T-1314/4
MA-129TA
LARS Contract Report 112877

CR151609

Final Report

Test of Spectral/Spatial Classifier

by J. L. Kast
B. J. Davis

November 1977

Principal Investigator
D. A. Landgrebe

Prepared for
National Aeronautics and Space Administration
Johnson Space Center
Earth Observation Division
Houston, Texas 77058
Contract No. NAS9-14970, Task 2
Technical Monitor: J. D. Erickson/SF3

Submitted by
The Laboratory for Applications of Remote Sensing
Purdue University
West Lafayette, Indiana 47906

STAR INFORMATION FORM

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
112877		

4. Title and Subtitle	5. Report Date
Test of Spectral/Spatial Classifier	November 1977
	6. Performing Organization Code
7. Author(s) James L. Kast and Barbara J. Davis	8. Performing Organization Report No. 112877
	10. Work Unit No.
9. Performing Organization Name and Address Laboratory for Applications of Remote Sensing 1220 Potter Drive Purdue University West Lafayette, Indiana 47906	11. Contract or Grant No. NAS9-14970
	13. Type of Report and Period Covered T-1314/4 Final Report
12. Sponsoring Agency Name and Address J.D. Erickson/SF3 NASA/Johnson Space Center Houston, Texas 77058	14. Sponsoring Agency Code

15. Supplementary Notes

D. A. Landgrebe was LARS principal investigator.

16. Abstract

This report provides the final results of the subtask to test the spectral/spatial classifier (ECHO). This document reports on:

1 the programming of the Nonsupervised ECHO algorithms,
2 tests of the effects of the input parameters on six performance measures, and
3 comparison of the Nonsupervised ECHO classifier with the Supervised ECHO classifier and the perpoint classifier.

The Nonsupervised ECHO classifier identifes objects without the benefit of class statistics. Statistics of the objects thus identified may be of value in training for the classifier. The Supervised ECHO classifier demonstrates superior classification accuracy, reduced variability of classification results, and requires less CPU time when compared to the perpoint classifier. The Nonsupervised ECHO processor requires less CPU time and produces less variable classification results than the perpoint classifier, but does not produce classification results which are superior to the perpoint classifier.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Spatial, classification, homogeneous clustering, verification, classifier, spatial/spectral classification, spatial/spectral clustering, object identification, sample classification	

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

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

Table of Contents

Test of Spectral/Spatial Classifier

Page

Introduction.. 2a - 1

Past Work at LARS.. 2a - 1

Objectives.. 2a - 3

Description of Work... 2a - 4

Documented Fortran Programs............................. 2a - 4

Test of Nonsupervised ECHO algorithms.................. 2a - 9

Data Sets Tested.. 2a - 9

Training Procedures... 2a - 13

Landsat Data Sets... 2a - 13

Simulated Thematic Mapper Data Sets..................... 2a - 13

Aircraft Data Sets.. 2a - 14

Dependent Variables.. 2a - 15

Results... 2a - 17

Landsat Results... 2a - 17

Simulated Thematic Mapper Results........................ 2a - 44

Aircraft Results... 2a - 64

Nonsupervised ECHO Object Map Integrity Assessment....... 2a - 69

Conclusions.. 2a - 71

Summary.. 2a - 71

Recommendations... 2a - 72

References... 2a - 73

Appendix A: Nonsupervised ECHO Program Abstracts

Appendix B: Nonsupervised ECHO Data Files

List of Figures

Test of Spectral/Spatial Classifier

Figure	Page
Figure 2a-1 General Flow of Nonsupervised ECHO (Phase 1)	2a-6
Figure 2a-2 General Flow of Nonsupervised ECHO (Phase 2)	2a-7
Figure 2a-3 Nonsupervised ECHO CPU Requirements for the LACIE/SRS Data Sets	2a-18
Figure 2a-4 Nonsupervised ECHO Field Center Pixel Performance for the LACIE/SRS Data sets	2a-18
Figure 2a-5 Nonsupervised ECHO Full Field Performance for the LACIE/SRS Data Sets	2a-19
Figure 2a-6 Nonsupervised ECHO Training Field Performance for the LACIE/SRS Data Sets	2a-19
Figure 2a-7 Nonsupervised ECHO RMS Proportion estimate error for the LACIE/SRS Data Sets	2a-20
Figure 2a-8 Nonsupervised ECHO Classification Variability for the LACIE/SRS Data Sets	2a-20
Figure 2a-9 Nonsupervised ECHO CPU Requirements for the CITARS Data Sets	2a-24
Figure 2a-10 Nonsupervised ECHO Field Center Pixel Performance for the CITARS Data Sets	2a-24
Figure 2a-11 Nonsupervised ECHO Full Field Performance for the CITARS Data Sets	2a-25
Figure 2a-12 Nonsupervised ECHO Training Field Performance for the CITARS Data Sets	2a-25

<u>Figures</u> <u>Page</u>

Figure 2a-13
Nonsupervised ECHO RMS Proportion Estimate Error for the CITARS Data Sets 2a-26

Figure 2a-14
Nonsupervised ECHO Classification Variability for the CITARS Data Sets 2a-26

Figure 2a-15
Nonsupervised ECHO CPU Requirements for all Landsat Data Sets 2a-30

Figure 2a-16
Nonsupervised ECHO Field Center Pixel Performance for all Landsat Data Sets 2a-30

Figure 2a-17
Nonsupervised ECHO Full Field Performance for all Landsat Data Sets 2a-31

Figure 2a-18
Nonsupervised ECHO Training Field Performance for all Landsat Data Sets 2a-31

Figure 2a-19
Nonsupervised ECHO RMS Proportion Estimate Error for all Landsat Data Sets 2a-32

Figure 2a-20
Nonsupervised ECHO Classification Variability for all Landsat Data Sets 2a-32

Figure 2a-21
Nonsupervised ECHO CPU Requirements for the Simulated Thematic Mapper Data Sets 2a-51

Figure 2a-22
Nonsupervised ECHO Field Center Pixel Performance for the Simulated Thematic Mapper Data Sets 2a-51

Figure 2a-23
Nonsupervised ECHO Full Field Performance for the Simulated Thematic Mapper Data Sets 2a-52

Figure 2a-24
Nonsupervised ECHO Training Field Performance for the Simulated Thematic Mapper Data Sets 2a-52

Figure 2a-25
Nonsupervised ECHO RMS Proportion Estimate Error for the Simulated Thematic Mapper Data Sets 2a-53

Figures | Page

Figure 2a-26
Nonsupervised ECHO Classification Variability for the Simulated Thematic Mapper Data Sets 2a-53

Figure 2a-27
Nonsupervised ECHO CPU Requirements for Aircraft Data Set 2a-66

Figure 2a-28
Nonsupervised ECHO Field Center Pixel Performance for the Aircraft Data Set 2a-66

Figure 2a-29
Nonsupervised ECHO Full Field Performance for Aircraft Data Set 2a-67

Figure 2a-30
Nonsupervised ECHO Training Field Performance for Aircraft Data Set 2a-67

Figure 2a-31
Nonsupervised ECHO Classification Variability for Aircraft Data Set 2a-68

Figure 2a-32
Nonsupervised ECHO Object Map 2a-70

List of Tables

Test of Spectral/Spatial Classifier

Table	Page
Table 2a-1 Nonsupervised ECHO Processor	2a-8
Table 2a-2 Data Sets	2a-10 - 2a-12
Table 2a-3 Overall Landsat Comparisons Effects of Parameter on LACIE Data	2a-21
Table 2a-4 Overall Landsat Comparisons Effects of Parameter on CITARS Data	2a-27
Table 2a-5 Overall Landsat Comparisons Effects of Parameters	2a-33
Table 2a-6 Results of Individual Landsat Data Set Parameter Anovas	2a-34
Table 2a-7 Overall Landsat Comparisons: Comparison of Perpoint, Supervised ECHO, and Nonsupervised ECHO Classifiers	2a-37
Table 2a-8 Overall Landsat Comparisons: Comparison of Perpoint, Supervised ECHO, and Nonsupervised ECHO Classifiers	2a-38
Table 2a-9 Optimal Results of Landsat Data Sets	2a-40 - 2a-41
Table 2a-10 Comparison of the Optimal Landsat Results for the Supervised ECHO, Nonsupervised ECHO and Perpoint Classifier	2a-42
Table 2a-11 Results of Individual Thematic Mapper Data Set Parameter Anovas	2a-45 - 2a-46

Table	Page
Table 2a-12 Overall Thematic Mapper Results Significant Effects of Parameters on Dependent Variables	2a-47
Table 2a-13 Overall Thematic Mapper Results	2a-55
Tables 2a-14 Overall Thematic Mapper Effects of Classifier on Variables	2a-56 - 2a-58
Table 2a-15 Optimal Results of Thematic Mapper Data Sets	2a-60 - 2a-61
Table 2a-16 Comparison of the Optimal Simulated Thematic Mapper Results for the Supervised ECHO, Nonsupervised ECHO, and Perpoint Classifiers	2a-62
Table 2a-17 T-tests Comparing the Perpoint Results to the Distribution of Nonsupervised ECHO Results for Six Variables	2a-65
Table 2a-18 Effects of Nonsupervised ECHO Homogeneity and Annexation Thresholds on Six Dependent Variables for One Aircraft Data Set	2a-65

TASK 2
TEST OF SPECTRAL/SPATIAL CLASSIFIER

INTRODUCTION

Contemporary classifiers for analysis of remotely sensed data compare spectral measurements from each feature of each pixel to class statistics, computing a likelihood discriminant function associated with each class, and categorizing the point according to the class with the largest discriminant function value. Each point is classified on the basis of its spectral measurements alone. One premise of this technique is that the objects of interest are large in comparison to the size of a point. If this were not so, a large proportion of points would be composites of several classes, making statistical pattern classification unreliable since pre-specified categories would be inadequate to describe actual states of nature. From this premise it follows that objects are represented by arrays of points, and that a statistical dependence exists between consecutive points. Contemporary classifiers fail to exploit the statistical dependence between adjacent points when assigning classes.

The ECHO processors benefit from spatial information by first aggregating into groups points whose spectral responses are not significantly different in a statistical sense, and then applying a maximum likelihood classification rule to these homogeneous groups. Homogeneous objects are identified in a three step process. First, cells are formed by systematically partitioning the data into N by N sized blocks of pixels. The statistics of each cell are then compared to a homogeneity criterion. Points which do not comprise homogeneous groups are classified on a point-by-point basis, just as contemporary classifiers classify all points. Statistics of adjoining homogeneous cells are then compared to annexation thresholds. Adjoining cells which appear to belong to the same statistical population are combined into a single object.

Two separate ECHO algorithms have been developed. The first, Supervised ECHO, makes use of pre-specified class statistics to identify homogeneous objects. The second, Nonsupervised ECHO, identifies homogeneous objects without the use of class statistics. Consequently, those objects identified by the Nonsupervised algorithm may be used to aid in the training process.

Past Work at LARS

Much of the background research on the ECHO concept was performed at LARS during 1975 and is documented in the Final Report for 1975[1], R. L. Kettig's doctoral thesis[2], a LARS Information Note[3], and in symposium proceedings[4].

---

*ECHO stands for Extraction and Classification of Homogeneous Objects.

The work related to the development, testing, and documentation of the Supervised ECHO algorithms is documented in Volume I of the Final Technical Report on NASA Contract NAS9-14970, June 1, 1976 - May 31, 1977 [5]. Included in that report are a more detailed background on the past development work on ECHO at LARS, an appendix containing program listings and documentation for the Supervised ECHO algorithms, the results of systematic tests of the Supervised processor on MSS data for agricultural regions as observed by the Landsat satellites, aircraft scanners, and simulated Thematic mapper data, and an example product of an object map enabling the determination of the utility of object maps to a LACIE Analyst Interpreter in the selection and labeling of training fields.

Objectives

The objectives for the contract extension were to:

* Complete the Fortran implementation and documentation of the Nonsupervised ECHO (Extraction and Classification of Homogeneous Objects), making the algorithms available for use by JSC.

* Classify, at different parameter settings, Landsat, simulated Thematic Mapper and aircraft data sets. Compare the resulting CPU time required, full field performance, field center pixel performance and classification variability of results with those achieved by the perpoint and Supervised ECHO processors.

* Evaluate the objects identified by the Nonsupervised algorithms in terms of two types of errors:

    1. more than one field on the ground being identified by ECHO as a single object, and

    2. a single field on the ground being subdivided into distinct objects.

DESCRIPTION OF WORK

Produce Documented Fortran Programs

Fortran listings and program abstracts for the Nonsupervised ECHO processor are presented in Appendix A. This processor performs field extraction without the benefit of class statistics. Statistics are necessary for classification of the objects identified, however. The software is designed to function in a two phase mode. The first phase is the Nonsupervised Field Extraction Algorithm. This phase of the processor partitions the data to be classified into a set of points and homogeneous fields and calculates the channel means and covariance matrix for each homogeneous field which is identified.

The Nonsupervised processor utilizes a homogeneity test which compares σ_{ij}/μ_{ij} to a threshold t_i; where i refers to feature or channel and j refers to the cell. If the standard deviation divided by the mean for feature i exceeds the user-specified threshold t_i, the cell is "singular" and elements of the cell will be classified on a point-by-point basis. Adjoining homogeneous cells are annexed to fields on the basis of a two-step test, 1) that the channel variances of the field are equivalent to the channel variances of the cell and 2) that the channel means of the field are equivalent to the channel means of the cell. Should either of these criteria not be met for any channel, annexation will not take place.

The research software was designed to run in two phases. The first phase merely calculated the cell mean and covariance matrices and wrote them on tape; the second phase proceeded to perform field extraction followed by classification. This process has two disadvantages. First, although an object map could be produced, it had to be produced in phase two, the same phase that required a class statistics deck for input in order to perform the classification. Second, since only cell mean and covariance matrices were written on the intermediate tape, when singular cells were encountered in phase two, they had to be classified as small samples, a cell at a time, rather than on a point-by-point basis, because data values for invididual points were not available to the phase two classifier.

The Nonsupervised ECHO software was restructured to move the field extraction algorithm into phase one of the processor and produce an intermediate tape containing:

* class means and covariance matrices for each homogeneous object identified.

* an object map containing the mean for channel i of object j in every pixel of object j and the original data values for those pixels belonging to singular cells.

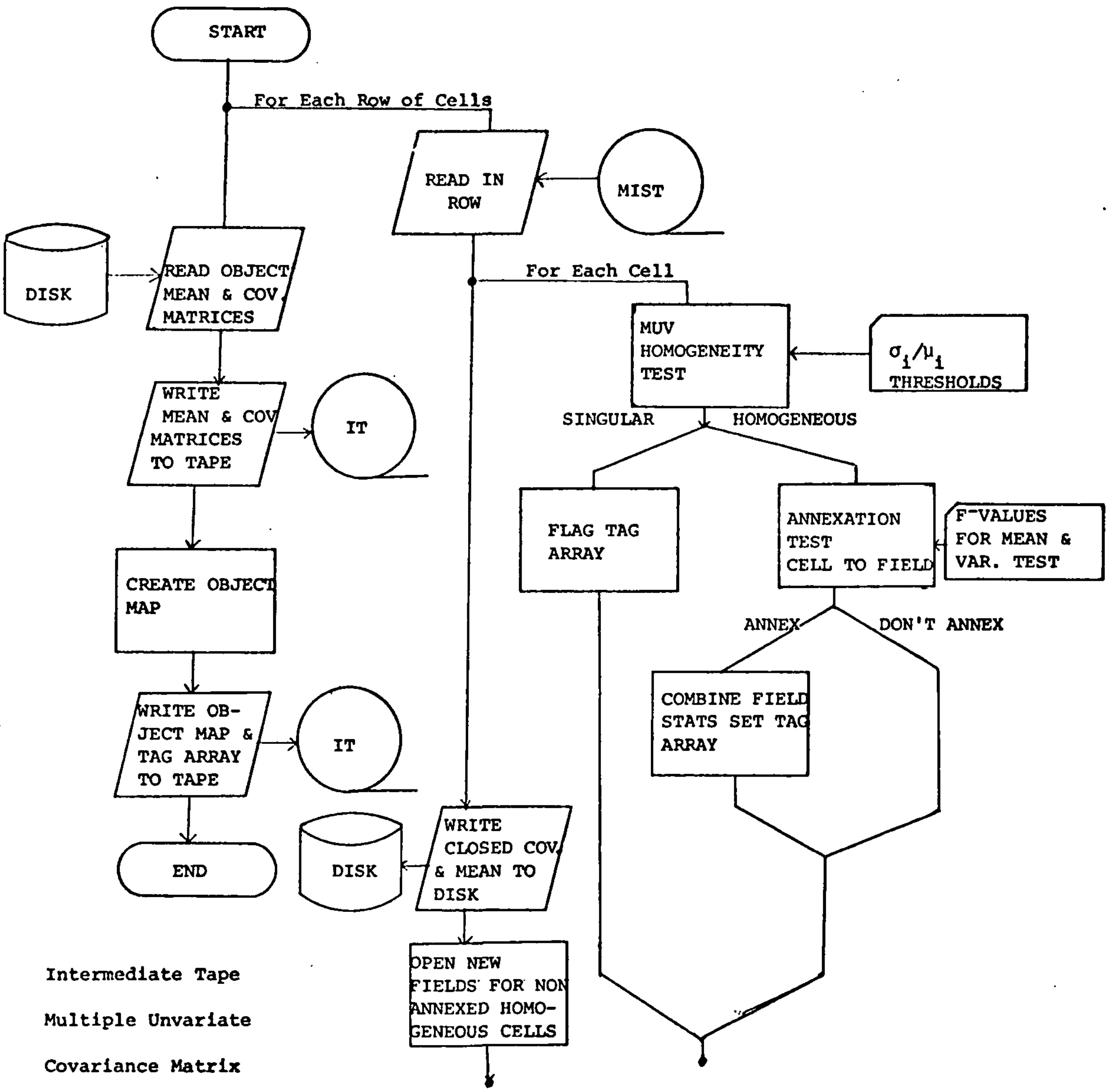
* a tag array identifying whether a pixel falls in a singular cell (and should be classified individually), or inaa homogeneous object and hence should receive a class assignment based on the sample classification of the object to which it belongs).

The second phase of the restructured Nonsupervised processor reads the intermediate tape and the class statistics deck, performs a maximum likelihood point-by-point classification on points falling in singular cells, and a maximum likelihood sample classification of the homogeneous objects identified. Figure 2a-1 presents the general processing flow for phase one (field extraction) of the Nonsupervised processor. The general processing flow for phase two (classification) is presented in Figure 2a-2. Inputs to outputs from the Nonsupervised ECHO processor's two phases are described in Table 2a-1.

The formats of the disk and tape files which support the Nonsupervised ECHO processors are presented in Appendix B. User documentation is available in the forms of an ECHO User's Guide[6] and an Echo Case Study[7].

Figure 2a-1

GENERAL FLOW OF
NONSUPERVISED ECHO
PHASE 1



Intermediate Tape

Multiple Unvariate

Covariance Matrix

Figure 2a-2

GENERAL FLOW OF

NONSUPERVISED ECHO

PHASE 2

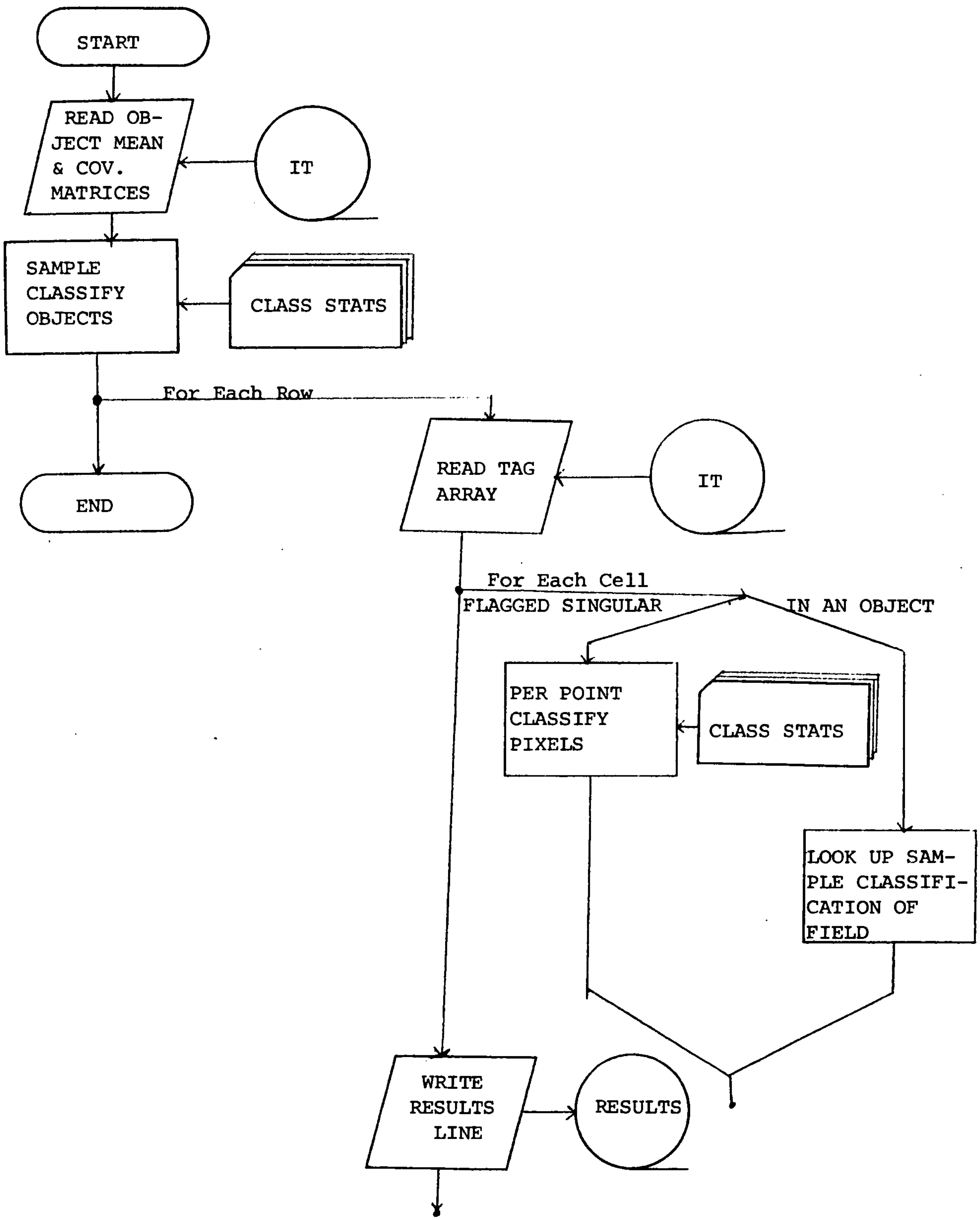


Table 2a-1

NONSUPERVISED ECHO

PROCESSOR

INPUT	OUTPUT
Phase One:	
Channel Selection	Intermediate Tape containing object map, object statistics and pixel tag array.
Cell Width	
Cell Homogeneity Thresholds (σ_i/μ_i for channel i)	
Annexation Thresholds (mean and variance)	
Multispectral Image Storage Tape	
Intermediate Results Tape and File Specification	
Phase Two:	
Class Statistics	Results File
Intermediate Tape from Phase One	
Specification of Results Tape and File	

Test the Nonsupervised ECHO Algorithms

1. Data Sets

The second objective of the ECHO Extension work is to test the Non-supervised ECHO algorithms on MSS data for agricultural regions. Data sets are to include Landsat, aircraft, and simulated Thematic Mapper data. Nine Landsat, one aircraft and six simulated Thematic Mapper data sets (form resolutions from one site and two resolutions from a second site) were selected for analysis. The data sets are summarized in Table 2a .

Table 2a-2

Data Sets

(LANDSAT)

CHANNEL NUMBER	WAVE BAND µM
1,5,9	.5-.6
2,6,10	.6-.7
3,7,11	.7-.8
4,8,12	.8-1.1

Area	Channles Used	Data Collected	Data Set
Graham County, Kansas	9, 10, 11, 12	5/26/74	LACIE/SRS
Grant County, Kansas	5, 6, 7, 8	5/19/74	LACIE/SRS
Haskell County, Kansas	9, 10, 11, 12	5/27/74	LACIE/SRS
Kearny County, Kansas	9, 10, 11, 12	5/27/74	LACIE/SRS
Huntington County, Indiana	1, 2, 3, 4	7/7/73	CITARS
Shelby County, Indiana	1, 2, 3, 4	9/7/73	CITARS
White County, Indiana	1, 2, 3, 4	8/21/73	CITARS
Fayette County, Illinois	1, 2, 3, 4	8/21/73	CITARS
Lee County, Illinois	1, 2, 3, 4	8/5/73	CITARS

Table 2a-2 (Continued)

Simulated Thematic Mapper

TM CHANNEL NUMBER	WAVE BAND µM
1	.45-.52
2	.52-.60
3	.63-.69
4	.74-.80
5	.80-.91
6	1.55-1.75
7	10.4-12.5
8	.74-.91

Area	Channels Used	Date	Resolution
Williams County, ND	1, 2, 3, 6, 8	8/15/75	30m
Williams County, ND	1, 2, 3, 6, 8	8/15/75	40m
Finney County, Kansas	2, 3, 5, 6, 7	7/6/75	30m
Finney County, Kansas	2, 3, 5, 6, 7	7/6/75	40m
Finney County, Kansas	2, 3, 5, 6, 7	7/6/75	50m
Finney County, Kansas	2, 3, 5, 6, 7	7/6/75	60m

Table 2a-2 (Continued)

Aircraft

Area	Wavebands Used (μM)	Data Collected
Finney County, KS	.4-.49, .59-.64, .65-.69, .82-.88, 1.53-1.62, 10.1-11.0	7/6/75

2. Training Procedures

a. Landsat Data Sets

The LACIE training sets were created using ground truth information provided by JSC for both test and training fields. Odd-numbered fields appearing in each subclass were used for training; even-numbered fields were used for test. For Graham and Grant counties, statistics were generated by using the STATISTICS processor. For Haskell and Kearny counties statistics were secured by clustering the training fields of each class into four subclasses which were then pooled on the basis of output from the SEPARABILITY processor. It should be noted that both training and test fields for the LACIE/SRS data are large enough to inset field boundaries approximately two pixels inside estimated field boundaries. This inset allows for any image misregistration which may occur between any two dates on the multitemporal input runs. Proportion estimate for the 1974 LACIE/SRS segments were available in ground truth packets provided by JSC.

The CITARS training sets were originally created by a supervised procedure using ground truth information provided by the Agricultural Stabilization and Conservation Services (ASCS) to select both the training and test fields|8|. Those training sets were used without change in the ECHO tests.

Five counties in the CITARS experiment were used as test sites for the Nonsupervised processor evaluation. A data set free of clouds which occured late in the growing season was required for each of the five counties. Dates from July, August or early September were selected. The classification results for all of the CITARS experiments are catalogued on a series of LARS tapes. The desired data sets were located on the catalogued CITARS tapes and the statistics which had been used for the CITARS experiment were obtained by using the LARSYS PUNCHSTATISTICS processor. The pooling of classes was determined by running the PRINTRESULTS processor and requesting training field results. This combination of requests produced a table of available classes in the statistics deck and also the informational names under which they were classified. By using this list it was possible to reconstruct the combinations of spectral classes and pooled spectral classes which were needed to reproduce the original classification.

Training and test fields for each CITARS classification were secured and appropriate control cards added to evaluate the ECHO classifications of CITARS data sets. The proportion estimates used for the CITARS evaluation are estimates of the proportions of the various classes for the entire county made by the SRS.

b. Simulated Thematic Mapper Data Sets

Training sets from the simulated Thematic Mapper tests performed at LARS in 1976 were used for ECHO analysis. These training sets were generated by selecting fields of known cover types and clustering each informational class separately to define subclasses.

Color infrared photographic mosaic prints were made from photographic data collected concurrently with the scanner data. Informational class information provided by ground observations was transferred to clear plastic overlays on the mosaic print. The analyst could then easily locate the corresponding fields in his cluster maps and assign the field coordinates to the informational classes.

Statistics were calculated for each training area and compared using the SEPARABILITY processor. Similar classes were combined, where indicated and the data set was used to classify the flightline. Training areas were not excluded from the test fields since the test fields had been pre-selected for the entire flightline.

Two to four subclasses were found in each informational class. The Kansas flight was an exception. Because of severe line-to-line changes in signal level in the original 6 meter scanner data, the analyst was forced to create more spectral classes to account for the within-class variations due to excessive noise. This was most apparent in the 30 and 40 meter resolution data. The effect was reduced but not eliminated in the 50 and 60 meter resolution data. Alternate fields were used for training and test decks, respectively. The procedure was repeated for each of the four resolution sizes. As resolution size increases, the number of spectral classes decreases.

The entire training set selection procedure was repeated for each resolution size so that any effects on training set selection which might be caused by data resolution would be included in the analysis results. An example is the increasing difficulty and eventual impossibility of selecting samples from small, or narrow, fields as the resolution size increases.

c. Aircraft

A 6-meter aircraft scanner data set used to generate the Finney County, Kansas simulated Thematic Mapper data set was used. The same training and test fields used for the Thematic Mapper were available in the six meter data set. Because of the very large number of data points in the full data set, only the first two miles (one third of the total flight line) were used for the Nonsupervised ECHO evaluation.

The data set was not corrected for sun and scanner angle effects. To compensate for these angle effects, training fields were distributed across the width of the flight line. At least six fields in each informational class were used in the training set. Fields from informational classes were clustered together into five spectral classes. All spectral classes from all informational classes were combined into a statistics deck and appropriate pooling was done based on SEPARABILITY results. The classification results were strongly influenced by angle effects.

3. Dependent Variables

There are six variables which were monitored to evaluate the ECHO algorithm.

- CPU time
- Field center pixel classification performance
- Training field classification performance
- Full field classification performance
- RMS proportion estimate error
- Classification variability

These variables are related to reasons for adopting a new classification technique: cost, accuracy, and usability of results. The CPU time required to perform a classification is one way to measure the cost of classification. Field center pixel, full field, and training field performances and RMS proportion estimate error are all wasy to evaluate the accuracy of the classifier. Classification variability is a measure of "salt and pepper effect" in classification results.

The CPU time required to execute each of the ECHO classifications has been recorded so that the effects of varying the cell homogeneity and annexation thresholds may be monitored. The CPU time required to perform the perpoint classifications have been adjusted to reflect the increased efficiency of the LARSYS perpoint classifier which is coded in assembly language. Thus, the CPU time recorded for a perpoint classification is what a FORTRAN classifier would have required to perform the classification.

The indices of classification performance were applies in several ways. Classification accuracy (identification) was evaluated utilizing field center pixel, "full field" and test field sample performances for all data sets. Proportion estimation was carried out for the Landsat and Simulated Thematic Mapper data sets.

The training performance is the overall classification accuracy (number of training pixels correctly classified divided by the total number of training pixels) of the pixels used to calculate the class statistics. Field center pixel performance is the overall classification accuracy of pixels inset at least one pixel from the field boundary. For the registered LACIE/SRS data the field center pixels are inset at least two pixels from the field boundary. Although this procedure insures that the pixels examined are not mixture pixels, it has the unfortunate effect of eliminating smaller fields from consideration. The third measure of classification accuracy, "full field" performance, includes those pixels on the boundaries of the fields in the classification performance. The "full field" pixels were generated by expanding the field center pixel boundaries one pixel in all directions.

The RMS error of informational class proportion estimates for each flightline was found by calculating the percent of the flightline classified as a particular class and comparing it with the ground-collected estimage using equation (1).

$$\text{RMS Error} = \sqrt{\frac{\sum_{i=1}^{N} (C_i - C'_i)^2}{N}} \qquad (1)$$

where, N = number of informational classes,
C_i= percent classified as informational class i, and
C'_i= percent of class i estimated from ground-collected data.

RMS error is calculated for the Landsat and Thematic Mapper data runs. The Agricultural Stabilization and Conservation Service (ASCS) provided the ground truth proportion estimates for the simulated Thematic Mapper data set. Proportion estimates for the 1974 LACIE/SRS segments were provided in ground truth packets received from JSC. The SRS county proportion estimates were used to calculate RMS proportion error for the CITARS data set.

Average variability is a measure of the rate of change from one information class to another. It should reflect the degree to which ECHO reduces the "salt and pepper effect" which is sometimes present in per-point classifications. Variability is calculated by systematically selecting 50 lines of the classified area, counting the number of information class changes, and dividing by the number of opportunities for class changes.

Variability - NCC/(50*(NS-1)) (2)

Where:

NCC = the number of class changes over the 50 selected lines, and
NS = the number of classified pixels/lines.

4. Results

This section outlines the results of the tests performed on the Nonsupervised ECHO classifications and the comparison of the Nonsupervised ECHO processor with the Supervised ECHO and the perpoint classifiers. Results are discussed separately for each scanner type, Landsat, simulated Thematic Mapper, and aircraft.

For all data sets considered, the Nonsupervised ECHO processor was run with the following parameter settings:

1. Cell width of 2.

2. Cell homogeneity thresholds of 0.05, 0.10, and 0.25.

3. Both the mean and the covariance annexation thresholds set at 0.001, 0.01, and 0.1.

a. Landsat Results

Training and test information for the nine Landsat test data sets were drawn from the 1974 Kansas LACIE/SRS data sets and the 1973 CITARS data sets. Four LACIE/SRS and five CITARS data sets were considered.

i LACIE/SRS Results

Figures 2a-3 through 2a-8 present the average results for the four LACIE/SRS sites examined at a cell width of two for three homogeneity and three annexation parameters. Figure 2a-3 plots the average CPU time in seconds required by the perpoint classifier (represented by the line of 'P's) versus the average CPU time required by the Nonsupervised ECHO routine to classify the four LACIE/SRS data sets at each of nine Nonsupervised ECHO parameter settings for 2 by 2 pixel cells. The cell homogeneity threshold is plotted along the horizontal axis. As this threshold increases, it becomes more likely a cell will be classified as a unit, less likely that a cell will be split and its constituent pixels classified individually. The dependent variable, CPU time, appears on the vertical axis. The cell-to-field annexation parameter for each cell homogeneity threshold is represented on the plot. A '1' appears in the position for the results achieved when both the mean and covariance annexation thresholds are set to 10^{-1}; a '2' when they are set to 10^{-2}; and a '3' when they are set to 10^{-3}. As the annexation thresholds become smaller, the likelihood that adjoining homogeneous cells will be annexed into a single field increases. When two or more annexation thresholds achieve the same performance, a star appears in the position on the plot.

The statistical significance of the effects of the Nonsupervised ECHO parameters on the LACIE/SRS results are presented in Table 2a-3.

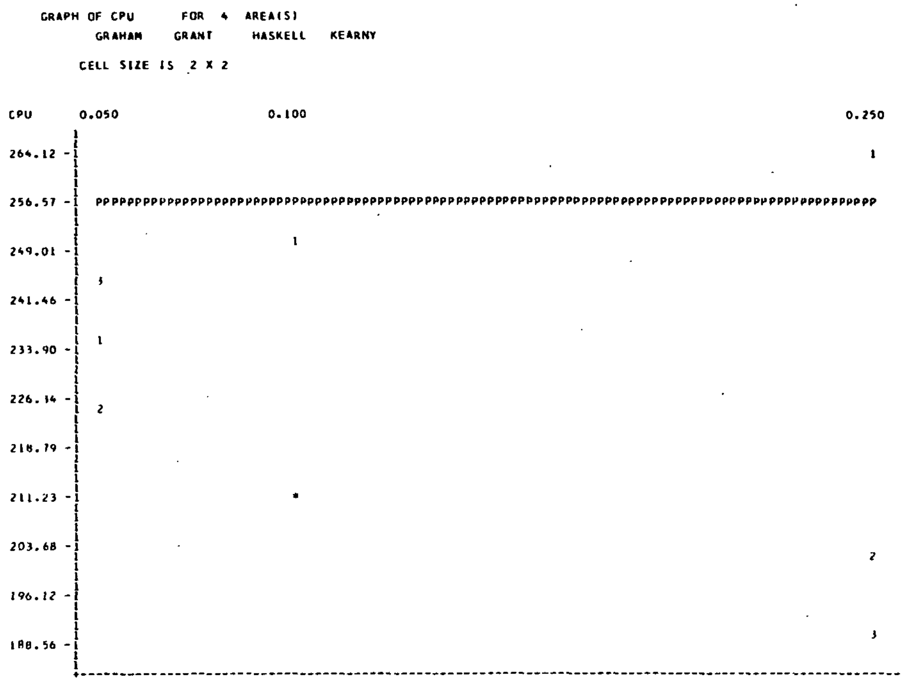


Figure 2a-3

Nonsupervised ECHO CPU Requirements
for the LACIE/SRS Data Sets

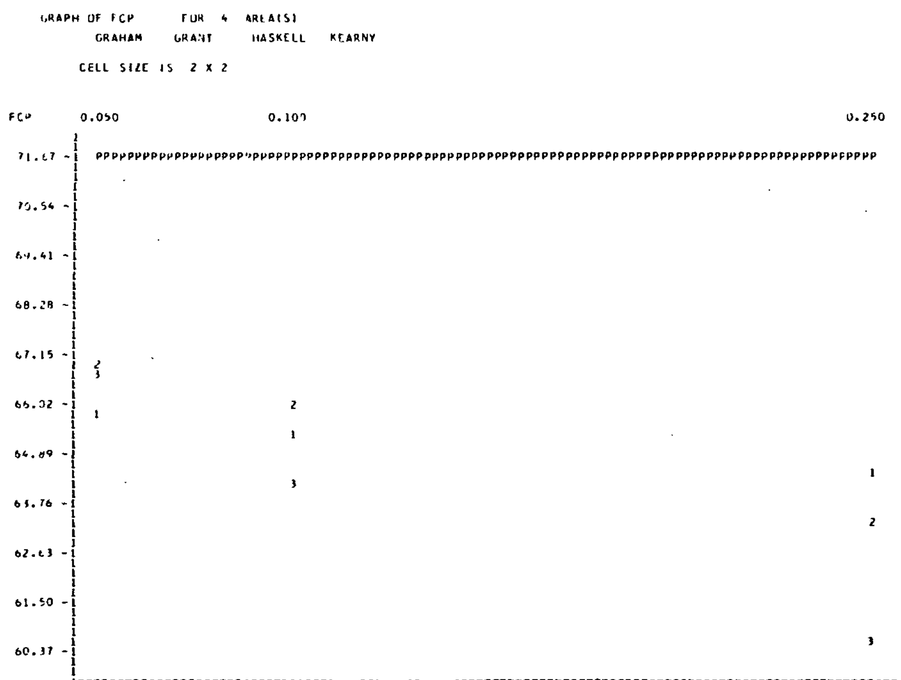


Figure 2a-4

Nonsupervised ECHO Field Center Pixel Performance
for the LACIE/SRS Data Sets

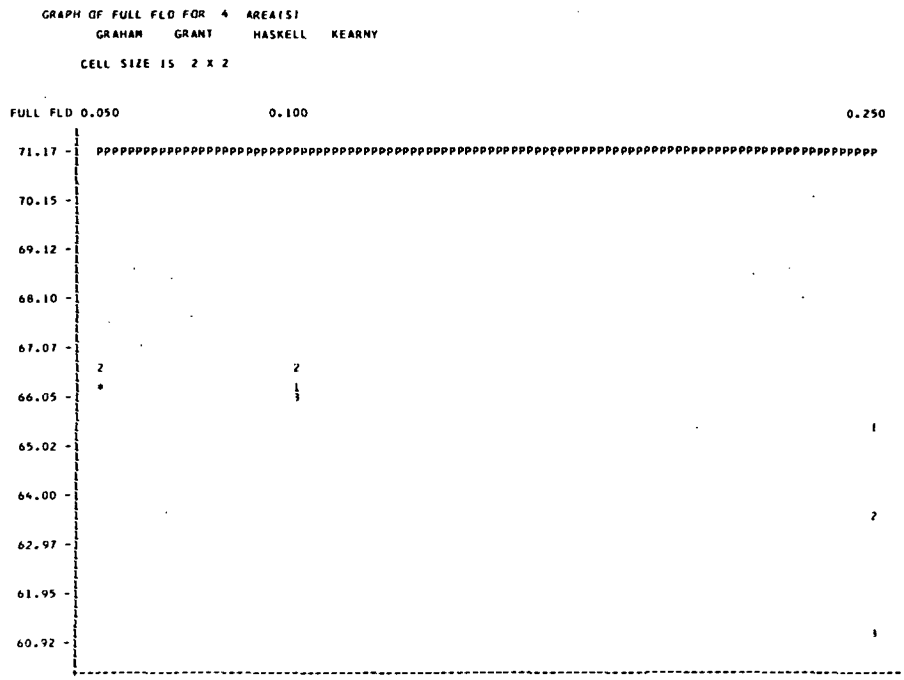


Figure 2a-5

Nonsupervised ECHO Full Field Performance
for the LACIE/SRS Data Sets

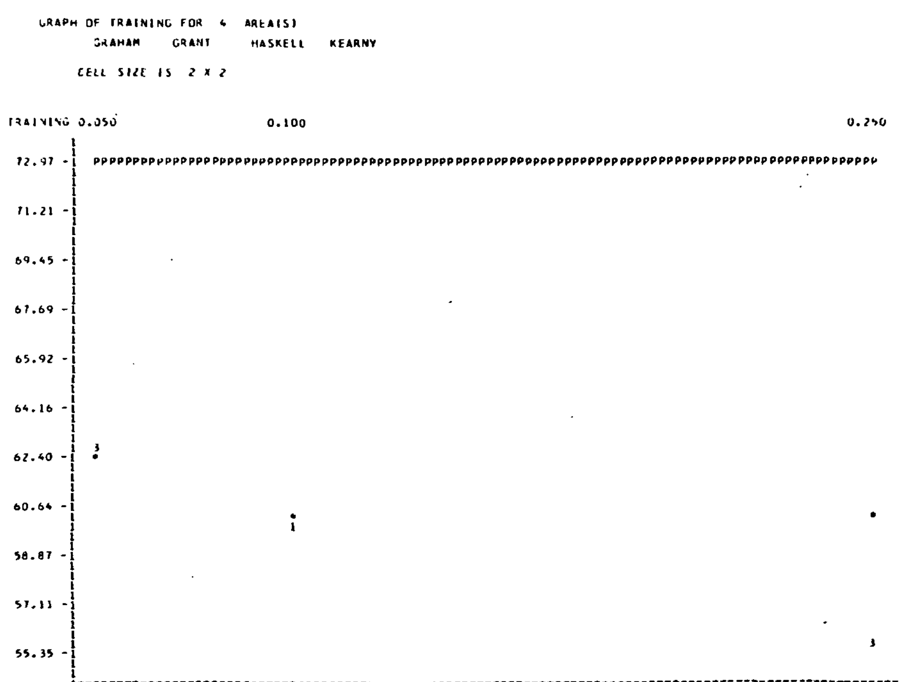


Figure 2a-6

Nonsupervised ECHO Training Field Performance
for the LACIE/SRS Data Sets

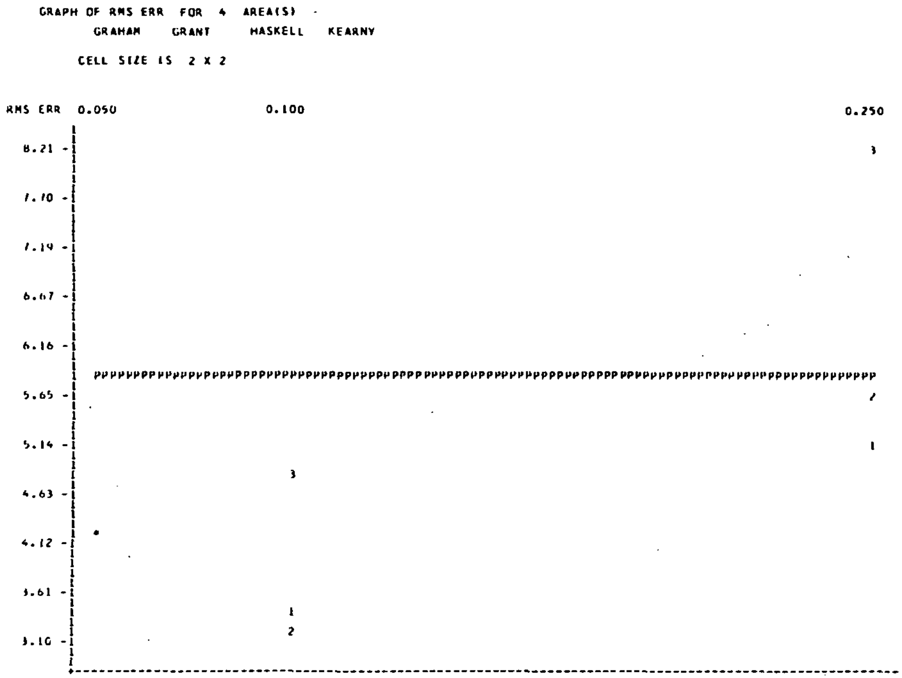


Figure 2a-7

Nonsupervised ECHO RMS Proportion Estimate Error
for the LACIE/SRS Data Sets

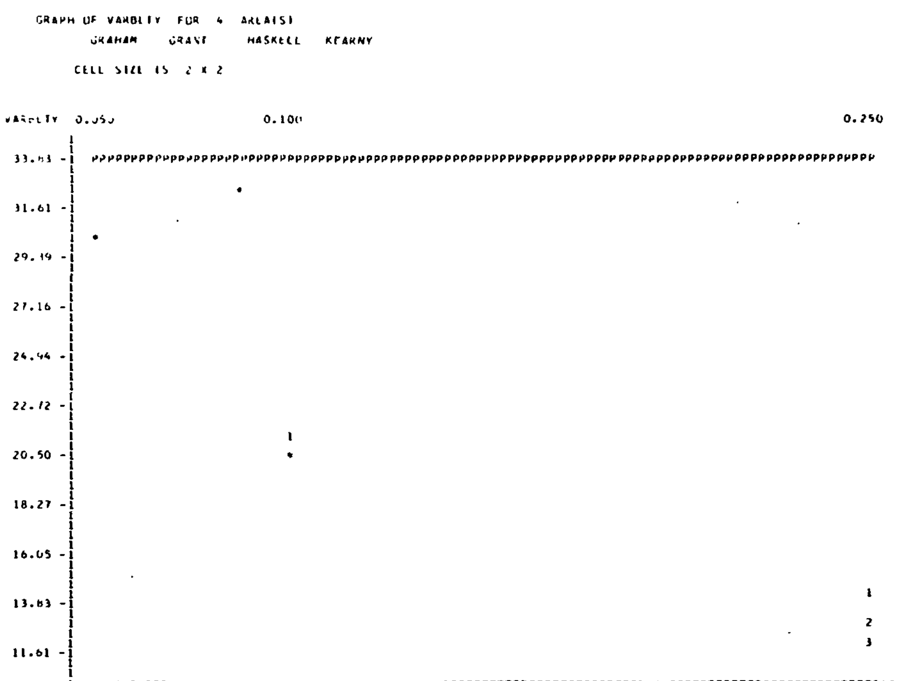


Figure 2a-8

Nonsupervised ECHO Classification Variability
for the LACIE/SRS Data Sets

Table 2a-3

Overall Landsat Comparisons
Effects of Parameters on LACIE Data

Variable	Homogeneity Thresholds	Annexation Thresholds	Homogeneity X Annexation
Degrees of Freedom	2,6	2,6	4,12
CPU Time	1.86	6.69*	21.37***
Field Center Pixel Performance	1.27	7.02*	1.90
Full Field Performance	1.31	11.09**	3.74*
Training Field Performance	.70	3.60+	5.89**
RMS Proportion Error	.86	4.03+	1.18
Classification Variability	13.64**	14.39**	12.59***

Significance Levels

+	10%
*	5%
**	1%
***	.1%

Table entries are F-values

It can be seen from Figure 2a-3 that, as the cell homogeneity parameter increases from .05 to .25, the CPU time required is effected by the interaction between the homogeneity and the annexation threshold. This interaction is probably the result of two separate influences. The LACIE/SRS, CITARS, and aircraft data runs of the Nonsupervised processor were produced utilizing the "MAP" option. When this option is specified, the data value for each point which falls within a field is replaced by the mean value for the field. This replacement requires a significant amount of computer time. Therefore, as more fields are identified, more CPU time will be used. As the homogeneity parameter increases, there is a tendency for more cells to be identified as homogeneous, and therefore, it is possible for more fields to be identified. The effect of this is reflected by the increase in CPU time of the runs with a 0.1 annexation parameter. However, with annexation parameters of 0.01 and 0.001, adjoining homogeneous cells are frequently annexed into fields reducing the number of passes through the classification equation and reducing the amount of computer time required.

