluster

D.1.4 OUTPUTS

The results are returned for use by the calling routine.

D.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

D.1.6 DESCRIPTION

For a description of subprogram SAEB1, see section 7.1.

D.1.7 FLOW CHART

The subprogram flow chart is provided in section 7.6.

D.1.8 LISTING

The subprogram listing is provided in section 7.7.

APPENDIX E

SUBPROGRAMS OF A85: BAYESIAN DOT ALLOCATION (NO PRIOR)

E.1 SAEB2

The subprogram SAEB2 computes a proportion estimate using Bayesian dot allocation (no prior). This subroutine has two entry points:

SAEB2 – to compute an estimate

INIT2 – to set the number of initial dots

E.1.1 LINKAGES

This routine calls the CLMPCS, GETDOT, CLMPXY, GTMPLB, and LBLITP subprograms. It is called by the MAIN program, A85.

E.1.2 INTERFACES

The SAEB2 subprogram interfaces with other routines through common block PRTFLG.

E.1.3 INPUTS

Calling sequence: CALL SAEB2(MAXDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
MAXDOT	I*2	1	In	Maximum number of dots to be assigned
PSG	R*2	Variable	Out	The proportion estimate for small grain

Entry calling sequence: INIT2(NNIT)

Parameter	Type	Dimension	In/Out	Definition
NNIT	I*2	1	In	Number of initial dots to be assigned to each cluster

E.1.4 OUTPUTS

The results are returned for use by the calling routine.

E.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

E.1.6 DESCRIPTION

For a description of subprogram SAEB2, see section 8.1.

E.1.7 FLOW CHART

The subprogram flow chart is provided in section 8.6.

E.1.8 LISTING

The subprogram listing is provided in section 8.7.

APPENDIX F

SUBPROGRAMS OF A87: BAYESIAN DOT ALLOCATION (QUADRATIC PRIOR)

F.1 SAEB3

The subprogram SAEB3 computes a proportion estimate using Bayesian dot allocation (quadratic prior). This subprogram has two entry points:

SAEB3 — to compute an estimate

INIT3 — to set the number of initial dots

F.1.1 LINKAGES

This routine calls the CLMPCS, GETDOT, CLMPXY, GTMPLB, TH1, and LBLITP subprograms. It is called by the MAIN program, A87.

F.1.2 INTERFACES

The SAEB3 subprogram interfaces with other routines through common block PRTFLG.

F.1.3 INPUTS

Calling sequence: CALL SAEB3(MAXDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
MAXDOT	I*2	1	In	Maximum number of dots to be assigned
PSG	R*2	Variable	Out	The proportion estimate for small grain

Entry calling sequence: CALL INIT3(NNIT)

Parameter	Type	Dimension	In/Out	Definition
NNIT	I*2	1	In	Number of initial dots to be assigned to each cluster

F.1.4 OUTPUTS

The results are returned for use by the calling routine.

F.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

F.1.6 DESCRIPTION

For a description of subprogram SAEB3, see section 9.1.

F.1.7 FLOW CHART

The subprogram flow chart is provided in section 9.6.

F.1.8 LISTING

The subprogram listing is provided in section 9.7.

F.2 TH1

The subprogram TH1 computes an estimate $\hat{\theta}$ (n,x) for use in connection with Bayesian dot allocation (quadratic prior).

F.2.1 LINKAGES

This routine does not call any other subprogram. It is called by subprogram SAEB3.

F.2.2 INTERFACES

The TH1 subprogram interfaces with other routines through the calling arguments.

F.2.3 INPUTS

Function usage: TH1(N,NX)

Parameter	Type	Dimension	In/Out	Definition
N	I*2	1	In	Number of dots assigned
NX	I*2	1	In	Number of small-grain dots assigned

F.2.4 OUTPUTS

The function's value is returned.

F.2.5 STORAGE REQUIREMENTS

This subprogram requires small storage.

F.2.6 DESCRIPTION

For a description of subprogram TH1, see section 9.1.

F.2.7 FLOW CHART

The subprogram flow chart is provided in section 9.6.

F.2.8. LISTING

The subprogram listing is provided in section 9.7.

APPENDIX G

SUBPROGRAMS OF A89: BAYESIAN DOT ALLOCATION (MODIFIED QUADRATIC PRIOR)

G.1 SAEB4

The subprogram SAEB4 computes a proportion estimate using Bayesian dot allocation (modified quadratic prior). This subprogram has two entry points:

SAEB4 – to compute an estimate

INIT4 – to set the number of initial dots

G.1.1 LINKAGES

This routine calls the CLMPCS, GETDOT, CLMPXY, GTMPLB, TH2, and LBLITP subprograms. It is called by the MAIN program, A89.