Figure 2a-4 graphs the field center pixel performance for the Nonsupervised ECHO classifier. The only significant effect is that as the annexation parameter goes from 10^{-1} to 10^{-3}, the field center pixel performance declines.

For training field and full field performance measures presented in Figures 2a-5 and 2a-6, as the annexation threshold becomes smaller the performance decreases and as the homogeneity value becomes larger the effects of the annexation parameters are increased.

Parameter selection seems to have an only slightly significant effect on the RMS proportion estimate error (Figure 2a-7). As the annexation parameter becomes smaller, the RMS error tends to become larger.

As Figure 2a-8 indicates, classification variability is strongly influenced by both the annexation and the homogeneity parameters and by their interaction.

ii CITARS Results

Figures 2a-9 through 2a-14 present plots for the 2 by 2 cell size Nonsupervised ECHO results achieved over the five CITARS data sets. Characteristic differences between the CITARS and the LACIE/SRS data sets include:

* The CITARS data set has a much smaller field size than the LACIE/SRS set.

* The information classes are different. CITARS information classes are corn, soybeans, and other; LACIE/SRS classes are wheat and other.

* the ground truth proportion estimates for the LACIE/SRS sites were for the area of the LACIE segment. The ground truth proportion estimates for the CITARS sites are for the whole county in which the data set lies, not for the area of the county which was actually sampled. Analysis of various results for the five CITARS data sets are presented in Table 2a-4.

Figure 2a-9 indicates that the CITARS data has the same interaction of annexation and homogeneity parameters as the LACIE/SRS data sets (Figure 2a-3) had with respect to CPU time required. This interaction is due to:

* the additional CPU time required to replace pixel values with field means as additional fields are identified, and

* the reduction of CPU time required by the classifier as larger fields are identified, resulting in fewer passes through the classification equation.

As the annexation thresholds go from 10^{-1} to 10^{-3}, the CPU time required to perform the Nonsupervised ECHO classification is reduced. For the CITARS data, as the homogeneity parameter increases, the decrease in average CPU time required is statistically significant.

There are no statistically significant parameter effects on the CITARS field center pixel or full field performance measures. Training field performance decreases as the annexation thresholds are decreased from 10^{-1} to 10^{-3} (Figure 2a-12). As the homogeneity parameter increases, the effect of the annexation parameters is increased for training field performance, RMS proportion error and classification variability measures.

As the homogeneity parameter increases, RMS proportion estimate error increases (Figure 2a-13) and classification variability decreases (Figure 2a-14). As the annexation thresholds decrease, the training field performance decreases, the RMS proportion error increases, and the classification variability decreases.

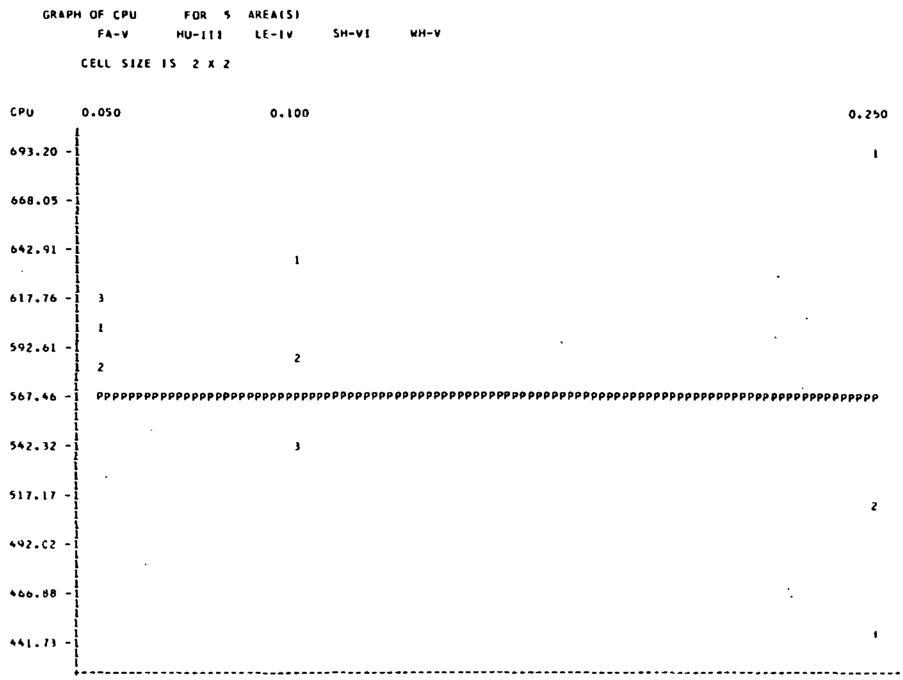


Figure 2a-9

Nonsupervised ECHO CPU Requirements
for the CITARS Data Sets

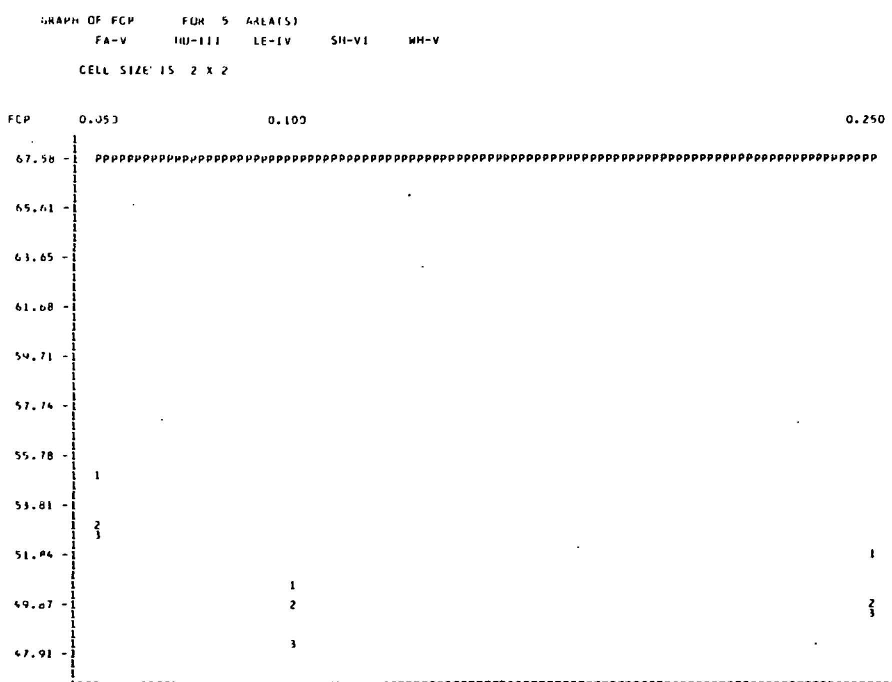


Figure 2a-10

Nonsupervised ECHO Field Center Pixel Performance
for the CITARS Data Sets

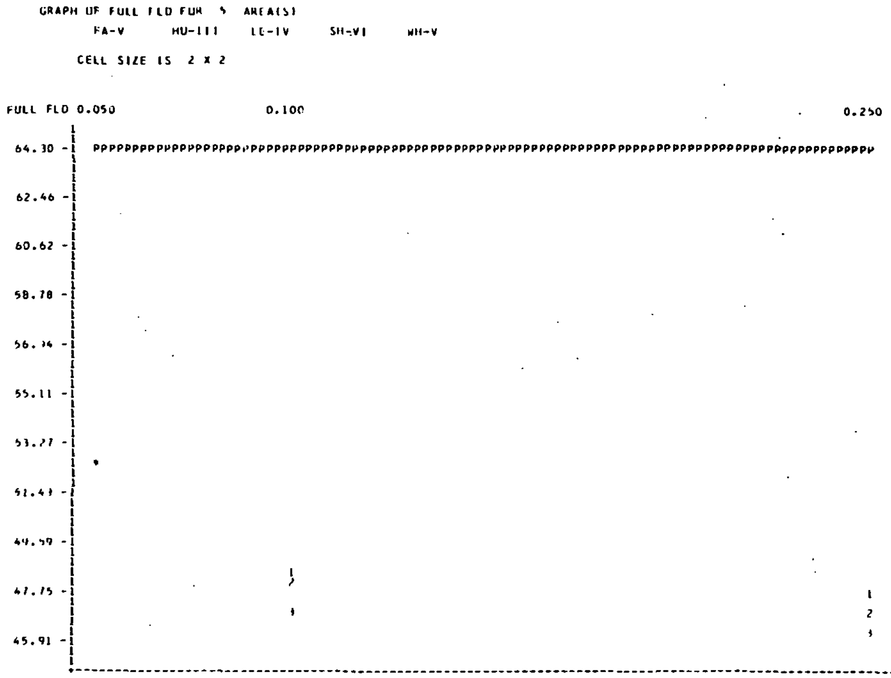


Figure 2a-11

Nonsupervised ECHO Full Field Performance for the CITARS Data Sets

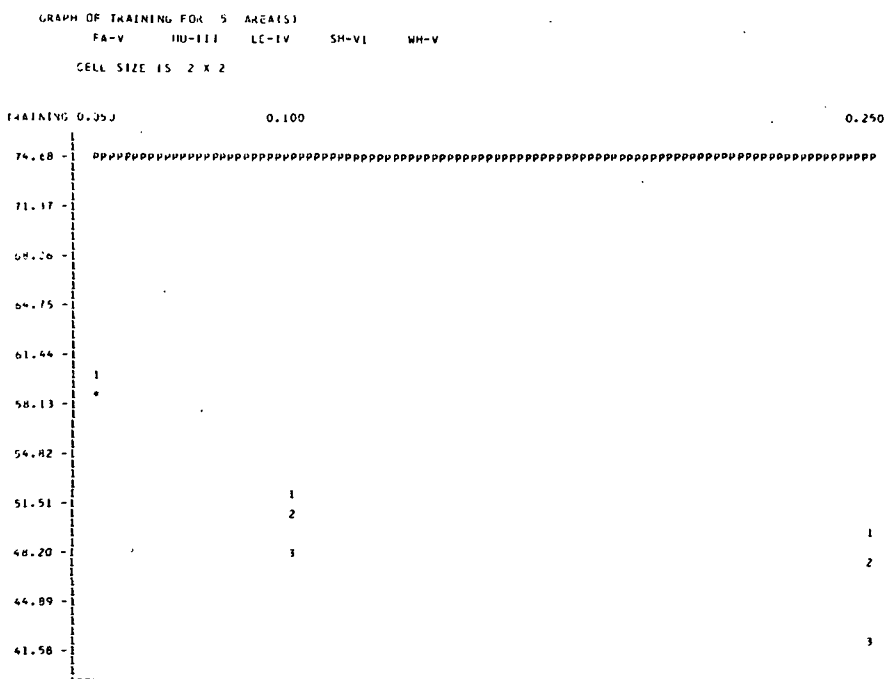


Figure 2a-12

Nonsupervised ECHO Training Field Performance for the CITARS Data Sets

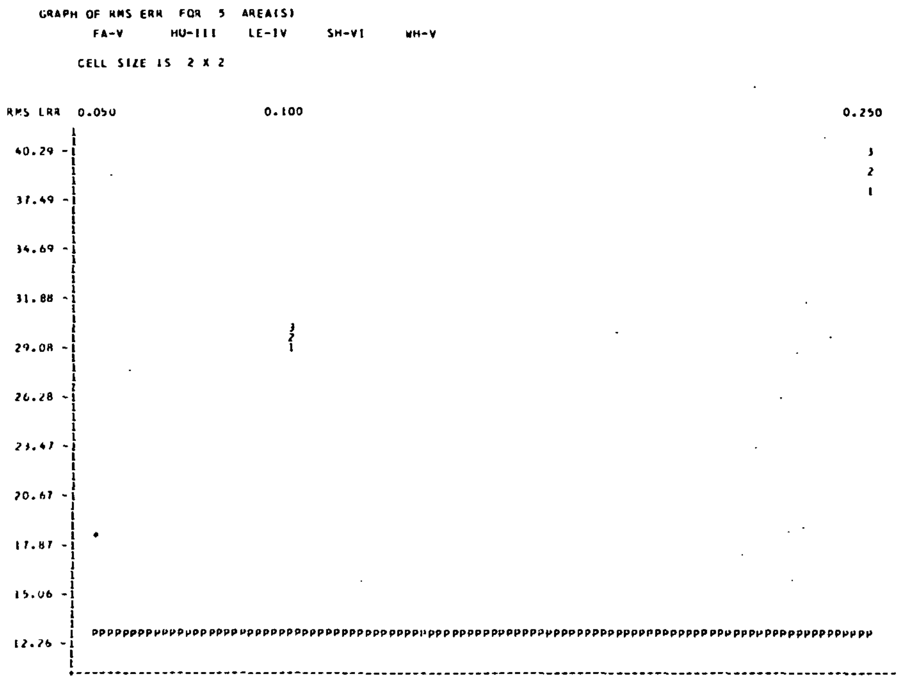


Figure 2a-13

Nonsupervised ECHO RMS Proportion Estimate Error
for the CITARS Data Sets

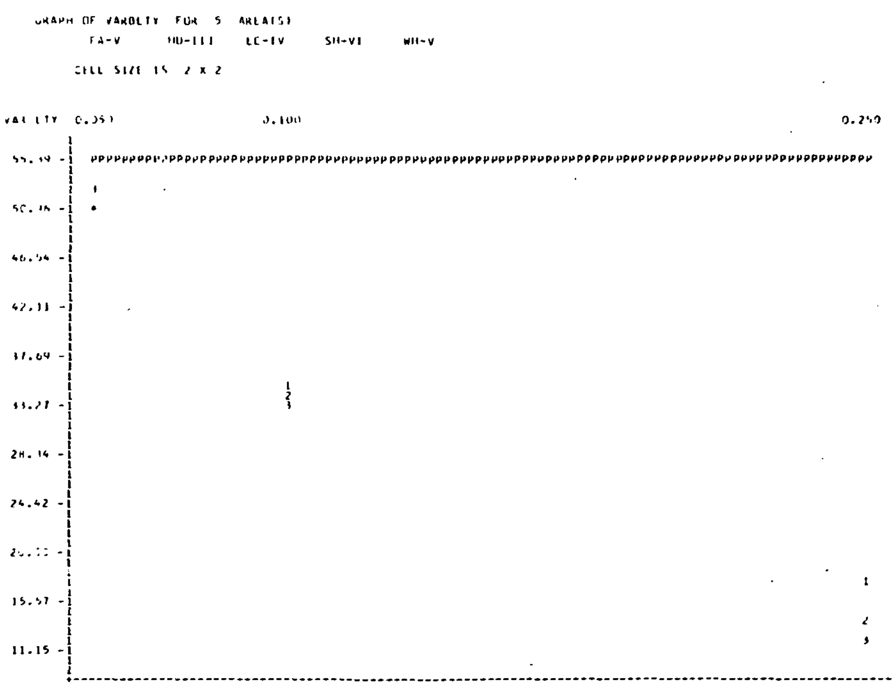


Figure 2a-14

Nonsupervised ECHO Classification Variability
for the CITARS Data Sets

Table 2a-4

Overall Landsat Comparisons
Effects of Parameters on CITARS Data

Variable	Homogeneity Thresholds	Annexation Thresholds	Homogeneity X Annexation
Degrees of Freedom	2,8	2,8	4,16
CPU Time	7.57*	13.73**	31.73***
Field Center Pixel Performance	.18	1.48	.16
Full Field Performance	.63	.84	.31
Training Field Performance	4.18+	8.62*	3.00*
RMS Proportion Error	10.29**	4.81*	4.22*
Classification Variability	158.47**	3.74+	7.81***

Significance Levels

+	10%
*	5%
**	1%
***	.1%

Table entries are F-values

In both the CITARS and the LACIE/SRS data sets, the classification performance measurements are markedly inferior to the perpoint performance over the same areas. This circumstance probably results from errors in object identification. It is of interest that while Nonsupervised ECHO performance measures for the LACIE/SRS data sets are 4 to 10 percentage points lower than the perpoint results, the average performance measures for the CITARS data sets are 10 to 30 percentage points inferior. This difference may be due to the effects of the smaller average field size for the CITARS data. The RMS proportion error for the Nonsupervised algorithm's results is superior to the perpoint algorithm results in the LACIE/SRS data, but inferior in CITARS case.

The Nonsupervised ECHO field center pixel performances are superior to the perpoint results in three of the five data sets (LE-IV, Hu-III, and Fa-V) by a few percentage points, but on the order of 17 to 20 points worse in the other two data sets (Sh-V and Wh-V). This indicates that the Nonsupervised processor has the potential to improve the classification accuracy achieved, but that it also may substantially decrease the accuracy.

iii Overall Landsat Results

Figures 2a-15 through 2a-20 summarize the effects of the ECHO parameter settings on the six dependent variables for nine Landsat data sets. Table 2a-5 summarizes the statistical significance of the effects of the ECHO parameter settings on these six dependent variables.

The homogeneity parameter has statistically significant effects on CPU time, training field performance, RMS proportion error, and classification variability. As the cell homogeneity threshold is increased, the amount of CPU time required to perform a classification decreases (Figure 2a-15). As the homogeneity parameter increases:

* the training field performance increases (Figure 2a-18),

* RMS proportion estimate error (Figure 2a-19) increases, and

* classification variability (Figure 2a-20) decreases.

Parameter settings of the annexation thresholds have a significant effect on five dependent variables, CPU time, full field performance, training field performance, RMS proportion error, and classification variability. As the annexation threshold goes from 10^{-1} to 10^{-3} (the tendency for cell-to-cell annexation increases), the amount of CPU time, the full field performance, the training field performance, and the classification variability all decrease. As the annexation thresholds decrease, the RMS proportion estimate error increases.

The interaction effects of the homogeneity and annexation parameters were significant for four of the dependent variables, CPU time, training field performance, RMS proportion error, and classification variability. For these four variables, as the cell homogeneity parameter was increased, the effects of the annexation parameters were increased. This result is expected since annexation may take place only when adjoining cells are homogeneous.

Table 2a-6 presents the significant effects of the homogeneity and annexation parameters on the six dependent variables for each of the nine individual Landsat data sets. Although the effects of the homogeneity threshold on the field center pixel and full field performances are not significant when the data sets are considered together, the effect of the homogeneity parameter is significant in seven and eight of the nine Landsat data sets, respectively, when the data sets are analyzed individually. This result indicates that the effects of the homogeneity parameter on the field center pixel and full field performances are opposite for differing data sets.

On the other hand, the effects of the homogeneity and the annexation thresholds on CPU time are not statistically significant for any Landsat data set, when the data sets are examined individually.

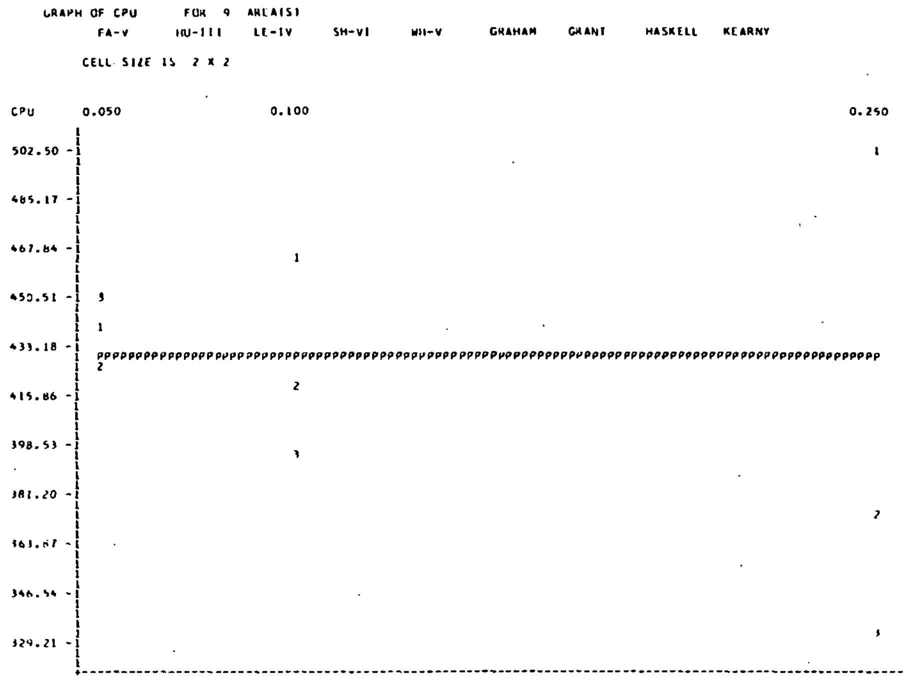


Figure 2a-15

Nonsupervised ECHO CPU Requirements
for all Landsat Data Sets

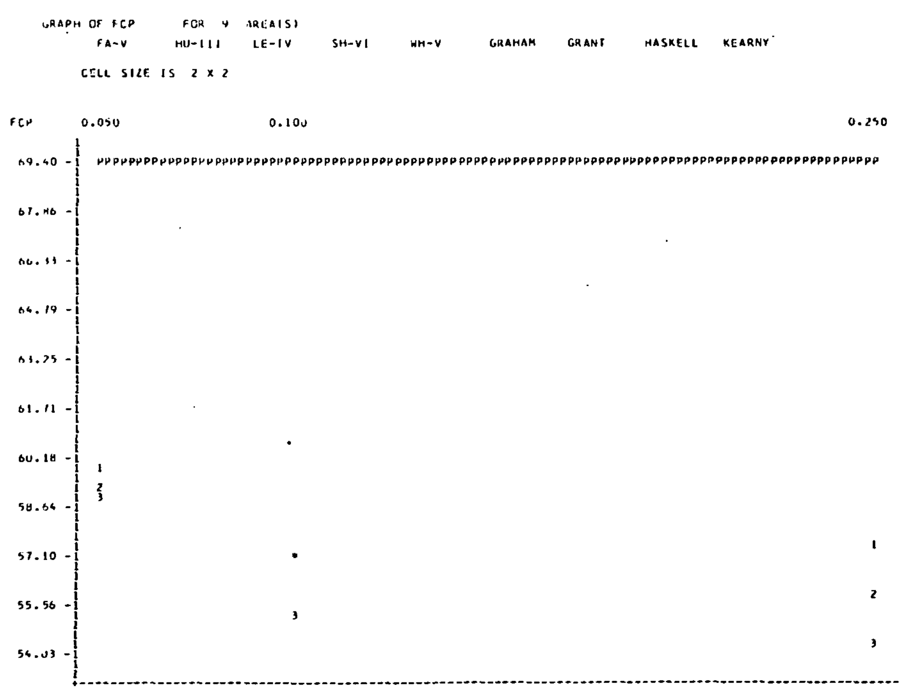


Figure 2a-16

Nonsupervised ECHO Field Center Pixel Performance
for all Landsat Data Sets

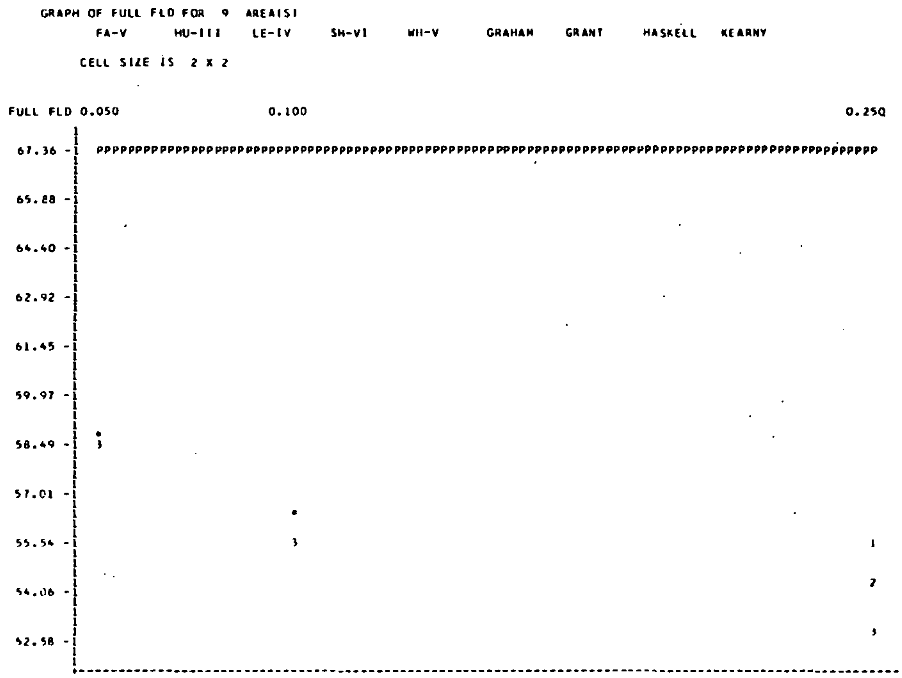


Figure 2a-17

Nonsupervised ECHO Full Field Performance for all Landsat Data Sets

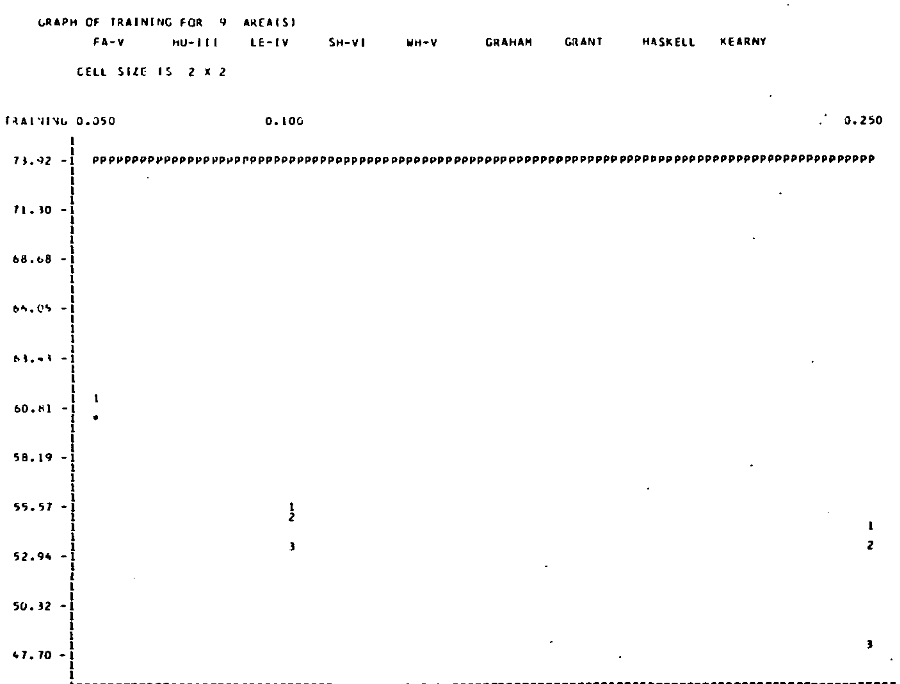


Figure 2a-18

Nonsupervised ECHO Training Field Performance for all Landsat Data Sets

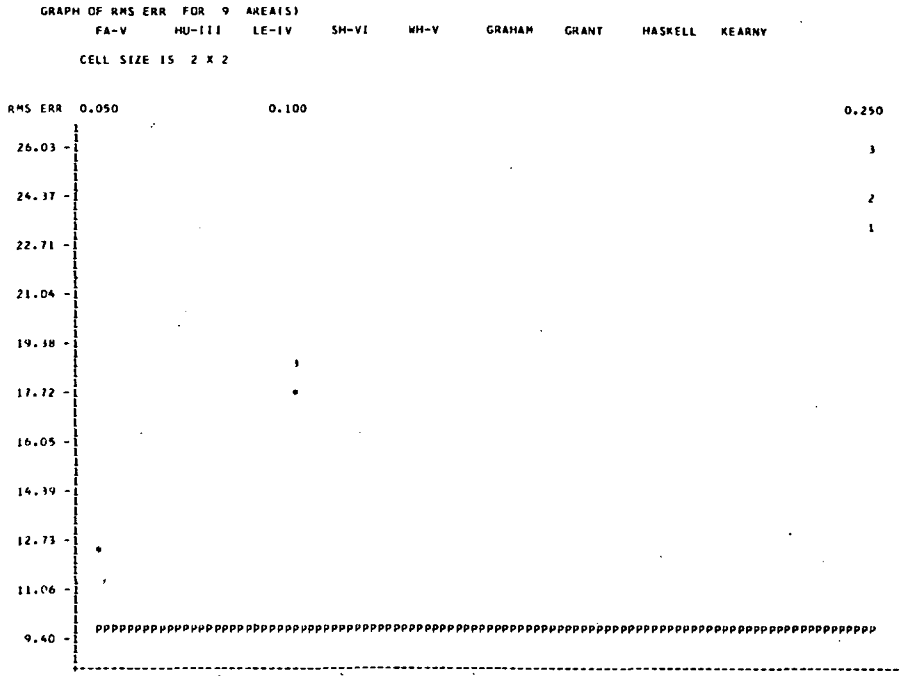


Figure 2a-19

Nonsupervised ECHO RMS Proportion Estimate Error for all Landsat Data Sets

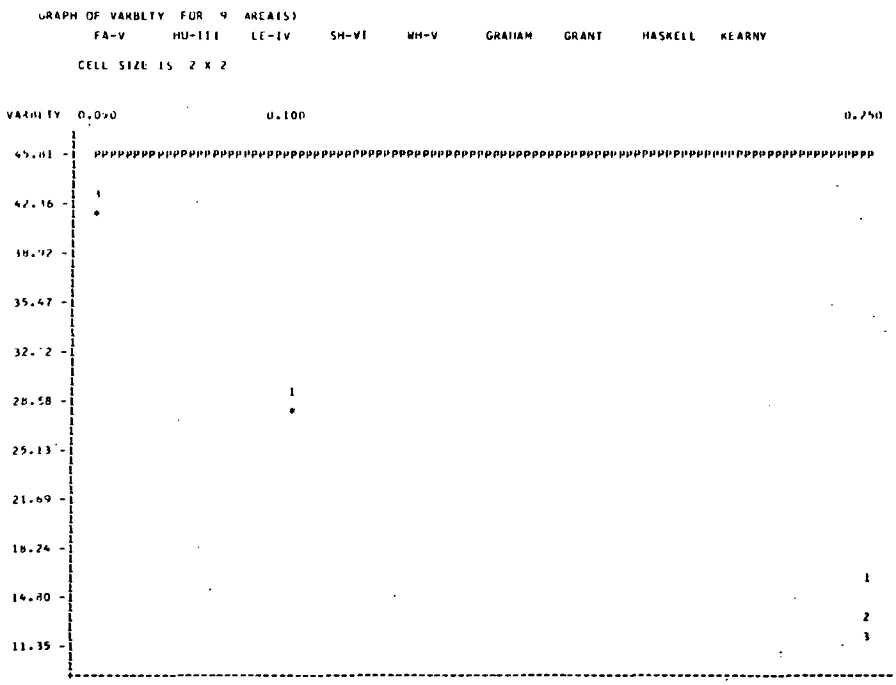


Figure 2a-20

Nonsupervised ECHO Classification Variability for all Landsat Data Sets

TABLE 2a-5

Overall Landsat Comparisons
Effects of Parameters

Variable	Homogeneity Thresholds	Annexation Thresholds	Homogeneity X Annexation
Degrees of Freedom	2,16	2,16	4,32
CPU Time	7.61**	13.89***	19.08***
Field Center Pixel Performance	.45	2.92+	.90
Full Field Performance	1.11	3.89*	2.06
Training Field Performance	4.56*	9.46**	7.54***
RMS Proportion Error	6.48**	8.26**	3.68**
Classification Variability	43.08***	8.87**	8.96***

Significance Levels

+	10%
*	5%
**	1%
***	.1%

Table entries are F-values

Table 2a-6

Results of Individual Landsat Data Set Parameter Anovas
(F Tests with 2,4 degrees of Freedom)

	CPU	FCP	FULL	TRAIN	RMS	VARI
GRAHAM						
Sell	ns	ns	10%	ns	ns	0.1%
Annl	10%	ns	ns	ns	25%	ns
GRANT						
Sell	ns	5%	5%	ns	0.1%	0.1%
Annl	25%	10%	10%	ns	ns	5%
HASKELL						
Sell	ns	25%	25%	ns	25%	0.1%
Annl	10%	ns	ns	ns	ns	10%
KEARNY						
Sell	ns	0.1%	0.1%	0.1%	1%	0.1%
Annl	25%	ns	ns	25%	ns	25%
FA-V						
Sell	ns	1%	1%	ns	0.1%	0.1%
Annl	ns	5%	10%	25%	25%	25%
HU-III						
Sell	25%	0.1%	1%	10%	0.1%	0.1%
Annl	25%	ns	ns	ns	25%	ns
LE-IV						
Sell	ns	0.1%	0.1%	25%	0.1%	0.1%
Annl	ns	ns	ns	ns	10%	25%
SH-VI						
Sell	ns	0.1%	0.1%	1%	0.1%	0.1%
Annl	25%	25%	25%	25%	25%	ns
WH-V						
Sell	ns	1%	0.1%	0.1%	0.1%	0.1%
Annl	ns	25%	25%	1%	25%	25%

Annl Annexation Thresholds
Sell Homogeneity Thresholds

However, the effects are consistent and the cumulative effects of both the annexation and the homogeneity parameters are statistically significant when the data sets are considered together.

iv Classifier Comparisons

Differences in the results achieved by the Supervised ECHO, the Nonsupervised ECHO, and the perpoint classifiers over the Landsat data sets will be examined in two ways. First, the classifiers will be compared by examining the results of all 768 observations of the Supervised ECHO classifier, versus all 81 observations of the Nonsupervised ECHO classifier, versus 10 observations of the perpoint classifier over the Landsat data. Then, the optimal results for each classifier, on each data set will be compared.

Comparison of all Landsat Observations

Table 2a-7 presents the effects of classifier on each of the six dependent variables. As can be seen from this table, the choice of classifier has a significant statistical effect on each of the six dependent variables. Table 2a-8 summarizes the results of pair wise comparisons of the classifiers. By examining this table, we may identify statistically significant differences between specific pairs of classifiers. By examining the average response over all ECHO parameters settings measured for the three classifiers, the following conclusions may be drawn:

* The Supervised ECHO classifier requires significantly less computer time than either the perpoint or Nonsupervised ECHO classifiers. There is not, however, a significant difference between the computer time required to classify an area by a perpoint classifier and the computer time required to classify the same area by the Nonsupervised ECHO classifier.

* The field center pixel performance of the Nonsupervised ECHO classifier is significantly lower than the field center pixel performance of the perpoint classifier and the field center pixel performance of the Supervised ECHO classifier, when all parameters settings are considered. The field center pixel performance achieved by the Supervised ECHO classifier is not significantly different from the field center pixel performance of the perpoint classifier when the results of the 24 Supervised ECHO classifications produced with the differing parameters settings for each data set are averaged.

* The full field performance of the Nonsupervised classifier is significantly lower than the full field performance of the Supervised ECHO and perpoint classifiers. There is no statistical difference between the performance of the Supervised ECHO classifier and the perpoint classifier when the results from all 24 ECHO parameter settings of the Supervised ECHO processor are averaged and compared to the perpoint result.

Table 2a-7

Overall Landsat Comparisons:
Comparison of Perpoint, Supervised ECHO,
and Nonsupervised ECHO Classifiers

Variable	Degrees of Freedom	F-Value
CPU Time	2,16	21.64***
FCP Performance	2,16	14.41***
Full Field Performance	2,16	12.85***
Training Performance	2,16	6.63**
RMS Error	2,16	4.16*
Classification Variability	2,16	10/54**

Significance Level

*	5%
**	1%
***	.1%

Table 2a-8

Overall Landsat Comparisons:
Comparison of Perpoint, Supervised ECHO, and Nonsupervised ECHO Classifiers
(859 Observations)

1	= Supervised ECHO	(768 Observations)
2	= Nonsupervised ECHO	(81 Observations)
3	= Perpoint	(10 Observations)

Range Tests
(T-tests with 17 degrees of Freedom)

CPU Time		1 3 2
	1,2	6.61***
	1,3	2.11*
	2,3	.30
Field Center Pixel Performance		2 3 1
	1,2	6.59***
	1,3	.04
	2,3	2.26*
Full Field Performance		2 1 3
	1,2	6.12***
	1,3	.06
	2,3	2.19*
Training Field Performance		2 3 1
	1,2	4.25***
	1,3	.01
	2,3	1.47
RMS Proportion Error		3 1 2
	1,2	3.35**
	1,3	.13
	2,3	1.29
Classification Variability		1 2 3
	1,2	1.67
	1,3	4.99***
	2,3	4.17***

* The training field performance of the Supervised ECHO classifier is significantly higher than the training field performance of the Nonsupervised ECHO classifier. The training field performance of the perpoint classifier is not significantly different from the training field performance of the Nonsupervised ECHO classifier; nor is the training field performance perpoint classifier significantly different from the training field performance of the Supervised ECHO classifier.

* The RMS proportion estimate error of the Nonsupervised ECHO classifier is significantly larger than the RMS proportion estimate error of either the perpoint or the Supervised ECHO classifiers. Again, the RMS proportion errors for the perpoint and the Supervised ECHO classification results are not significantly different.

* Both the Nonsupervised and the Supervised ECHO classifiers have significantly lower variability of classification results than the perpoint classifier. The variability of the classification results for the two ECHO algorithms are not significantly different, however.

These results indicate that, with random selection of homogeneity and annexation parameters, both ECHO classifiers perform better than the perpoint classifier only with respect to classification variability, and the Supervised ECHO classifier will require less time than the perpoint or the Nonsupervised ECHO classifier. On the other hand, with random selection of parameter settings, the Nonsupervised ECHO classifier has inferior performance with respect to field center pixel accuracy, full field accuracy and RMS proportion error to those of the Supervised ECHO and the perpoint classifiers.

Comparison of the Optimal Landsat Results

Table 2a-9 summarizes the optimal results for each dependent variable of each classifier for each Landsat data set. In addition, this table lists the difference between each pair of classifiers to be examined. The difference between the conclusions reached in this section and the conclusions reached in the previous section is that, in the previous section, the results for all the homogeneity and annexation parameter settings were considered for the Nonsupervised and Supervised ECHO classifiers and these classifiers were then compared to each other and the perpoint classifier; in this section only the results yielded by the optimal parameter settings of the Supervised and Nonsupervised ECHO classifiers are considered.

Table 2a-10 summarizes the results of paired T-tests between the optimal performances of each pair of classifiers over the Landsat data sets.