G.1.2 INTERFACES

The SAEB4 subprogram interfaces with other routines through common block PRTFLG.

G.1.3 INPUTS

Calling sequence: CALL SAEB4(MAXDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
MAXDOT	I*2	1	In	Maximum number of dots to be assigned
PSG	R*2	Variable	Out	The proportion estimate for small grain

Entry calling sequence: CALL INIT4(NNIT)

Parameter	Type	Dimension	In/Out	Definition
NNIT	I*2	1	In	Number of initial dots to be assigned to each cluster

G.1.4 OUTPUTS

The results are returned for use by the calling routine.

G.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

G.1.6 DESCRIPTION

For a description of subprogram SAEB4, see section 10.1.

G.1.7 FLOW CHART

The subprogram flow chart is provided in section 10.6.

G.1.8 LISTING

The subprogram listing is provided in section 10.7.

G.2 TH2

The subprogram TH2 computes an estimate $\hat{\theta}$ (n,x) for use in connection with Bayesian dot allocation (modified quadratic prior). It has two entries:

TH2 – return the value of the function
TH2PAR – to set the parameters a, b, c

G.2.1 LINKAGES

This routine does not call any other subprogram. It is called by subprogram SAEB4.

G.2.2 INTERFACES

The TH2 subprogram interfaces with other routines through the calling arguments.

G.2.3 INPUTS

Function usage: TH2(N,NX)

Parameter	Type	Dimension	In/Out	Definition
N	I*2	1	In	Number of dots assigned
NX	I*2	1	In	Number of small-grain dots assigned

Entry calling sequence: CALL TH2PAR(AX,BX,CX)

Parameter	Type	Dimension	In/Out	Definition
AX	R*4	1	In	Set parameter a to value in AX
BX	R*4	1	In	Set parameter b to value in BX
CX	R*4	1	In	Set parameter c to value in CX

G.2.4 OUTPUTS

The function's value is returned.

G.2.5 STORAGE REQUIREMENTS

This subprogram requires small storage.

G.2.6 DESCRIPTION

For a description of subprogram TH2, see section 10.1. The entry point TH2PAR provides a means of modifying the parameters a, b, c, which are initially set to a = 6, b = -7.877, and c = 2.9345.

G.2.7 FLOW CHART

The subprogram flow chart is provided in section 10.6.

G.2.8 LISTING

The subprogram listing is provided in section 10.7.

APPENDIX H

SUBPROGRAMS OF A91: BAYESIAN DOT ALLOCATION (ADAPTIVE PRIOR)

H.1 SAEB5

The subprogram SAEB5 computes a proportion estimate using Bayesian dot allocation (adaptive prior). This routine has two entry points:

SAEB5 – to compute an estimate
INIT5 – to set the number of initial dots

H.1.1 LINKAGES

This routine calls the CLMPCS, GETDOT, CLMPXY, GTMPLB, TH3, and LBLITP subprograms. It is called by the MAIN program, A91.

H.1.2 INTERFACES

The SAEB5 subprogram interfaces with other routines through common block PRTFLG.

H.1.3 INPUTS

Calling sequence: CALL SAEB5(MAXDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
MAXDOT	I*2	1	In	Maximum number of dots to be assigned
PSG	R*2	Variable	Out	The proportion estimate for small grain

Entry calling sequence: CALL INIT5(NNIT)

Parameter	Type	Dimension	In/Out	Definition
NNIT	I*2	1	In	Number of initial dots to be assigned to each cluster

H.1.4 OUTPUTS

The results are returned for use by the calling routine.

H.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

H.1.6 DESCRIPTION

For a description of subprogram SAEB5, see section 11.1.

H.1.7. FLOW CHART

The subprogram flow chart is provided in section 11.6.

H.1.8 LISTING

The subprogram listing is provided in section 11.7.

H.2 TH3

The subprogram TH3 computes an estimate $\hat{\theta}$ (n,x) for use in connection with Bayesian dot allocation (adaptive prior). It has four entries:

- TH3 – to return the value of the function
- TH3PAR – to set the parameters a, b, c, for the quadratic prior
- SETEP – to set the parameter α for the exponential prior
- SETPRI – to select the prior

H.2.1 LINKAGES

This routine does not call any other subprogram. It is called by subprogram SAEB5.

H.2.2 INTERFACES

The TH3 subprogram interfaces with other routines through the calling arguments.