TABLE 2a-9

OPTIMAL RESULTS OF LANDSAT DATA SETS

Data Set	Variable	Perpoint	Super.	Nonsup.	Perpoint -Sup.	Perpoint -Nonsup	Nonsup. -Sup.
GRAHAM	CPU	121.00	74.00	154.00	47.00	-33.00	80.00
	FCP	89.90	93.90	95.10	-4.00	-5.20	1.20
	FF	87.40	90.00	92.00	-2.60	-4.60	2.00
	TRAIN	90.40	98.50	94.50	-8.10	-4.10	-4.00
	RMS	1.20	.10	.10	1.10	1.10	0.00
	VAR	20.85	10.80	10.49	10.05	10.36	-.31
GRANT	CPU	122.00	98.00	149.00	24.00	-27.00	51.00
	FCP	64.50	68.10	60.80	-3.60	3.70	-7.30
	FF	66.40	70.80	64.90	-4.40	1.50	-5.90
	TRAIN	56.20	58.40	54.00	-2.20	2.20	-4.40
	RMS	6.50	.10	2.28	-1.60	4.22	-5.82
	VAR	24.62	8.60	12.81	16.02	11.82	4.20
KEARNY	CPU	389.00	100.51	205.00	288.49	184.00	104.49
	FCP	65.00	65.40	51.00	-0.40	14.00	-14.40
	FF	63.90	64.40	52.30	-0.50	11.60	-12.10
	TRAIN	67.60	68.20	45.20	-0.60	22.40	-23.00
	RMS	12.30	12.28	8.00	.02	4.30	-4.28
	VAR	44.12	16.16	11.36	27.96	32.76	-4.80
HASKELL	CPU	396.00	166.82	223.00	229.18	173.00	56.18
	FCP	67.30	77.20	65.40	-9.90	1.90	-11.80
	FF	67.00	76.40	64.50	-9.40	2.50	-11.90
	TRAIN	77.70	88.50	63.80	-10.80	13.90	-24.70
	RMS	3.50	.98	.10	2.52	3.40	-0.88
	VAR	45.73	14.07	13.55	31.66	32.18	-0.52
LE-IV	CPU	284.00	180.22	306.00	103.78	-22.00	125.78
	FCP	50.70	57.20	55.10	-6.50	-4.40	-2.10
	FF	50.60	55.70	54.20	-5.10	-3.60	-1.50
	TRAIN	69.00	73.00	60.60	-4.00	8.40	-12.40
	RMS	16.20	19.37	23.78	-3.17	-7.58	4.41
	VAR	46.38	8.54	3.29	37.84	43.09	-5.25
HU-III	CPU	852.00	480.00	696.00	372.00	156.00	216.00
	FCP	59.10	61.70	61.90	-2.60	-2.80	.20
	FF	59.50	59.70	52.00	-0.20	7.50	-7.70
	TRAIN	85.20	90.80	40.00	-5.60	45.20	-50.80
	RMS	14.70	11.59	6.83	3.11	7.87	-4.76
	VAR	48.90	18.90	17.80	30.00	31.10	-1.10

TABLE 2a-9 (cont'd)

OPTIMAL RESULTS OF LANDSAT DATA SETS

Data Set	Variable	Perpoint	Super.	Nonsup.	Perpoint -Sup.	Perpoint -Nonsup.	Nonsup. -Sup.
SH-VI	CPU	526.00	296.75	473.00	229.25	53.00	176.25
	FCP	66.90	63.02	50.60	3.80	16.30	-12.42
	FF	62.40	60.20	50.10	2.20	12.30	-10.10
	TRAIN	59.20	65.70	53.10	-6.50	6.10	-12.60
	RMS	21.60	20.50	19.24	1.10	2.36	-1.26
	VAR	48.45	13.09	9.53	35.36	38.92	-3.56
WH-V	CPU	458.00	239.13	425.00	218.87	33.00	185.87
	FCP	74.60	75.50	57.30	-0.90	17.30	-18.20
	FF	70.60	70.51	57.60	.09	13.00	-12.91
	TRAIN	87.80	91.00	65.90	-3.20	21.90	-25.10
	RMS	6.90	7.20	23.13	-0.30	-16.23	15.93
	VAR	47.87	19.30	10.75	28.57	37.12	-8.55
LIVSTON	CPU	513.00	253.00		260.00		
	FCP	71.30	75.20		-3.90		
	FF	68.90	69.40		-0.50		
	TRAIN	85.10	88.20		-3.10		
	RMS	12.80	7.00		5.80		
	VAR	53.07	18.00		35.07		
FA-V	CPU	348.00	198.88	325.50	149.12	22.50	126.62
	FCP	86.60	89.20	89.50	-2.60	-2.90	.30
	FF	78.40	80.10	74.30	-1.70	4.10	-5.80
	TRAIN	72.20	78.00	88.50	-5.80	-16.30	10.50
	RMS	4.70	5.03	16.05	-0.33	-11.53	-11.02
	VAR	44.30	18.28	18.80	26.02	25.50	.52

CPU times are in seconds.
Field center pixel (FCP), full field (FF), and training field (TRAIN) performances, RMS proportion estimate error, and classification variabilities (VAR) are in percentages.

TABLE 2a-9 (cont'd)

OPTIMAL RESULTS OF LANDSAT DATA SETS

Data Set	Variable	Perpoint	Super.	Nonsup.	Perpoint -Sup.	Perpoint -Nonsup.	Nonsup. -Sup.
SH-VI	CPU	526.00	296.75	473.00	229.25	53.00	176.25
	FCP	66.90	63.02	50.60	3.80	16.30	-12.42
	FF	62.40	60.20	50.10	2.20	12.30	-10.10
	TRAIN	59.20	65.70	53.10	-6.50	6.10	-12.60
	RMS	21.60	20.50	19.24	1.10	2.36	-1.26
	VAR	48.45	13.09	9.53	35.36	38.92	-3.56
WH-V	CPU	458.00	239.13	425.00	218.87	33.00	185.87
	FCP	74.60	75.50	57.30	-0.90	17.30	-18.20
	FF	70.60	70.51	57.60	.09	13.00	-12.91
	TRAIN	87.80	91.00	65.90	-3.20	21.90	-25.10
	RMS	6.90	7.20	23.13	-0.30	-16.23	15.93
	VAR	47.87	19.30	10.75	28.57	37.12	-8.55
LIVSTON	CPU	513.00	253.00		260.00		
	FCP	71.30	75.20		-3.90		
	FF	68.90	69.40		-0.50		
	TRAIN	85.10	88.20		-3.10		
	RMS	12.80	7.00		5.80		
	VAR	53.07	18.00		35.07		
FA-V	CPU	348.00	198.88	325.50	149.12	22.50	126.62
	FCP	86.60	89.20	89.50	-2.60	-2.90	.30
	FF	78.40	80.10	74.30	-1.70	4.10	-5.80
	TRAIN	72.20	78.00	88.50	-5.80	-16.30	10.50
	RMS	4.70	5.03	16.05	-0.33	-11.53	-11.02
	VAR	44.30	18.28	18.80	26.02	25.50	.52

CPU times are in seconds.

Field center pixel (FCP), full field (FF), and training field (TRAIN) performances, RMS proportion estimate error, and classification variabilities (VAR) are in percentages.

Table 2a-10

Comparison of the Optimal Landsat Results for the Supervised ECHO, Nonsupervised ECHO and Perpoint Classifier

Variable	Perpoint versus Supervised	Perpoint versus Nonsupervised	Nonsupervised versus Supervised
<u>Observations</u>	10	9	9
CPU			
T	5.53	2.03	6.41
Significance Level	.1%	5%	.1%
FCP			
T	2.65	1.37	2.94
Significance Level	1%	NS	1%
FF			
T	2.1	2.22	4.31
Significance Level	5%	5%	1%
Training			
T	5.24	1.88	2.81
Significance Level	.1%	5%	5%
RMS Error			
T	1.03	.48	.33
Significance Level	NS	NS	NS
Classification Variability			
T	10.08	7.67	1.71
Significance Level	.1%	.1%	NS

At optimal parameter settings, both the Nonsupervised and Supervised ECHO classifiers require significantly less CPU time than the perpoint classifier. This is true even though the Nonsupervised classifier was run with the "MAP" option specified for the Landsat data. In addition, the Supervised ECHO classifier requires significantly less time than the Nonsupervised ECHO classifier for these data sets. The specification of the "MAP" option for the Nonsupervised classifier contributed to this result.

The Supervised classifier demonstrates a significantly superior field center pixel performance to either the perpoint or the Nonsupervised ECHO classifier. However, the Nonsupervised classifier, which identifies objects without the benefit of class statistics, demonstrates no statistically significant difference from the perpoint classifier with respect to field center pixel accuracy, at optimal annexation and homogeneity parameter settings.

The full field performance of the Supervised ECHO classifier is significantly superior to that of the perpoint and Nonsupervised ECHO classifiers. On the other hand, the full field performance of the Nonsupervised ECHO classifier at its optimal parameter settings is still significantly inferior to that of the perpoint classifier.

For the Landsat data sets, the Supervised ECHO classifier has a statistically significant advantage in training performance to both the Nonsupervised ECHO and the perpoint classifiers. Again, even at optimal parameter settings the Nonsupervised ECHO classifications are at a statistically significant disadvantage when compared to the perpoint results.

There are no statistically significant differences between the perpoint, the Supervised ECHO and the Nonsupervised ECHO classifiers with respectto RMS proportion estimate errors.

While not being significantly different from each other, the results of both the Supervised and the Nonsupervised ECHO classifiers at their optimal parameter settings demonstrate significantly less classification variability than the perpoint results.

b. Simulated Thematic Mapper Results

The analyses of variance for the effects of the Nonsupervised ECHO parameters on the six simulated Thematic Mapper data sets (two resolutions from Williams County, ND, four resolutions from Finney County, Kansas) are presented in Table 2a-11. Results for the six data sets considered together are presented in Table 2a-12.

i Cell Width Parameter

For the Landsat and aircraft data sets the Nonsupervised ECHO cell width parameter was set to two in all cases. For those data sets the cell width setting of two was selected to economize on the CPU time required to perform the verification tests based on the observation that, for Supervised ECHO results, larger cell width parameters did not improve ECHO performances. The six simulated Thematic Mapper data sets were classified by the Nonsupervised ECHO classifier at cell widths of two and three. There were several statistically significant effects of the cell width parameter on the Nonsupervised ECHO classifications of simulated Thematic Mapper data:

* When the cell width parameter is set to three, less CPU time is required than when it is set to two (significant at a 10% confidence level).

* The training performance at cell width three is significantly poorer than at cell width two (5% confidence level).

* The RMS proportion estimate error is less at cell width three than at cell width two (10% confidence level).

* The classification variability is less for cell width two than cell width three (10% confidence level).

The fact that the proportion estimates are better for cell width three than for cell width two for the simulated Thematic Mapper data indicates that the partitioning using the larger cell size may, indeed, be of value with the Nonsupervised ECHO processor.

Because the likelihood that a cell will be homogeneous when it contains nine scene elements is smaller than the likelihood that a cell containing only four scene elements will be homogeneous, it makes sense that, for a given homogeneity setting, more cells at cell width three, than at cell width two, would be identified as singular and split with constituent points classified individually. This situation may explain why the classification variability at cell width two is smaller than the classification variability at cell width three. On the other hand, when large homogeneous areas are present in a scene, fewer calculations of the classification equation will be necessary at larger cell widths. Thirty-six pixels in 2 by 2 pixel cells require nine calculations of class probability for each spectral class while, in 3 by 3 pixel cells, only four calculations of class probability for each spectral class are required.

Table 2a-11

Results of Individual Thematic Mapper Data Set
Parameter Anovas
(108 Observations)

	CPU	FCP	FULL	TRAIN	RMS	VARI	DF
1730							
Celw	1%	ns	5%	25%	1%	1%	1,4
Sell	0.1%	5%	0.1%	1%	1%	0.1%	2,4
Annl	1%	25%	25%	ns	10%	0.1%	2,4
Celw X Sell	1%	ns	10%	10%	ns	0.1%	2,4
Celw X Annl	1%	ns	ns	ns	25%	5%	2,4
Sell X Annl	25%	ns	ns	ns	10%	5%	4,4
1740							
Celw	0.1%	5%	10%	ns	1%	1%	1,4
Sell	0.1%	5%	1%	1%	0.1%	0.1%	2,4
Annl	1%	25%	25%	5%	1%	0.1%	4,4
Celw X Sell	5%	ns	ns	ns	5%	0.1%	2,4
Celw X Annl	5%	ns	ns	25%	ns	5%	2,4
Sell X Annl	5%	25%	25%	25%	1%	1%	4,4
3730							
Celw	0.1%	ns	ns	ns	ns	ns	1,4
Sell	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	2,4
Annl	0.1%	5%	5%	25%	ns	5%	4,4
Celw X Sell	1%	ns	ns	ns	10%	5%	2,4
Celw X Annl	1%	25%	10%	25%	25%	25%	2,4
Sell X Annl	5%	5%	5%	25%	ns	25%	4,4
3740							
Celw	1%	ns	10%	ns	25%	5%	1,4
Sell	0.1%	ns	0.1%	0.1%	0.1%	0.1%	2,4
Annl	0.1%	ns	5%	ns	25%	1%	2,4
Celw X Sell	1%	ns	1%	ns	0.1%	1%	2,4
Celw X Annl	1%	ns	ns	ns	ns	5%	2,4
Sell X Annl	5%	ns	25%	ns	ns	10%	4,4
3750							
Celw	5%	0.1%	0.1%	0.1%	0.1%	0.1%	1,4
Sell	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	2,4
Annl	1%	ns	25%	5%	10%	10%	2,4
Celw X Sell	5%	1%	5%	5%	5%	1%	2,4
Celw X Annl	1%	ns	ns	ns	ns	5%	2,4
Sell X Annl	10%	25%	25%	10%	25%	25%	4,4

Table 2a-11 (Continued)

	CPU	FCP	FULL	TRAIN	RMS	VARI	DF
3760							
Celw	25%	ns	ns	ns	10%	ns	1,4
Sell	0.1%	1%	1%	0.1%	1%	5%	2,4
Annl	1%	ns	ns	ns	ns	ns	2,4
Celw X Sell	5%	ns	ns	ns	ns	ns	2,4
Celw X Annl	5%	25%	25%	25%	ns	ns	2,4
Sell X Annl	5%	ns	ns	ns	ns	ns	4,4

Table 2a-12

OVERALL THEMATIC MAPPER RESULTS
Significant Effects of Parameters on Dependent Variables
(108 Observations)

Parameters	Degrees of Freedom	Variables: CPU Time	FCP Perform.	Full Field Perform.	Training Perform.	RMS Error	Classi. Vari.
Resolution	3,1	8.35	57.90+	19.44	509.24*	5.41	2.85
Cell Width	1,1	91.73+	3.34	0.76	345.90*	119.46+	72.44+
Homogeneity Threshold	2,2	25.51*	32.50*	16.26+	164.70**	68.63*	73.41*
Annexation Threshold	2,2	178.21**	4.05	3.45	13.46+	9.78+	65.95*
Resolution X Cell Width	3,1	92.01+	14.51	4440.00*	64.71+	19.90	38.07
Resolution X Homogeneity	6,2	3.68	50.21*	22.59*	12.70+	43.59*	48.58*
Resolution X Annexation	6,2	.68	1.72	3.05	.31	8.07	1.87
Cell Width X Homogeneity	2,2	45.59*	479.36**	7.68	44.74*	5.83	12.87+
Cell Width X Annexation	2,2	30.27*	19.66*	5.74	11.05+	286.00**	63.04**
Homogeneity X Annexation	4,4	227.36***	2.89	3.2	29.19**	4.21+	96.25***

Significance Level

+	10%
*	5%
**	1%
***	.1%

Table entries are for F-values

There are several interaction effects of cell width and resolution. As resolution increases, the effect of the cell width parameter on CPU time decreases (10% confidence level) and the effects of the cell width parameter on the full field and the training performances decrease (10% and 5% confidence levels, respectively). As resolution elements become larger, fields will be sampled by fewer pixels. Therefore, the number of truly homogeneous cells at a given cell size will decrease. Since there will be fewer homogeneous cells at 60 meter resolution than at 30 meter resolution, more cell splitting will take place and, hence, the effects of aggregating pixels into cells will be smaller at 60 meter resolution than at 30 meter resolution. This effect expands as the cell size becomes larger since the number of homogeneous 3 by 3 cells will shrink faster than the number of homogeneous 2 by 2 cells as the resolution elements become larger.

The interaction of the cell width parameter and the homogeneity parameter is statistically significant with respect to CPU time, field center pixel performance, training field performance, and variability of classification results. In all cases, as the cell width parameter increases, the effect of the homogeneity parameter increases. This results is logical since, as the cell become larger, the number of pixels affected by the outcome of each homogeneity test is larger.

There are also significant cell width-annexation threshold interactions. As the cell width becomes larger, the effect of the annexation parameter becomes smaller for CPU time, training field performance, field center pixel performance, RMS proportion estimate error, and classification variability. This results reflects the facts that as the partition becomes larger, there will be fewer and fewer homogeneous cells, and that, as the partition becomes larger, it becomes less likely that a neighboring homogeneous cell will be contained in a single object.

ii Resolution

As resolution increases, both field center pixel and traing field performances show a statistically significant decrease. The 50 meter Thematic Mapper data is probably responsible for this circumstance. Both the field center pixel and the training field performances on the 50 meter data sets are significantly below the performance levels of the 40 and the 60 meter data sets. Reasons for this degradation in performance at 50 meters are unclear. One would expect a tailing off of classification accuracy as the size of the cell approaches the size of the objects on the ground. However, in this data set, even at 60 meter resolution, cell sizes do not approach the size of the agricultural fields.

The interaction of the resolution with the homogeneity parameter is statistically significant for the field center pixel performance, the full field performance, the training performance, the RMS proportion estimate error, and the classification variability. For the 30 meter resolution the field center pixel and the full field performances improve as the cell homogeneity parameter increases from .05 to .10, and then fall somewhat as the homogeneity parameter goes from .10 to .25. At the larger resolutions, however, these variables fall steadily as the homogeneity parameter is increased. The effects of the homogeneity parameter increase for: the RMS proportion estimate error, the training field performance, and the classification variability as the resolution element size increases.

Interaction of the resolution and the annexation thresholds is not significant for any of the six dependent variables monitored.

iii Homogeneity Threshold

The homogeneity threshold has a significant effect on all six dependent variables. As the homogeneity threshold increases:

* the CPU time required to perform the Nonsupervised ECHO classification decreases,

* the field center pixel performance decreases,

* the full field performance decreases,

* the training field performance decreases,

* the RMS proportion estimate error increases, and

* the classification variability decreases.

Figures 2a-21 through 2a-26 graph the effects of the homogeneity and annexation parameters on the six dependent variables for 2 by 2 pixel cells. Unlike the Landsat results (See figure 2a-15), for the simulated Thematic Mapper data sets, the effect of homogeneity parameter on CPU time is statistically significant. The "MAP" option was not used on the simulated Thematic Mapper data. Therefore, there was no data replacement when homogeneous objects were identified in the field extraction phase, and consequently, the CPU time required goes down as the number of homogeneous cells increases due to the reduction in the number of passes through the classification equation made possible by classifying a field, rather than a point at a time.

The results of the field center pixel performance, full field performance, training field performance, and RMS proportion estimate error measurements indicate that for simulated Thematic Mapper data, at resolutions above 30 meters, homogeneity parameter specifications of .05 or less are appropriate.

Interaction between the homogeneity parameter and the annexation thresholds is significant for CPU time, training field performance, RMS proportion estimate error, and classification variability. As the homogeneity parameter increases, the effects of the annexation also increase for each of the dependent variables listed above. Since annexation can take place only when adjoining cells are homogeneous and since the number of homogeneous fields increases as the homogeneity parameter increases, it makes sense for the annexation parameter to have a greater effect when the homogeneity parameter is larger.

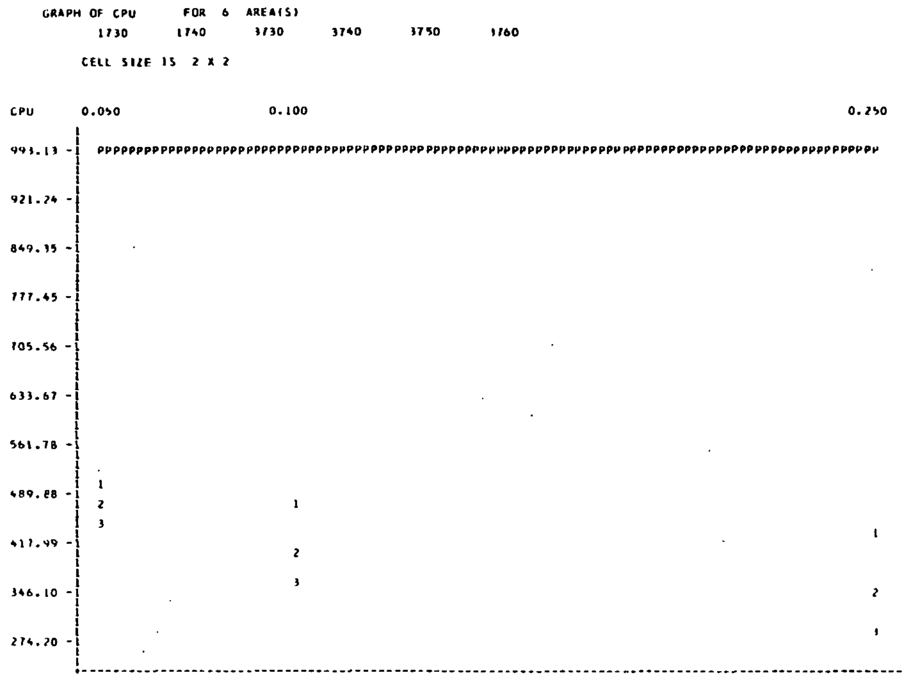


Figure 2a-21

Nonsupervised ECHO CPU Requirements
for the Simulated Thematic Mapper Data Sets

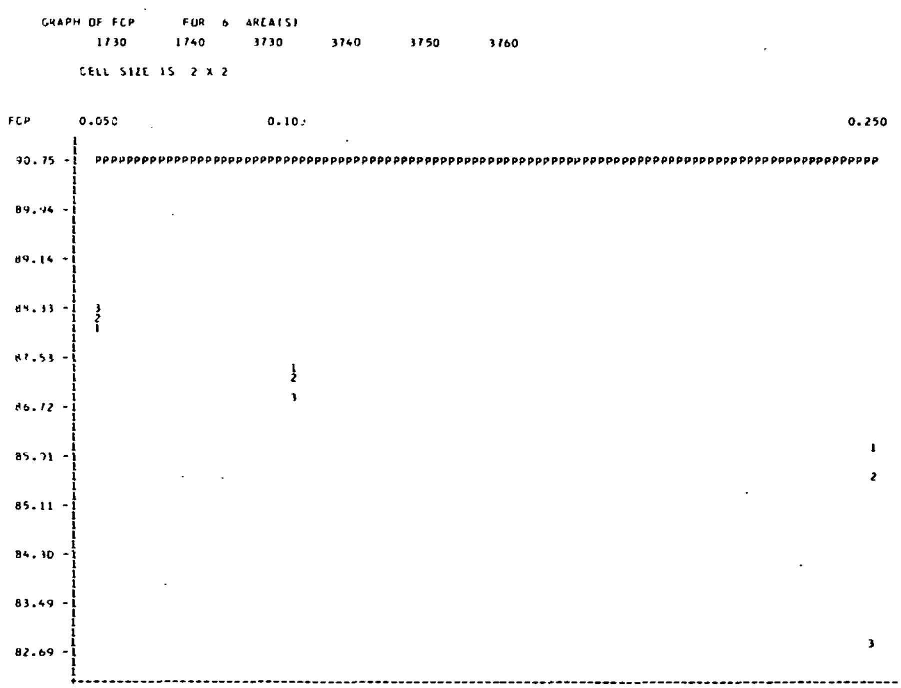


Figure 2a-22

Nonsupervised ECHO Field Center Pixel Performance
for the Simulated Thematic Mapper Data Sets

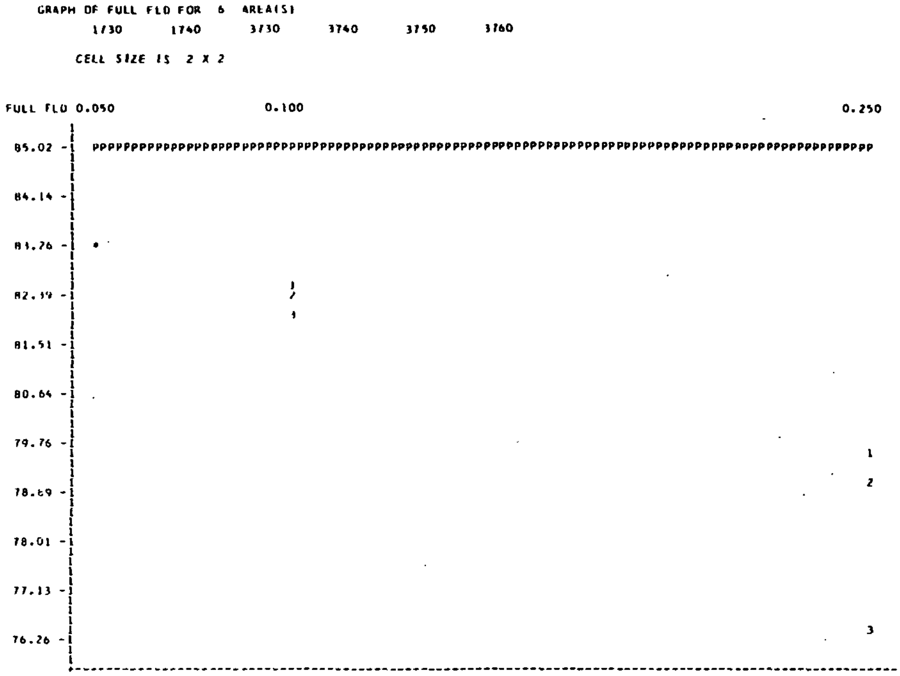


Figure 2a-23

Nonsupervised ECHO Full Field Performance for the Simulated Thematic Mapper Data Sets

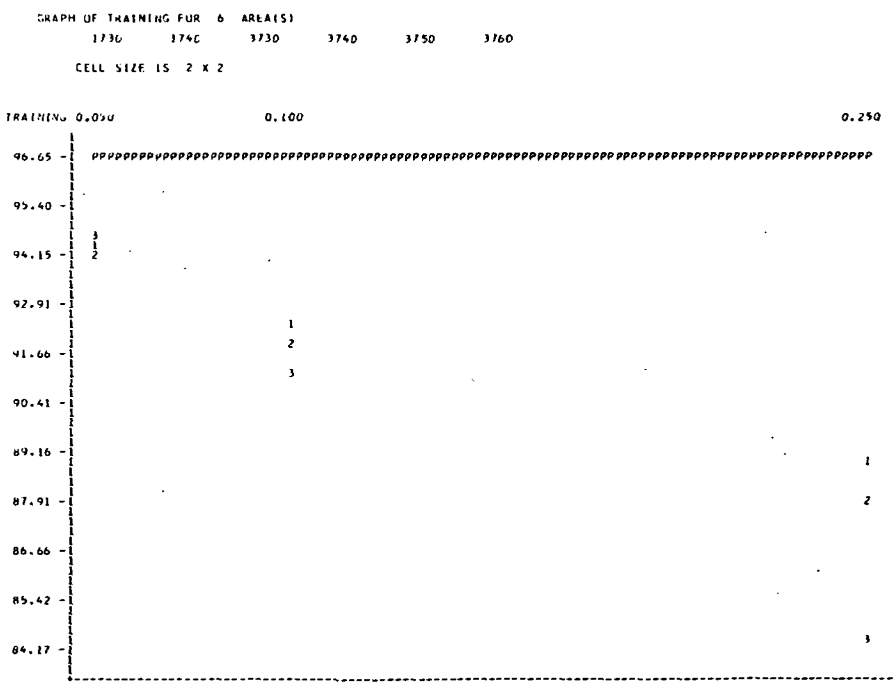


Figure 2a-24

Nonsupervised ECHO Training Field Performance for the Simulated Thematic Mapper Data Sets

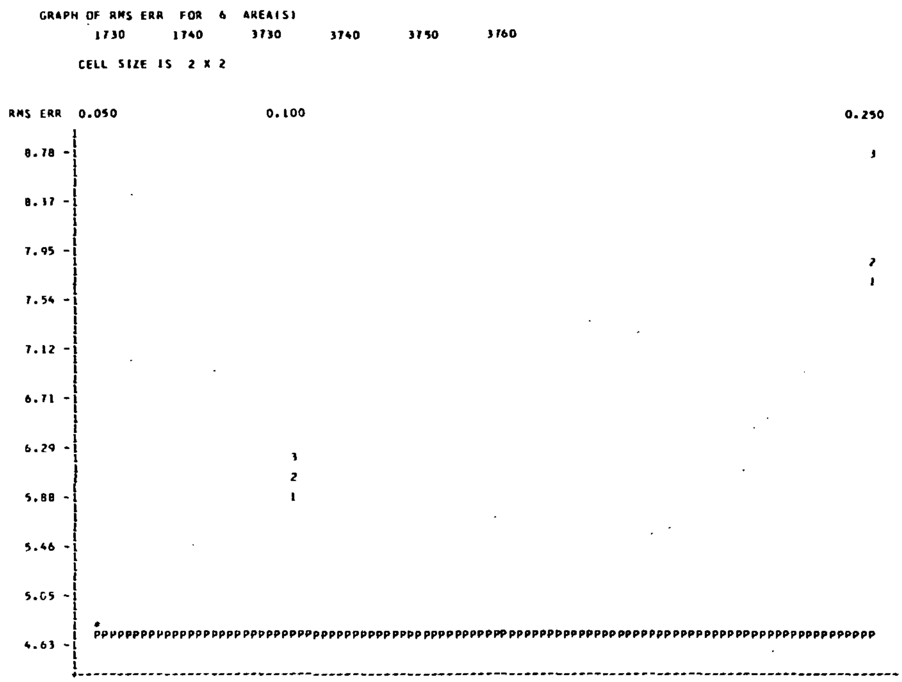


Figure 2a-25

Nonsupervised ECHO RMS Proportion Estimate Error for the Simulated Thematic Mapper Data Sets

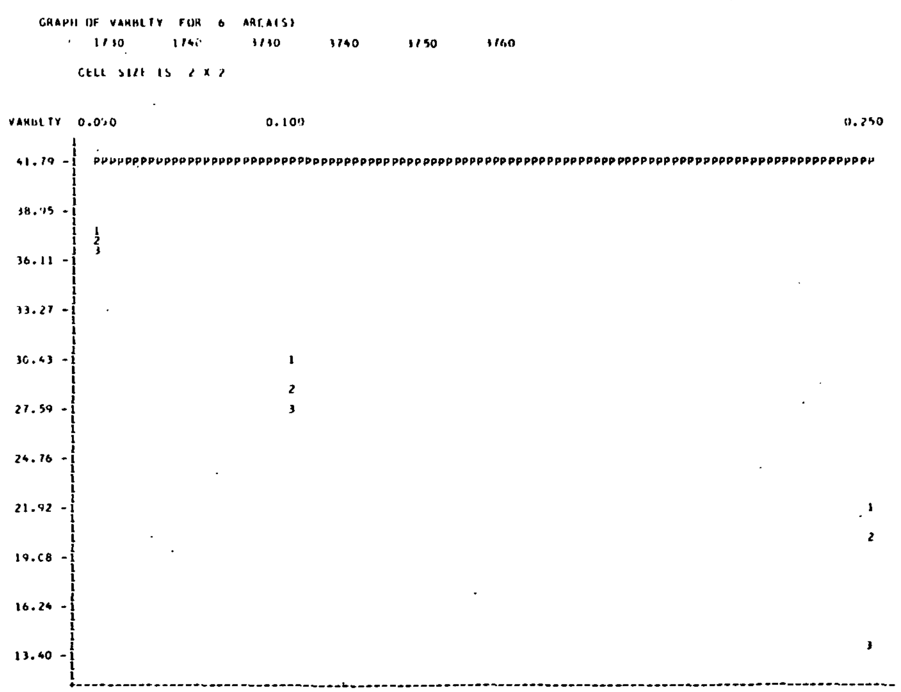


Figure 2a-26

Nonsupervised ECHO Classification Variability for the Simulated Thematic Mapper Data Sets iv Annexation Threshold

The annexation threshold has a significant effect on the CPU time required to perform classification, on the training performance, on the RMS proportion estimate error, and on the classification variability. As the annexation threshold goes from 10^{-1} to 10^{-3}, the time required to perform a classification decreases, the training performance decreases, the RMS proportion estimate error increases, and the classification variability decreases. This result indicates that for the simulated Thematic Mapper data, annexation parameters in the neighborhood of 10^{-1} are most appropriate.

v Classifier Comparisons

Differences in the results achieved by the Supervised ECHO, the Nonsupervised ECHO, and the perpoint classifiers over the simulated Thematic Mapper data sets will be examined in two ways. First, the classifiers will be compared by examining the results of 8 observations of the perpoint classifier versus the 108 observations of the Nonsupervised ECHO processor versus the 576 observations of the Supervised ECHO processor over the simulated Thematic Mapper data sets. Then, the optimal results for each classifier, on each data set will be compared.

Comparison of all Simulated Thematic Mapper Observations

Table 2a-13 presents the effects of classifier and resolution on the simulated Thematic Mapper data sets for the six dependent variables when all observations are considered for the ECHO classifiers. The effect of the classifier is significant for all six of these variables. Resolution has a statistically significant effect on field center pixel performance, full field performance, and classification variability. The interaction of the classifier and the resolution is statistically significant for training field performance, field center pixel performance, full field performance, RMS proportion estimate error, and classification variability. Table 2a-14 illustrates some of the effects of the three classifiers. The CPU time required by the Nonsupervised classifier is significantly less than the CPU time required by either the Supervised ECHO or the perpoint classifiers. On the other hand, the Nonsupervised ECHO classifier demonstrates inferior field center pixel, full field, and training field performances when all ECHO observations are considered.

The poor performance of the Nonsupervised ECHO processor on the 50 meter data may be responsible for many of the significant effects. By examining the graphs of the average field center pixel and full field performance in Table 2a-14, it can be seen that the 50 meter resolution is causing the significant interaction between resolution and classifier and the statistically significant differences between classifiers, for these two variables. The 50 meter results are also causing the significant effect of resolution on field center pixel performance, and the interaction effect between resolution and classifier for the training field performance and classification variability variables.

Table 2a-13

Overall Thematic Mapper Results

692 Observations

EFFECT	DEGREES OF FREEDOM	CPU TIME	FCP	FULL FIELDS	TRAINING FIELDS	RMS PROPORTION	CLASSIFICATION VARIABILITY
		Performance					
CLASSIFIER	2,2	18.72+	229.11**	122.61**	514.46**	80.47**	69.45*
RESOLUTION	3,3	5.02	9.51*	10.35*	2.06	2.90	7.409+
CLASSIFIER X RESOLUTION	6,4	.65	9.94*	6.82*	10.77*	5.37+	7.989*

Confidence Level:

+	10%
*	5%
**	1%
***	.1%

Table entries are F-values

Table 2a-14

Overall Thematic Mapper
Effects of Classifier on Variables

1 = Supervised ECHO Results
2 = Nonsupervised ECHO Results
3 = Perpoint Results

CPU Time	2 1 3
1,2	3.52*
1,3	2.48
2,3	3.42*

RMS Proportion Error	1 3 2
1,2	4.49**
1,3	.03
2,3	1.26

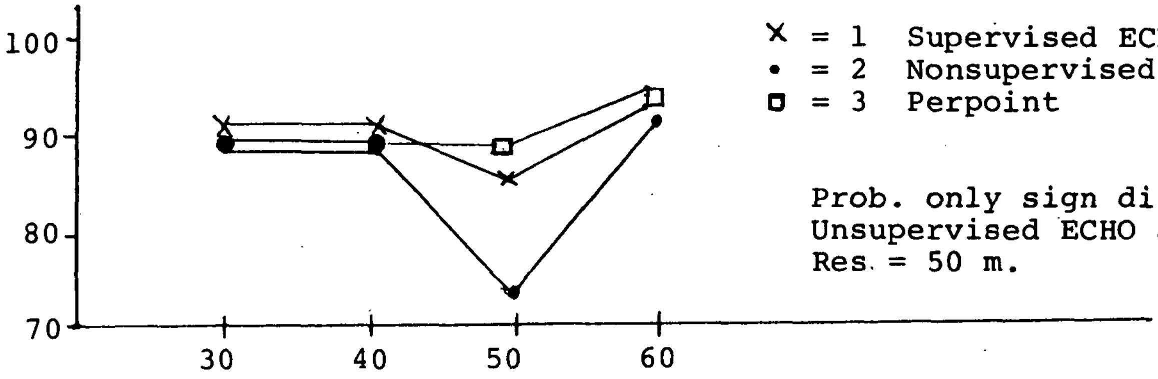
Significance Level

*	5%
**	1%

Table entries are T-values

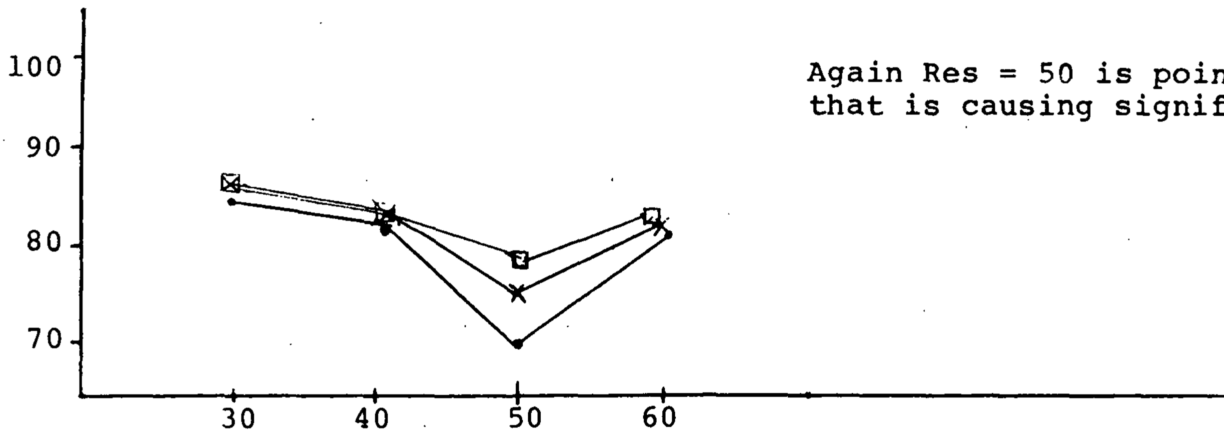
Table 2a-14 (Continued)

Field Center Pixel Performance



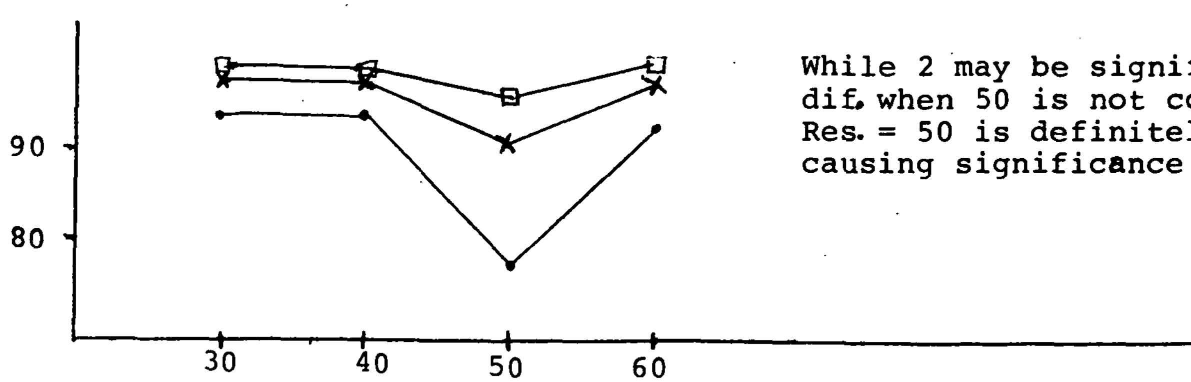
Prob. only sign diff. Unsupervised ECHO at Res. = 50 m.

Full Field Performance



Again Res = 50 is point that is causing significance

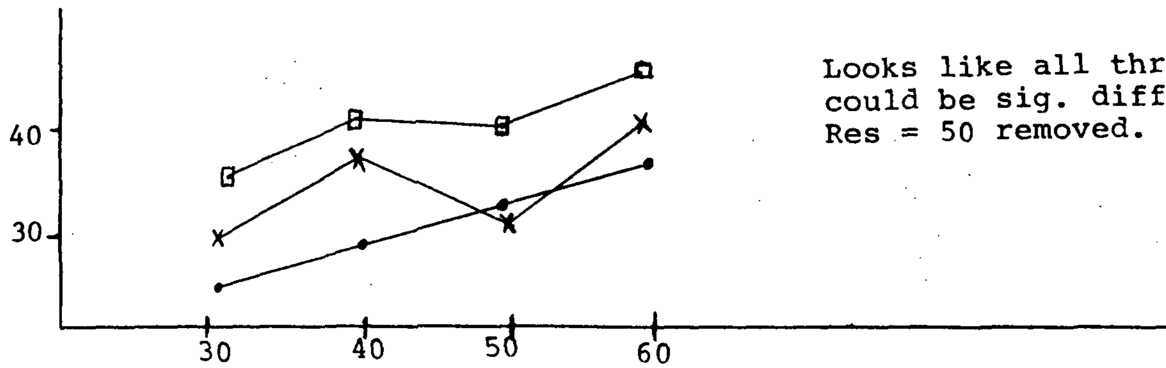
Training Field Performance



While 2 may be significant dif. when 50 is not considered Res. = 50 is definitely causing significance

Table 2a-14 (Continued)

Classification Variability



Looks like all three could be sig. diff. if Res = 50 removed.

As the resolution elements increase in size, the classification variability increases. This reflects the fact that it becomes less likely that adjacent pixels will be of the same class as the size of resolution elements increase.

There is a statistically significant difference between classifiers with respect to classification variability. The perpoint classification results are more variable than Supervised ECHO classification results which are, in turn, more variable than the Nonsupervised ECHO classification results. The difference between the two ECHO processors indicates that the homogeneity parameters used for the Nonsupervised ECHO tests result in more cells being identified as homogeneous than the homogeneity parameters in the Supervised ECHO experiments. It is possible that for the simulated Thematic Mapper data sets, the optimal homogeneity parameter may be less than the smallest Nonsupervised ECHO homogeneity parameter tested (which was 0.05).

The RMS proportion estimate error for the Supervised ECHO processor is significantly lower than that of the Nonsupervised processor. The proportion estimate error of the perpoint classifier is significantly different from neither the Supervised nor the Nonsupervised ECHO processor.

Comparison of Classifier Optimums

Table 2a-15 presents the optimal dependent variable measurement (lowest CPU, RMS proportion estimate error, classification variability, and highest field center pixel, full field and training field performance) for each of the three classifiers, together with the differences between pairs of optimal responses, for each of the simulated Thematic Mapper data sets. Table 2a-16 presents the results of a paired T-test to identify the significant differences between each pair of classifiers with respect to each dependent variable.

Comparing optimal results, the ECHO classifiers require significantly less CPU time to classify an area than the perpoint classifier requires. The CPU time required to produce a classification by the Nonsupervised and the Supervised ECHO processors are not statistically different, however.

At optimal parameter settings, the Supervised classifier has significantly higher field center pixel and full field classification performances than either the Nonsupervised ECHO or the perpoint classifier. The Nonsupervised ECHO classifier is not significantly different from the perpoint classifier at optimal parameter settings of the Nonsupervised processor with respect to these two variables.

The Supervised ECHO results have training field performance statistically superior to those of the perpoint classifier, which in turn, have the training field performance statistically higher than the results of the Nonsupervised ECHO processor.

The RMS proportion estimate error of the perpoint classifier was not statistically different from either of the ECHO classifiers. The Nonsupervised ECHO classifier had significantly higher proportion estimate error than the Supervised processor, however.

Table 2a-15

OPTIMAL RESULTS OF THEMATIC MAPPER DATA SETS

Data Set	Variable	Perpoint	Super.	Nonsup.	Perpoint -Sup.	Perpoint -Nonsup.	Nonsup. -Sup.
1730	CPU	1170.00	335.20	370.90	834.80	799.10	35.70
	FCP	92.10	94.70	91.20	-2.60	.90	-3.50
	FF	88.10	89.60	87.40	-1.50	.70	-2.20
	TRAIN	97.60	98.50	96.20	-0.90	1.40	-2.30
	RMS	3.40	3.61	3.77	-0.21	-0.37	.16
	VAR	35.93	14.27	12.66	21.66	23.27	-1.61
1740	CPU	732.00	228.94	256.80	503.06	475.20	27.86
	FCP	88.60	91.20	91.70	-2.60	-3.10	.50
	FF	82.60	84.20	85.00	-1.60	-2.40	.80
	TRAIN	97.20	97.70	97.30	-0.50	-0.10	-0.40
	RMS	8.10	7.80	7.98	.30	.12	.18
	VAR	37.68	18.31	13.28	19.37	24.40	-5.03
1750	CPU	436.40	144.97		291.43		
	FCP	88.20	88.60		-0.40		
	FF	74.70	74.90		-0.20		
	TRAIN	98.70	98.90		-0.20		
	RMS	7.60	7.66		-0.06		
	VAR	35.22	15.71		19.51		
1760	CPU	358.80	111.66		247.14		
	FCP	94.50	95.50		-1.00		
	FF	81.00	81.90		-0.90		
	TRAIN	98.70	98.70		0.00		
	RMS	8.50	8.60		-0.10		
	VAR	42.31	22.30		20.01		
3730	CPU	1200.00	353.16	402.50	846.84	797.50	49.34
	FCP	87.50	91.40	89.30	-3.90	-1.80	-2.10
	FF	84.10	87.70	85.80	-3.60	-1.70	-1.90
	TRAIN	97.90	98.60	97.60	-7.00	.30	-1.00
	RMS	7.30	2.40	2.34	4.90	4.96	-0.06
	VAR	34.75	18.47	17.51	16.28	17.24	-0.96
3740	CPU	1222.00	418.43	312.00	803.57	910.00	-106.43
	FCP	91.60	92.80	90.50	-1.20	1.10	-2.30
	FF	86.20	87.60	85.70	-1.40	.50	-1.90
	TRAIN	96.60	98.00	95.90	-1.40	.70	-2.10
	RMS	2.70	2.71	3.17	-0.01	-0.47	.46
	VAR	45.93	27.02	24.63	18.91	21.30	-2.39

Table 2a-15 (cont'd)

OPTIMAL RESULTS OF THEMATIC MAPPER DATA SETS

Data Set	Variable	Perpoint	Super.	Nonsup.	Perpoint -Sup.	Perpoint -Nonsup.	Nonsup. -Sup.
3750	CPU	855.20	233.47	218.60	621.73	636.60	-14.87
	FCP	91.50	91.46	78.10	.04	13.40	-13.36
	FF	85.00	84.68	75.80	.32	9.20	-8.88
	TRAIN	92.90	92.90	84.90	0.00	8.00	-8.00
	RMS	3.80	3.79	7.69	.01	-3.89	3.90
	VAR	46.08	21.20	14.70	24.88	31.38	-6.50
3760	CPU	779.60	322.56	170.40	457.04	609.20	-152.16
	FCP	93.20	93.80	92.10	-0.60	1.10	-1.70
	FF	84.10	84.70	83.30	-0.60	.80	-1.40
	TRAIN	97.70	98.20	96.20	-0.50	1.50	-2.00
	RMS	3.00	2.61	3.53	.39	-0.53	.92
	VAR	50.35	33.97	2.73	16.38	47.62	-31.24

Table 2a-16

Comparison of the Optimal Simulated Thematic Mapper Results for the Supervised ECHO, Nonsupervised ECHO, and Perpoint Classifiers

Variable	Perpoint versus Supervised	Perpoint versus Nonsupervised	Nonsupervised versus Supervised
Observations			
CPU			
T	6.79	10.87	1.87
Significance Level	.1%	.1%	NS
Field Center Pixel Performance			
T	3.18	.80	1.87
Significance Level	1%	NS	10%
Full Field Performance			
T	2.85	.70	1.93
Significance Level	1%	NS	10%
Training Field Performance			
T	1.58	1.60	2.36
Significance Level	10%	10%	5%
RMS Proportion Estimate Error			
T	1.08	.03	1.52
Significance Level	NS	NS	10%
Classification Variability			
T	19.97	6.21	1.68
Significance Level	.1%	1%	10%

The results of the Nonsupervised ECHO processor were significantly less variable than those of either the perpoint or the Supervised ECHO processor. In addition, the Supervised ECHO classification results were less variable than the perpoint classifier's results at optimal parameter settings.