H.2.3 INPUTS

Function usage: TH3(N,NX)

Parameter	Type	Dimension	In/Out	Definition
N	I*2	1	In	Number of dots assigned
NX	I*2	1	In	Number of small-grain dots assigned

Entry calling sequence: CALL TH3PAR(AX,BX,CX)

Parameter	Type	Dimension	In/Out	Definition
AX	R*4	1	In	Set parameter a to value in AX
BX	R*4	1	In	Set parameter b to value in BX
CX	R*4	1	In	Set parameter c to value in CX

Entry calling sequence: CALL SETEP(ALPHA)

Parameter	Type	Dimension	In/Out	Definition
ALPHA	R*4	1	In	Set parameter α to value in ALPHA

Entry calling sequence: CALL SETPRI(JFPR)

Parameter	Type	Dimension	In/Out	Definition
JFPR	I*2	1	In	Prior select: 1 – quadratic prior 2 – exponential prior

H.2.4 OUTPUTS

The function's value is returned.

H.2.5 STORAGE REQUIREMENTS

This subprogram requires small storage.

H.2.6 DESCRIPTION

For a description of subprogram TH3, see section 11.1. The ability of the processor to select one prior (quadratic) or the other (exponential) at any instant comes from this subprogram, TH3. By calling SETPRI before using the function TH3, one can switch from one prior to the other. Notice that for quadratic prior, the parameters are initially set as follows: $a = 6$, $b = -7.877$, $c = 2.9345$. For the exponential prior, the parameter α is not initially set to any value. The calling program must set α by calling SETEP(ALPHA) before using the function.

H.2.7 FLOW CHART

The subprogram flow chart is provided in section 11.6.

H.2.8 LISTING

The subprogram listing is provided in section 11.7.

APPENDIX I

SUBPROGRAMS OF A88: BAYESIAN DOT ALLOCATION, MAJORITY RULE LABELING

I.1 SAEBMR

The subprogram SAEBMR computes a proportion estimate using Bayesian dot allocation and majority rule labeling.

I.1.1 LINKAGES

This routine calls the CLMPCS and BMR1 subprograms. It is called by the MAIN program, A88.

I.1.2 INTERFACES

The subprogram interfaces with other routines through common block PRTFLG.

I.1.3 INPUTS

Calling sequence: CALL SAEBMR(NDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
NDOT	I*2	1	Out	Total number of dots that are allocated
PSG	R*4	1	Out	The proportion estimate for small grain

I.1.4 OUTPUTS

The results are returned for use by the calling routine.

I.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

I.1.6 DESCRIPTION

For a description of subprogram SAEBMR, see section 12.1. This subroutine uses the subroutine BMR1 (which in turn uses subroutine BMRSUP) to achieve the computation of the estimate.

Subroutine SAEBMR is the executive routine to drive the rest of the routines.

I.1.7 FLOW CHART

The subprogram flow chart is provided in section 12.6.

I.1.8 LISTING

The subprogram listing is provided in section 12.7.

I.2 BMR1

The subprogram BMR1 returns a small-grain or non-small-grain label for a cluster.

I.2.1 LINKAGES

This routine calls the BMRSUP subprogram. It is called by subprogram SAEBMR.

I.2.2 INTERFACES

The subprogram interfaces with other routines through common block PRTFLG.

I.2.3 INPUTS

Calling sequence: CALL BMR1(ISG,NS,NDARY,K1,MM,NN,LL,M)

Parameter	Type	Dimension	In/Out	Definition
ISG	I*2	1	Out	Small-grain indicator: 1 – yes, 0 – no
NS	I*2	2	Out	Split of small-grain and non-small-grain dots
NDARY	I*2	Variable	In/Out	Dot array
K1	I*2	1	In	The kth cluster the dots will be allocated to
MM	I*2	Variable	In	Array containing sizes of all clusters
NN	I*2	Variable	In/Out	Array containing numbers of dots assigned to all clusters
LL	I*2	Variable	In	Array containing the clusters' numbers (the first cluster is 1, second cluster is 2, etc.)
M	I*2	1	In	Total number of clusters

I.2.4 OUTPUTS

The results are returned for use by the calling routine.

I.2.5 STORAGE REQUIREMENTS

This subprogram requires small storage.

I.2.6 DESCRIPTION

For a description of subprogram BMR1, see section 12.1. This routine serves as a middleman between the executive subroutine SAEBMR and the workhorse routine BMRSUP. If the user wants to change the prior (from uniform prior), only this routine, BMR1, needs reprogramming.

I.2.7 FLOW CHART

The subprogram flow chart is provided in section 12.6.

I.2.8 LISTING

The subprogram listing is provided in section 12.7.

I.3 BMRSUP

The subprogram BMRSUP performs the essential function of dot assignment and determination of the small-grain and non-small-grain split.

I.3.1 LINKAGES

This routine calls the GETDOT, GTMPLB, and LBLITP subprograms. It is called by subprogram BMR1.