For the optimal parameter settings of the ECHO classifiers, the performance of the Nonsupervised ECHO processor is superior to the performance of the perpoint classifier only with respect to the CPU time required to perform the classification and the variability of the classification results. The Nonsupervised ECHO processor had training field performance which was worse than the perpoint classifier, for all the parameter settings which were tested (performances would have been the same if no homogeneous cells had been identified). These results indicate that there may be little or not advantage in classifying with the Nonsupervised ECHO processor with respect to classification performances. However, the classification variability results indicate that for the five channel simulated Thematic Mapper data, the homogeneity parameters selected for testing of the Nonsupervised ECHO caused more cells to be identified as homogeneous than were identifed by the Supervised processor as homogeneous. All homogeneity parameters for the Nonsupervised ECHO tests over this data set may have been too high. Simply lowering the homogeneity parameter may improve the performance of the Nonsupervised processor. In any case, it seems fair to conclude that the Supervised ECHO processor performs better than both the Nonsupervised ECHO and the perpoint classifiers. Its superiority may be attributable to the use of class statistics in object identification.

c. Aircraft Results

Figures 2a-27 through 2a-31 present the CPU time, the field center pixel performance, the full field performance, the training field performance, and the classification variability achieved by the Nonsupervised ECHO processor for the single aircraft run tested. RMS proportion estimate error values were not calculated.

There are insufficient samples to perform an analysis of variance comparing these results to the single perpoint classification of the area. However, results of T-tests of the probability of each dependent variable's perpoint result falling in the distribution of the nine measurements of that dependent variable for the Nonsupervised ECHO results is presented in Table 2a-17. It appears from these results that, for the aircraft data set, the Nonsupervised ECHO classifier:

* requires less CPU time than the perpoint classifier,

* has inferior field center pixel performance to the perpoint classifier, and

* has superior full field performance when compared to the perpoint classifier.

These results are weak since so few samples are available. On the basis of Landsat and simulated Thematic Mapper results, it seems unlikely that the field center pixel performance is significantly superior for the same data set in which full field performance is significantly inferior.

Table 2a-18 presents the analysis of variance results for the effects of the Nonsupervised ECHO homogeneity and annexation parameters on each of the dependent variables. There are insufficient data points for any effects to be statistically significant at a 10% confidence level.

TABLE 2a-17

T-tests Comparing the Perpoint Results to the Distribution of Nonsupervised ECHO Results for Six Variables

	CPU Time	FCP	Full Field	Training Field	Classification Variability
Mean of the Nonsupervised Results	279.22	70.53	70.52	85.22	14.94
Standard Deviation of Nonsupervised Results	39.43	.77	.72	1.01	5.65
Perpoint Results	434.00	71.80	69.50	86.40	22.29
Perpoint result is different than the ECHO result at a 10% confidence level	yes	yes	yes	no	no

TABLE 2a-18

Effects of Nonsupervised ECHO Homogeneity and Annexation Thresholds on Six Dependent Variables for One Aircraft Data Set

	CPU	FCP	Full Field	Training Field	Classification Variability
Homogeneity	NS	NS	NS	NS	NS
Annexation	NS	NS	NS	NS	NS

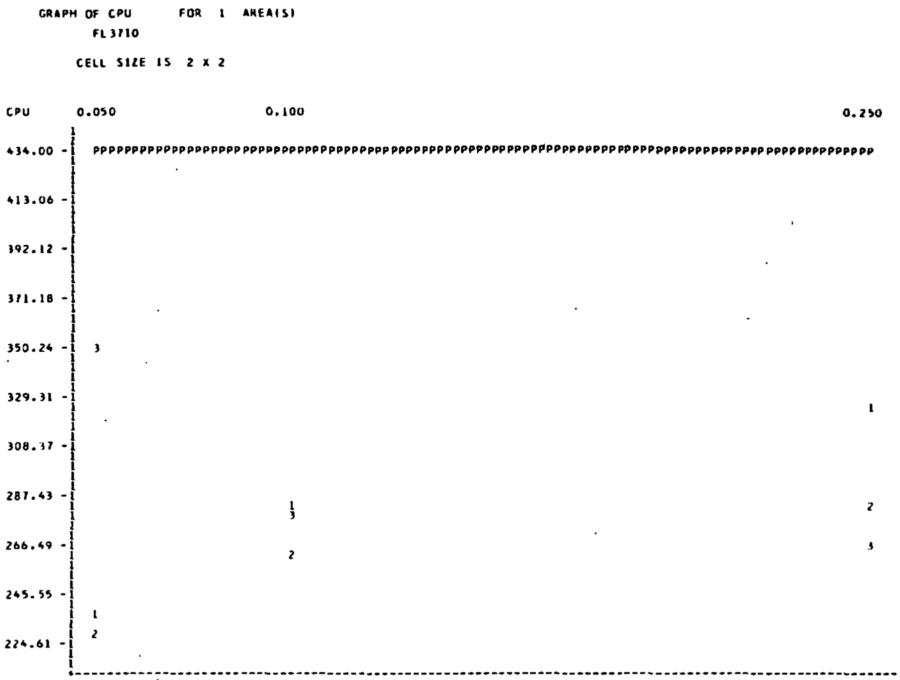


Figure 2a-27

Nonsupervised ECHO CPU Requirements for Aircraft Data Set

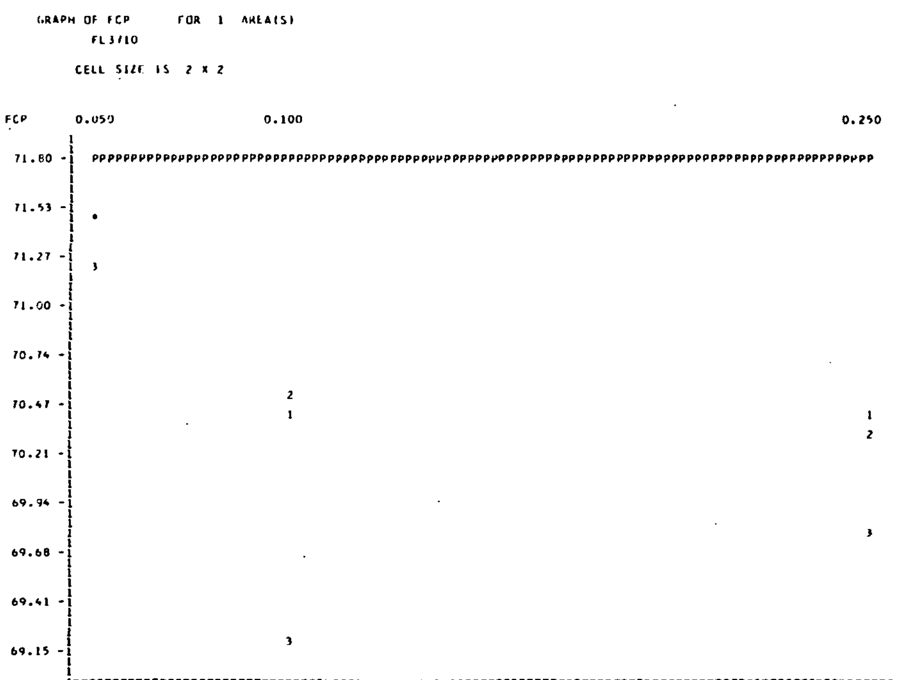


Figure 2a-28

Nonsupervised ECHO Field Center Pixel Performance for Aircraft Data Set

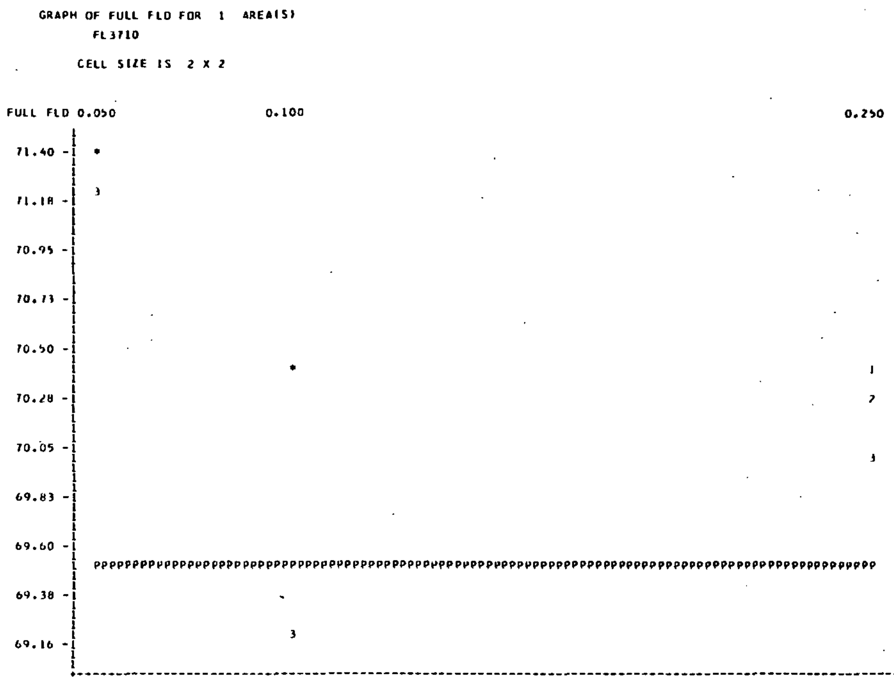


Figure 2a-29

Nonsupervised ECHO Full Field Performance for Aircraft Data Set

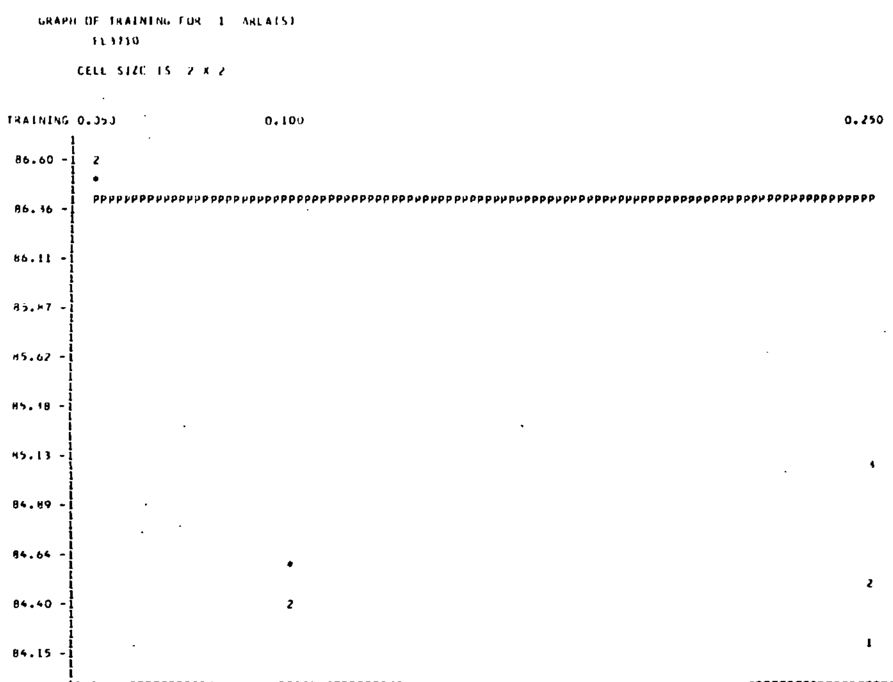


Figure 2a-30

Nonsupervised ECHO Training Field Performance for Aircraft Data Set

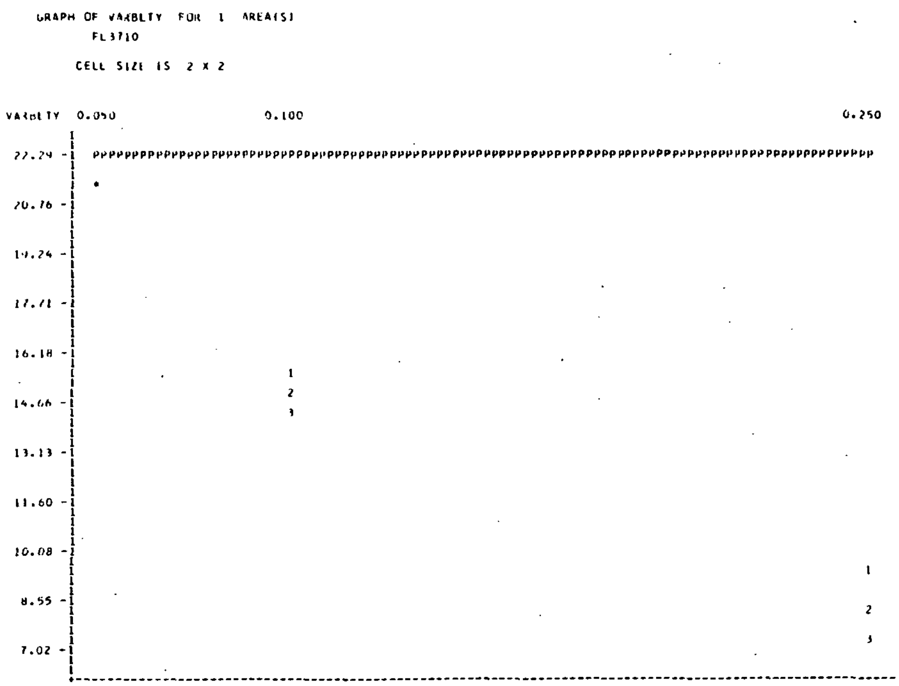


Figure 2a-31

Nonsupervised ECHO Classification Variability for aircraft data set

Nonsupervised ECHO Object Map Integrity Assessment

The Object Map Integrity Assessment task called for the comparison of the objects found by the Nonsupervised ECHO with the known fields in the scene. Several problems arose which made it impossible to obtain quantitative results. The program written to construct object boundaries from the intermediate results tape took longer than anticipated to debug, and the delay strained the time and personnel resources allotted to this task. The display of the objects in a form suitable for comparison with the field maps of the LACIE segments also encountered difficulties which have not been completely resolved. As a result, no assessment of the object map integrity has been possible to date.

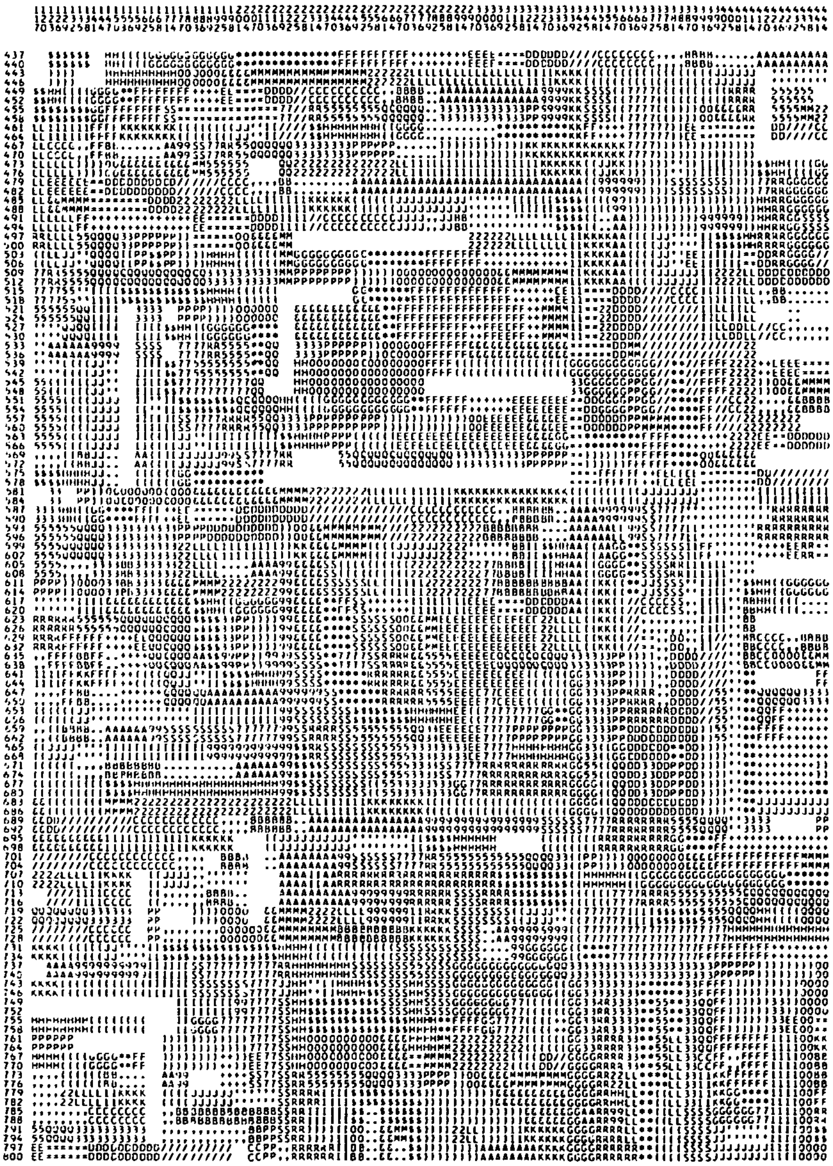
```
ECHOP6                         LABORATORY FOR APPLICATIONS OF REMOTE SENSING            NOV  4,1977
CARLOS ARTURO                              PURDUE UNIVERSITY                            04 34 00 PM
                                                                                        LARSYS VERSION 3
AIRCRAFT

SAMPLE CLASSIFICATION STUDY ..

          RUN NUMBER............ 75003710               DATE DATA TAKEN... JULY  6,1975
          FLIGHT LINE... LINE 3 R1 KANSAS               TIME DATA TAKEN..... 1553 HOURS
          DATA TAPE/FILE NUMBER.. 3230/ 1               PLATFORM ALTITUDE..  10000 FEET
          REFORMATTING DATE. APR   5,1976               GROUND HEADING.....   0 DEGREES

                                    CHANNELS USED
                                    -------------

     CHANNEL  1   SPECTRAL BAND  0.59 TO  0.64 MICROMETERS   CALIBRATION CODE = 1   CO = 0.0
     CHANNEL  2   SPECTRAL BAND  0.65 TO  0.69 MICROMETERS   CALIBRATION CODE = 1   CO = 0.0
     CHANNEL  3   SPECTRAL BAND  0.72 TO  0.76 MICROMETERS   CALIBRATION CODE = 1   CO = 0.0
     CHANNEL  5   SPECTRAL BAND  0.82 TO  0.88 MICROMETERS   CALIBRATION CODE = 1   CO = 0.0
     CHANNEL  6   SPECTRAL BAND  0.98 TO  1.04 MICROMETERS   CALIBRATION CODE = 1   CO = 0.0
     CHANNEL  7   SPECTRAL BAND 10.10 TO 11.00 MICROMETERS   CALIBRATION CODE = 1   CO = 0.0
AREA PROCESSED.....LINES    437-  800  (BY  3)
                   COLUMNS  117-  444  (BY  3)
```



```
10103  CPU TIME USED WAS  186.885 SECONDS.     (LARSMN)

ECHOP6                         LABORATORY FOR APPLICATIONS OF REMOTE SENSING            NOV  4,1977
CARLOS ARTURO                              PURDUE UNIVERSITY                            04 34 00 PM
                                                                                        LARSYS VERSION 3
AIRCRAFT

10004  END OF INPUT DECK - RUN COMPLETED    (LARSMN)
10050  TOTAL CPU TIME FOR THIS RUN WAS  186.942 SECONDS.     (LARSMN)
```

Figure 2a-32

Nonsupervised ECHO Object Map

CONCLUSIONS

Summary

The Supervised ECHO processor (which utilizes class statistics for object identification) successfully exploits the redundancy of states characteristics of sampled imagery of ground scenes to achieve better classification accuracy, reduce the number of classifications required and reduce the variability of classification results. The Supervised ECHO processor requires cell size, cell-to-field annexation, and cell homogeneity parameters, input data, and a class-conditional marginal density statistics deck for both object identification and classification. The improvement in classification performance the Supervised ECHO classifier provides (over a perpoint classifier) expands as the size of the objects expands in terms of numbers of pixels.

The Nonsupervised ECHO processor (which identifies objects without the benefit of class statistics) successfully reduces the number of classifications required and the variability of the classification results. It is unsuccessful in improving classification performance, however. The Nonsupervised ECHO processor runs in two phases, a field extraction phase, then a classification phase. Cell size, cell-to-field annexation, and cell homogeneity parameters, along with input data are required by the field extraction phase of the Nonsupervised processor. The classification phase requires the intermediate results produced by the field extraction phase of the Nonsupervised ECHO processor and a class-conditioned marginal density statistics deck to classify the objects identified in the field extraction phase.

Both ECHO processors provide information which may be of value in the training process. The Supervised processor produces a singular cell map which may be used to assess the adequacy of training for the area classified. Cells are identified as singular by the Supervised processor when the likelihood of the cell belonging to the most likely of the available classes falls below a threshold. Therefore, when a cell is categorized as "singular", it either contains pixels from more than one class or the spectral class of the pixels of the cell is not represented in the available class statistics. Groups of contiguous singular cells will occur when one or more spectral classes have been omitted. The singular cell map may indicate where additional training statistics should be collected.

Using the "MAP" option causes the intermediate results tape produced by the field extraction phase of the Nonsupervised ECHO processor to replace the response value in each channel of a data vector by the average response value for that channel of the field to which the data vector belongs. A false color photo of these data channels from such an intermediate results tape may help the analyst identify areas which are spectrally homogeneous and may reduce such noise effects as the six line banding which is encountered in some Landsat data sets. Figure 2a-32 is an example object map produced by the field extraction phase of the Nonsupervised ECHO processor. Blanks on this map indicate singular cells. Each pixel falling in an object identified by the Nonsupervised field extraction algorithm is represented by a symbol which has arbitrarily been assigned for that object.

The greatest potential aid to the training process is the intermediate tape produced by the Nonsupervised ECHO field extraction routine. This tape contains the covariance and mean matrices of each field identified (without the benefit of class statistics) by the Nonsupervised ECHO processor. This spectral/spatial cluster may contain all the information an analyst needs to produce a more representative class statistics deck for a given area in less time than would be required to produce a statistics deck by conventional methods.

Recommendations

The ECHO processors are documented in this report, the Final Technical Report for May, 1977 [5], the ECHO User's Guide [6], and an ECHO case study [7]. All the ECHO algorithms are available to JSC (and other remote terminal sites) via the LARS remote terminal. It is recommended that:

* JSC personnel should be encouraged (or assigned) to use the ECHO algorithms via the remote terminal. Only through this type of experience will NASA personnel develop experience and confidence in using this approach to classification. Only in this way will they gain sufficient insight into the characteristics of the ECHO classifiers to appreciate the potential impact of the ECHO approach in the context of large area surveys.

* In order to further evaluate the utility of the Nonsupervised field extraction algorithm as a method for spectral/spatial clustering, tests should be initiated involving it and ERIM's "Glob and Blob" algorithm. These tests could be carried out either by LARS or by personnel at JSC via the LARS remote terminal.

* The general utility of the ECHO processors and training procedures and classification analyses should be investigated. For example, the singular cell map produced by the Supervised ECHO processor provides an indicator of the adequacy of the training statistics for a given area; groups of contiguous singular cells, where fields or other objects are known to exist, indicate the omission of one or more spectral classes from the statistics. Such investigation could be included in the training area selection/pixel labeling investigations. Also needed are training procedures geared specifically for the training of sample classifiers. Although these procedures have been evolving gradually, present methods are still more appropriate for training perpoint classifiers.

REFERENCES FOR TEST OF SPECTRAL/SPATIAL CLASSIFIER

1. Landgrebe, D. A. 1975.
   Final Report, NASA Contract NAS9-14016, June 1, 1974-May 31, 1975. pp. X-1 - X-11.

2. Kettig, R. L. 1975.
   Computer Classification of Remotely Sensed Multispectral Image Data by Extraction and Classification of Homogeneous Objects.
   Ph.D. Thesis, School of Electrical Engineering, Purdue University, West Lafayette, Indiana.

3. Kettig, R. L. and Landgrebe, D. A. 1973.
   Automatic Boundary Finding and Sample Classification of Remotely Sensed Multispectral Data.
   LARS Information Note 041773, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana.

4. Kettig, R. L. and Landgrebe, D. A. 1975.
   Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects.
   Proceedings Symposium on Machine Processing of Remotely Sensed Data, Purdue University, IEEE Catalog No. 75CH1009-0-C.

5. Landgrebe, D. A. 1977.
   Final Technical Report, NASA Contract NAS9-14970, June 1, 1976-May 31, 1977, Vol. 1.

6. Kast, J. L., Swain, P. H., and Davis, B. J. 1977.
   ECHO User's Guide.
   LARS Publication 083077, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana.

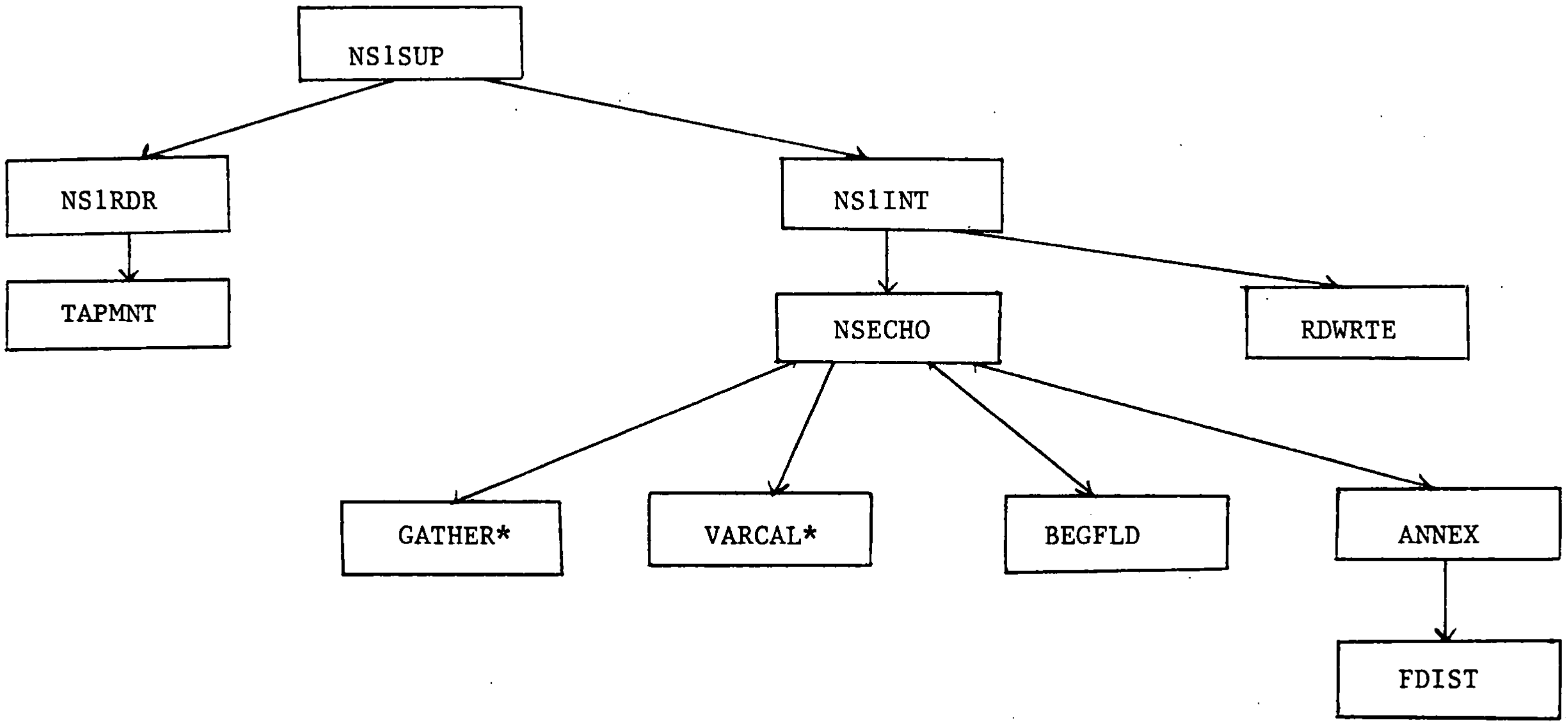
7. Scholz, D., Russell, J., Lindenlaub, J., Swain, P. 1977.
   A Case Study Using ECHO for Analysis of Multispectral Scanner Data.
   LARS Publication 090177, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana.

8. Bauer, M. E., Cary, T. K., Davis, B. J., and Swain, P. H. 1975.
   Crop Identification Technology Assessment for Remote Sensing (CITARS), Volume VI: Data Processing at the Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana.
   National Aeronautics and Space Administration, Houston, Texas.
   JSC 09389.

APPENDIX A

Fortran Program Documentation for the Nonsupervised ECHO Processor

MAIN SUBROUTINES TREE OF THE FIELD EXTRACTION PHASE
OF THE NONSUPERVISED ECHO PROCESSOR



*Delivered with Supervised ECHO Documentation in the Final Technical Report on NASA Contract NAS9-14970, May 1977

LARS Program Abstract ________

MODULE IDENTIFICATION

Module Name: NS1SUP Function Name: NS1ECHO

Purpose: Supervisor for NS1ECHO

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza Date: 8/20/77

Latest Revisor: ________ Date: ________

MODULE ABSTRACT

Supervisor for the field extraction phase (phase 1) of Nonsupervised ECHO (Extraction and Classification of Homogeneous Objects).

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NS1SUP

CALL NS1SUP

NS1SUP is called by LARSMN to execute the field extraction phase of nonsupervised ECHO. There are no parameters involved.

2. Internal Description

NS1SUP receives control from LARSMN to perform Nonsupervised field extraction. NS1SUP calls NS1RDR to read and interpret the function control cards, then calls NS1INT to complete the initialization and compute array bases used to separate storage in the arrays for the subroutine that performs the field extraction process (NSECHO).

3. Input Description

Not applicable

4. Output Description

Two messages are produced and written to unit TYPEWR (the console).

UNSUPERVISED ECHO FUNCTION (PHASE 1)REQUESTED

signifies beginning of the function.

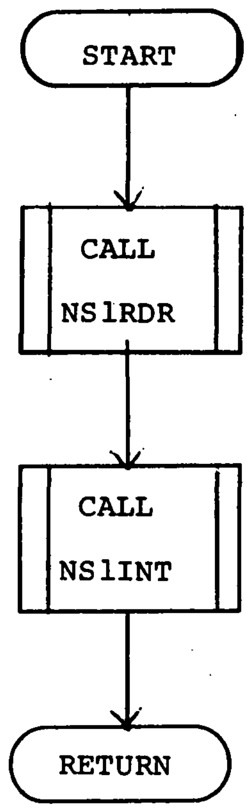
UNSUPERVISED ECHO FUNCTION (PHASE 1) COMPLETED

signifies end of the function.

5. Supplemental Information

See LARSYS System Manual for supervisor requirements.

6. Flowchart



FILE NSISUP

```
                 C       NSISUP    LARS  XXXX
                 C                 WRITTEN             BY C. A. POMALAZA
                 C
                 C*****************************************************************
                 C
                 C SUPERVISOR ROUTINE FOR FIELD EXTRACTION PHASE OF NONSUPERVISED ECHO
                 C
                 C*****************************************************************
                 C
0001                   SUBROUTINE NSISUP
0002                   IMPLICIT INTEGER * 4 (A-Z)
                 C
                 C
                 C*****************************************************************
                 C
0003                   COMMON /GLCCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX,
                      1     CLUSTX, CONPUT, CPYOUT, CRDRDR, CRDSEQ, DATAPE,
                      2     DUPLTP, DUPRUN, ERRMSG, FBPNT,
                      3     FILESV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,
                      4     IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,
                      5     PAGSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,
                      6     RESTRT, RUNFIL, RUNTAB(10,3),
                     .7     SDATA, SEPARX, SEPTPX, SPARE(10), TEMPAS(30),
                      8     TPSTAT(6), TTFLDX ,TYPEWR,
                      9     TOP,  ARRAY(12500)
                 C
                 C
0004                   REAL * 8 ARRAY
0005                   REAL * 4 FROCAL(5,30)
0006                   INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)
0007                   INTEGER * 2 BLANK2
0008                   LOGICAL * 4 CHKOUT
0009                   LOGICAL * 1 BLANK1
0010                   EQUIVALENCE (DATSAV,ID(1)), (CURRUN,ID(3)), (FROCAL(1),ID(51)),
                      1          (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),
                      2          (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),
                      3          (MAPSAV,TPSTAT(1)),
                      4          (SLPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),
                      5          (CCPSER,TPSTAT(5)), (TRAOUT,TPSTAT(6)),
                      6          (BLANK,BLANK2,BLANK1)
                 C
0011                   COMMON /NSICOM/
                      1 ANFLAG, ANN1, ANN2, BUFPTS, BUFRCZ,
                      2 C1, C4, CELSIZ, CELWTH,
                      3 CSET(3,30), CSET3(3,30), INFO(17), JPTS,
                      4 K9, L1, MAPFLG, MAXSIZ, MINSIZ, MTXSIZ, ND(200),
                 5    5 NEWRUN, NCCLS, NOFEAT, NSR, NUMCLS, NVR,
                      6 PTS, RESULT, ROWSIZ, RQFILE, RQTAPE, SELEC1(30),
                      7 TTITLE(30), TOTPTS, VARSIZ, VECSIZ,
                      8 CSEL(30), CSEL3(30), FETVC3(30), FETVEC(30), ZDUM,
                      9 ARRAY4(4000)
                 C
0012                   REAL*8 ARRAY4
                 C
0013                    REAL*4 CSET,CSET3,SELEC1,ANN1,ANN2
                 C
0014                   INTEGER * 2 CSEL,CSEL3, FETVEC, FETVC3
                 C
0015                    LOGICAL * 4 MAPFLG
0016                   EQUIVALENCE (VECSIZ,NOFET3),(MTXSIZ,VARSZ3),(CELWTH,GRSIZE)
                 C
                 C
                 C
                 C
                 C     CALL READER AFTER INITIAL MESSAGE PRODUCED
                 C
0017                   WRITE(TYPEWR, 100)
0018             100   FORMAT( ' I0000  UNSUPERVISED ECHO FUNCTION (PHASE1) ',
                     * 'REQUESTED (NSISUP)' )
0019                   CALL NSIRDR
                 C
                 C     CALL PROCESSING ROUTINE , THEN PRINT TERMINATION MESSAGE
                 C
0020                   CALL NSIINT
0021                   WRITE(TYPEWR, 9000)
0022             9000  FORMAT(' I0000 UNSUPERVISED ECHO FUNCTION (PHASE1) COMPLETED ',
                     * '(NSISUP)')
```

LARS Program Abstract ________

MODULE IDENTIFICATION

Module Name: NS1RDR **Function Name:** NS1ECHO

Purpose: Reads function control cards

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza **Date:** 8/21/77

Latest Revisor: **Date:**

MODULE ABSTRACT

NS1RDR reads and interprets all function control for the field extraction phase of Nonsupervised ECHO (Extraction and Classification of Homogeneous Objects). Checks are made for data validity. Also an intermediate results tape is readied for passing results to the classification phase of the Nonsupervised ECHO algorithm (phase 2).

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NS1RDR

CALL NS1RDR

This section lists the actions taken when the following control cards are read.

INTERMEDIATE-TAPE	The variable RQTAPE is set to the given tape number.
INTERMEDIATE-FILE	The variable RQFILE is set to the given file number.
INTERMEDIATE-INITIALIZE	The local flag INITFG is set to TRUE. This flag is used to trigger a call to TAPMNT to initialize the tape.
INTERMEDIATE-NEWRUN	A new run number is submitted to be used in the ID record of the intermediate tape.
CHANNELS	Subroutine channel is called to return VECSIZ, CSEL3, CSET3 and FETVC3 based upon interpretation of the channels card.
ANNEXATION-MEANS	Significance level 1 of unsupervised mode.
ANNEXATION-VARIANCE	Significance level 2 of unsupervised mode.
CELL-HOMOGENEITY	Cell selection (Level 1) thresholds.
CELL-WIDTH	Value of the width of a (square) cell.
PRINT-MAP	The flag MAPFLG in NS1COM is set to TRUE.

2. Internal Description

NS1RDR uses standard card reader logic in using LARSYS system subroutines CTLWRD, CTLPRM, IVAL and FVAL in reading and interpreting the control cards. Subroutine CHANNEL is used to interpret the CHANNELS card.

NS1RDR begins by calling TSTREQ to clear the stop/suspend flag. Then flags and arrays which will convey control card information are initialized. From this point the program, functions in a loop of reading and interpreting control cards until a DATA or END card is read indicating the card of function control cards. After the control cards have been read, several checks are made on the data. A CHANNELS card must be supplied and the value of the cell width is checked to be equal to or greater than 2. The intermediate tape is then mounted and positioned. Finally a list of selected options is printed.

3. <u>Input Description</u>

Function control cards for NS1ECHO are read by LARSYS system routine CTLWRD.

4. <u>Output Description</u>

Control card error messages are written to both the printer (PRNTR) and the console (TYPEWR). A brief list of these follows:

ERROR ON RESULTS CARD (TAPE OR FILE PARAMETER) - TYPE IN CORRECT CARD

Syntax error in the TAPE or FILE specification. Standard corrective action is taken (requesting the user retype the card from the interactive terminal).

ERROR ON CHANNELS CARD - TYPE CORRECT CARD

Standard corrective action is requested.

NO AUTO OR CHANNELS CARD - TYPE IN CHANNELS CARD

Channels must be given via the CHANNELS card. After this message the keyboard will unlock to accept a CHANNELS card. A response of carriage return will cause the keyboard to unlock again.

CELL SIZE MUST BE GREATER THAN OR EQUAL TO 2 - DEFAULT OF 2 ASSUMED

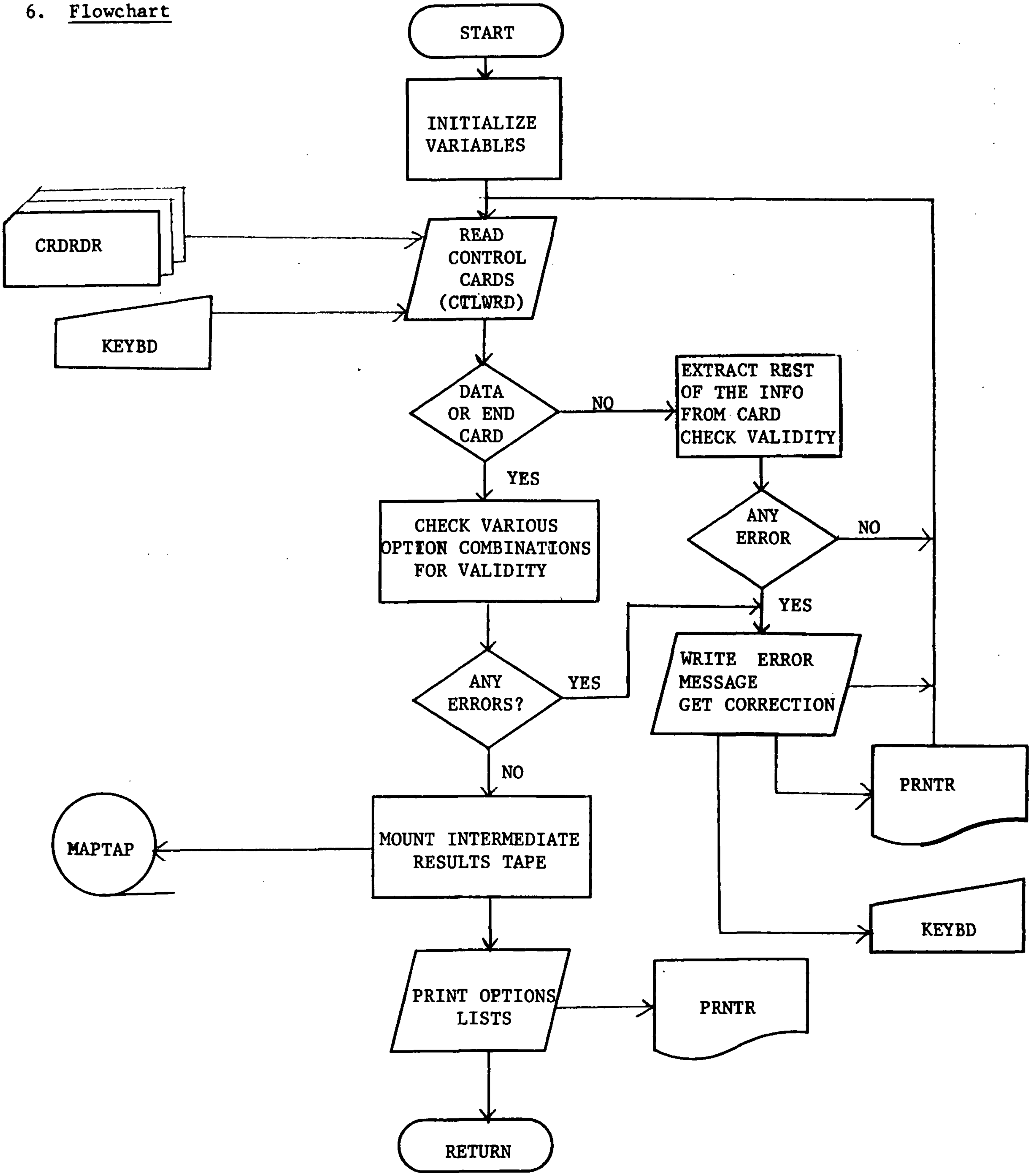
BOTH FILE AND INITIALIZATION OPTION REQUESTED FILE REQUESTED IGNORED. FUNCTION CONTINUES.

The intermediate tape is initialized. Only file 1 can be initialized. In addition to this message the list of options selected is printed.

5. <u>Supplemental Information</u>

See LARSYS System Manual for card reader requirements.

6. Flowchart


START
INITIALIZE VARIABLES
CRDRDR
READ CONTROL CARDS (CTLWRD)
KEYBD
DATA OR END CARD
NO
EXTRACT REST OF THE INFO FROM CARD CHECK VALIDITY
YES
CHECK VARIOUS OPTION COMBINATIONS FOR VALIDITY
ANY ERROR
NO
YES
WRITE ERROR MESSAGE GET CORRECTION
ANY ERRORS?
YES
NO
PRNTR
MAPTAP
MOUNT INTERMEDIATE RESULTS TAPE
KEYBD
PRINT OPTIONS LISTS
PRNTR
RETURN

FILE NSIRDR

```
              C        NSIRDR     EARS  XXXX
              C                   WRITTEN            BY C. V. PUMALAZA
              C
              C*************************************************************************
              C
              C READER ROUTINE FOR FIELD EXTRACTION PHASE OF NONSUPERVISED LOG.
              C
              C*************************************************************************
              C
0001                SUBROUTINE NSIRDR
0002                IMPLICIT INTEGER * 4 (N-Z)
              C
              C NEWRUN = NEW RUN NUMBER TO BE ASSIGNED TO 'INTERMEDIATE' DATA TAPE
              C ROTAPE = TAPE NUMBER OF INTERMEDIATE DATA TAPE
              C ROFILE = FILE WHERE NEW DATA IS TO BE WRITTEN
              C
              C*************************************************************************
              C
0003                COMMON /GLCCOM/ BLANK, CARD(20), CHKOUT, CCPFIL, CLASSR, CLASSX,
                   1     CLUSTX, CONPUT, CPYOUT, CRDRDR, CRDSEG, DATAPE,
                   2     DUPLIP, DUPRUN, ERRMSG, FBPNT,
                   3     FILESV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,
                   4     IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,
                   5     PAGSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,
                   6     RESTRT, RUNFIL, RUNTAB(10,3),
                   7     SDATA, SEPARX, SEPTPX, SPARE(10), TEMPAS(30),
                   8     TPSTAT(6), TTFLDX ,TYPEWR,
                   9     TOP,  ARRAY(12500)
              C
0004                REAL * 8 ARRAY
0005                REAL * 4 FRQCAL(5,30)
0006                INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)
0007                INTEGER * 2 BLANK2
0008                LOGICAL * 4 CHKOUT
0009                LOGICAL * 1 BLANK1
0010                EQUIVALENCE (DATSAV,ID(1)), (CURRUN,ID(3)), (FRQCAL(1),ID(51)),
                   1          (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),
                   2          (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),
                   3          (MAPSAV,TPSTAT(1)),
                   4          (SCPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),
                   5          (COPSER,TPSTAT(5)), (TRAULT,TPSTAT(6)),
                   6          (BLANK,BLANK2,BLANK1)
              C
0011                COMMON /NSICOM/
                   1 ANFLAG, ANN1, ANN2, BUFPTS, BUFRUZ,
                   2 C1, C4, CELSIZ, CELWTH,
                   3 CSET(3,30), CSET3(3,30), INFO(17), JPTS,
                   4 K9, L1, MAPFLG, MAXSIZ, MINSIZ, MTXSIZ, NC(200),
                   5 NEWRUN, NOCLS, NOFEAT, NSR, NUMCLS, NVR,
                   6 PTS, RESULT, ROWSIZ, ROFILE, ROTAPE, SELECT(30),
                   7 TITLE(30), TOTPTS, VARSIZ, VECSIZ,
                   8 CSEL(30), CSEL3(30), FETVC3(30), FETVEC(30), ZDUM,
                   9 ARRAY4(4000)
              C
0012                REAL*8 ARRAY4
              C
0013                REAL*4 CSET,CSET3,SELECT,ANN1,ANN2
              C
0014                INTEGER*2 CSEL,CSEL3, FETVEC, FETVC3
              C
0015                LOGICAL * 4 MAPFLG
              C
0016                EQUIVALENCE (VECSIZ,NOFET3),(MTXSIZ,VARSZ3),(CELWTH,GRSIZE)
              C
              C
              C LOCAL VARIABLES
              C
0017                REAL * 4 AVEC(30), C6
0018                INTEGER*4 KEYWRD(7),INTCOD(4),  TITBCD(12), TITLE(20)
0019                INTEGER*4 ANNCOD(2),CELCOD(2)
0020                INTEGER*2 FETVC3
0021                LOGICAL*1 INITFG
0022                EQUIVALENCE (TITLE(2), TITLE(1))
              C
              C
```


FILE NSIRDR

```
              C LOCAL VARIABLES DESCRIPTION
              C
              C*************************************************************************
              C
              C VECSIZ = NUMBER OF CHANNELS
              C*************************************************************************
              C
              C
0023                DATA KEYWRD /'CHAN','INTE','CELL','ANNE','PRIN','DATA','END '/,
                   *        INTCOD /'NEWR','TAPE','FILE','INIT'/,
                   *        ANNCOD/'MEAN','VARI'/,
                   *        CELCOD/'WIDT','HOMO'/,
                   *        TITBCO/'1'' ',' SA','MPLE',' CLA','SSIF','ICAT','ION ',
                   *               'STUD','Y ..','''//',')   ',' '    '/
              C
              C*************************************************************************
              C
              C
              C        INITIALIZE
              C
              C*************************************************************************
              C
0024                MAPFLG = .FALSE.
0025                INITFG = .FALSE.
0026                NUMCLS = 0
0027                MINSIZ = 10
0028                MAXSIZ = 90000
0029                DO 5 I = 1,12
0030                TITLE(I) = TITBCD(I)
0031          5     CONTINUE
0032                CELWTH = 0
0033                VECSIZ = 0
0034                DO 20 J = 1, 30
0035                CSEL3(J) = 0
0036                FETVC3(J) = 0
0037                DO 10 I = 1,3
0038                CSET3(I,J) = -50000.
0039          10    CONTINUE
0040          20    CONTINUE
0041                ROTAPE = 0
0042                ROFILE = 0
0043                RESULT = MAPTAP
0044                NEWRUN = 0
0045                ERRCOR = 0
              C
              C*************************************************************************
              C
              C        DECODE CONTROL CARDS
              C
              C*************************************************************************
              C
0046                ASSIGN 100 TO GOVEC
0047          101   GO TO GOVEC, (100,600)
0048          100   CALL CTLWRD(CARD,COL,KEYWRD,7,CODE,CRDRDR,ERRCOR)
0049                IF(CODE .EQ. 7) SPARE(2) = 1
0050                IF(ERRCOR .EQ. 4) CALL RTMAIN
0051                IF(ERRCOR .EQ. 2)CALL ERPRNT(139,'STOP')
              C
              C*************************************************************************
              C
              C        DECODE KEYWORD AND GO ACCORDINGLY
              C
              C*************************************************************************
              C
0052                GO TO (200,300,400,500,800,600,600), CODE
              C
              C        ERROR ON CONTROL CARD
              C
0053          110   WRITE(TYPEWR,9110) CARD
0054          9110  FORMAT(5X,20A4)
0055          115   CALL ERPRNT(ERRNUM, 'GO  ')
0056          111   IF(ERRCOR .NE. 3) ERRCOR = 1
0057                GO TO 100
              C
              C*************************************************************************
              C
```

```
              C       CHANNELS CARD
              C
              C*********************************************************************
              C
0058          200     CALL CHANEL(CARD, COL, VECSIZ, CSEL1, CSEL4, FETVC1,&250)
0059                  GO TO 101
              C
              C       ERROR ON CHANNELS CARD
              C
0060          250     ERRNUM = 171
0061                  WRITE(6,1291) VECSIZ
0062          1291    FORMAT('    VECSIZ IS NOW   ',I5)
0063                  GO TO 110
              C
              C*********************************************************************
              C
              C       DECODE INTERMEDIATE TAPE ESPECIFICATIONS
              C
              C*********************************************************************
              C
0064          300     IF(COL .EQ. 72) GO TO 101
0065                  CALL CTLPRM(CARD, COL, INTCOD, 4, CODE, &111)
0066                  IF(CODE .EQ. 4) GO TO 310
0067                  VECSZ = 1
0068                  CALL IVAL(CARD, COL, 1, VECSZ, &350)
0069                  IF(VECSZ .EQ. 0) GO TO 350
0070                  IF(CODE .EQ. 1) NEWRUN = I
0071                  IF(CODE .EQ. 2)ROTAPE = I
0072                  IF(CODE .EQ. 3)ROFILE = I
0073                  GO TO 300
0074          310     INITFG = .TRUE.
0075                  GO TO 300
              C
              C       ERROR ON TAPE CARD
              C
0076          350     ERRNUM = 348
0077                  GO TO 110
              C
              C*********************************************************************
              C
              C       CELL    CARD
              C
              C*********************************************************************
              C
0078          400     IF(COL .EQ. 72)GO TO 100
0079                  CALL CTLPRM(CARD, COL, CELCOD, 2,CODE, &111)
0080                  IF(CODE .EQ. 2)GO TO 420
0081                  CALL IVAL(CARD,COL, 1, I,&111)
0082                  CELWTH = I
0083                  GO TO 400
0084          420     IV=30
0085                  CALL FVAL(CARD, COL, AVEC, IV, &111)
0086                  C6 =AVEC(1)
0087                  ISEL = IV
0088                  DO 450 J=1,ISEL
0089          450     SELECI(J) = AVEC(J)
0090                  GO TO 400
              C
              C*********************************************************************
              C
              C       DECODE ANNEXATION  CARD
              C
              C*********************************************************************
              C
0091          500     IF(COL .EQ. 72) GO TO 100
0092                  CALL CTLPRM(CARD, COL, ANNCOD, 2, CODE, &111)
0093                  IV = 1
0094                  CALL FVAL(CARD, COL, AVEC, IV, &111)
0095                  C6 = AVEC(1)
0096          560     IF(CODE .EQ.1) ANN2 =C6
0097                  IF(CODE .EQ. 2 )ANN1 = C6
0098                  GO TO 500
              C
              C*********************************************************************
              C
              C       DECODE PRINT CARD
```



```
              C
              C*********************************************************************
              C
0099          800     CALL CTLPRM(CARD, COL, PRICOD, 1, CODE, &111)
0100                  MAPFLG = .TRUE.
0101                  GO TO 100
              C
              C
              C*********************************************************************
              C
              C       DATA CARD ENCOUNTERED.. CHECK CELWTH AND CHANNELS
              C
              C*********************************************************************
              C
0102          600     IF(VECSIZ .GT. 0) GO TO 620
0103                  ASSIGN 600 TO GOVEC
0104                  ERRNUM = 343
0105                  ERRCOR = 3
0106                  GO TO 115
              C
0107          620     IF(VECSIZ .LE. ISEL) GO TO 630
0108                  IF(ISEL .GT. 0) GO TO 640
0109                  ASSIGN 620 TO GOVEC
0110                  GO TO 645
0111          640     I = ISEL + 1
0112                  DO 642 J = I, VECSIZ
0113                  SELECI(J) = SELECI(ISEL)
0114          642     CONTINUE
0115                  GO TO 630
              C
              C       ERROR ON CHANNEL OR PARAMETER CARD
              C
0116          645     ERRNUM = 124
0117                  ERRCOR = 3
0118                  GO TO 115
              C
0119          630     IF(CELWTH .GE. 2)GO TO 650
0120                  CELWTH = 2
0121                  WRITE(TYPEWR, 9620)
0122                  WRITE(PRNTR, 9620)
0123          9620    FORMAT('   EXXX CELL SIZE MUST BE GREATER THAN OR EQUAL TO 2-
                     *DEFAULT OF 2 ASSUMED (NSIRDR) ')
              C
              C*********************************************************************
              C
              C       MOUNT INTERMEDIATE TAPE
              C
              C*********************************************************************
              C
0124          650     MODE = 1
0125                  IF( .NOT. INITFG) GO TO 680
0126                  IF( ROFILE .EQ. 0) GO TO 660
0127                  WRITE(TYPEWR, 9650)
0128                  WRITE(PRNTR, 9650)
0129          9650    FORMAT(' IC050  BOTH FILE AND INITIALIZATION OPTION REQUESTED. ',
                     *        'FILE REQUESTED IGNORED. FUNCTION CONTINUES (NSIRDR)')
0130                  ROFILE = 0
0131          660     CALL TAPMNT(ROTAPE, ROFILE, MODE)
0132                  GO TO 700
0133          680     CALL TAPMNT(ROTAPE, ROFILE, MODE)
              C
              C*********************************************************************
              C
              C       PRINT OUT PROCESSING PARAMETERS
              C
              C*********************************************************************
              C
0134          700     WRITE(PRNTR,9700) VECSIZ, CELWTH, NEWRUN,ROTAPE,ROFILE,
                     * ANN2,ANN1,(SELECI(J), J= 1, VECSIZ)
0135          9700    FORMAT(///'0READER INFORMATION 0'//
                     * '        NUMBER OF CHANNELS =', I3//
                     * '        CELL WIDTH =',I2//
                     * '        NEW RUN NUMBER =',I9,'  TAPE =',I5,'  FILE =',I3//
                     * '        MEAN AND VARIANCE THRESHOLDS ARE ',F10.5,'  AND',F10.5//
                     * '        CELL HOMOGENEITY THRESHOLDS ARE ',(T40,5F10.5))
0136                  RETURN
```

LARS Program Abstract ____________

MODULE IDENTIFICATION

Module Name: TAPMNT   Function Name: Non-Supervised ECHO Functional Support

Purpose: Mounts and positions intermediate tapes

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza   Date: 8/20/77

Latest Revisor: ____________   Date: ____________

MODULE ABSTRACT

TAPMNT mounts and positions the intermediate tape used in Phase 1 and Phase 2 of Nonsupervised ECHO.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

TAPMNT

CALL TAPMNT (RQTAPE, RQFILE, MODE)

Input Arguments

RQTAPE	I*4 Number of requested tape. A tape number of 0 is a request for a scratch tape.
RQFILE	I*4 Number of requested file. If RQFILE = 0 then the tape will be initialized by writing a record type 1 (ID record) on the intermediate tape with filetype = 0.
MODE	I*4 Indicating usage of TAPMNT. MODE = 0 indicates that an intermediate results tape is being mounted with the ring out (only for reading).

Output Arguments

RQFILE	Current file position of tape.

2. Internal Description

See Output description.

3. Input Description

The record type of the intermediate results tape is read for each file up to and including the file needed.

4. Output Description

The following information messages are issued under the circumstances listed:

10042 - is typed when a tape has been mounted and before TAPMNT positions it. This message is not typed when the tape is being initialized or when the correct tape number was already mounted.

10043 - is typed when MODE = 1 and the file tape has results in it (Check $\neq$ 0).

10045 - is typed when the tape is correctly positioned. This is not typed when initializing a tape.

After 10043 the user is asked whether he wished to overwrite the file, respecify a new intermediate results card or terminate the function.

10100 - is typed to allow entry of the new intermediate results card. This occurs when the user requests to respecify the intermediate results card.

The following error messages are typed under the conditions listed:

E361 - is written when the tape is being filed forward and a file is encountered with filetype other than zero (Check $\neq$ 0) before the requested file is reached and MODE = 0.

E362 - is written when the circumstances for E361 occurs and MODE = 1.

For message texts refer to the User's Manual.

5. Supplemental Information

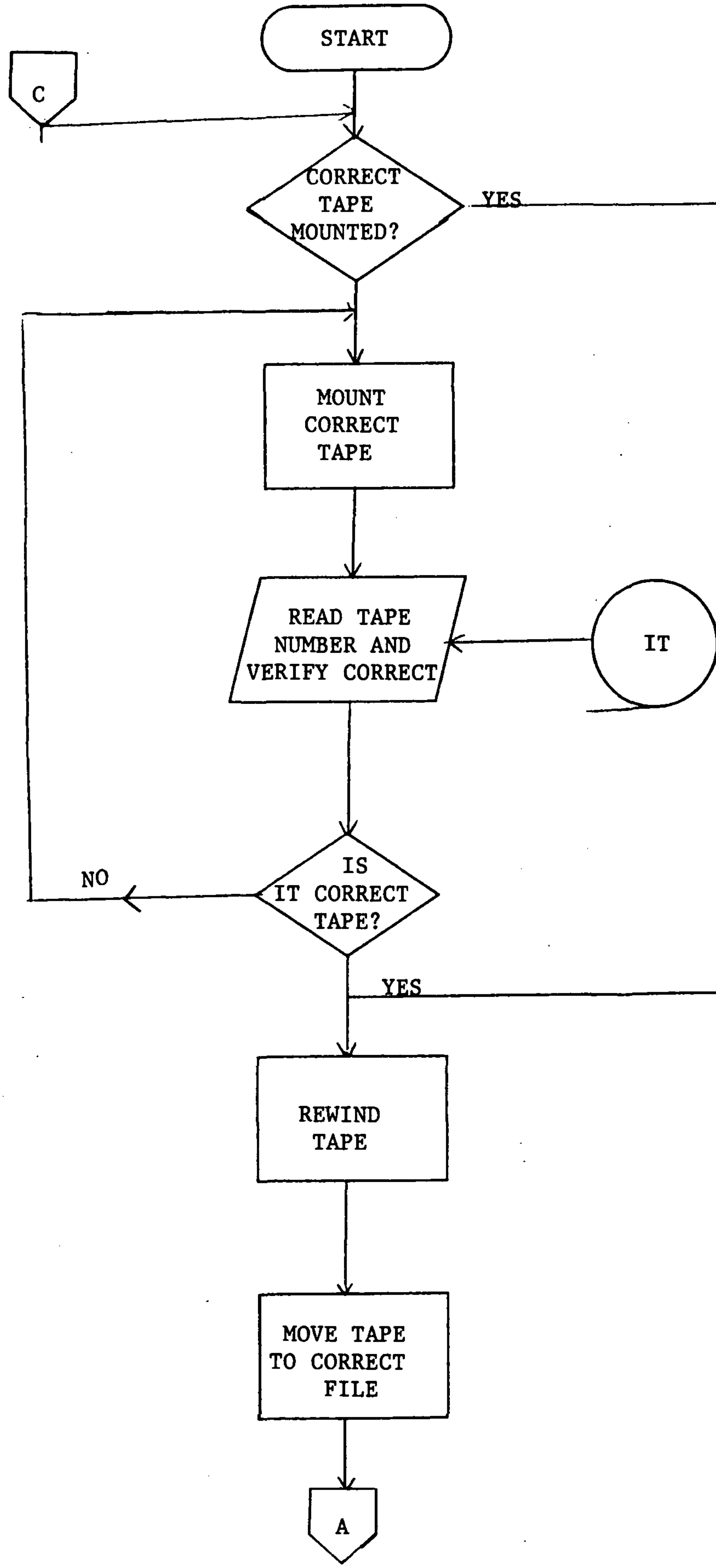
Input

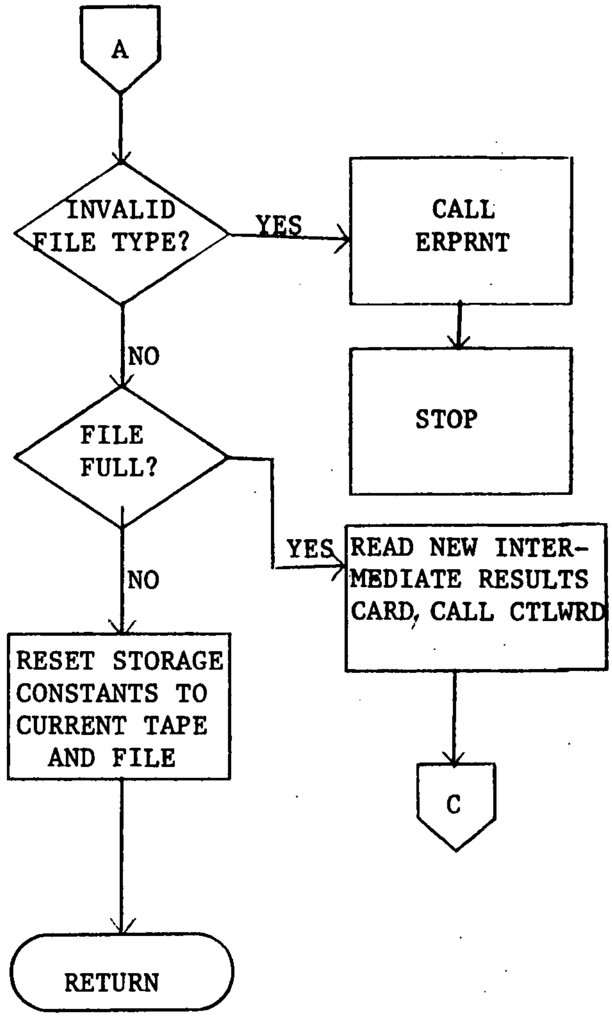
If a tape is mounted on the device and it is the incorrect tape number (as noted from the appropriate status words in CLODOM) TOPRV is called to unload the tape before the correct tape is mounted. If the correct tape is mounted TAPMNT assumes that the file number (as recorded in GLOCOM) is correct and moves the tape backwards or forwards to find the requested file.

Output

The tape is mounted with ring in for MODE = 1 and with ring out for MODE - 0. The tape is left positioned at the beginning of the requested file. When the tape is initialized LARSYS System routine TOPRW is used to do this.

6. Flowchart


START
C
CORRECT TAPE MOUNTED?
YES
MOUNT CORRECT TAPE
READ TAPE NUMBER AND VERIFY CORRECT
IT
NO
IS IT CORRECT TAPE?
YES
REWIND TAPE
MOVE TAPE TO CORRECT FILE
A


A
INVALID FILE TYPE?
YES
CALL ERPRNT
STOP
NO
FILE FULL?
YES
READ NEW INTERMEDIATE RESULTS CARD, CALL CTLWRD
C
NO
RESET STORAGE CONSTANTS TO CURRENT TAPE AND FILE
RETURN

FILE TAPMNT

```
                C        TAPMNT    LARS  OXXX                                    MMT00010
                C                                                                MMT00020
                C****************************************************************MMT00030
                C                                                                MMT00040
                C TAPMNT      MOUNTS AND POSITIONS INTERMEDIATE RESULTS TAPES    MMT00050
                C             WRITTEN BY C. A. PUMALAZA                          MMT00060
                C                                                                MMT00070
                C****************************************************************MMT00080
                C                                                                MMT00090
 0001                 SUBROUTINE TAPMNT (ROTAPE,ROFILE,MODE)                     MMT00100
                C                                                                MMT00110
 0002                 IMPLICIT INTEGER * 4 (A-Z)                                 MMT00120
 0003                 COMMON /GLOCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX, MMT00130
                     1     CLUSTX, CONPUT, CPYOUT, CRDROR, CRDSEQ, DATAPE,       MMT00140
                     2     DUPLTP, DUPRUN, ERRMSG, FBPNT,                        MMT00150
                     3     FILESV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,     MMT00160
                     4     IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,                 MMT00170
                     5     PACSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,           MMT00180
                     6     RESTRT, RUNFIL, RUNTAB(10,1),                         MMT00190
                     7     SDATA, SLPARX, SLPTPX, SPARE(10), TIMPAS(10),         MMT00200
                     8     TPSTAT(6), TTFLDX ,TYPEWR,                            MMT00210
                     9     TOP, ARRAY(12500)                                     MMT00220
                C                                                                MMT00230
 0004                 REAL * 8 ARRAY                                             MMT00240
 0005                 REAL * 4 FROCAL(5,10)                                      MMT00250
 0006                 INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5) MMT00260
 0007                 INTEGER * 2 BLANK2                                         MMT00270
 0008                 LOGICAL * 4 CHKOUT                                         MMT00280
 0009                 LOGICAL * 1 BLANK1                                         MMT00290
 0010                 EQUIVALENCE (DATSAV,ID(1)), (CURRUN,ID(3)), (FROCAL(1),ID(51)), MMT00300
                     1        (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),MMT00310
                     2        (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),          MMT00320
                     3        (MAPSAV,TPSTAT(1)),                                MMT00330
                     4        (SLPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),MMT00340
                     5        (COPSCR,TPSTAT(5)), (TRAUCT,TPSTAT(6)),            MMT00350
                     6        (BLANK,BLANK2,BLANK1)                              MMT00360
                C                                                                MMT00370
                C                                                                MMT00380
 0011                 INTEGER * 4 RESCOD(3), CLACOD(3), IDD(200)                 MMT00390
 0012                 EQUIVALENCE  (IDD(1),TNO), (IDD(2), TFO),(IDD(3), CHECK)   MMT00400
 0013                 DATA IDD/200*0/                                            MMT00410
 0014                 DATA  INTCOD/'INTE'/, RESCOD/'OVER','RESP','TERM'/,        MMT00420
                     1     CLACOD/'TAPE','FILE','TMIT'/                          MMT00430
                C                                                                MMT00440
                C****************************************************************MMT00450
                C                                                                MMT00460
                C ANS    = REPLY FROM USER AS TO WHETHER TO CONTINUE WITH RUN    MMT00470
                C ERROR  = ERROR CODE FROM CPFUNC (NOT USED)                      MMT00480
                C K      = USED AS DUMMY READING VERSION NUMBER FROM TAPE         MMT00490
                C MODE   = INDICATES FUNCTION OF TAPMNT, SUPPLIED FROM CALLER     MMT00500
                C          = 1 MEANS WISH TO GET TAPE READY FOR WRITING RESULTS  MMT00510
                C          = 0 MEANS WISH TO GET TAPE READY FOR READING RESULTS  MMT00520
                C ROFILE = REQUESTED FILE NUMBER IN TAPE TO BE READIED. IF ROFILE MMT00530
                C          IS INPUT AS A 0, THEN TAPMNT IS TO WRITE AN INITIAL   MMT00540
                C          RECORD ONTO THE TAPE.                                 MMT00550
                C ROTAPE = REQUESTED TAPE NUMBER TO BE MOUNTED.A ZERO MEANS SCRATCH TAPEMMT00560
                C TFO    = FILE NUMBER AS READ FROM INTERMEDIATE RESULTS TAPE    MMT00570
                C TNO    = TAPE NUMBER AS READ FROM INTERMEDIATE RESULTS TAPE    MMT00580
                C UNIT   = DSRN OF TAPE BEING MOUNTED. THIS WILL BE MAPTAP OR CPYOUT MMT00590
                C                                                                MMT00600
                C****************************************************************MMT00610
                C                                                                MMT00620
 0015                 INITFG = 0                                                 MMT00630
                C                                                                MMT00640
                C                                                                MMT00650
 0016           100   UNIT = MAPTAP                                              MMT00660
 0017                 IF(MODE .EQ. 0) GO TO 108                                  MMT00670
 0018                 FILNO = FILESV                                             MMT00680
                C                                                                MMT00690
                C        IF CORRECT TAPE IS MOUNTED GO TO POSITIONING            MMT00700
                C                                                                MMT00710
 0019                 IF (MAPSAV .EQ. ROTAPE) GO TO 130                          MMT00720
 0020                 CALL TCPRU(MAPTAP)                                         MMT00730
 0021                  GO TO 110                                                 MMT00740
 0022           108   UNIT = CPYCUT                                              MMT00750
 0023                 FILNO = COPFIL                                             MMT00760
```


FILE TAPMNT

```
 0024                 IF(COPSCR .EQ. ROTAPE) GO TO 130                           MMT00770
 0025                 CALL TCPRU(CPYOUT)                                         MMT00780
 0026             110 IF (MODE .EQ. 1) CALL MOUNT (ROTAPE,MAPTAP,'RI')           MMT00790
 0027                 IF (MODE .EQ. 0) CALL MOUNT (ROTAPE,CPYOUT,'RO')           MMT00800
                C                                                                MMT00810
                C                                                                MMT00820
 0028                 FILNO = 1                                                  MMT00830
                C                                                                MMT00840
                C        ROFILE OF 0 MEANS THE CALL IS TO INITIALIZE THE TAPE    MMT00850
                C                                                                MMT00860
 0029                 IF (ROFILE .NE. 0) GO TO 120                               MMT00870
 0030                 JCOUNT = 800                                               MMT00880
 0031                 IDD(1) = ROTAPE                                            MMT00890
 0032                 IDD(2) = 1                                                 MMT00900
 0033                 CALL TCPWR(UNIT,JCOUNT,ERROR,IDD)                          MMT00910
 0034                 ROFILE = 1                                                 MMT00920
 0035                 CALL TCPRW(UNIT)                                           MMT00930
 0036                 GO TO 390                                                  MMT00940
 0037             120 WRITE (TYPEWR,9120) ROTAPE, ROFILE                         MMT00950
 0038            9120 FORMAT (' 10042  POSITIONING INTERMEDIATE RESULTS TAPE',I5, MMT00960
                     1        ' TO FILE',I4,' (TAPMNT)')                         MMT00970
 0039                 CALL TCPRW(UNIT)                                           MMT00980
                C                                                                MMT00990
                C        READ FIRST RECORD AND SEE IF CORRECT TAPE IS MOUNTED    MMT01000
                C                                                                MMT01010
 0040                 JCOUNT = 800                                               MMT01020
 0041                 CALL TCPRD(UNIT, JCOUNT, ERROR, ICO,2, WLF)                MMT01030
 0042                 CALL TCPRW(UNIT)                                           MMT01040
 0043                 IF (TNO .EQ. ROTAPE) GO TO 130                             MMT01050
 0044                 CALL CPFUNC (15,'M CP WRONG TAPE',ERROR)                   MMT01060
 0045                 CALL TOPRU(UNIT)                                           MMT01070
 0046                 GO TO 110                                                  MMT01080
                C                                                                MMT01090
                C        POSITION RESULTS TAPE TO CORRECT FILE                   MMT01100
                C                                                                MMT01110
 0047           130   IF (ROFILE .GT. 1) GO TO 140                               MMT01120
                C                                                                MMT01130
                C                                                                MMT01140
 0048             135 FILNO = 1                                                  MMT01150
 0049                 CALL TCPRW(UNIT)                                           MMT01160
 0050                 GO TO 200                                                  MMT01170
 0051             140 J = ROFILE - FILNO                                         MMT01180
                C                                                                MMT01190
                C        J IS THE NUMBER OF FILES THE TAPE MUST BE MOVED         MMT01200
                C                                                                MMT01210
 0052                 IF (J) 150, 200, 170                                       MMT01220
 0053             150 J = -J                                                     MMT01230
                C                                                                MMT01240
                C        THE DO 160 LOOP BACKS THE TAPE UP TO THE CORRECT FILE   MMT01250
                C                                                                MMT01260
 0054                 DO 160 I = 1,J                                             MMT01270
 0055                 FILNO = FILNO -1                                           MMT01280
 0056             160 CALL TCPBF(UNIT)                                           MMT01290
 0057                 GO TO 200                                                  MMT01300
                C                                                                MMT01310
                C        THE DO 180 LOOP FORWARD FILES THE TAPE AND CHECKS       MMT01320
                C        TO BE SURE THAT FILES ARE REALLY THERE TO SPACE OVER    MMT01330
                C                                                                MMT01340
 0058             170 DO 180 I = 1, J                                            MMT01350
 0059                 JCOUNT = 800                                               MMT01360
 0060                 CALL TCPRD(UNIT, JCOUNT, ERROR, ICO, 2, WLF)               MMT01370
 0061                 IF (CHECK .EQ. 0) GO TO 210                                MMT01380
 0062                 FILNO = FILNO + 1                                          MMT01390
 0063             180 CALL TCPFF(UNIT)                                           MMT01400
                C                                                                MMT01410
                C        IF FILNO AND ROFILE ARE UNEQUAL , THE PROGRAM HAS GCCURED. MMT01420
                C                                                                MMT01430
 0064             200 IF (FILNO .NE. ROFILE) CALL ERPRNT (ICO,'STOP')            MMT01440
                C                                                                MMT01450
                C        NOW READ THE FIRST RECORD OF THE FILE AND CHECK VALIDITY MMT01460
                C                                                                MMT01470
 0065                 JCOUNT = 800                                               MMT01480
 0066                 CALL TCPRD(UNIT, JCOUNT, ERROR, ICO, 2, WLF)               MMT01490
                C                                                                MMT01500
                C                                                                MMT01510
 0067                 IF (MODE .EQ. 0) GO TO 300                                 MMT01520
```

FILE TAPMNT

```
              IF(CHECK .NE. 0) GO TO 230
              GO TO 300
      210     CALL ERPRNT(361+MODE,'GO')
              WRITE(PRNTR,9210) I
              WRITE(TYPEWR, 9210) I
     9210     FORMAT(10X,'FILE',I3)
              CALL RTMAIN
C
C
      230     WRITE (TYPEWR, 9230)
              WRITE (PRNTR, 9230)
        9230 FORMAT (' 10043  FILE HAS RESULTS IN IT - DO YOU WISH TO ',
            1          'OVERWRITE THE FILE, RESPECIFY THE ENTIRE RESULTS CARD,',/
            2            ' OR TERMINATE THE JOB (OVERWRITE, RESPECIFY, TERMINATE) '
            3          ,'      (TAPMNT)')
        235 COL = 0
            ERRCOR = 4
            CALL CTLWRF (CARD, COL, RESCOD, 3, CODE, KEYBD, ERRCOR)
            IF (ERRCOR .EQ. 4) CALL RTMAIN
            GO TO (300,240,244), CODE
        240 WRITE(TYPEWR,9250)
            WRITE (KEYBD, 9250)
       9250 FORMAT (' 10100  ENTER INTERMEDIATE CARD          (TAPMNT)')
            COL = 0
            ERRCOR = 3
        242 CALL CTLWRF (CARD, COL, INTCOD, 1, CODE, KEYBD, ERRCOR)
            IF (ERRCOR .EQ. 4) CALL RTMAIN
        243 IF (COL .EQ. 72) GO TO 100
            CALL CTLPRM (CARD, COL, CLACOD, 3, CODE, &245)
            GO TO (250,260,270),CODE
C
C TERMINATE OPTION
C
        244 CALL RTMAIN
C
C RETYPE RESULTS CARD AFTER ERROR IN ENTRY
C
        245 ERRCOR = 1
            GO TO 242
C
C PROCESS PARAMETER ERRORS
C
        248 WRITE (TYPEWR, 9248) CARD
       9248 FORMAT (5X,20A4)
            CALL ERPRNT (ERRNUM, 'GO')
            GO TO 242
C
C DECODE TAPE SPECIFICATION
C
        250 VECSZ = 1
            CALL IVAL (CARD, COL, RQTAPE, VECSZ, &252)
            IF (VECSZ .EQ. 0) GO TO 252
            GO TO 243
        252 ERRNUM = 348
            GO TO 248
C
C FILE SPECIFICATION
C
        260 VECSZ = 1
            CALL IVAL (CARD, COL, RQFILE, VECSZ, &252)
            IF (VECSZ .EQ. 0) GO TO 252
            GO TO 243
        265 ERRNUM = 347
            COL = 0
            ERRCOR = 3
            GO TO 242
C
C INITIALIZE SPECIFICATION
C
        270 INITFG = 1
            GO TO 243
C
C
        300 WRITE (TYPEWR,9300)
       9300 FORMAT (' 10045  RESULTS TAPE MOUNTED AND POSITIONED',
            1          '      (TAPMNT)')
```


FILE TAPMNT

```
              CALL TCPRF(UNIT)
      390     IF(MODE .EQ. 0) GO TO 400
              MAPSAV = RQTAPE
              FILESV = FILNO
              RETURN
      400     CDPSER = RQTAPE
              CDPFIL = FILNO
              RETURN
          500 CALL ERPRNT(365,'STOP')
          530 CALL ERPRNT (261,'STOP')
              END
```

LARS Program Abstract ______

MODULE IDENTIFICATION

Module Name: NS1INT **Function Name:** NS1ECHO

Purpose: Initialization Including array base computation

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza **Date:** 8/23/77

Latest Revisor: ______ **Date:** ______

MODULE ABSTRACT

NS1INT reads the area to be processed and computes the array bases for storage allocation of the variables used by the subroutine that performs the field extraction phase (NSECHO).

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NS1INT

CALL NS1INT

There are no arguments passed to NS1INT. It is called by NS1SUP to allocate storage prior to the call of NSECHO.

2. Internal Description

NS1INT performs the following functions:

1 - The data card of the area to be processed is read and interpreted using LARSYS system routine LAREAD.

2 - The new ID record to be used in the intermediate results tape is built using part of the ID record from the multispectral image storage tape, (MIST) and information read by NS1RDR.

3 - Check the validity of the channels requested comparing them with the ones present on the MIST.

4 - Compute the space needed for arrays to be used.

5 - Prints data summary.

6 - Call NSECHO to perform the field extraction phase.

7 - Call RDWRTE which transfer results written on disk by NSECHO to the intermediate results tape.

3. Input Description

The field description card of the area to be processed is read by a call to LARSYS system routine LAREAD.

4. Output Description

Information concerning the parameters and the field selected are written on the printer (PRNTR).

Information and error messages are also printed.

LAST SAMPLE NUMBER XXXXX OF FIELD XXXXXXXX EXCEEDS LAST SAMPLE ON TAPE.
LAST SAMPLE RESET TO XXXXX.

The last sample number is reset to the one present on the MIST tape.

YOU HAVE REQUESTED A CHANNEL NOT AVAILABLE IN THIS RUN. REQUEST CANCELLED.

A channel number was requested which does not exist on the data run.

FIELDS EXCEEDS LIMITS OF DATA. FIELD IGNORED. FIELD DESIGNATION FOLLOWS:

The first column of the field is greater than the last column of data on the tape.

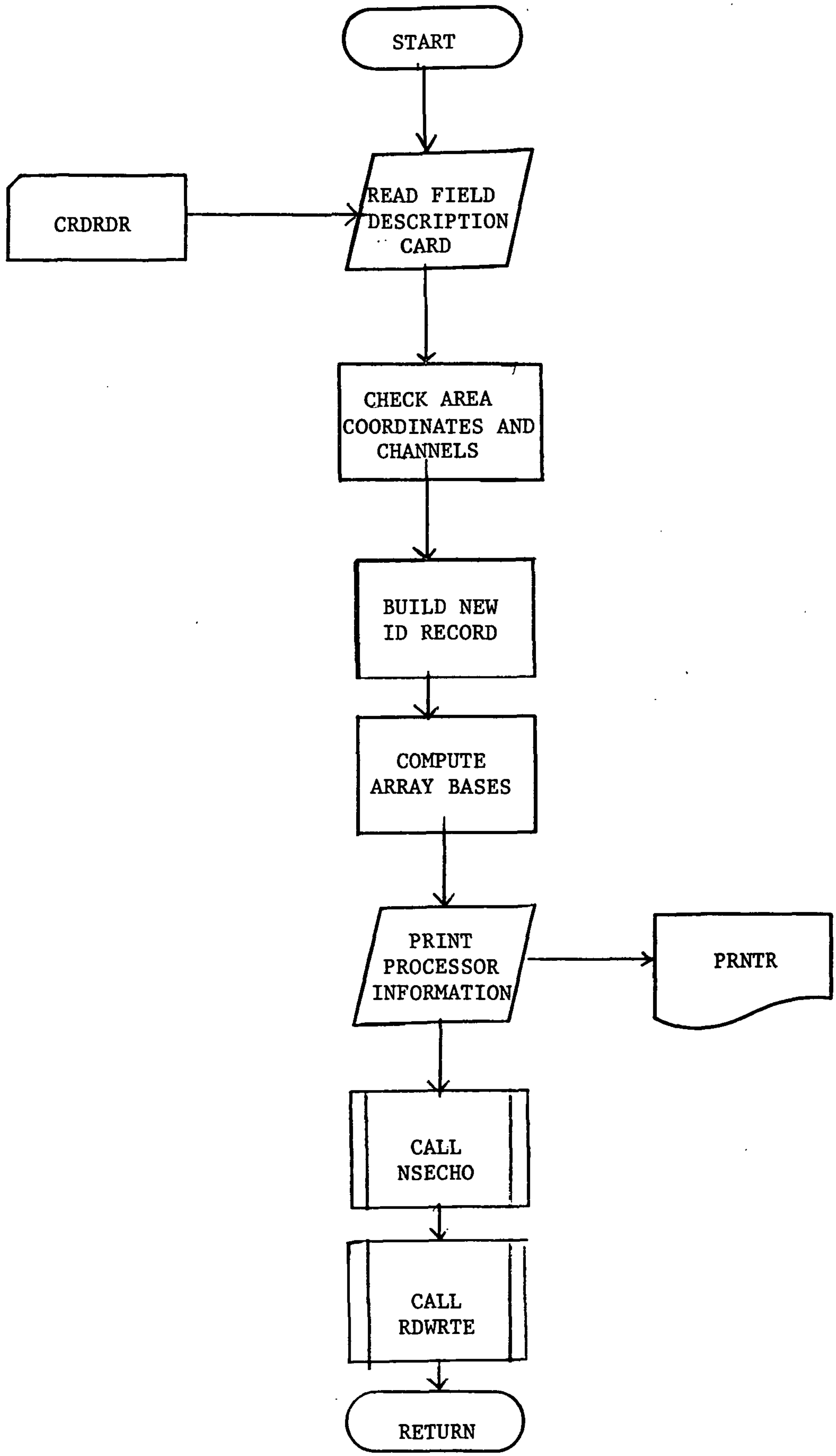
A DATA DECK IS MISSING - FUNCTION TERMINATED

The user has forgotten to include the field description card.

5. Supplemental Information

Not applicable.