I.3.2 INTERFACES

The subprogram interfaces with other routines through common block PRTFLG.

I.3.3 INPUTS

Calling sequence: CALL BMRSUP(JDOT,NS,MR,NDARY,KOLD,MM,NN,LL,M)

Parameter	Type	Dimension	In/Out	Definition
JDOT	I*2	1	In	Number of dots requested to be assigned
NS	I*2	2	Out	Split of small-grain and non-small-grain dots
MR	I*2	1	In/Out	Majority indicator: 1 – small grain 2 – other
NDARY	I*2	Variable	In/Out	Dot array
KOLD	I*2	1	In	The kth cluster the dots will be allocated to

Parameter	Type	Dimension	In/Out	Definition
MM	I*2	Variable	In	Array containing sizes of all clusters
NN	I*2	Variable	In/Out	Array containing numbers of dots assigned to all clusters
LL	I*2	Variable	In	Array containing the clusters' numbers (the first cluster is 1, second 2, etc.)
M	I*2	1	In	Total number of clusters

I.3.4 OUTPUTS

The results are returned for use by the calling routine.

I.3.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

I.3.6 DESCRIPTION

For a description of subprogram BMRSUP, see section 12.1. The subroutine BMRSUP is the workhorse of the entire program A88; it assigns dots and determines labels.

I.3.7 FLOW CHART

The subprogram flow chart is provided in section 12.6.

I.3.8 LISTING

The subprogram listing is provided in section 12.7.

APPENDIX J

SUBPROGRAMS OF A90: NON-BAYESIAN SEQUENTIAL DOT ALLOCATION, MAJORITY RULE LABELING

J.1 SAENBS

The subprogram SAENBS computes a proportion estimate using non-Bayesian sequential dot allocation and majority rule labeling.

J.1.1 LINKAGES

This routine calls the CLMPCS and NBSMR subprograms. It is called by the MAIN program, A90.

J.1.2 INTERFACES

The subprogram interfaces with other routines through common block PRTFLG.

J.1.3 INPUTS

Calling sequence: CALL SAENBS(NDOT,PSG)

Parameter	Type	Dimension	In/Out	Definition
NDOT	I*2	1	Out	Total number of dots that are allocated
PSG	R*4	1	Out	The proportion estimate for small grain

J.1.4 OUTPUTS

The results are returned for use by the calling routine.

J.1.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

J.1.6 DESCRIPTION

For a description of subprogram SAENBS, see section 13.1. This subroutine uses NBSMR to achieve the computation of the estimate.

J.1.7 FLOW CHART

The subprogram flow chart is provided in section 13.6.

J.1.8 LISTING

The subprogram listing is provided in section 13.7.

J.2 NBSMR

The subprogram NBSMR assigns dots to a cluster according to non-Bayesian sequential majority rule.

J.2.1 LINKAGES

This routine calls the GETDOT, CLMPXY, GTMPLB, and LBLITP subprograms. It is called by subprogram SAENBS.

J.2.2 INTERFACES

The subprogram interfaces with other routines through common block PRTFLG.

J.2.3 INPUTS

Calling sequence: CALL NBSMR(ISG,NS,NDARY,K1,MM,NN,LL,NX,M)

Parameter	Type	Dimension	In/Out	Definition
ISG	I*2	1	Out	Small-grain indicator: 1 – yes, 0 – no
NS	I*2	2	Out	Split of small-grain and non-small-grain dots
NDARY	I*2	Variable	In/Out	Dot array

Parameter	Type	Dimension	In/Out	Definition
K1	I*2	1	In	The kth cluster the dots will be assigned to
MM	I*2	Variable	In	Array containing sizes of all clusters
NN	I*2	Variable	In/Out	Array containing numbers of dots assigned to all clusters
LL	I*2	Variable	In	Array containing the clusters' numbers (the first cluster is 1, second 2, etc.)
NX	I*2	1	In/Out	Array containing numbers of small-grain dots assigned to the clusters
M	I*2	1	In	Total number of clusters

Entry calling sequence: CALL INIT(NNIT)

Parameter	Type	Dimension	In/Out	Definition
NNIT	I*2	1	In	Number of initial dots desired

J.2.4 OUTPUTS

The results are returned for use by the calling routine.

J.2.5 STORAGE REQUIREMENTS

This subprogram requires a fair amount of storage.

J.2.6 DESCRIPTION

For a description of subprogram NBSMR, see section 13.1. This subroutine is the workhorse used by subroutine SAENBS to compute the estimate.

J.2.7 FLOW CHART

The subprogram flow chart is provided in section 13.6.

J.2.8 LISTING

The subprogram listing is provided in section 13.7.