6. Flowchart


START
CRDRDR
READ FIELD DESCRIPTION CARD
CHECK AREA COORDINATES AND CHANNELS
BUILD NEW ID RECORD
COMPUTE ARRAY BASES
PRINT PROCESSOR INFORMATION
PRNTR
CALL NSECHO
CALL RDWRTE
RETURN

```
          C     NSLINT     LARS XXXX
          C
          C
          C      WRITTEN BY C. A. POMALAZA
          C
          C*********************************************************************
          C
          C   PERFORMS ARRAY ALLOCATIONS FOR NONSUPERVISED FIELD EXTRACTION
          C
          C*********************************************************************
          C
0001             SUBROUTINE NSLINT
0002             IMPLICIT INTEGER * 4 (A-Z)
          C
0003             COMMON /GLCCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX,
               1     CLUSTX, CONPUT, CPYOUT, CRDRDR, CRDSEQ, DATAPE,
               2     DUPLTP, DUPRUN, ERRMSG, FBPNT,
               3     FILESV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,
               4     IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,
               5     PAGSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,
               6     RESTRT, RUNFIL, RUNTAB(10,3),
               7     SDATA, SEPARX, SEPTPX, SPARE(10), TEMPAS(30),
               8     TPSTAT(6), TTFLOX ,TYPCWR,
               9     TOP, ARRAY(12500)
          C
0004             REAL * 8 ARRAY
0005             REAL * 4 FROCAL(5,30)
0006             INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)
0007             INTEGER * 2 BLANK2
0008             LOGICAL * 4 CHKOUT
0009             LOGICAL * 1 BLANK1
0010             EQUIVALENCE (DATSAV,ID(1)), (CURRUN,ID(3)), (FROCAL(1),ID(51)),
               1         (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),
               2         (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),
               3         (MAPSAV,TPSTAT(1)),
               4         (SCPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),
               5         (COPSER,TPSTAT(5)), (TRAOUT,TPSTAT(6)),
               6         (BLANK,BLANK2,BLANK1)
          C
          C
0011             COMMON /NSICOM/
               1 AMFLAG, ANN1, ANN2, BUFPTS, BUFROZ,
               2 C1, C4, CELSIZ, CELWTH,
               3 CSET(5,30), CSET3(3,30), INFO(17), JPTS,
               4 K9, L1, MAPFLG, MAXSIZ, MINSIZ, MTXSIZ, ND(200),
          5    5 NEWRUN, NCCLS, NOFEAT, NSR, NUMCLS, NVR,
               6 PTS, RESULT, ROWSIZ, RQFILE, RQTAPE, SELEC1(30),
               7 TTITLE(30), TOTPTS, VARSIZ, VECSIZ,
               8 CSEL(30), CSEL3(30), FETVC3(30), FETVCC(30), ZDUM,
               9 ARRAY4(4000)
          C
0012             REAL*8 ARRAY4
          C
0013              REAL*4 CSET,CSET3,SELEC1,ANN1,ANN2
          C
0014             INTEGER * 2 CSEL, CSEL3, FETVEC, FETVC3
          C
0015              LOGICAL * 4 MAPFLG
          C
0016             EQUIVALENCE (VECSIZ,NOFET3),(MTXSIZ,VARSZ3),(CELWTH,GRSIZE)
          C
          C
          C
          C DEFINE LOCAL VARIABLES
          C
0017             REAL*4 ARRAY8(8000)
0018             EQUIVALENCE (ARRAY4(1),ARRAY8(1))
          C
0019             INTEGER * 4 TITLE(20)
0020             REAL*4 ABS, ALOG
          C
0021             REAL*4 FROC2(5,30)
0022              EQUIVALENCE (FROC2(1), ND(51))
0023             EQUIVALENCE (TTITLE(2), TITLE),
               & (FSTLIN,INFO(4)) , (LSTLIN,INFO(5)) , (LININT,INFO(6)),
               & (FSTCOL,INFO(7)) , (LSTCOL,INFO(8)) , (COLINT,INFO(9)),
```



```
               & (LINEA,ID(21)) , (LINEB,ID(22)) , (COLA,ID(24)) , (COLB,ID(25))
0024             DATA END/'END '/
          C
          C*********************************************************************
          C
          C VARIABLES DESCRIPTION
          C
          C*********************************************************************
          C
          C SELOP = 2 -- UNSUPERVISED MUV CELL SELECTION  VARIANCE
          C
          C
          C ANNOP  = 2 -- UNSUPERVISED MUV G.L.R. ANNEXATION
          C RECSKP = HOW MANY RECORDS TO DESIRED AREA
          C SCCR   = NUMBER OF COLUMNS TO NEXT CELL DATA
          C NOCELL = NO OF CELLS
          C
0025             CONTINUE
          C
          C ARRAY SIZES
          C
          C STACK    1 HALFWORD FOR EACH CELL HORIZONTALLY
          C CLOSED   1 BYTE     FOR EACH CELL HORIZONTALLY
          C OPEN     1 BYTE     FOR EACH CELL HORIZONTALLY
          C FLDSIZ   1 FULLWORD FOR EACH CELL HORIZONTALLY
          C BDATA    (SAMP*CHAN+8)/8 BYTES FOR EACH LINE (DATA BUFFER)
          C HDATA    (PTS+1)/2      (DATA BUFFER)
          C
          C*********************************************************************
          C
          C
          C*********************************************************************
          C
0026             CALL TSTREC(NOTE)
          C*********************************************************************
          C                                                                   *
          C      READ IN AREA TO BE CLASSIFIED
          C*********************************************************************
0027             IF (CARD(1) .EQ. END) CALL ERPRNT(254,'STOP')
0028             CALL LAREAD(INFO, CODE, CRDRDR, 0, 0)
0029             IF(CODE .NE. 0)CALL ERPRNT(104,'STOP')
0030             CALL GETRUN(INFO(1), DATAPE, ID, ERROR, RUNTAB, IMARK)
0031             IF(ERROR .GT. 0) CALL RUNERR(ERROR,INFO(1))
          C*********************************************************************
          C                                                                   *
          C      INITIALIZE NEW ID                                            *
          C                                                                   *
          C*********************************************************************
0032             DO 100 I = 1,200
0033             ND(I) = ID(I)
0034      100    CONTINUE
0035             SAVID = ND(6)
          C*********************************************************************
          C                                                                   *
          C      SET UP CALIBRATION VECTOR                                    *
          C                                                                   *
          C*********************************************************************
0036             DO 120 J=1, 30
0037             DO 130 I=1,3
0038             IF(CSET3(I,J) .EQ. -50000.) CSET3(I,J) = FROCAL(I+2,J)
0039             IF(CSET3(I,J) .NE. -50000.) FROC2(I+2,J) = CSET3(I,J)
0040      130    CONTINUE
0041      120    CONTINUE
          C*********************************************************************
          C                                                                   *
          C      CHECK AREA COORDINATES
          C                                                                   *
          C*********************************************************************
0042             LSAMNO = ID(6)- 6
0043             IF(INFO(7) .GT. LSAMNO) GO TO 160
0044             IF(INFO(8) .LE.  LSAMNO) GO TO 140
0045             WRITE(TYPEWR, 9140)INFO(8), INFO(2), INFO(3), LSAMNO
0046     9140    FORMAT(' IC048 LAST SAMPLE NUMBER (',I5,' ) OF FIELD',
                *1X,2A4,' EXCEEDS LAST '/8X,' SAMPLE ON TAPE. LAST SAMPLE RESET ',
                *'TO ',I5,'   (NSINT) ')
0047             INFO(8) = LSAMNO
```

```
0048     140  IF(INFO(4) .GT. ID(20)) GO TO 160
0049          IF(INFO(5) .GT. ID(20)) INFO(5) = ID(20)
         C*********************************************************************
         C
         C        CHECK CHANNELS
         C
         C*********************************************************************
0050          N = NOFLT3
0051          DO 150 I = 1,N
0052          IF( FETVC3(I) .LE. ID(5)) GO TO 150
0053          NOFLT3 = NOFLT3 -1
0054          WRITE(PRNTR, 9150)
0055          WRITE(TYPEWR, 9150)
0056     9150 FORMAT( ' I0046  YOU HAVE REQUESTED A CHANNEL NOT AVAILABLE ',
             *'IN THIS RUN, REQUEST CANCELLED (NSIINT)')
0057          STOP 10
0058     150  CONTINUE
0059          GO TO 200
         C*********************************************************************
         C
         C        ILLEGAL AREA REQUESTED FOR CLASSIFICATION
         C
         C*********************************************************************
0060     160  WRITE (PRNTR, 9160)INFO
0061          WRITE(TYPEWR, 9160) INFO
0062     9160 FORMAT('0',1X,I8,2X,2A4,2X,6I5,9A4,A2)
0063          CALL ERPRNT(242,'STOP')
         C*********************************************************************
         C
         C        SET UP THE NUMBER OF POINTS AND REDEFINE INFO VECTOR
         C
         C*********************************************************************
0064     200  JPTS = (INFO(8) - INFO(7) +INFO(9))/(INFO(9)*CELWTH)
0065          PTS = JPTS*CELWTH
0066          INFO(8) = INFO(7) + (PTS-1)*INFO(9)
0067          GLCR = INFO(6)*CELWTH
0068          INFO(5) = INFO(4) + GLCR*((INFO(5)-INFO(4)+INFO(6))/GLCR)- INFO(6)
0069          NSR = PTS + 6
0070          NVR = NSR*VECSIZ
         C*********************************************************************
         C
         C        MODIFY THE NEW ID RECORD
         C
         C*********************************************************************
0071          NC(1) = ROTAPE
0072          NC(2) = ROFILE
0073          NC(3) = NEWRUN
         C*********************************************************************
         C
         C        COMPUTE ARRAY SIZES (FOR ARRAY4)
         C
         C*********************************************************************
0074          CELSIZ =CELWTH*CELWTH
0075          BDATA = 1
0076          RDATA = BDATA + (NC(6)*VECSIZ + 3)/4
0077          TAPBUF =RDATA + NVR*CELWTH
0078          CELSUM = TAPBUF + 2
0079          CELCOR = CELSUM +VECSIZ*JPTS
0080          MTXSIZ = VECSIZ*(VECSIZ + 1)/2
0081          PIXVAL = CELCOR + MTXSIZ*JPTS
0082          PIXCOR = PIXVAL + VECSIZ*CELSIZ
0083          CATBAS = PIXCOR + MTXSIZ*CELSIZ
0084          IF(CATBAS-1 .GT. 8000) CALL ERPRNT( 176, 'STOP')
         C*********************************************************************
         C
         C        ALLOCATE CORE FOR ARRAY USED IN ANNEXATION
         C
         C*********************************************************************
0085          C4 = VECSIZ*CELSIZ
0086          ROWSIZ = JPTS*CELSIZ
0087          SIZ1 = (JPTS+3)/4
0088          SIZ2 = (JPTS+7)/8
0089          SIZ3 = (JPTS+1)/2
0090          SIZ4 = (VECSIZ + 1)/2
0091          SIZ5 = (VECSIZ * JPTS + 1)/2
0092          SIZ6 = (MTXSIZ + 1)/2
```



```
0093          SIZ7 = (MTXSIZ*JPTS +1)/2
0094          SIZ8 = (NUMCLS*JPTS + 1)/2
0095          STACK = 1
0096          CLOSED = STACK + SIZ1
0097          OPEN = CLOSED + SIZ2
0098          FLDSIZ = OPEN + SIZ2
0099          BDATA1 = FLDSIZ + SIZ3
0100          RDATA1 = BDATA1 + (PTS+1)/2
0101          PIXCOR = RDATA1
0102          PIXVAL = PIXCOR + (MTXSIZ*CELSIZ + 1)/2
0103          SIZE = C4
0104          FLDVAL = PIXVAL + (SIZE+1)/2
0105          LINNO = FLDVAL
0106          FLDSUM = SIZ7 + 1 + LINNO
0107          FLDCOR =FLDSUM + SIZ5
0108          CELVAR = FLDCOR + SIZ7
0109          FLDVAR = CELVAR + SIZ5
0110          CELAVE = FLDVAR + SIZ5
0111          AUXSUM = CELAVE + SIZ4
0112           AUXVAR = AUXSUM + SIZ4
0113          WORK = AUXVAR + SIZ4
0114          BUFFER = WORK
0115          BUFROZ = (TOP/8 - (BUFFER -1))/JPTS
0116          IF(BUFROZ .LT. 2) CALL ERPRNT(376, 'STOP')
         C*********************************************************************
         C
         C        PROCESSING PARAMETERS
         C
         C*********************************************************************
         C
0117          WRITE(PRNTR,9170) BUFROZ,MAXSIZ
0118     9170 FORMAT(////' PROCESSING PARAMETERS'//
             * '    NUMBER OF ROWS OF CELLS IN BUFFER = ',I6//
             * '    MAXIMUM FIELD SIZE (NO. OF POINTS) = ',I7)
         C
         C*********************************************************************
         C
         C        PRINT RUN HEADER
         C
         C*********************************************************************
0119          CALL GETIME(ITIME)
0120          WRITE(PRNTR,HEAD)
0121          WRITE(PRNTR,TITLE)
0122          CALL HEADER(VECSIZ,FETVC3,CSEL3,CSET3,ID,FRJCAL)
         C
         C*********************************************************************
         C
         C        CLASSIFY THE REGION
         C
         C
         C        ARRAY REFER TO ANNEX AREAS
         C        ARRAY4 REFER TO ORIGINAL STATS AREAS
         C
         C*********************************************************************
         C
0123          CALL NSECHI(
             * ARRAY(STACK), ARRAY(CLOSED), ARRAY(OPEN),
             * ARRAY(FLDSIZ), ARRAY(BDATA1), ARRAY(RDATA1),ARRAY8(PIXCOR),
             * ARRAY8(PIXVAL), ARRAY8(CELSUM), ARRAY8(CELCOR),
             * ARRAY(FLDSUM), ARRAY(FLDCOR), ARRAY(CELVAR), ARRAY(FLDVAR),
             * ARRAY(CELAVE), ARRAY(AUXSUM), ARRAY(AUXVAR),
             * ARRAY(WORK), ARRAY(BUFFER),
             * ARRAY(LINNO), ARRAY8(BDATA), ARRAY8(RDATA), ARRAY8(TAPBUF))
         C
0124          CALL TSTREW(OTE)
0125          BDATS = 1
0126          RDATS = (VECSIZ*ID(6) + 3)/4 + 1
0127          TOTPTS = (VECSIZ*ID(6) + 1)/2 + 2
0128          NROWPT = JPTS*CELWTH + 6
0129          MTPTS = JPTS*4
0130          FUFPTS = JPTS*4
0131          KPTS = JPTS + 1
0132          CALL REWRTE(ARRAY(BDATA), ARRAY(RDATA), NROWPT, ARRAY4(1),
             * ARRAY4(KPTS),ARRAY4(MTPTS))
         C
         C
```



```
         C
         C
0133          RETURN
0134          END
```

LARS Program Abstract ______

MODULE IDENTIFICATION

Module Name: NSECHO **Function Name:** NS1ECHO

Purpose: Perform field extraction and disk filing for NS1ECHO

System/Language: CMS/FORTRAN

Author: ______ **Date:** ______

Latest Revisor: C. A. Pomalaza **Date:** 8/21/77

MODULE ABSTRACT

NSECHO computes cell mean and variances row by row, performs cell splitting, annexation and field closures. As fields are annexed this information is sequentially stored on disk. As fields are closed, field statistics information is stored in a random access file on disk.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NSECHO

CALL NSECHO (STACK, CLOSED, OPEN, FLDSIZ, PRTBUF, RDATAT, PIXCOR, PIXVAL, CELSUM, CELCOR, FLDSUM, FLDCOR, CELVAR, FLDVAR, CELAVE, AUXSUM, AUXVAR, WORK, BUFFER, LINBUF, BDATA, RDATA, TAPBUF)

All arguments are array bases computed by NS1INT.

STACK	I*4 Array dimensioned JPTS, where JPTS is the number of cell widths/line being processed. It is a revolving queue to store flags about the field status (which field storage areas are in use).
CLOSED OPEN	L*1 Arrays each dimensioned JPTS. Logical flags for each field, used to ascertain closure status.
FLDSIZ	I*4 Array dimensioned JPTS. Keeps record of number of pixels in each field.
PRTBUF	I*2 Array used to store output symbols to print object map (dimensioned by core constraints).
PIXVAL	R*4 Array dimensioned VECSIZ * CELSIZ where VECSIZ is the number of channels used and CELSIZ is the number of pixels/cell used to store cell values during processing.
PIXCOR	R*4 Dimensioned VECSIZ * MTXSIZ where MTXSIZ = VECSIZ x $\frac{(VECSIZ + 1)}{2}$. Used to store intermediate correlation matrices.
CELSUM	R*4 Array dimensioned VECSIZ * JPTS and used to store cell means one row at a time.
CELCOR	R*4 Array dimensioned MTXSIZ * JPTS and used to store cell correlation matrices one row at a time.
FLDSUM	R*4 Array dimensioned VECSIZ * JPTS used to accumulate field means.
FLDCOR	R*4 Array dimensioned MTXSIZ * JPTS used to accumulate field correlation.
CELVAR FLDVAR	R*4 Arrays dimensioned JPTS * VECSIZ and used to store cell and field variances.
CELAVE	R*4 Array dimensioned VECSIZ used as intermediate cell mean buffer.

AUXCOR	R*4 Arrays dimensioned JPTS * VECSIZ used as auxiliary storage during variance processing for candidate annexations.
AUXSUM	R*4 Array dimensioned JPTS * VECSIZ used to store means for candidate annexations.
WORK	R*4 Working area. What remains of ARRAY in GLOCOM after ZCOV.
BUFFER	I*2 Hold field assignments from current and previous line of data. Dimensioned 2*JPTS*2. Buffer (1,x) contains the field number of cell x, BUFFER (2,x) contains the relative field pointer into FLDCOV (covariance of open fields).
BDATA	L*1 Array dimensioned VECSIZ * ID(6). Receives raw data values.
RDATA	R*4 Dimensioned VECSIZ * JPTS * CELSIZ holds calibrated data values from the tape.
TAPBUF	I*4 Symbol storage area.
LINBUF	I*4 Output storage area.

2. Internal Description

NSECHO operates in a loop over the lines requested. It retrieves a row of data, annexes cells into fields on a column basis. It then writes the buffer line to disk. Reads the next row (cell width lines) annexes it by columns and by rows to the first row. Fields that are closed are written with the field number to the intermediate tape and the buffer is written to disk. The slots in FLDCOV are then released and a new row is read. This processing continues until the lines are completed. NSECHO uses GATHER to compute cell variances and means, ANNEX to perform annexation and BEGFLD to open new fields.

3. Input Description

Data is read from tape via GADLIN.

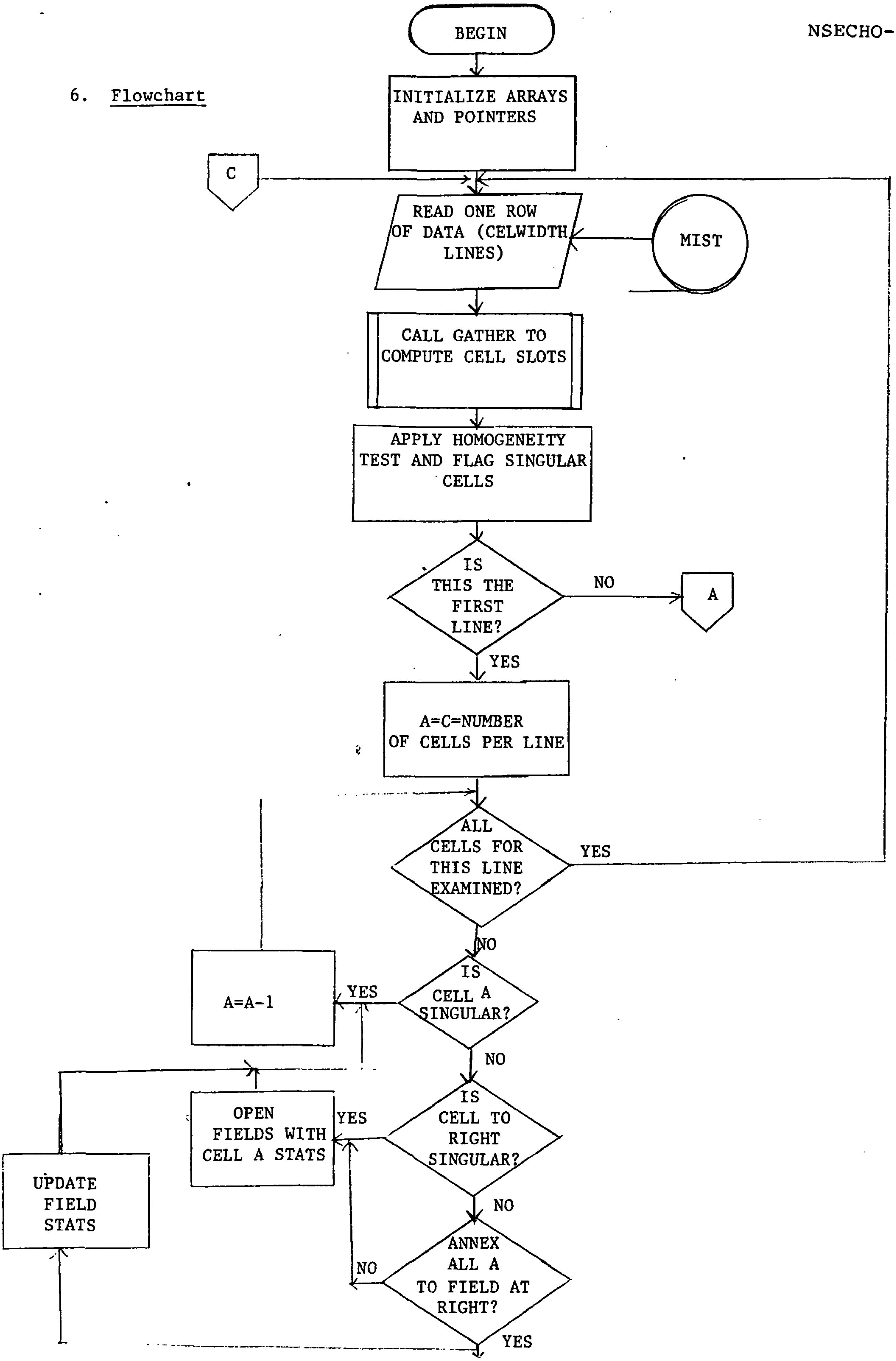
4. Output Description

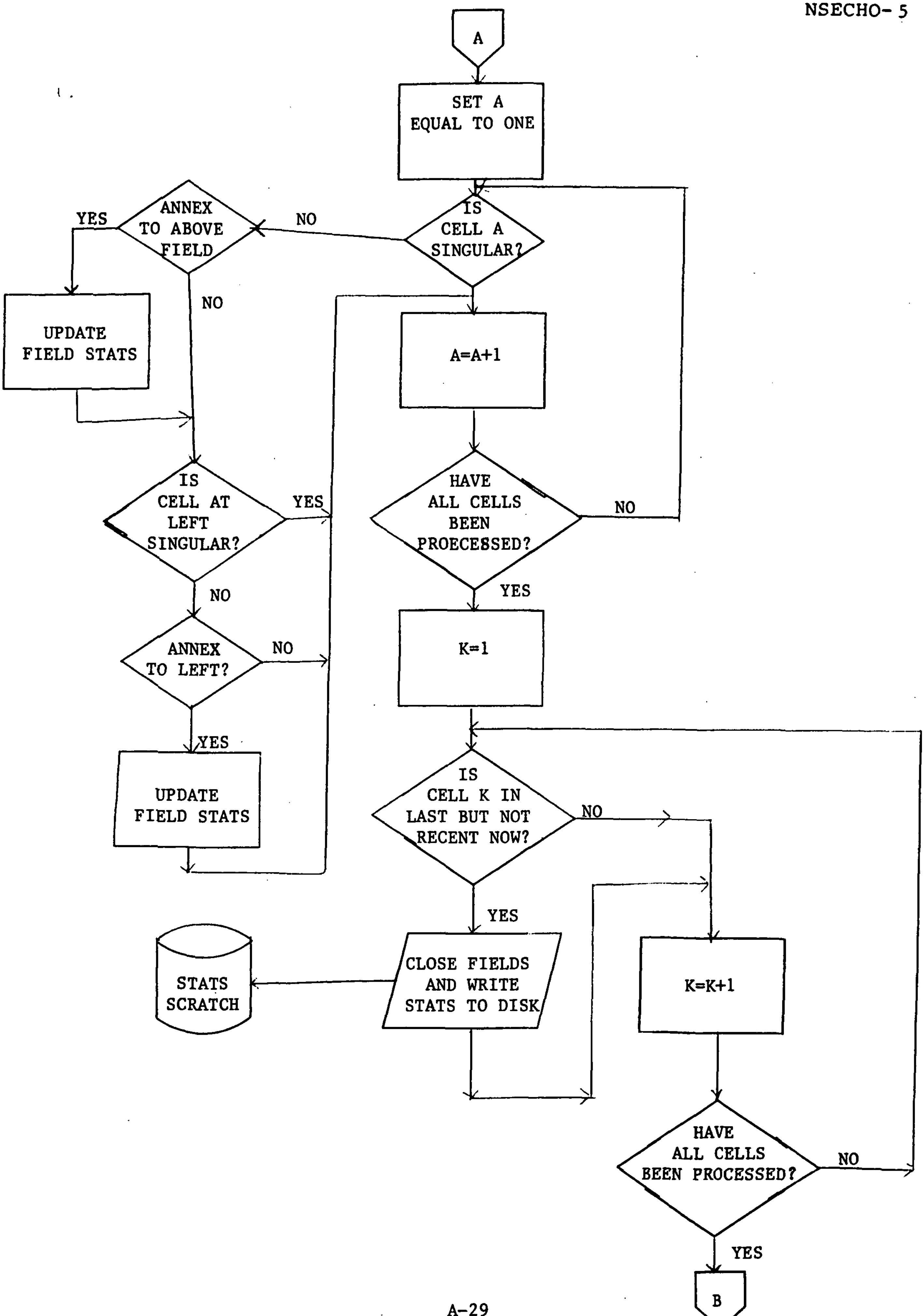
Information is written on disk using unformatted write.

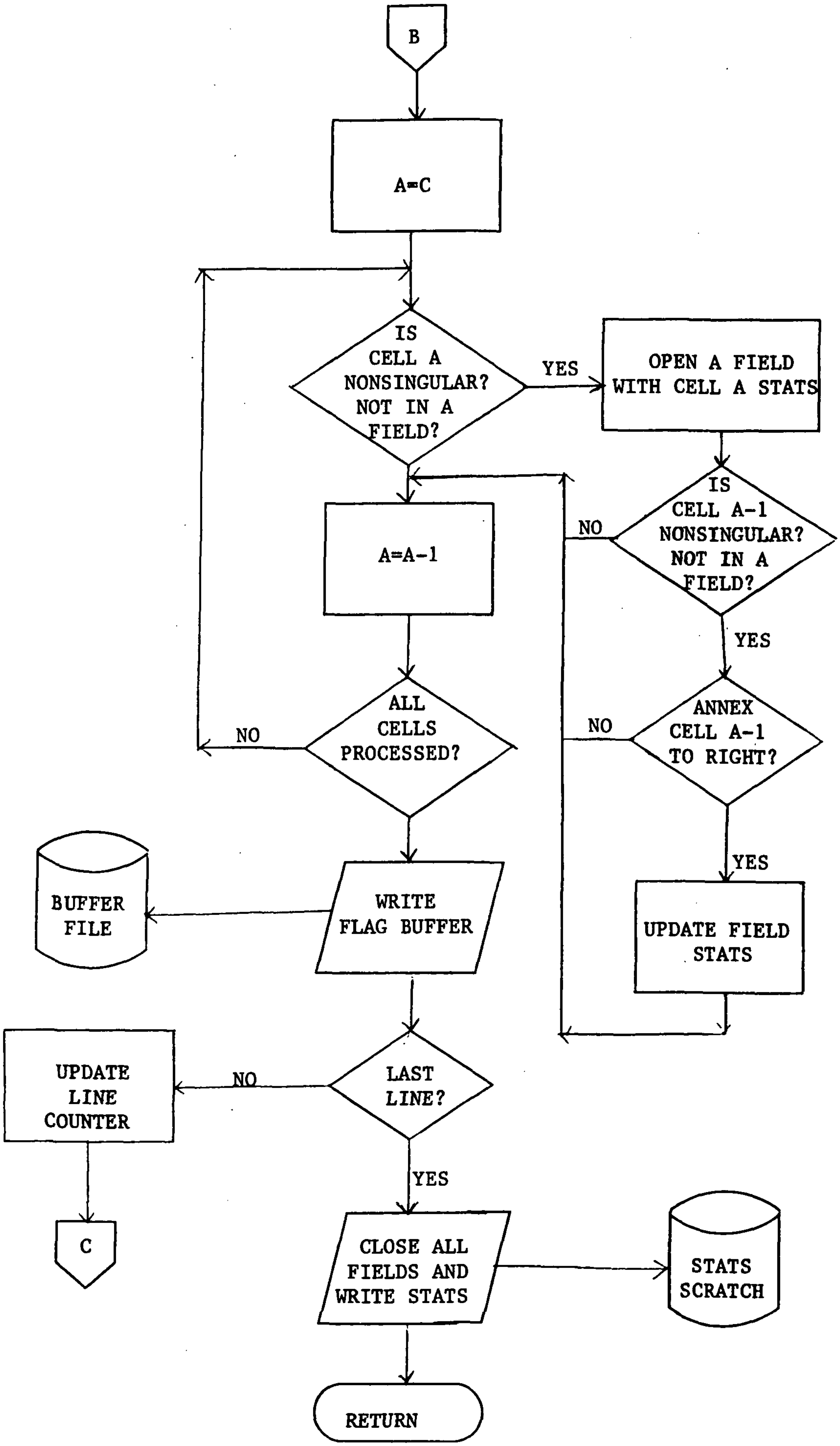
5. Supplemental Information

See BUFFER and SCRATCH file description.

6. Flowchart


BEGIN
INITIALIZE ARRAYS AND POINTERS
C
READ ONE ROW OF DATA (CELWIDTH LINES)
MIST
CALL GATHER TO COMPUTE CELL SLOTS
APPLY HOMOGENEITY TEST AND FLAG SINGULAR CELLS
IS THIS THE FIRST LINE?
NO
A
YES
A=C=NUMBER OF CELLS PER LINE
ALL CELLS FOR THIS LINE EXAMINED?
YES
NO
IS CELL A SINGULAR?
YES
A=A-1
NO
IS CELL TO RIGHT SINGULAR?
YES
OPEN FIELDS WITH CELL A STATS
UPDATE FIELD STATS
NO
ANNEX ALL A TO FIELD AT RIGHT?
NO
YES


A
SET A EQUAL TO ONE
IS CELL A SINGULAR?
NO
ANNEX TO ABOVE FIELD
YES
UPDATE FIELD STATS
NO
A=A+1
HAVE ALL CELLS BEEN PROECESSED?
NO
YES
IS CELL AT LEFT SINGULAR?
YES
NO
ANNEX TO LEFT?
NO
YES
UPDATE FIELD STATS
K=1
IS CELL K IN LAST BUT NOT RECENT NOW?
NO
YES
CLOSE FIELDS AND WRITE STATS TO DISK
STATS SCRATCH
K=K+1
HAVE ALL CELLS BEEN PROCESSED?
NO
YES
B


B
A=C
IS CELL A NONSINGULAR? NOT IN A FIELD?
YES
OPEN A FIELD WITH CELL A STATS
IS CELL A-1 NONSINGULAR? NOT IN A FIELD?
NO
YES
A=A-1
ALL CELLS PROCESSED?
NO
ANNEX CELL A-1 TO RIGHT?
NO
YES
UPDATE FIELD STATS
BUFFER FILE
WRITE FLAG BUFFER
UPDATE LINE COUNTER
NO
LAST LINE?
YES
C
CLOSE ALL FIELDS AND WRITE STATS
STATS SCRATCH
RETURN

FILE NSECHO

```
C
C     NSECHO      LARS     XXXX
C                 MODIFIED  BY C.A. POMALAZA
C
C*********************************************************************
C
C     PROGRAM TO PERFORM FIELD EXTRACTION PHASE OF NONSUPERVISED ECHO
C
C*********************************************************************
C
      SUBROUTINE NSECHO(
     & STACK, CLOSED, OPEN, FLDSIZ, PRTBUF, RDATA1, PIXCOR,
     & PIXVAL, CELSUM, CELCOR, FLDSUM, FLDCOR, CELVAR, FLDVAR,
     &   CELAVE, AUXSUM, AUXVAR, WORK, BUFFER, LINBUF, BDATA,
     & HDATA, TAPBUF)
      IMPLICIT INTEGER*4 (A-Z)
C
      COMMON /GLOCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX,
     1    CLUSTX, CONPUT, CPYOUT, CRDRDR, CRDSEC, DATAPE,
     2    CUPETP, DUPRUN, ERRMSG, ERPRT,
     3    FILESV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,
     4    IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,
     5    PACSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,
     6    RESTRT, RUNFIL, RUNTAB(10,3),
     7    SDATA, SEPARX, SEPTPX, SPARE(10), TEMPAS(30),
     8    TPSTAT(6), TTELDX ,TYPEWR,
     9    TOP, ARRAY(12500)
C
      REAL * 8 ARRAY
      REAL * 4 FRQCAL(5,30)
      INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)
      INTEGER * 2 BLANK2
      LOGICAL * 4 CHKOUT
      LOGICAL * 1 BLANK1
      EQUIVALENCE (DATSAV,ID(1)), (CURRUN,ID(4)), (FRQCAL(1),ID(51)),
     1         (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(49)),
     2         (TIME(1),HEAD(54)), (COMENT(1),HEAD(72)),
     3         (MAPSAV,TPSTAT(1)),
     4         (SUPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),
     5         (COPSTR,TPSTAT(5)), (TRAOUT,TPSTAT(6)),
     6         (BLANK,BLANK2,BLANK1)
C
C
C
C
      COMMON /NSICOM/
     1 ANFLAG, ANN1, ANN2, BUFPTS, BUFRCZ,
     2 C1, C4, CLLSIZ, CELWTH,
     3 CSET(3,30), CSET3(3,30), INFO(17), JPTS,
     4 K9, L1, MAPFLG, MAXSIZ, MINSIZ, MTXSIZ, NC(200),
     5 NEWRUN, NOCLS, NOFEAT, NSR, NUMCLS, NVR,
     6 PTS, RESULT, ROWSIZ, ROFILE, RUTAPE, SELEC1(30),
     7 TTITLE(30), TOTPTS, VARSIZ, VECSIZ,
     8 CSEL(30), CSEL3(30), FETVC3(30), FETVEC(30), ZOUM,
     9 ARRAY4(4000)
C
      REAL*8 ARRAY4
C
      REAL*4 CSET,CSET3,SELEC1,ANN1,ANN2
C
      INTEGER * 2 CSEL,CSEL3,FETVEC,FETVC3
C
      LOGICAL * 4 MAPFLG
C
      EQUIVALENCE (VECSIZ,NOFET3),(MTXSIZ,VARSZ3),(CELWTH,GRSIZE)
C
C - - - - - - - - - - - - - - - - -
C
C
C DEFINE LOCAL VARIABLES
      INTEGER * 4 LINBUF(1), TAPBUF(1), STAT(2), SCRTCH(2)
      INTEGER * 2 BLOCK(4), BDATA(1), AL2(2)
      EQUIVALENCE (AL2(1), ALP)
      REAL*4 RDATA1(1), CELSUM(VECSIZ,JPTS), CELCOR(MTXSIZ,JPTS)
      REAL* 4 PIXVAL(VECSIZ,CELSIZ), PIXCOR(MTXSIZ, CELSIZ)
      LOGICAL * 1 CLOSED(JPTS), OPEN(JPTS), STOP
```


FILE NSECHO

```
      INTEGER * 2 BUFFER(1),PRTBUF(1),STACK(JPTS),CLASS2,JPTS2,
     & SINGUL, VERTI, HORIZ, PLUS, EX, CLDBUF(500)
      INTEGER * 2 SYM(36)
      INTEGER*4 FLDSIZ(JPTS)
      REAL*4
     3 FLDSUM(VECSIZ,JPTS), FLDCOR(MTXSIZ,JPTS),
     5  CELVAR(VECSIZ,JPTS), FLDVAR(VECSIZ,JPTS),
     6 CELAVE(VECSIZ), WORK(1),
     8 AUXSUM(VECSIZ), AUXVAR(VECSIZ), RDATA(NVR, CELWTH),
     A CHISQR, MAXLIK, A1, DLTERM, FLOAT, R9(30)
      EQUIVALENCE (FSTLIN,INFO(4)),(LSTLIN,INFO(5)),(LININT,INFO(6))
      DATA SING /'  '/
      DATA STAT /'STAT','    '/,SCRTCH /'SCTH','    '/
      DATA SINGUL,VERTI,HORIZ,PLUS,EX /'0','I','-','+','X'/
      DATA SYM/'A ',' ','B ',' ','C ',' ','D ',' ','E ',
     1        ' ','F ',' ','G ',' ','H ',' ','I ',' ',
     2        'J ',' ','K ',' ','L ',' ','M ',' ','O ',
     3        ' ','P ',' ','Q ',' ','R ',' ','S ',' ','T '/
C
C*********************************************************************
C
C SET UP VECTOR FOR PROPER READ
C
C*********************************************************************
C
      LINES = (LSTLIN-FSTLIN)/LININT + 1
C
C*********************************************************************
C
C PRINT HEADER.
C
C*********************************************************************
C
      L2 = 400
      N = 1
      DSR=22
      DEFINE FILE 22(10000,200,L,K9)
      REWIND 30
      K9=1
      L=1
      ALP = 1
      DO 6 K7=1,400
    6 WRITE(22'ALP)A
      WRITE(PRNTR,9125) (INFO(I),I=4,9)
 9125 FORMAT('0AREA PROCESSED.....LINES ',I5,'-',I5,'  (BY ',I2,')'/
     & T21,'COLUMNS ',I4,'-',I5,'  (BY ',I2,')'/)
C     N = INFO(9)*CELWTH
      N = INFO(9)
      L = INFO(7)
      IX = JPTS*CELWTH
      JPTS1 = MIN0(IX,110)
      M = L + (JPTS1-1)*N
      I1 = 1
      IF(M .GT. 999) I1=10
      I = L
      DO 30 K=1,JPTS1
      PRTBUF(K) = I/(100*I1)
      PRTBUF(K+JPTS1) = MOD(I,100*I1)/(10*I1)
      PRTBUF(K+2*JPTS1) = MOD(I,10*I1)/I1
   30 I = I+N
      I1 = 2*JPTS1+1
      L1 = 0
      DO 40 I=1,I1,JPTS1
      J = I+JPTS1- 1
   40 WRITE(PRNTR,9040) (PRTBUF(K),K=I,J)
 9040 FORMAT(' ',8X,111I1)
      IF(M .LE. 999) GO TO 50
      I = L
      DO 45 J=1,JPTS1
      PRTBUF(J) = MOD(I,10)
   45 I = I+N
      WRITE(PRNTR,9040) (PRTBUF(I),I=1,JPTS1)
   50 WRITE(PRNTR,9120)
 9120 FORMAT('0')
```

```
            C
            C**********************************************************************
            C
            C  INITIALIZE
            C
            C**********************************************************************
            C
0072               JPTS2 = JPTS-
0073               NEXT=1
0074               BOTTOM=JPTS
0075               DO 100 I=2,4
0076           100 BLOCK(I) = INFO(I+5)
0077               DO 110 I = 1,JPTS
0078               STACK(I) = I
0079               PRTBUF(I) = BLANK2
0080               CLOSED(I) = .TRUE.
0081           110 OPEN(I) = .FALSE.
0082               GLCR = LININT*CELWTH
0083               C1 = CELSIZ - 1
0084               DETCON = FLOAT(C1**VECSIZ)
0085               R6 = (VECSIZ*(2*VECSIZ+3)-1)/(6.*(VECSIZ+1))
0086               R7 = (VECSIZ-1)*(VECSIZ+2)/6.
0087               DO 120 I = 1, VECSIZ
0088               R9(I) = SELEC(I) * SELEC(I)
0089          120  CONTINUE
0090               NOLINE = 0
0091               PLINE = FSTLIN
0092               NEXT = 1
0093               ISYM = 0
0094               PRTRW = 1
0095               JPT = JPTS - 1
0096               OLDRW = 0
0097               NEWRW = 1
0098               CURLIN = FSTLIN
            C  THIS IS THE BEGINNING OF THE OLD SIBI LOOP
0099           150 J = NOLINE*CELWTH
0100               NOLINE = NCLINE + 1
0101               I = CURLIN
0102               DO 155 J = 1,CELWTH
0103               BLOCK(1) = I
0104               CALL GADLIN(BLOCK(1),CSEL3,CSET3,ID,DATAPE,VECSIZ,NSR,BDATA,
                  & RDATA(1,J),ROLL,ERROR)
0105               IF (ERROR.GT.0) CALL LINERR(ERROR,I,&155,&155,&155)
0106           155 I = I + INFO(6)
0107           160 H = 1
0108               DO 201 CELNUM = 1,JPTS
0109               CALL GATHER(RDATA(H,1),PIXVAL,PIXCOR,CELSUM(1,CELNUM),
                  & CELCOR(1,CELNUM),CELWTH,VECSIZ,MTXSIZ,NSR,NVR)
0110           201 H = H + CELWTH
            C
            C**********************************************************************
            C
            C  ENTER MAIN LOOP --  READ CELL SUMS AND CORRELATIONS.
            C
            C**********************************************************************
            C
0111           151 J = NOLINE*CELWTH
0112               IF(MOD(NOLINE,25) .EQ. 0) WRITE(TYPEWR,9235) J
0113          9235 FORMAT(I6,' LINES CLASSIFIED.  (ECHOB3)')
0114               CROWBS= 1 + (OLDRW-1)*(4*JPTS)
0115               NROWBS = 1 + (NEWRW-1)*(JPTS*4)
0116               FROWBS =  (NEWRW*JPTS*4)
            C
            C  GET ONE ROW OF DATA.
            C
            C
            C  COMPUTE CELL STATISTICS AND SCREEN OUT BAD CELLS.
            C
0117               J = NRCWBS
0118               I1 = 1
0119               I2 = 1
0120               I3 = 1
0121               I4 = 1
0122               I5 = 1
0123               I6 = 1
0124               I7 = 1
```



```
0125               DO 200 I=1,JPTS
0126               BUFFER(J) = 0
            C
            C  RETURN VARIANCES FROM SUM AND CROSS PRODUCT MTX
            C
0127           170 CALL VARCAL(CELSUM(1,I2),CELAVE,CELCOR(1,I2),
                  & CELVAR(1,I5),CELSIZ,VECSIZ)
0128           176 DO 177 K=1,VECSIZ
0129               IF(CELVAR(K,I5) .GT. R9(K) *CELSUM(K,I2)*CELAVE(K)) GO TO 190
0130           177 CONTINUE
0131               GO TO 198
            C
            C**********************************************************************
            C
            C  INHOMOGENEOUS CELL.  CLASSIFY IT SEPARATELY.
            C
            C**********************************************************************
            C
0132           190 CONTINUE
0133               K = J + 3
0134               DO 196 L=J,K
0135           196 BUFFER(L) = -1
0136               PRTBUF(I) = SING
0137           198  I2 = I2 + L
0138               I5 = I5+1
0139           200 J = J + 4
0140               IF(OLDRW .EQ. 0) GO TO 265
            C
            C**********************************************************************
            C
            C  SCAN OLDRW FOR VERTICAL BOUNDARIES AND SINGULARITIES.
            C
            C**********************************************************************
            C
0141               K = OROWBS
0142               GO TO 228
            C
            C**********************************************************************
            C
            C  COMPARE CELLS WITH FIELDS.
            C
            C**********************************************************************
            C
            C  COMPARE CURRENT ROW WITH PREVIOUS ROW.
            C
0143           228 J = OROWBS
0144               K = NROWBS
0145               DO 240 I=1,JPTS
0146               H = BUFFER(J)
0147               M = BUFFER(J+1)
0148               IF(H .LE. JPTS .AND. H .GE. 0) CLOSED(H)=.FALSE.
0149               IF(BUFFER(K) .NE. 0) GO TO 239
0150               IF(H .LT. 0) GO TO 230
0151               ASSIGN 229 TO RETURN
0152               GO TO 1000
0153           229 IF(ANFLAG) 230,230,227
0154           227 OPEN(H) = .TRUE.
0155               BUFFER(K) = H
0156               BUFFER(K+1)=M
0157               PRTBUF(I) = OLDBUF(I)
0158               GO TO 239
            C
            C  COMPARE WITH FIELD TO LEFT, IF ANY.
            C
0159           230 IF(I .EQ. 1) GO TO 239
0160               H = BUFFER(K-4)
0161               M = BUFFER(K-3)
0162               IF (H.LE.0) GO TO 239
0163               ASSIGN 232 TO RETURN
0164               GO TO 1000
0165           232 IF(ANFLAG) 2385,2385,238
0166           235 OPEN(H) = .TRUE.
0167           238 BUFFER(K) = H
0168               BUFFER(K+1) = M
0169               PRTBUF(I) = PRTBUF(I-1)
0170               GO TO 239
```

```
      2345 ISYM = 1 + MOD(ISYM,36)
           PRTBUF(I) = SYM(ISYM)
       230 J = J + 4
       240 K = K + 4
C
C************************************************************************
C
C CLOSE FIELDS THAT ARE IN OLDRW BUT NOT IN NEWRW.
C
C************************************************************************
C
           LAST = ORDWBS + (JPTS-1)*4
           DO 260 H=1,JPTS
           IF(CLOSED(H) .OR. OPEN(H)) GO TO 250
           BOTTOM = 1 + MOD(BOTTOM,JPTS)
           STACK(BOTTOM) = H
C
C CLASSIFY CLOSED FIELDS.
C
           DO 414 Q=ORDWBS,LAST,4
           IF (BUFFER(Q).EQ.H) GOTO 4151
       414 CONTINUE
           ALP=BUFFER(Q+1)
      4151 ALP=BUFFER(Q+1)
           WRITE(22'ALP)BUFFER(Q),BUFFER(Q+1),FLDSIZ(H),(FLDSUM(IL,H),IL
         & =1,VECSIZ),(FLDCOR(IL,H),IL=1,MTXSIZ)
C       WRITE(PRNTR,1)BUFFER(Q),BUFFER(Q+1),FLDSIZ(H)
       250 CLOSED(H) = .TRUE.
       260 OPEN(H) = .FALSE.
           KAPP=ORDWBS+1
           KUPP=NROWBS
C
C************************************************************************
C
C BACKSCAN ACROSS ROW.  BEGIN NEW FIELDS.
C
C************************************************************************
C
       275 K = NROWBS + (JPTS-1) * 4
           I = JPTS
           DO 280 M=1,JPTS
           J = 1
           I = I-1
           L = K - 4
           H = BUFFER(K)
           CO = BUFFER(K+1)
           IF(H .LT. 0) GO TO 280
           IF(H .NE. 0) GO TO 270
           L1 = L1 + 1
           H = STACK(NEXT)
           NEXT = 1 + MOD(NEXT,JPTS)
           BUFFER(K) = H
           BUFFER(K+1) = L1
           CO = L1
           ISYM = 1+MOD(ISYM,36)
           PRTBUF(J) = SYM(ISYM)
           CALL BEGFLD(CELSUM(1,N),FLDSUM(1,H),CELCOR(1,N),FLDCOR(1,H),
         & CELVAR(1,N), FLDVAR(1,H),FLDSIZ(H))
       270 IF(L.EQ.J .OR. BUFFER(L).NE.0) GO TO 280
           ASSIGN 272 TO RETURN
           GO TO 1000
       272 IF(ANFLAG) 280,280,275
       275 BUFFER(L) = H
           BUFFER(L+1) = CO
           PRTBUF(I) = PRTBUF(N)
       280 K=L
C
C************************************************************************
C
C WRITE OUT PRINTER LINE
C
C************************************************************************
           PRTCEL = (JPTSI+1)/CELWTH
           DO 320 III = 1, CELWTH
           L = CURLIN + (III - 1)*INFO(6)
       320 WRITE(PRNTR,9290) L,((PRTBUF(I),J=1,CELWTH),I=1,PRTCEL)
```



```
      9290 FORMAT(I7,1X,111A1)
           DO 330 JK = 1,JPTS
       330 CLDBUF(JK) = PRTBUF(JK)
       340 CURLIN = CURLIN + GLCR
C  THIS IS THE END CONDITION
           Y=U
       342 WRITE(30)(BUFFER(I),I=NROWBS,FROWBS)
           IF(CURLIN .GT. LSTLIN) GO TO 400
C
C POINT TO NEXT ROW IN BUFFER
C
           OLDRW = NEWRW
       360 NEWRW = MOD(NEWRW,BUFROZ) + 1
C THIS IS THE RETURN OR END OF ROW PROCESSING CONDITION
           IF(NEWRW .NE. PRTRW) GO TO 150
C
C************************************************************************
C
C CLASSIFY AND PRINT ONE ROW OF CELLS.
C
C************************************************************************
C
           K = 1 + (PRTRW-1)*ROWSIZ
           L = K + ROWSIZ-CELSIZ
           DO 365 I=K,L,CELSIZ
           H = BUFFER(I)
           IF(H .GT. JPTS) GO TO 365
       365 CONTINUE
           ASSIGN 150 TO RETURN
           GO TO 4000
C
       400 CONTINUE
C************************************************************************
C
C CLASSIFY ALL OPEN FIELDS.
C
C************************************************************************
C
       410 L = NROWBS + (JPTS - 1) * 4
           DO 420 I=NROWBS,L,4
           H = BUFFER(I)
           IF(H .LT. 0) GO TO 420
           AL2(2)=BUFFER(I+1)
           WRITE(22'ALP)BUFFER(I),BUFFER(I+1),FLDSIZ(H),(FLDSUM(IL,H),IL
         & =1,VECSIZ),(FLDCOR(IL,H),IL=1,MTXSIZ)
       415 ROW = NEWRW
           FIELD = H
       420 CONTINUE
C
C FINISH PRINTING BUFFER.
C
           RETURN
C************************************************************************
C
C ANNEXATION ROUTINE
C
C************************************************************************
C
      1000 ANFLAG = 0
           IF(FLDSIZ(H) .GE. MAXSIZ) GO TO 1006
           CALL ANNEX(CELSUM(1,I),FLDSUM(1,H),AUXSUM,CELCOR(1,I),
         & FLDCOR(1,H),CELVAR(1,I),FLDVAR(1,H), AUXVAR,FLDSIZ(H))
      1006 GO TO RETURN, (229,232,272)
C
C************************************************************************
C
C PRINT ONE ROW OF CELLS.
C
C************************************************************************
C
      4000 I1 = 1 + (PRTRW-1)*ROWSIZ
           I2 = I1 + CELSIZ - CELWTH
           DO 4002 I3=I1,I2,CELWTH
           I5 = 0
           I4 = I3 + ROWSIZ-CELSIZ
           DO 4001 I6=I3,I4,CELSIZ
```



```
           I7 = I6 + CELWTH - 1
           DO 4001 I8=I6,I7
           I5 = I5 + 1
           GO TO 4002
      4001 PRTBUF(I5) = BUFFER(I8) - JPTS2
      4002 PLINE = PLINE + LINIST
           PRTRW = MOD(PRTRW,BUFROZ) + 1
           GO TO 150
           END
```

LARS Program Abstract ____________

MODULE IDENTIFICATION

Module Name: BEGFLD Function Name: NS1ECHO

Purpose: Initialization of field statistics

System/Language: CMS/FORTRAN

Author: ____________ Date: ____________

Latest Revisor: C. A. Pomalaza Date: 8/21/77

MODULE ABSTRACT

BEGFLD initializes field statistics from cell statistics for the field extraction phase of unsupervised ECHO.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

BEGFLD

CALL BEGFLD (CELSUM, FLDSUM, CELCOR, FLDCOR, CELVAR, FLDVAR, FLDSIZ)

Input Arguments

CELSUM	R*4 Array dimensioned VECSIZ and used to store cell means.
CELCOR	R*4 Array dimensioned MTXSIZ and used to store cell correlation matrix.
CELVAR	R*4 Array dimensioned VECSIZ and used to store cell variances.

Output Arguments

FLDSUM	R*4 Array dimensioned VECSIZ and used to store field means.
FLDCOR	R*4 Array dimensioned MTXSIZ and used to store field correlation matrix.
FLDVAR	R*4 Array dimensioned VECSIZ and used to store field variances.
FLDSIZ	I*4 Variable to keep record of the number of pixels in each field.

2. Internal Description

BEGFLD initializes FLDSUM, FLDCOR, FLDVAR and FLDSIZ to be equal to CELSUM, CELCOR, CELVAR, and CELSIZ.

3. Input Description

Not applicable.

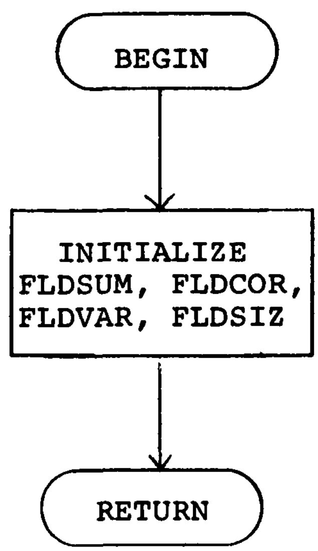
4. Output Description

Not applicable.

5. Supplemental Information

Not applicable.

6. Flowchart



FILE BEGFLD

```
              C BEGFLD         MODIFIED BY C.A. POMALAZA                             EC400010
              C INITIALIZES STATISTICS FOR A NEW FIELD                               EC400020
0001                SUBROUTINE BEGFLD(CELSUM,FLDSUM,CELCOR,FLDCOR,CELVAR,FLDVAR,     EC400030
                  &                 FLDSIZ)                                          EC400040
0002                IMPLICIT INTEGER * 4 (A-Z)                                       EC400050
              C---------------------------------------------------------------------EC400060
              C                                                                      EC400070
0003                COMMON /NSICOM/
                  1 ANFLAG, ANN1, ANN2, BUFPTS, BUFROZ,
                  2 C1, C4, CELSIZ, CELWTH,
                  3 CSET(3,30), CSET3(3,30), INFO(17), JPTS,
                  4 K9, L1, MAPFLG, MAXSIZ, MINSIZ, MTXSIZ, NC(200),
              5   5 NEWRUN, NOCLS, NOFEAT, NSR, NUMCLS, NVR,
                  6 PTS, RESULT, ROWSIZ, RQFILE, RQTAPE, SELEC1(30),
                  7 TTITLE(30), TOTPTS, VARSIZ, VECSIZ,
                  8 CSEL(30), CSEL3(30), FETVC3(30), FETVEC(30), ZDUM,
                  9 ARRAY4(4000)
              C
0004                REAL*8 ARRAY4
              C
0005                 REAL*4 CSET,CSET3,SELEC1,ANN1,ANN2
              C                                                                      EC400240
0006                INTEGER * 2 CSEL,CSEL3,FETVEC,FETVC3                             EC400250
              C                                                                      EC400260
0007                 LOGICAL * 4 MAPFLG                                              EC400270
              C                                                                      EC400280
0008                EQUIVALENCE (VECSIZ,NOFET3),(MTXSIZ,VARSZ3),(CELWTH,GRSIZE)      EC400290
              C---------------------------------------------------------------------EC400300
0009                REAL * 4 CELSUM(VECSIZ),FLDSUM(VECSIZ),CELCOR(MTXSIZ),           EC400310
                  & FLDCOR(MTXSIZ),CELVAR(VECSIZ),FLDVAR(VECSIZ)                     EC400320
              C---------------------------------------------------------------------EC400330
0010                FLDSIZ = CELSIZ                                                  EC400340
0011                DO 1 I=1,VECSIZ                                                  EC400350
0012                FLDVAR(I) = CELVAR(I)                                            EC400360
0013              1 FLDSUM(I) = CELSUM(I)                                            EC400370
0014                DO 2 I=1,MTXSIZ                                                  EC400380
0015              2 FLDCOR(I) = CELCOR(I)                                            EC400390
0016                RETURN                                                           EC400400
0017                END                                                              EC400410
```

LARS Program Abstract ______

MODULE IDENTIFICATION

Module Name: ANNEX Function Name: NS1ECHO

Purpose: Carry out annexation tests and processing

System/Language: CMS/FORTRAN

Author: ______ Date: ______

Latest Revisor: C. A. Pomalaza Date: 8/21/77

MODULE ABSTRACT

ANNEX performs the multiple-univariate tests for annexation of cells to fields and updates field statistics when the annexation criteria are achieved.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

ANNEX

CALL ANNEX (CELSUM, FLDSUM, AUXSUM, CELCOR, FLDCOR, CELVAR, FLDVAR, AUXVAR, FLDSIZ)

Input Argument

CELSUM FLDSUM	R*4 Arrays dimensioned VECSIZ and used to store cell and field means respectively.
CELCOR FLDCOR	R*4 Arrays dimensioned MTXSIZ and used to store cell and field correlation matrices.
CELVAR FLDVAR	R*4 Arrays dimensioned VECSIZ and used to store cell and field variances.
FLDSIZ	I*4 Variable that records the number of pixels in each field.

Output Arguments

FLDSUM	R*4 See Above.
AUXSUM	R*4 Array dimensioned VECSIZ and used to store means for candidate field annexations.
FLDCOR	R*4 Array. See above.
FLDVAR	R*4 Array. See above.
AUXVAR	R*4 Array dimensioned VECSIZ and used to store variances for candidate annexations.

2. Internal Descriptions

ANNEX performs the multiple-univariate tests for equivalent mean vectors and for equivalent covariance matrices between a cell and a field. If both tests are successful the cell is annexed to the field and the field statistics are updated. ANNEX calls the function FDIST to find the decision threshold values of the F-distribution necessary to perform the tests.

3. Input Description

Not applicable

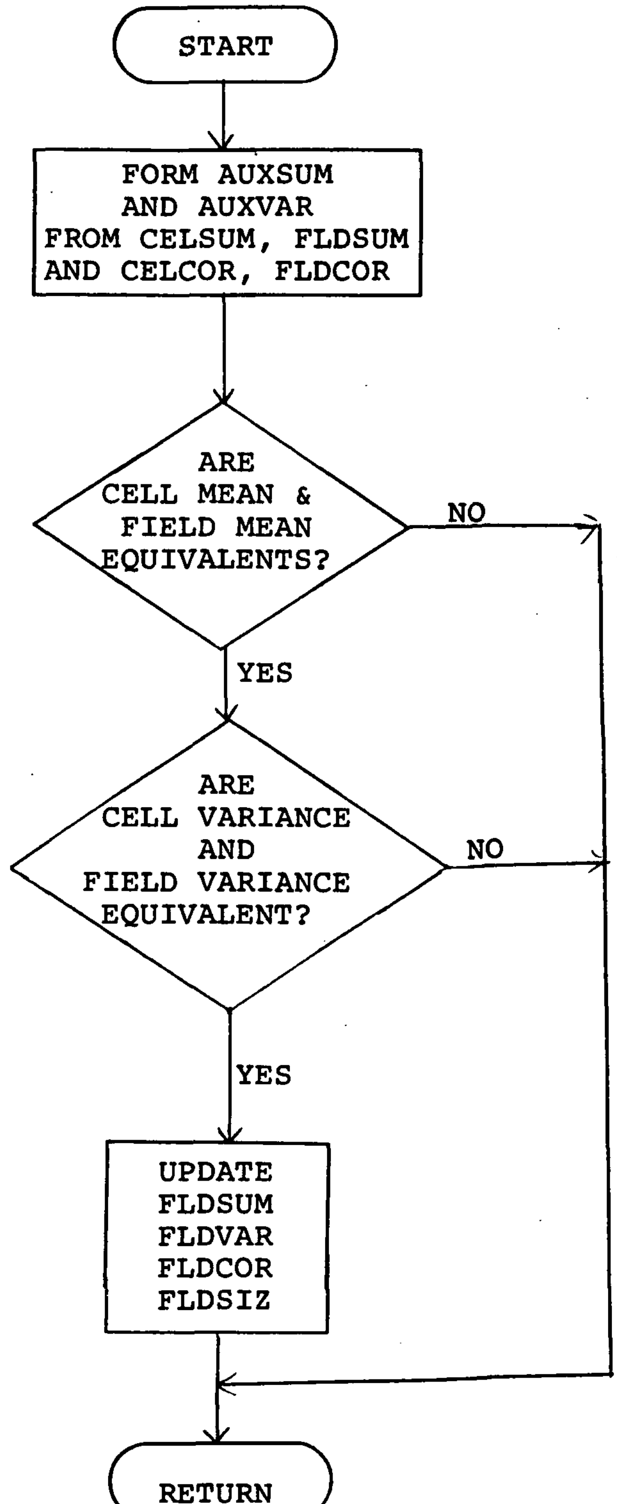
4. Output Description

Not applicable.

5. <u>Supplemental Information</u>

Not applicable.

6. Flowchart


START
FORM AUXSUM
AND AUXVAR
FROM CELSUM, FLDSUM
AND CELCOR, FLDCOR
ARE
CELL MEAN &
FIELD MEAN
EQUIVALENTS?
NO
YES
ARE
CELL VARIANCE
AND
FIELD VARIANCE
EQUIVALENT?
NO
YES
UPDATE
FLDSUM
FLDVAR
FLDCOR
FLDSIZ
RETURN

FILE ANNEX

```
               C                                                                    JEC00010
               C                                                                    JEC00020
               C   ANNEX        MODIFIED BY C.A. POMALAZA                           JEC00030
               C                                                                    JEC00040
0001                 SUBROUTINE ANNEX(CELSUM,FLDSUM,AUXSUM,CELCOR,FLDCOR,CELVAR,    JEC00050
                   &              FLOVAR,AUXVAR,FLDSIZ)                             JEC00060
0002                IMPLICIT INTEGER * 4 (A-Z)                                      JEC00070
               C--------------------------------------------------------------------JEC00080
               C                                                                    JEC00090
0003                COMMON /NSICOM/
                   1 ANFLAG, ANN1, ANN2, BUFPTS, BUFROZ,
                   2 C1, C4, CELSIZ, CELWTH,
                   3 CSET(3,30), CSET3(3,30), INFO(17), JPTS,
                   4 K9, L1, MAPFLG, MAXSIZ, MINSIZ, MTXSIZ, ND(200),
               5   5 NEWRUN, NCCLS, NOFEAT, NSR, NUMCLS, NVR,
                   6 PTS, RESULT, ROWSIZ, ROFILE, ROTAPE, SELEC1(30),
                   7 TTITLE(30), TOTPTS, VARSIZ, VECSIZ,
                   8 CSEL(30), CSEL3(30), FETVC3(30), FETVEC(30), ZDUM,
                   9 ARRAY4(4000)
               C
0004                REAL*8 ARRAY4
               C
0005                 REAL*4 CSET,CSET3,SELEC1,ANN1,ANN2
               C                                                                    JEC00260
0006                INTEGER * 2 CSEL,CSEL3,FETVEC,FETVC3                            JEC00270
               C                                                                    JEC00280
0007                 LOGICAL * 4 MAPFLG                                             JEC00290
               C                                                                    JEC00300
0008                EQUIVALENCE (VECSIZ,NOFET3),(MTXSIZ,VARSZ3),(CELWTH,GRSIZE)     JEC00310
               C                                                                    JEC00320
               C--------------------------------------------------------------------JEC00330
0009                REAL * 4 CELSUM(VECSIZ),FLDSUM(VECSIZ),AUXSUM(VECSIZ),          JEC00340
                   & CELCOR(MTXSIZ),FLDCOR(MTXSIZ),                                 JEC00350
                   & CELVAR(VECSIZ),FLOVAR(VECSIZ),AUXVAR(VECSIZ),                  JEC00360
                   & A1,A2,A3,A4,A5,A6,FDIST,DETERM,ALOG,SORT,FLOAT                 JEC00370
               C--------------------------------------------------------------------JEC00380
               C--------------------------------------------------------------------JEC00390
               C UNSUPERVISED MUV G.L.R. TEST                                        JEC00400
               C MULTIPLE-UNIVARIATE TEST FOR EQUIVALENT MEAN VECTORS                JEC00410
0010                C5 = FLDSIZ - 1                                                 JEC00420
0011                C8 = CELSIZ + FLDSIZ                                            JEC00430
0012            120 J = 0                                                           JEC00440
0013                A1 = 1. + FDIST(1,C1+C5,ANN2)/(C1+C5)                           JEC00450
0014                DO 130 I=1,VECSIZ                                               JEC00460
0015                J = J+1                                                         JEC00470
0016                AUXSUM(I) = CELSUM(I) + FLDSUM(I)                               JEC00480
0017                AUXVAR(I) = CELCOR(J) + FLDCOR(J) -AUXSUM(I)*AUXSUM(I)/C8       JEC00490
0018                IF(AUXVAR(I) .GT. (CELVAR(I)+FLDVAR(I))*A1) RETURN              JEC00500
0019            130 CONTINUE                                                        JEC00510
               C MULTIPLE-UNIVARIATE TEST FOR EQUIVALENT COVARIANCE MATRICES.       JEC00520
0020            100 IF(ANN1 .LE. 1.E-25) GO TO 135                                  JEC00530
0021                A1 = FLOAT(C1 + C5)                                             JEC00540
0022                A2 = (1./C1 + 1./C5 - 1./A1)/3.                                 JEC00550
0023                A3 = 3./(A2*A2)                                                 JEC00560
0024                A2 = (1. - A2 + 2./A3)/A3                                       JEC00570
0025                A4 = FDIST(1,INT(A3+.01),ANN1)/A3                               JEC00580
0026                DO 110 I=1,VECSIZ                                               JEC00590
0027                A3 = CELVAR(I)/C1                                               JEC00600
0028                IF(A3 .LE. 0.) RETURN                                           JEC00610
0029                A5 = FLDVAR(I)/C5                                               JEC00620
0030                IF(A5 .LE. 0.) RETURN                                           JEC00630
0031                A3= A2*(A1*ALOG((CELVAR(I)+FLDVAR(I))/A1)-C1*ALOG(A3)-C5*ALOG(A5))JEC00640
0032                IF(A3 .GE. 1.) RETURN                                           JEC00650
0033                A5 = 1. - A3                                                    JEC00660
0034                IF(A3 .GT. A5*A4) RETURN                                        JEC00670
0035            110 CONTINUE                                                        JEC00680
0036            135 DO 140 I=1,VECSIZ                                               JEC00690
0037                FLDSUM(I) = AUXSUM(I)                                           JEC00700
0038            140 FLDVAR(I) = AUXVAR(I)                                           JEC00710
0039                DO 150 I=1,MTXSIZ                                               JEC00720
0040            150 FLDCOR(I) = FLDCOR(I) + CELCOR(I)                               JEC00730
0041                FLDSIZ = C8                                                     JEC00740
0042                ANFLAG = 1                                                      JEC00750
0043                RETURN                                                          JEC00760
0044                END                                                             JEC00770
```

LARS Program Abstract ______________

MODULE IDENTIFICATION

Module Name: FDIST   Function Name: NS1ECHO

Purpose: Computes decision threshold of a F distribution

System/Language: CMS/FORTRAN

Author: ______________   Date: ______________

Latest Revisor: C. A. Pomalaza   Date: 8/21/77

MODULE ABSTRACT

FDIST returns the decision threshold of the F distribution for a given significance level and a given number of degrees of freedom.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

FDIST

CALL FDIST (NUMBER, DENOM, SIGLEV)

Arguments:

NUMBER DENOM	I*4	degrees of freedom of the F distribution
SIGLEV	R*4	significance level

FDIST is a R*4 function.

2. Internal Description

FDIST looks at a table to get the value corresponding to the significance level. The significane level may be one of the following values: .1, .05, .025, .01, .005, .001

3. Input Description

Not applicable

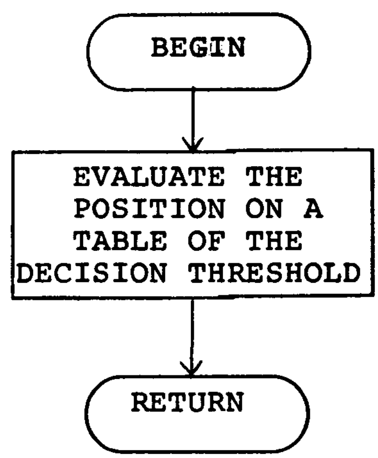
4. Output Description

Not applicable.

5. Supplemental Information

Not applicable.

6. Flowchart


BEGIN
EVALUATE THE POSITION ON A TABLE OF THE DECISION THRESHOLD
RETURN

LARS Program Abstract

MODULE IDENTIFICATION

Module Name: RDWRTE   Function Name: NS1ECHO

Purpose: Writes intermediate tape

System/Language: CMS/FORTRAN

Author: Date:

Latest Revisor: C.A. Pomalaza   Date: 8/20/77

MODULE ABSTRACT

RDWRTE converts the disk files written by NSECHO into a intermediate tape file which is the input to NS1ECHO (the classification phase).

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

RDWRTE

CALL RDWRTE (BDATA, RDATA, NROWPT, BUFFER, MEAN, COVAN)

All arguments but NROWPT are array bases computed by NS1INT.

BDATA	I*2 Array dimensioned as VECSIZ * ID(6) and used as in GADLIN (for raw data).
RDATA	R*4 Array dimensioned as NROWPT * VECSIZ and used as in GADLIN (holding calibrated data values from MIST).
NROWPT	I*4 Number of pixels per line + 6.
BUFFER	I*2 Array dimensioned JPTS * 4 where JPTS is the number of cell widths/line. It is used to read the flag buffer from disk.
MEAN	R*4 Array dimensioned as VECSIZ. Used to store cell means.
COVAN	R*4 Array dimensioned as VECSIZ * (VECSIZ + 1)/2. Used to store cell covariance matrices while processing.

2. Internal Description

RDWRTE performs the following functions:

1. The new ID record, the processing parameters and the field description card are written on the intermediate tape file using LARSYS System routine TOPWR.
2. The statistics file (mean and covariance) fromeeach field are read from a disk file and written on tape using TOPWR.
3. The program starts a loop getting a line of data via LARSYS system routine GADLIN. It then checks the MAP option and if it is on, the appropriate flag line is read from disk, and the data read by GADLIN is altered to reflect field means if the pixels were annexed to fields. The means are read from the disk stat file. If the MAP option is off the data read by GADLIN is unaltered.
4. The line when finished is output to tape using TOPWR. The flag buffer line is also written using TOPWR.
5. The loop started in 5 continues until the area is finished.
6. An additional file (only the ID record) is written in the intermediate tape.

3. Input Description

Multispectral Data is read from the tape using GADLIN.

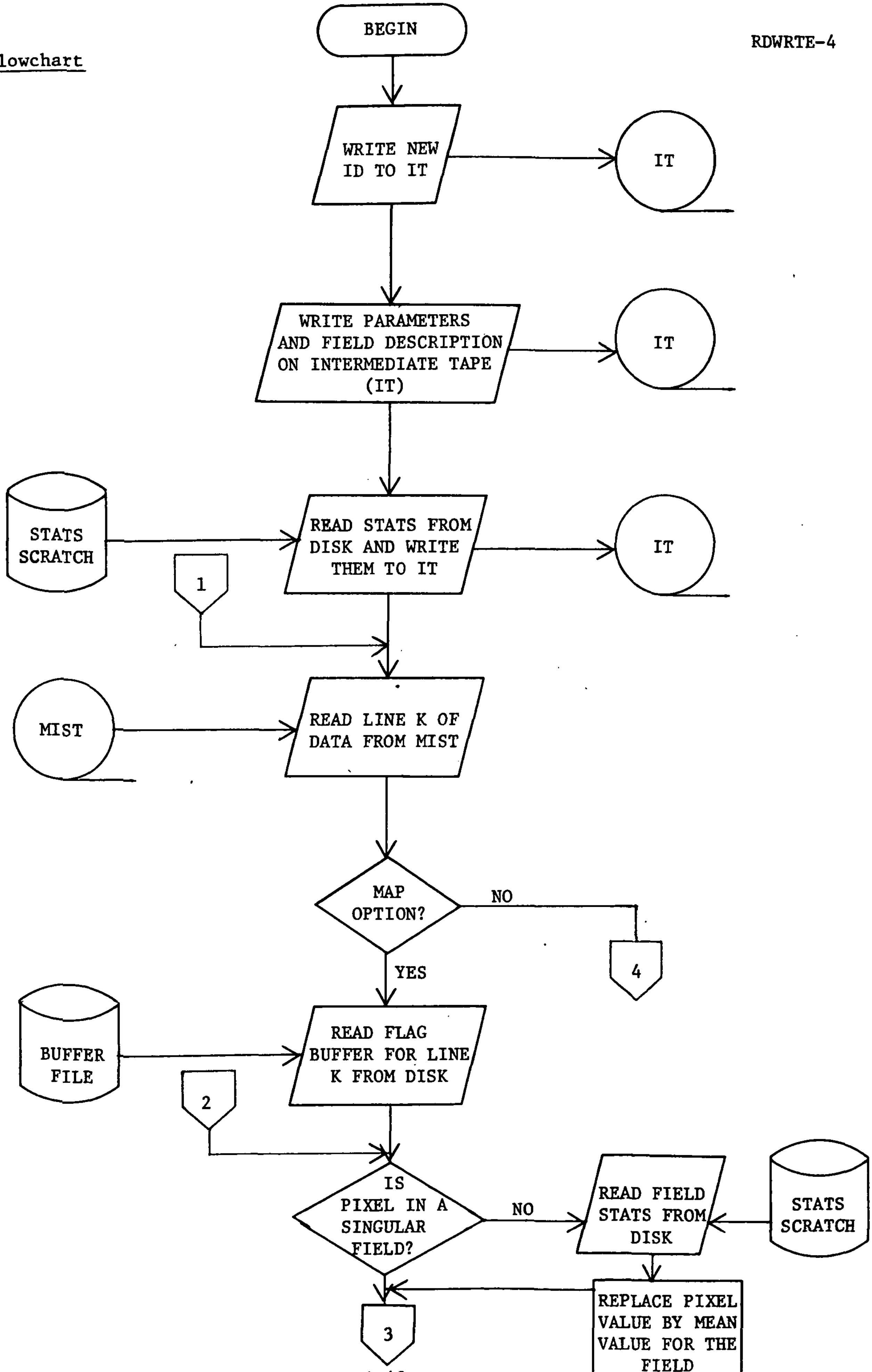
4. Output Description

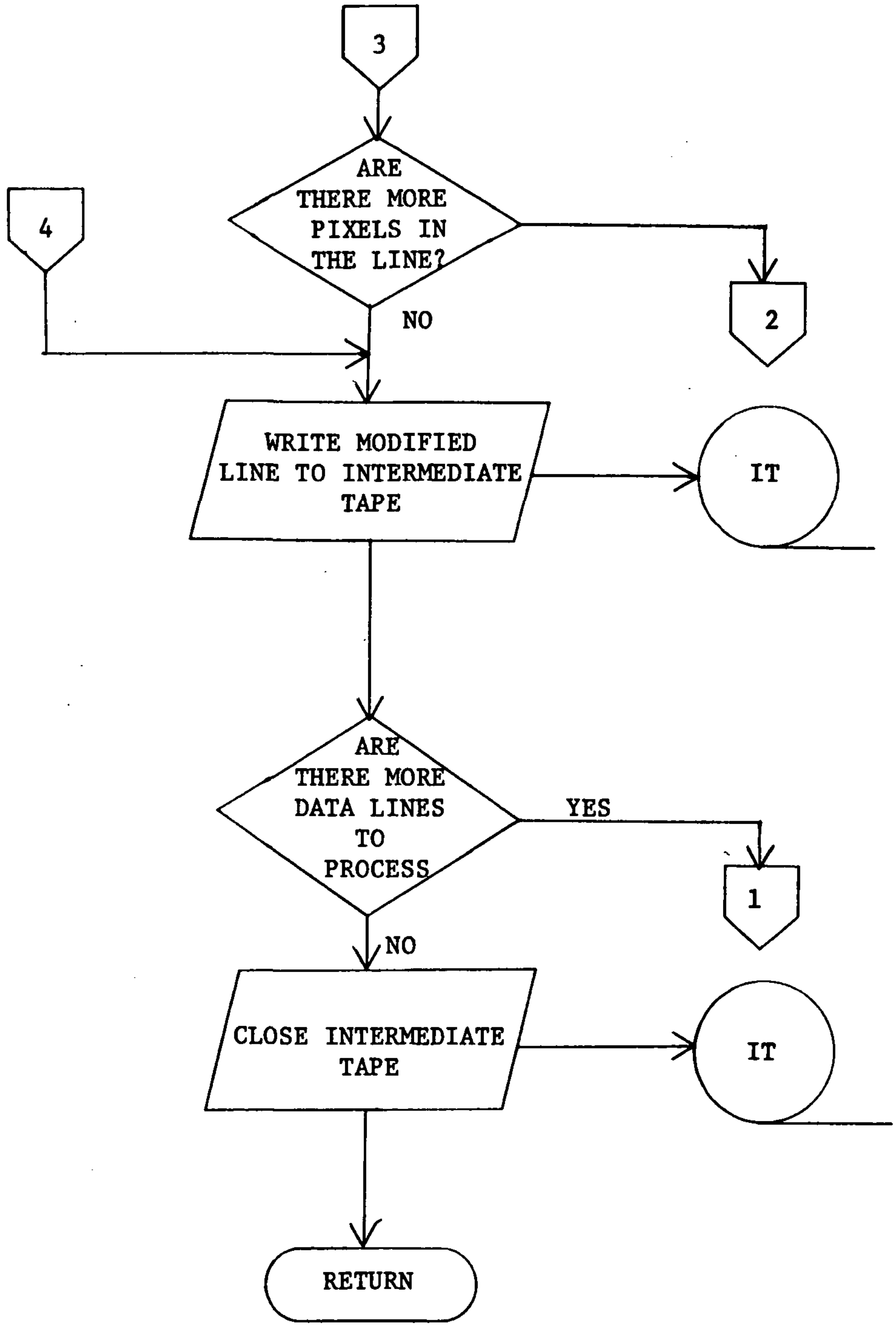
Intermediate tape for input to NS2ECHO (classification phase) is written.

5. Supplemental Information

See intermediate tape file description.

6. Flowchart


BEGIN
WRITE NEW ID TO IT
IT
WRITE PARAMETERS AND FIELD DESCRIPTION ON INTERMEDIATE TAPE (IT)
IT
STATS SCRATCH
READ STATS FROM DISK AND WRITE THEM TO IT
IT
1
MIST
READ LINE K OF DATA FROM MIST
MAP OPTION?
NO
4
YES
BUFFER FILE
READ FLAG BUFFER FOR LINE K FROM DISK
2
IS PIXEL IN A SINGULAR FIELD?
NO
READ FIELD STATS FROM DISK
STATS SCRATCH
REPLACE PIXEL VALUE BY MEAN VALUE FOR THE FIELD
3


3
4
ARE THERE MORE PIXELS IN THE LINE?
2
NO
WRITE MODIFIED LINE TO INTERMEDIATE TAPE
IT
ARE THERE MORE DATA LINES TO PROCESS
YES
1
NO
CLOSE INTERMEDIATE TAPE
IT
RETURN

FILE ROWRTE

```
              C                                                                  RDW00010
              C         ROWRTE    WRITES INTERMEDIATE TAPE RECORDS               RDW00020
              C         MODIFIED BY C.A. POMALAZA                                RDW00030
              C                                                                  RDW00040
0001                SUBROUTINE ROWRTE(BDATA,RDATA,NROWPT,BUFFER,MEAN,COVAN)      RDW00050
0002                IMPLICIT INTEGER*4 (A-Z)                                     RDW00060
              C                                                                  RDW00070
0003                COMMON /GLCCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSH, CLASSX, RDW00080
                   & CLUSTX, CCNPUT, CPYOUT, CRDROR, CRDSEQ, DATAPE,             RDW00090
                   & DUPLTP, DUPRUN, ERRMSG, FBPNT,                              RDW00100
                   & FILLSV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,           RDW00110
                   & IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,                       RDW00120
                   & PAGSIZ, PNCH, POINT, PRCSUX, PRNTR, READIN,                 RDW00130
                   & RESTAT, RUNFIL, RUNTAB(10,3),                               RDW00140
                   & SDATA, SEPARX, SEPIPX, SPARE(10), TEMPAS(10),               RDW00150
                   & TPSTAT(6), TTFLOX, TYPEWR,                                  RDW00160
                   & TOP, ARRAY(12500)                                           RDW00170
              C                                                                  RDW00180
              C                                                                  RDW00190
0004                REAL * 8 ARRAY                                               RDW00200
              C                                                                  RDW00210
0005                REAL * 4 FRUCAL(5,30)                                        RDW00220
              C                                                                  RDW00230
0006                INTEGER *4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)  RDW00240
              C                                                                  RDW00250
0007                INTEGER * 2 BLANK2                                           RDW00260
              C                                                                  RDW00270
              C                                                                  RDW00280
0008                LOGICAL * 4 CHKOUT                                           RDW00290
              C                                                                  RDW00300
              C                                                                  RDW00310
              C                                                                  RDW00320
0009                LOGICAL * 1 BLANK1                                           RDW00330
              C                                                                  RDW00340
0010                EQUIVALENCE (DATSAV,ID(1)),(CURRUN,ID(3)),(FRUCAL(1),ID(51)), RDW00350
                   & (HED1(1),HEAD(8)),(DATE(1),HEAD(26)),(HED2(1),HEAD(39)),   RDW00360
                   & (TIME(1),HEAD(58)),(COMENT(1),HEAD(72)),                   RDW00370
                   & (MAPSAV,TPSTAT(1)),                                        RDW00380
                   & (SEPSCR,TPSTAT(2)),(DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),  RDW00390
                   & (COPSCR,TPSTAT(5)), (TRAOUT,TPSTAT(6)),                    RDW00400
                   & (BLANK,BLANK2,BLANK1)                                      RDW00410
              C                                                                  RDW00420
              C                                                                  RDW00430
0011                COMMON /NSICUM/
                   1 ANFLAG, ANN1, ANN2, BUFPTS, BUFRCZ,
                   2 C1, C4, CCLSIZ, CELWTH,
                   3 CSET(3,30), CSET3(3,30), INFO(17), JPTS,
                   4 K9, L1, MAPFLG, MAXSIZ, MINSIZ, MTXSIZ, ND(200),
              5    5 NEWRUN, NCCLS, NOFEAT, NSR, NUMCLS, NVR,
                   6 PTS, RESULT, ROWSIZ, ROFILE, ROTAPE, SELECI(30),
                   7 TTITLE(30), TOTPTS, VARSIZ, VECSIZ,
                   8 CSEL(30), CSEL3(30), FETVC3(30), FETVEC(30), ZDUM,
                   9 ARRAY4(4000)
              C
0012                REAL*8 ARRAY4
              C
0013                 REAL*4 CSET,CSET3,SELECI,ANN1,ANN2
              C                                                                  RDW00600
0014                INTEGER * 2 CSEL,CSEL3,FETVEC,FETVC3                         RDW00610
              C                                                                  RDW00620
0015                LOGICAL * 4 MAPFLG                                           RDW00630
              C                                                                  RDW00640
0016.               EQUIVALENCE (VECSIZ,NOFET3),(MTXSIZ,VARSZ3),(CELWTH,GRSIZE)  RDW00650
              C                                                                  RDW00660
              C                                                                  RDW00670
              C                                                                  RDW00680
0017                 INTEGER * 2 BUFFER(BUFPTS), K12(2), D, B, NNID(400), AR2(250) RDW00690
0018                 INTEGER * 2 BLOCK(4), BDATA(TOTPTS)                         RDW00700
0019                REAL*4 RDATA(NROWPT,VECSIZ),MEAN(VECSIZ),COVAN(MTXSIZ)       RDW00710
0020                REAL* 4 S, L,AR(125)                                         RDW00720
0021                 INTEGER* 4 AR1(125)                                         RDW00730
0022                EQUIVALENCE (K12(1), K16), (NNID(1), ND(1))                  RDW00740
0023                EQUIVALENCE (AR(1), AR1(1),AR2(1))                           RDW00750
              C                                                                  RDW00760
0024                LOGICAL*1 C(4)                                               RDW00770
              C                                                                  RDW00780
```


FILE ROWRTE

```
0025                EQUIVALENCE(C4,C(1))                                         RDW00790
              C                                                                  RDW00800
              C                                                                  RDW00810
0026                RZ=JPTS*CELWTH+6                                             RDW00820
              C                                                                  RDW00830
              C     SET UP INTERMEDIATE TAPE ID RECORD                           RDW00840
              C                                                                  RDW00850
0027                ND(4)=L1                                                     RDW00860
0028                ND(5)=VECSIZ                                                 RDW00870
0029                ND(6)=(INFO(8)-INFO(7)+1)/INFO(9) + 6                        RDW00880
0030                ND(20)=(INFO(5)-INFO(4)+1)/INFO(6)                           RDW00890
0031                DO 170 K=1,VECSIZ                                            RDW00900
0032                NNID(39+2*K)=FETVC3(K)                                       RDW00910
0033            170 NNID(40+2*K)=CSEL3(FETVC3(K))                                RDW00920
0034                COUNT=800                                                    RDW00930
0035                ERROR=0                                                      RDW00940
0036                CALL TOPWR(MAPTAP,COUNT,ERROR,ND)                            RDW00950
0037                CALL TOPWR(MAPTAP,68,ERROR,INFO)                             RDW00960
0038                AR(1)=CELWTH                                                 RDW00970
0039                AR(2)=ANN1                                                   RDW00980
0040                AR(3)=ANN2                                                   RDW00990
0041                L2=INFO(4)                                                   RDW01000
0042                L3=INFO(5)                                                   RDW01010
0043                DO 120 J = 1,30                                              RDW01020
0044                AR(3+J) = SELECI(J)                                          RDW01030
0045          120   CONTINUE                                                     RDW01040
0046                IC = 132                                                     RDW01050
0047                CALL TCPWR(MAPTAP,IC,ERROR,AR)                               RDW01060
0048                REWIND 30                                                    RDW01070
0049                Y=1                                                          RDW01080
0050                BLOCK(2)=INFO(7)                                             RDW01090
0051                BLOCK(3)=INFO(8)                                             RDW01100
0052                BLOCK(4)=INFO(9)                                             RDW01110
0053                RSZ=JPTS*4                                                   RDW01120
              C  STATISTICS WRITTEN TO TAPE                                      RDW01130
              C                                                                  RDW01140
              C READ AND WRITE OBJECT COV AND MEAN MATRICES                      RDW01150
              C                                                                  RDW01160
0054                DO 1000 K19=1,L1                                             RDW01170
0055                READ(22'K19)D,B,F,(MEAN(I),I=1,VECSIZ),(COVAN(J),J=1,MTXSIZ) RDW01180
0056                QCOUNT=(2+VECSIZ+MTXSIZ)*4                                   RDW01190
0057                AR2(1)=D                                                     RDW01200
0058                AR2(2)=B                                                     RDW01210
0059                AR1(2)=F                                                     RDW01220
0060                DO 451 KO=1,VECSIZ                                           RDW01230
0061            451 AR(2+KO)=MEAN(KO)                                            RDW01240
0062                DO 452 KO=1,MTXSIZ                                           RDW01250
0063            452 AR(2+VECSIZ+KO)=COVAN(KO)                                    RDW01260
0064           1000 CALL TOPWR(MAPTAP,QCOUNT,ERROR,AR)                           RDW01270
0065                KLIN = 0                                                     RDW01280
0066                 L4= CELWTH*INFO(6)                                          RDW01290
0067                DO 100 IA=L2,L3,L4                                           RDW01300
0068                READ(30)(BUFFER(I),I=1,RSZ)                                  RDW01310
0069                DO 100 JA = 1, CELWTH                                        RDW01320
0070                A = IA + JA - 1                                              RDW01330
0071                BLOCK(1)=A                                                   RDW01340
0072                CALL GADLIN(BLOCK(1),CSEL3,CSET3,ID,DATAPE,VECSIZ,RZ,BDATA,RDATA(1RDW01350
                   &,1),ROLL,ERROR)                                              RDW01360
0073                K3=0                                                         RDW01370
0074                INM=2                                                        RDW01380
              C                                                                  RDW01390
              C                                                                  RDW01400
0075          102   DO 122 A2=1,RSZ,4                                            RDW01410
0076                K3=K3+1                                                      RDW01420
0077                IF (BUFFER(A2).EQ.-1.OR. .NOT.MAPFLG) GO TO 118              RDW01430
0078                K16=BUFFER(A2+1)                                             RDW01440
0079                READ(22'K16)D,B,F,(MEAN(I),I=1,VECSIZ),(COVAN(J),J=1,MTXSIZ) RDW01450
0080                K16=1                                                        RDW01460
0081                DO 115 K2=1,CELWTH                                           RDW01470
0082                DO 115 K=1,VECSIZ                                            RDW01480
0083                MN=FETVC3(K)                                                 RDW01490
0084                CALL RLCP(MN,S,L)                                            RDW01500
0085            115 RDATA(K3*CELWTH-CELWTH+K2,K)=(MEAN(K)/F-L)/S                 RDW01510
0086                GO TO 122                                                    RDW01520
0087            118 DO 121 K2=1,CELWTH                                           RDW01530
0088                DO 121 K=1,VECSIZ                                            RDW01540
```


FILE ROWRTE

```
0089                MN=FETVC3(K)                                                 RDW01550
0090                CALL RLCP(MN,S,L)                                            RDW01560
0091            121 RDATA(K3*CELWTH-CELWTH+K2,K)=(RDATA(K3*CELWTH-CELWTH+K2,K)-L)/S RDW01570
0092            122 CONTINUE                                                     RDW01580
0093                INM=3                                                        RDW01590
0094                C4=A                                                         RDW01600
0095                 KLIN = KLIN + 1                                             RDW01610
0096                BDATA(1)=KLIN                                                RDW01620
0097                BDATA(2)=ROLL                                                RDW01630
0098                DO 130 K7=1,VECSIZ                                           RDW01640
0099                DO 130 K8=1,RZ                                               RDW01650
0100            130 BDATA(4+(K7-1)*RZ+K8)=RDATA(K8,K7)                           RDW01660
0101                INM=4                                                        RDW01670
0102                COUNT=VECSIZ*K8*4                                            RDW01680
0103                COUNT2=VECSIZ*K8                                             RDW01690
0104                 COUNT4 = 4                                                  RDW01700
0105                 CALL MOVBYT(BDATA,0,1,RDATA,0,1,COUNT4)                     RDW01710
0106                CALL MOVBYT(BDATA,4,2,RDATA,4,1,COUNT2)                      RDW01720
0107                INM=5                                                        RDW01730
0108            110 CALL TOPWR(MAPTAP,COUNT,ERROR,RDATA)                         RDW01740
0109          111   JCOUNT=8*JPTS                                                RDW01750
0110                CALL TCPWR(MAPTAP,JCOUNT,ERROR,BUFFER)                       RDW01760
0111            100 CONTINUE                                                     RDW01770
              C                                                                  RDW01780
              C     END INTERMEDIATE TAPE                                        RDW01790
              C                                                                  RDW01800
0112                CALL TOPEF(MAPTAP)                                           RDW01810
0113                JCOUNT = 800                                                 RDW01820
0114                DO 101 JJ = 3,200                                            RDW01830
0115                ND (JJ) = 0                                                  RDW01840
0116          101    CONTINUE                                                    RDW01850
0117                ND(2) = ND(2) + 1                                            RDW01860
0118                 FILESV = FILESV + 1                                         RDW01870
0119                CALL TOPWR(MAPTAP,JCOUNT,ERROR,ND)                           RDW01880
0120                CALL TCPEF(MAPTAP)                                           RDW01890
0121                CALL TCPBF(MAPTAP)                                           RDW01900
0122                CALL TOPEF (MAPTAP)                                          RDW01910
0123                CALL TCPBF(MAPTAP)                                           RDW01920
0124                CALL TOPBF(MAPTAP)                                           RDW01930
0125                RETURN                                                       RDW01940
0126                END                                                          RDW01950
```

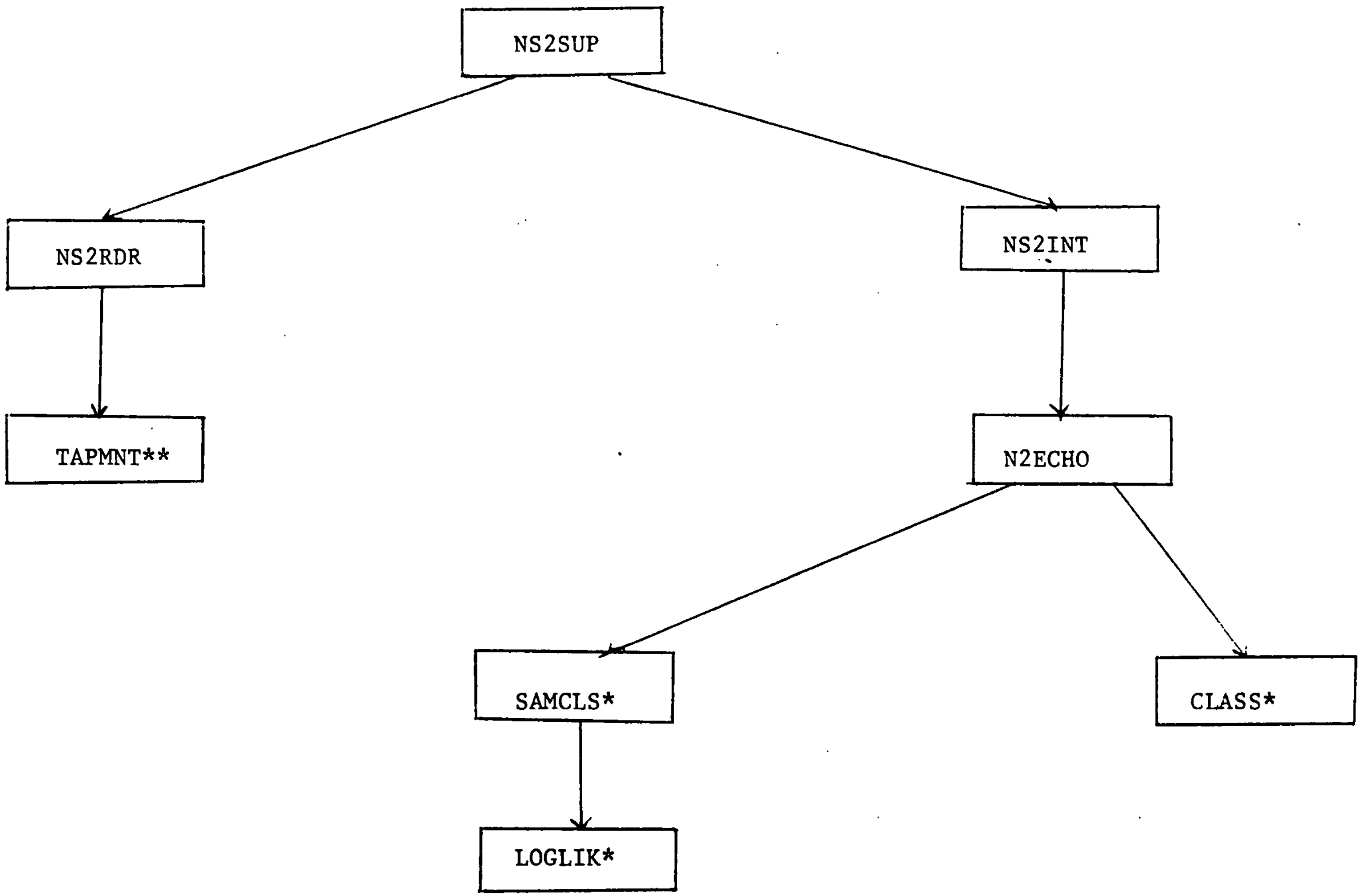
```
FORTRAN IV G LEVEL  20.?        RDWRIT        DATE = ?????        ??/??/??        PAGE 0001
FILE RDWRIT

      C        RDWRIT    WRITES INTERMEDIATE TAPE RECORDS                          RDW00010
      C        MODIFIED BY C.A. [illegible]                                        RDW00020
      C                                                                            RDW00030
0001           SUBROUTINE RDWRIT(RDATA,*DATA,[illegible])                          RDW00040
0002           IMPLICIT INTEGER*4 (A-Z)                                            RDW00050
      C                                                                            RDW00060
0003           COMMON /[illegible]/ BLANK, CARD(20), CHROUT, [illegible], CLASSA,
             & [illegible], CPYINT, [illegible], CRDSTU, DATAPT,                   RDW00090
             & DUPLTP, DUPNUM, ERRMSG, FBPNT,                                      RDW00100
             & FILESV, FIRMND, ODATA, HEADIRA, IDI200), [illegible],               RDW00110
             & [illegible], KEYBD, MAPTAP, MATCHA, MAXCLS,                         RDW00120
             & PAGSIZ, PACH, POINT, PRCSUX, PRNIR, READIN,                         RDW00130
             & [illegible], RDWFIL, NUMTAB(10,3),                                  RDW00140
             & SDATA, SEPARA, SEPIPA, SPARE(10), TEMPAS(10),                       RDW00150
             & TPSTAT(6), TIFLDX, TYPCHR,                                          RDW00160
             & TOP, ARRAY(12500)                                                   RDW00170
      C                                                                            RDW00180
      C                                                                            RDW00190
0004           REAL * 8 ARRAY                                                      RDW00200
      C                                                                            RDW00210
0005           REAL * 4 FAUCAL(5,30)                                               RDW00220
      C                                                                            RDW00230
0006           INTEGER *4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)          RDW00240
      C                                                                            RDW00250
0007           INTEGER * 2 BLANK2                                                  RDW00260
      C                                                                            RDW00270
      C                                                                            RDW00280
0008           LOGICAL * 4 CHROUT                                                  RDW00290
      C                                                                            RDW00300
      C                                                                            RDW00310
      C                                                                            RDW00320
0009           LOGICAL * 1 BLANK1                                                  RDW00330
      C                                                                            RDW00340
0010           EQUIVALENCE (DATSAV,IDI(1)),(CURRUN,IDI(3)),(PROCAL(1),IDI(5)),     RDW00350
             & (HE1(1),HEAD(1)),(DATE(1),HEAD(2)),(HED2(1),HEAD(39)),              RDW00360
             & (TIME(1),HEAD(5)),(COMENT(1),HEAD(72)),                             RDW00370
             & (MAPSAV,TPSTAT(1)),                                                 RDW00380
             & (SEPSCH,TPSTAT(2)),(DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),           RDW00390
             & (CUPSCH,TPSTAT(5)), (TRAOUT,TPSTAT(6)),                             RDW00400
             & (BLANK,BLANK2,BLANK1)                                               RDW00410
[illegible]
```

```
FORTRAN IV G LEVEL  20.?        RDWRIT        DATE = ?????        ??/??/??        PAGE 0002
FILE RDWRIT
[illegible]
```

```
FORTRAN IV G LEVEL  20.?        RDWRIT        DATE = ?????        ??/??/??        PAGE 0003
FILE RDWRIT
[illegible]
```

MAIN SUBROUTINES TREE OF CLASSIFICATION PHASE
OF THE NONSUPERVISED ECHO PROCESSOR



*Delivered with Supervised ECHO documentation in the Final Technical Report on NASA Contract NAS9-14970, May 31, 1977

**Delivered with Field Extraction Phase of the Nonsupervised ECHO processor

LARS Program Abstract ______

MODULE IDENTIFICATION

Module Name: NS2SUP Function Name: NS2ECHO

Purpose: Supervisor for NS2ECHO

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza Date: 8/21/77

Latest Revisor: ______ Date: ______

MODULE ABSTRACT

NS2SUP receives control from LARSMN. This supervisor performs no computation, but instead makes call to the subroutines which really make up the processor.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NS2SUP

There are no parameters to NS2SUP. It is called by LARSMN when NS2ECHO is requested. Control returns to LARSMN when the function is completed.

2. Internal Description

NS2SUP first calls the subroutine NS2RDR to read in all the control cards, and then calls NS2INT to read the statistics data and compute the array bases. NS2INT calls NS2ECHO which performs the classification.

3. Input Description

Not applicable.

4. Output Description

Two messages are produced and written to unit TYPEWR (the console).

UNSUPERVISED ECHO FUNCTION (PHASE 2) REQUESTED

Signifies beginning of the function.

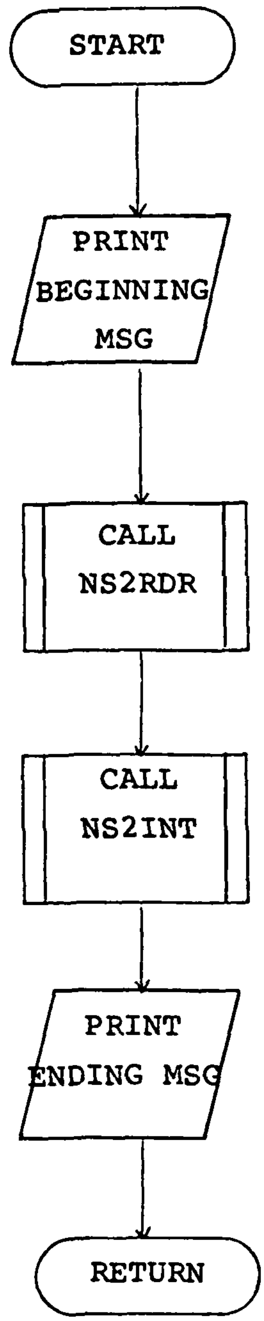
UNSUPERVISED ECHO FUNCTION (PHASE 2) COMPLETED

Signifies end of function.

5. Supplemental Information

Not applicable.

6 - Flowchart


START
PRINT
BEGINNING
MSG
CALL
NS2RDR
CALL
NS2INT
PRINT
ENDING MSG
RETURN

FILE NS2SUP

```
              C                                                                         *NS100010
              C       NS2SUP  LARS XXXX                                                 *NS100020
              C       WRITTEN BY C.A. POMALAZA                                           NS100030
              C                                                                         *NS100040
              C*************************************************************************NS100050
              C                                                                         *NS100060
   0001               SUBROUTINE NS2SUP                                                  NS100070
              C                                                                         *NS100080
   0002               IMPLICIT INTEGER*4 (A-Z)                                           NS100090
   0003               COMMON /GLCCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX,   NS100100
                     1    CLUSTX, CONPUT, CPYOUT, CRDRDR, CRDSEQ, DATAPE,                NS100110
                     2    DUPLTP, DUPRUN, ERRMSG, FBPNT,                                 NS100120
                     3    FILESV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,              NS100130
                     4    IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,                          NS100140
                     5    PAGSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,                    NS100150
                     6    RESTRT, RUNFIL, RUNTAB(10,3),                                  NS100160
                     7    SDATA, SEPARX, SEPTPX, SPARE(10), TEMPAS(30),                  NS100170
                     8    TPSTAT(6), TTFLDX, TYPEWR,                                     NS100180
                     9    TOP, ARRAY(12500)                                              NS100190
   0004               REAL * 8 ARRAY                                                     NS100200
   0005               REAL * 4 FRQCAL(5,30)                                              NS100210
   0006               INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)       NS100220
   0007               INTEGER * 2 BLANK2,NNID(400)                                       NS100230
   0008               LOGICAL * 4 CHKOUT                                                 NS100240
   0009               LOGICAL * 1 BLANK1                                                 NS100250
   0010               EQUIVALENCE (DATSAV,ID(1)), (CURRIN,ID(3)), (FRQCAL(1),ID(51)),    NS100260
                     1        (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),NS100270
                     2        (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),                  NS100280
                     3        (MAPSAV,TPSTAT(1)),                                        NS100290
                     4        (SEPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),NS100300
                     5        (COPSER,TPSTAT(5)), (TRAOUT,TPSTAT(6)),                    NS100310
                     6        (BLANK,BLANK2,BLANK1)                                      NS100320
              C                                                                         *NS100330
   0011               COMMON /NS2COM/ CSET(3,30), MTXSIZ, NOFET3, NOPOOL, OFILE, OTAPE,
                     1 POLNAM(2,60), RESULT, RQFILE, RQTAPE,STKPTR, VECSIZ, WRKSIZ,
                     2 CSEL(30), POLPTR(2,60), POLSTK(60), POLNM1(60)
              C
   0012               REAL*4 CSET
              C
   0013               INTEGER*4 POLNAM
              C
   0014               INTEGER*2 CSEL, POLPTR, POLSTK
              C
   0015               LOGICAL*1 POLNM1
              C
              C
   0016               LOGICAL*1 POLNM1
              C
              C                                                                         *NS100400
              C*************************************************************************NS100410
              C                                                                         *NS100420
              C       CALL READER AFTER INITIAL MESSAGE PRODUCED                        *NS100430
              C                                                                         *NS100440
              C*************************************************************************NS100450
              C                                                                         *NS100460
   0017               WRITE(TYPEWR, 100)                                                 NS100470
   0018       100     FORMAT(' ICOO UNSUPERVISED ECHO FUNCTION (PHASE 2)',               NS100480
                     * ' REQUESTED (NS2SUP)' )                                           NS100490
              C                                                                         *NS100500
   0019               CALL NS2RDR                                                        NS100510
              C                                                                         *NS100520
              C*************************************************************************NS100530
              C                                                                         *NS100540
              C       CALL PROCESSING ROUTINE THEN PRINT TERMINATION MESSAGE            *NS100550
              C                                                                         *NS100560
              C*************************************************************************NS100570
              C                                                                         *NS100580
   0020               CALL NS2INT                                                        NS100590
              C                                                                         *NS100600
   0021               WRITE(TYPEWR,9000)                                                 NS100610
   0022       9000    FORMAT( ' I000UNSUPERVISED ECHO FUNCTION (PHASE2) COMPLETED ',     NS100620
                     * '(NS2SUP)')                                                       NS100630
              C                                                                         *NS100640
   0023               RETURN                                                             NS100650
   0024               END                                                                NS100660
```

LARS Program Abstract ______

MODULE IDENTIFICATION

Module Name: NS2RDR   Function Name: NS2ECHO

Purpose: Read functions control card

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza   Date: 8/21/77

Latest Revisor: ______   Date: ______

MODULE ABSTRACT

NS2RDR reads and interprets all function control cards for NS2ECHO. Also a results tape is readied.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NS2RDR

CALL NS2RDR

This section lists the actions taken when the following control cards are read:

RESULTS--TAPE	- The variable RQTAPE is set to the given tape number.
RESULTS--FILE	- The variable RQFILE is set to the given file number.
RESULTS--INITIALIZE	- The local flag INITFG is set to TRUE. This flag is used by NS2RDR to call MMTAPE to initialize a tape.
INTERMEDIATE--TAPE	- The variable OTAPE is set to the given tape number.
INTERMEDIATE--FILE	- The variable OFILE is set to the given file number.
CLASSES	- LARSYS System Subroutine POLSCN is called to set up the arrays POLNAM, POLPTR, POLSTK, and POLNM1 and compute NOPPOL and STKPTR based on the interpretation of the classes card.

2. Internal Description

NS2RDR uses standard card reader logic in using LARSYS system routines CTLWRD, CTLPRM, and IVAL in reading and interpreting the control cards. After initializing flags and arrays which will convey control card information, NS2RDR functions in a loop of reading and interpreting control cards until DATA or END card is read indicating the end of function control card. After control cards have been read some checks are made on the data. If the classes are pooled, the pooling is checked for validity. Then the results tape is mounted and, if requested, initialized.

3. Input Description

Function control cards for NS2ECHO are read via LARSYS System routine CTLWRD.

4. Output Description

Control card error messages are written to both the printer (PRNTR) and the console (TYPEWR). A brief list of these follows:

ERROR IN CLASSES CARD. CORRECT ALL CLASSES CARDS AND START OVER.

ERROR IN RESULTS CARD (TAPE OR FILE PARAMETER) - TYPE IN CORRECT CARD.

Syntax error in the TAPE or FILE specification for the intermediate or results tape. Standard corrective action.

A POOL HAS NOT BEEN DEFINED. CORRECT CLASSES CARD. POOL NUMBER IS

The pool number is written on the next line. The function terminates. Pool numbers must be consecutive and start at 1.

BOTH FILE AND INITIALIZATION OPTION REQUESTED. FILE REQUESTED IGNORED. FUNCTION CONTINUES.

The results tape is initialized. Only file 1 can be initialized.

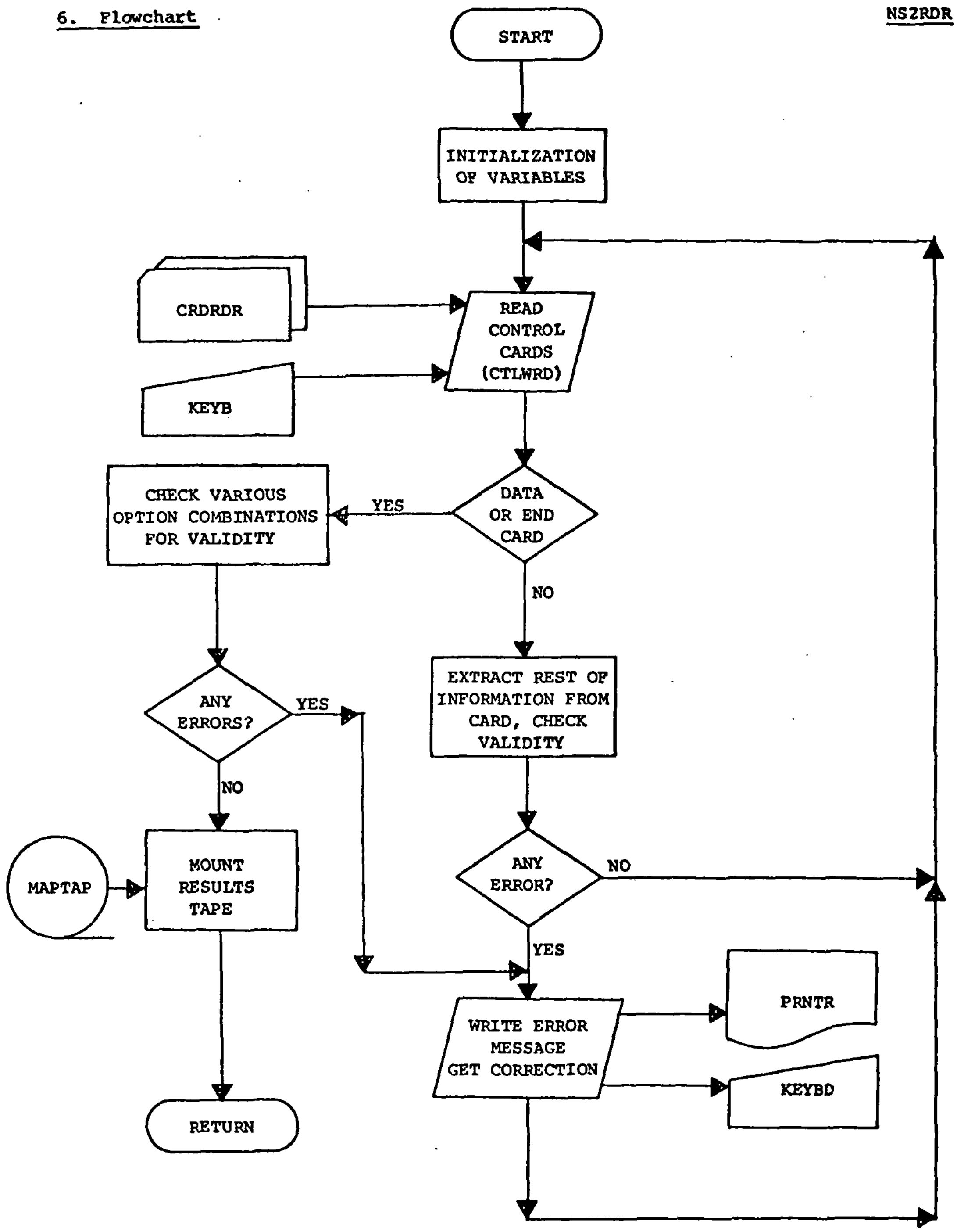
UNEXPECTED END OF FILE ON INPUT DATA

The end of the input deck was reached while reading cards for the function. This normally means that the END card was omitted.

5. Supplemental Information

See LARSYS System Manual for card reader requirements.

6. Flowchart


START
INITIALIZATION OF VARIABLES
CRDRDR
KEYB
READ CONTROL CARDS (CTLWRD)
CHECK VARIOUS OPTION COMBINATIONS FOR VALIDITY
YES
DATA OR END CARD
NO
ANY ERRORS?
YES
NO
EXTRACT REST OF INFORMATION FROM CARD, CHECK VALIDITY
MAPTAP
MOUNT RESULTS TAPE
ANY ERROR?
NO
YES
PRNTR
WRITE ERROR MESSAGE GET CORRECTION
KEYBD
RETURN

FILE NS2RDR

```
                C
                C      NS2RDR          LARS  XXXX
                C                      WRITTEN BY C.A. POMALAZA
                C
                C*********************************************************************
                C
0001                   SUBROUTINE NS2RDR
                C
0002                   IMPLICIT INTEGER*4 (A-Z)
                C
0003                   COMMON /GLBCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX,
                      1    CLUSTX, CONPUT, CPYOUT, CRDRDR, CRDSEQ, DATAPE,
                      2    DUPLTP, DUPRUN, ERRMSG, ERPNT,
                      3    FILESV, FLDBNO, HDATA, HEAD(88), ID(200), IMAGEX,
                      4    IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,
                      5    PAGSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,
                      6    RESTRT, RUNFIL, RUNTAB(10,3),
                      7    SDATA, SEPARK, SEPTPX, SPARE(10), TEMPAS(30),
                      8    TPSTAT(6), TTFLOX, TYPEWR,
                      9    IOF, ARRAY(12500)
0004                   REAL * 8 ARRAY
0005                   REAL * 4 FROCAL(5,30)
0006                   INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)
0007                   INTEGER * 2 BLANK2,NNID(400)
0008                   LOGICAL * 4 CHKOUT
0009                   LOGICAL * 1 BLANK1
0010                   EQUIVALENCE (DATSAV,ID(1)), (CURRIN,ID(3)), (FROCAL(1),ID(51)),
                      1          (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),
                      2          (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),
                      3          (MAPSAV,TPSTAT(1)),
                      4          (STPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),
                      5          (CMPSCR,TPSTAT(5)), (TRAOUT,TPSTAT(6)),
                      6          (BLANK,BLANK2,BLANK1)
                C
0011                   COMMON /NS2COM/ CSET(3,30), MTXSIZ, NOFET3, NOPOOL, OFILE, OTAPE,
                      1 POLNAM(2,60), RESULT, ROFILE, ROTAPE,STKPTR, VECSIZ, WRKSIZ,
                      2 CSEL(30), POLPTR(2,60), POLSTK(60), POLNM1(60)
                C
0012                   REAL*4 CSET
                C
0013                   INTEGER*4 POLNAM
                C
0014                   INTEGER*2 CSEL, POLPTR, POLSTK
                C
0015                   LOGICAL*1 POLNM1
                C
                C
0016                   LOGICAL*1 POLNM1
                C
                C
                C
0017                   INTEGER*4 NOCNT(60), KEPPTS(60), REDBUF(20), PREFIX(2)
0018                   INTEGER*4 KEYWRD(5),RESCOD(3)
0019                   LOGICAL*4 INITFG
0020                   INTEGER*2 BINFLG,FETVEC(30), FETVC3(30)
0021                   DATA KEYWRD /'CLAS','RESU','INTE','DATA','END '/,
                      &     RESCOD /'TAPE','FILE','INIT'/,
                      &     EOS/'EOS'/,END/'END'/
                C
                C*********************************************************************
                C
                C      INITIALIZE
                C
                C*********************************************************************
                C
0022                   INITFG = .FALSE.
0023                   STOPFG = 0
0024                   RESULT = MAPTAP
0025                   ROTAPE = 0
0026                   ROFILE = 0
0027                   NOFET3 = 0
0028                   NOPOOL = 0
0029                   STKPTR = 0
0030                   STATKY = 0
0031                   DO 10 I=1,30
0032                   CSEL(I) = 0
```


FILE NS2RDR

```
0033                10 FETVC3(I) = 0
0034                   DO 11 I=1,60
0035                   POLSTK(I)=0
0036                   KEPPTS(I) = 0
0037                   POLNM1(I)=.TRUE.
0038                   POLPTR(1,I)=0
0039              11   POLPTR(2,I) = 0
0040                   DO 15 I=1,3
0041                   DO 15 J=1,30
0042                15 CSET(I,J)=-50000.
                C
                C*********************************************************************
                C
                C      DECODE CONTROL CARDS
                C
                C*********************************************************************
                C
0043                   ASSIGN 100 TO GOVEC
0044              101  GO TO GOVEC,(100, 540)
0045               100 CALL CTLWRD(CARD,COL,KEYWRD,5,CODE,5,ERRCOR)
0046                   IF(CODE .EQ. 5) SPARE(2) = 1
0047                   IF(ERRCOR .EQ. 4) CALL RTMAIN
0048                   IF(ERRCOR .EQ. 2)CALL ERPRNT(139,'STOP')
                C
                C*********************************************************************
                C
                C      DECODE KEYWORD AND GO ACCORDINGLY
                C
                C*********************************************************************
                C
0049                   GO TO (200,300,400,500,500),CODE
                C  ERROR RECTIFICATION
0050               110 WRITE(TYPEWR,9110)CARD
0051              9110 FORMAT(5X,20A4)
0052              115  CALL ERPRNT(ERRNUM,'GO  ')
0053              111  IF(ERRCOR .NE. 3) ERRCOR = 1
0054                   GO TO 100
                C
                C*********************************************************************
                C
                C      CLASSES CARD
                C
                C*********************************************************************
                C
0055              200  CALL POLSCN(POLNAM, POLPTR, POLSTK, POLNM1, NOPOOL, STKPTR, COL,
                      *          CARD, &210)
0056                   GO TO 100
0057              210  STOPFG = 1
0058                   ERRNUM = 341
0059                   CALL ERPRNT(341, 'GOTO', &101)
                C
                C*********************************************************************
                C
                C      RESULT TAPE ESPECIFICATIONS
                C
                C*********************************************************************
                C
0060              300  IF(COL .EQ. 72) GO TO 101
0061                   CALL CTLPRM(CARD, COL, RESCOD, 3, CODE, &111)
0062                   GO TO(310, 320, 330), CODE
                C
                C      TAPE SPECIFICATIONS
                C
0063              310  VECSZ = 1
0064                   CALL IVAL(CARD, COL, ROTAPE, VECSZ, &315)
0065                   IF(VECSZ .EQ. 0) GO TO 315
0066                   GO TO 300
0067              315  ERRNUM = 348
0068                   GO TO 110
                C
                C      FILE SPECIFICATION
                C
0069              320  VECSZ = 1
0070                   CALL IVAL(CARD, COL, ROFILE, VECSZ, &315)
0071                   IF(VECSZ .EQ.0) GO TO 315
0072                   GO TO 300
```

FILE NS2RDR

```
                C                                                                  *NS201470
                C      INITIALIZE SPECIFICATION                                    *NS201480
                C                                                                  *NS201490
 0073      330       INITFG = .TRUE.                                               NS201500
 0074                GO TO 300                                                     NS201510
                C                                                                  *NS201520
                C*******************************************************************NS201530
                C                                                                  *NS201540
                C      INTERMEDIATE TAPE SPECIFICATIONS                            *NS201550
                C                                                                  *NS201560
                C*******************************************************************NS201570
                C                                                                  *NS201580
 0075      400       IF(COL .EQ. 72) GO TO 100                                     NS201590
 0076                CALL CTLPRM(CARD,COL, RESCOD,2,CODE, &111)                    NS201600
 0077                GO TO(410,420), CODE                                          NS201610
                C                                                                  *NS201620
                C      TAPE SPECIFICATIONS                                         *NS201630
                C                                                                  *NS201640
 0078      410       VECSZ = 1                                                     NS201650
 0079                CALL IVAL(CARD, COL, OTAPE, VECSZ, &415)                      NS201660
 0080                IF(VECSZ .EQ. 0) GO TO 415                                    NS201670
 0081                GO TO 400                                                     NS201680
 0082      415       ERRNUM = 348                                                  NS201690
 0083                GO TO 110                                                     NS201700
                C                                                                  *NS201710
                C      FILE SPECIFICATION                                          *NS201720
                C                                                                  *NS201730
 0084      420       VECSZ = 1                                                     NS201740
 0085                CALL IVAL(CARD, COL, OFILE, VECSZ, &415)                      NS201750
 0086                IF(VECSZ .EQ. 0) GO TO 415                                    NS201760
 0087                GO TO 400                                                     NS201770
                C                                                                  *NS201780
                C*******************************************************************NS201790
                C                                                                  *NS201800
                C      CHECK FOR PRESENCE AND VALIDITY OF ALL INFORMATION IN CORE  *NS201810
                C                                                                  *NS201820
                C*******************************************************************NS201830
                C                                                                  *NS201840
 0088      500       IF(STOPFG .EQ. 1) CALL RTMAIN                                 NS201850
 0089      520       IF (NOPOOL .LE. 0) GO TO 540                                  NS201860
 0090                DO 525 I = 1, NOPOOL                                          NS201870
 0091                IF(POLPTR(1,I) .EQ. 0)GO TO 530                               NS201880
 0092      525       CONTINUE                                                      NS201890
 0093                GO TO 540                                                     NS201900
 0094      530       CALL ERPRNT(344,'GO')                                         NS201910
 0095                WRITE(PRNTR,9530) I                                           NS201920
 0096                WRITE(TYPEWR,9530) I                                          NS201930
 0097     9530       FORMAT('          ',I4)                                       NS201940
 0098                CALL RTMAIN                                                   NS201950
                C                                                                  *NS201960
                C      CHECK RESULT, CARD ENTRIES, ANY FILE ENTRY WITH RESULTS      NS201970
                C      INIT WILL RESULT IN CHANGING THE FILE NUMBER OF THE FIRST    NS201980
                C      FILE WRITTEN ON THE INITIALIZED TAPE. NS2INT WRITES THIS     NS201990
                C      RECORD                                                       NS202000
                C                                                                  *NS202010
 0099      540       IF(.NOT. INITFG) GO TO 545                                    NS202020
 0100                IF(ROFILE .EQ. 0) GO TO 542                                   NS202030
 0101                WRITE(TYPEWR,9540)                                            NS202040
 0102                WRITE(PRNTR,9540)                                             NS202050
 0103     9540       FORMAT( ' 10050 BOTH FILE AND INITIALIZATION OPTION REQUESTED. ', NS202060
                    *   'FILE REQUESTED IGNORED. FUNCTION CONTINUES (NS2RDR)')     NS202070
 0104      542       ROFILE = 1                                                    NS202080
 0105                CALL MMTAPE(ROTAPE, 0, 1)                                     NS202090
 0106                GO TO 900                                                     NS202100
 0107      545       CALL MMTAPE(ROTAPE,ROFILE,1)                                  NS202110
 0108                IF(ROFILE .GE. 0)GO TO 900                                    NS202120
 0109                ASSIGN 540 TO GOVEC                                           NS202130
 0110                ERRCOR = 3                                                    NS202140
 0111                ROFILE = IABS(ROFILE)                                         NS202150
 0112                GO TO 100                                                     NS202160
 0113      900       RETURN                                                        NS202170
 0114                END                                                           NS202180
```

LARS Program Abstract ______

MODULE IDENTIFICATION

Module Name: NS2INT Function Name: NS2ECHO

Purpose: Initialization including array bases computation

System/Language: CMS/FORTRAN

Author: C. A. Pomalaza Date: 8/20/77

Latest Revisor: ______ Date: ______

MODULE ABSTRACT

NS2INT carries out required initialization of the rest of the variables used by NS2ECHO and finishes reading the statistics to be used by the processor. Also the intermediate tape is readied and the subroutine that performs the classification is called.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

NS2INT

CALL NS2INT

The program NS2INT is called with no arguments. Any variables to be used or changed are contained in common blocks GLOCOM and NS2COM.

2. Internal Description

NS2INT performs the following functions:

1. Uses LARSYS system routine STAT to read the statistics from cards.

2. Computes space needed for arrays to hold original statistics.

3. Use LARSYS system routine CLSCHK to check class validity.

4. The intermediate tape is mounted and positioned via TAPMNT.

5. With the information read from the intermediate tape it computes array bases for reduced statistics.

6. Reduce statistics using LARSYS system routine REDSAV.

7. The first three record types are written on the results file. The first record has a 0 in the sixth full word to indicate the absence of weights in the file.

8. Some processor information is printed out.

9. With the information from the Intermediate tape the array bases for calling N2ECHO are computed.

10. N2ECHO is called for performing the classification when finished it writes the needed tape marks and the check record.

3. Input Description

Statistics are read via a call to LARSYS system routine STATS. Information from the ID record of the intermediate tape is obtained via a call to LARSYS system routine TOPRD.

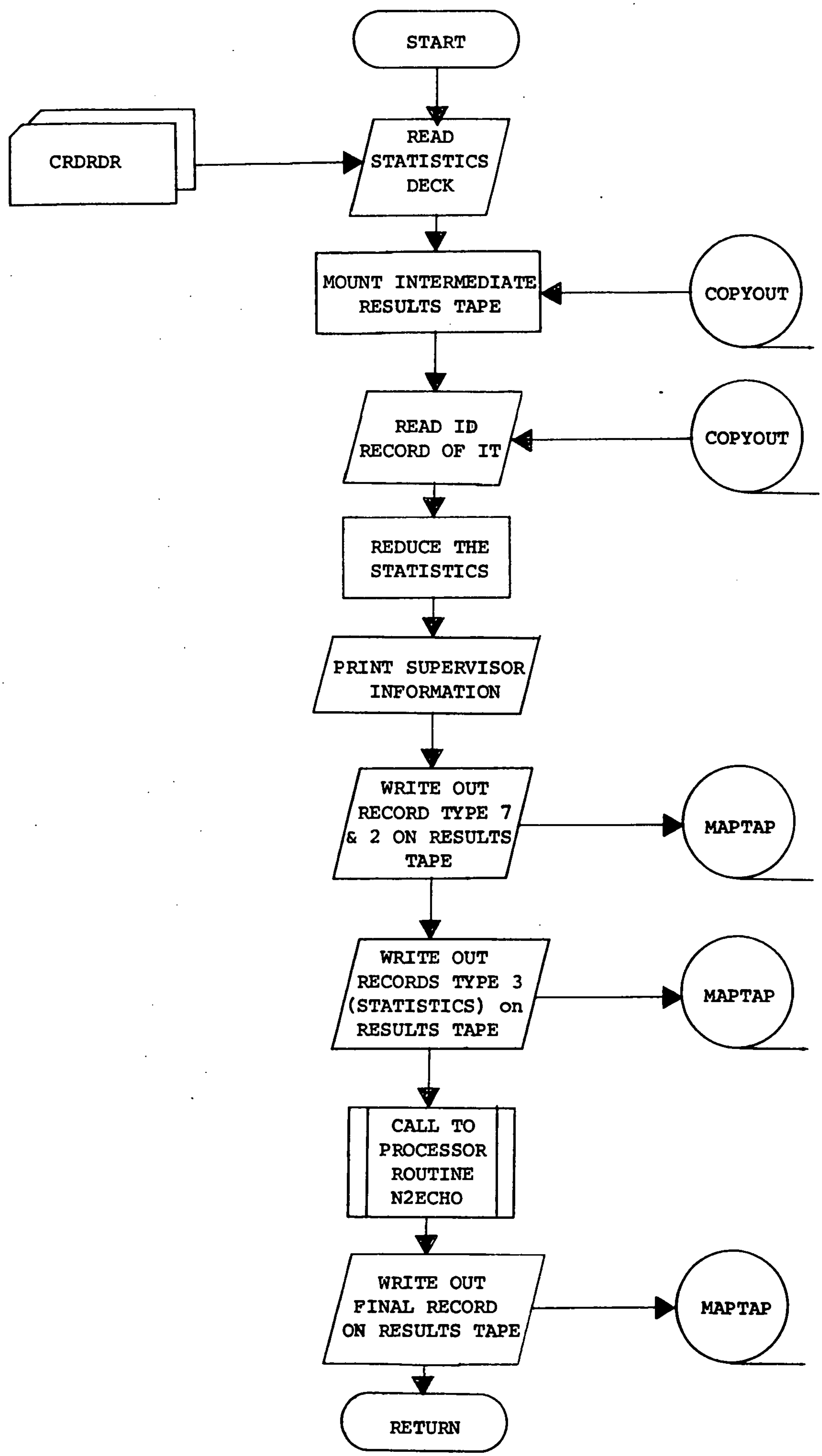
4. Output Description

Information concerning serial number, classes, field and channels is written on the printer (PRNTR). The results tape is written using unformatted FORTRAN write statement.

5. Supplemental Information

Not applicable.

6. Flowchart


START
CRDRDR
READ STATISTICS DECK
MOUNT INTERMEDIATE RESULTS TAPE
COPYOUT
READ ID RECORD OF IT
COPYOUT
REDUCE THE STATISTICS
PRINT SUPERVISOR INFORMATION
WRITE OUT RECORD TYPE 7 & 2 ON RESULTS TAPE
MAPTAP
WRITE OUT RECORDS TYPE 3 (STATISTICS) on RESULTS TAPE
MAPTAP
CALL TO PROCESSOR ROUTINE N2ECHO
WRITE OUT FINAL RECORD ON RESULTS TAPE
MAPTAP
RETURN

FILE NS2INT

```
                 C                                                                       *NS200010
                 C       NS2INT            LARS  XXXX                                     *NS200020
                 C                         WRITTEN BY C.A. POPALAZA                        NS200030
                 C                                                                       *NS200040
                 C***********************************************************************NS200050
                 C                                                                       *NS200060
0001                     SUBROUTINE NS2INT                                                 NS200070
                 C                                                                       *NS200080
0002                     IMPLICIT INTEGER*4 (A-Z)                                          NS200090
                 C                                                                       *NS200100
0003                     COMMON /GLOCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX,  NS200110
                        1    CLUSTX, CONPUT, CPYOUT, CRDROR, CRDSEQ, DATAPE,               NS200120
                        2    DUPLTP, DUPRUN, ERRMSG, FDPNT,                                NS200130
                        3    FILESV, FLDHND, HDATA, HEAD(88), ID(200), IMAGEX,             NS200140
                        4    IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,                         NS200150
                        5    PAGS(2), PNCH, POINT, PRESUX, PRNTR, READIN,                  NS200160
                        6    RESTRT, RUNFIL, RUNTAB(10,3),                                 NS200170
                        7    SDATA, SEPARX, SEPTPX, SPARE(10), TEMPAS(30),                 NS200180
                        8    TPSTAT(6), TIFLDX, TYPEWR,                                    NS200190
                        9    TOP, ARRAY(12500)                                             NS200200
0004                     REAL * 8 ARRAY                                                    NS200210
0005                     REAL * 4 EQUAL(5,30)                                              NS200220
0006                     INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)      NS200230
0007                     INTEGER * 2 BLANK2                                                NS200240
0008                     LOGICAL * 4 CHKOUT                                                NS200250
0009                     LOGICAL * 1 BLANK1                                                NS200260
0010                     EQUIVALENCE (DATSAV,ID(1)), (CURRIN,ID(3)), (FROCAL(1),ID(5)),    NS200270
                        1        (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),NS200280
                        2        (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),                 NS200290
                        3        (MAPSAV,TPSTAT(1)),                                       NS200300
                        4        (SUPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),NS200310
                        5        (COMPSFR,TPSTAT(5)), (TRADLT,TPSTAT(6)),                  NS200320
                        6        (BLANK,BLANK2,BLANK1)                                     NS200330
                 C                                                                       *NS200340
                 C                                                                         NS200350
                 C       THE NEXT 7 STATEMENTS ARE ONLY FOR THE PURPOSES OF THIS FUNCTION  NS200360
                 C       AND DO NOT BELONG TO THE STANDARD GLOCOM VARIABLES OF LARSYS      NS200370
                 C                                                                       *NS200380
0011                     INTEGER*2 NNID(400)                                               NS200390
0012                     EQUIVALENCE (ID(1), NNID(1))                                      NS200400
                 C                                                                       *NS200410
0013                     COMMON /NS2COM/ CSET(3,30), MTXSIZ, NOFET3, NOPOOL, CFILE, OTAPE,
                        1 POLNAM(2,60), RESULT, ROFILE, RUTAPE,STKPTR, VECSIZ, WRKSIZ,
                        2 CSEL(30), POLPTR(2,60), POLSTK(60), POLNM1(60)
                 C
0014                     REAL*4 CSET
                 C
0015                     INTEGER*4 POLNAM
                 C
0016                     INTEGER*2 CSEL, POLPTR, POLSTK
                 C
0017                     LOGICAL*1 POLNM1
                 C
                 C
0018                     LOGICAL*1 POLNM1
                 C
                 C                                                                       *NS200510
                 C                                                                       *NS200520
0019                     INTEGER*4 NOCNT(60), KEPPTS(60), REDBUF(20), PREFIX(2)            NS200530
0020                     INTEGER*2 BINFLG,FETVEC(30),FETVC3(30)                            NS200540
0021                     DATA EOS/'EOS'/,END/'END'/                                        NS200550
                 C                                                                       *NS200560
                 C***********************************************************************NS200570
                 C                                                                       *NS200580
                 C       READ AND SAVE STATISTICS                                          NS200590
                 C                                                                       *NS200600
                 C***********************************************************************NS200610
                 C                                                                       *NS200620
0022                     NOFLD3 = 0                                                        NS200630
0023                     NOFLDS = 0                                                        NS200640
0024                     UNIT = CRDRDR                                                     NS200650
0025                     IF(CARD(1) .EQ. END)CALL ERPRNT(254, 'STOP')                      NS200660
0026                     CALL STAT(UNIT,NOCLS,NOFLD,NOFEAT,NUM,BINFLG)                     NS200670
0027                     REWIND SDATA                                                      NS200680
0028                     MAXCLS = NOCLS                                                    NS200690
0029                     VARSIZ = NOFEAT*(NOFEAT+1)/2                                      NS200700
```


FILE NS2INT

```
0030                     COVAR1 = 1                                                        NS200710
0031                     AVAR1 = COVAR1+(VARSIZ*MAXCLS + 1)/2                              NS200720
0032                     CLSID1 = AVAR1 + (NOFEAT*MAXCLS+1)/2                              NS200730
0033                     DATBAS = CLSID1 + MAXCLS                                          NS200740
0034                     CORE = TOP -(DATBAS-1)*8                                          NS200750
0035                     IF(CORE .LE. 0) CALL ERPRNT(371,'STOP')                           NS200760
0036                     CALL REDSTA(ARRAY(COVAR1),ARRAY(AVAR1), ARRAY(CLSID1),KEPPTS,     NS200770
                        *     BINFLG, NOCNT, NOCLS, CSET, CSEL, FETVEC, VARSIZ, NOFEAT)    NS200780
                 C                                                                       *NS200790
                 C***********************************************************************NS200800
                 C                                                                       *NS200810
                 C       CHECK  CLASSES VALIDITY                                         * NS200820
                 C                                                                       *NS200830
                 C***********************************************************************NS200840
                 C                                                                       *NS200850
0037                     CALL CLSCHK(ARRAY(CLSID1), NOCLS, NOPOOL, POLPTR, POLSTK, POLNAM, NS200860
                        *       POLNM1, STKPTR, NO)                                        NS200870
                 C                                                                       *NS200880
                 C***********************************************************************NS200890
                 C                                                                       *NS200900
                 C       MOUNT THE INTERMEDIATE RESULTS TAPE                              *NS200910
                 C                                                                       *NS200920
                 C***********************************************************************NS200930
                 C                                                                       *NS200940
0038                     MODE = 0                                                          NS200950
0039                     CALL TAPMNT(OTAPE, OFILE, MODE)                                   NS200960
0040                     CALL TPRD(DATAPE, 800, ERROR, ID, 2, &EF)                         NS200970
0041                     NOFET3=ID(5)                                                      NS200980
0042                     VECSIZ= NOFET3                                                    NS200990
0043                     MTXSIZ=VECSIZ*(VECSIZ+1)/2                                        NS201000
0044                     DO 120 K = 1, NOFET3                                              NS201010
0045               120   FETVC3(K) = NNID(39+2*K)                                          NS201020
0046                     L = 1                                                             NS201030
0047                     DO 130 K2 = 1,30                                                  NS201040
0048                     CSEL(K2) = 0                                                      NS201050
0049                     IF(FETVC3(L) .NE. K2) GO TO 130                                   NS201060
0050                     CSEL(K2) = NNID(40+2*L)                                           NS201070
0051                     L = L+1                                                           NS201080
0052               130   CONTINUE                                                          NS201090
                 C                                                                       *NS201100
                 C***********************************************************************NS201110
                 C                                                                       *NS201120
                 C       CALCULATE BASES OF REDUCED ARRAYS                                *NS201130
                 C                                                                       *NS201140
                 C***********************************************************************NS201150
                 C                                                                       *NS201160
0053                     VARSZ3 = NOFET3*(NOFET3+1)/2                                      NS201170
0054                     COVAR3 = COVAR1                                                   NS201180
0055                     AVAR3 = COVAR3 + (VARSZ3*NOPOOL +1)/2                             NS201190
0056                     SCALAR = AVAR3 + (NOFET3*NOPOOL+1)/2                              NS201200
0057                     DETAR = SCALAR + (NOPOOL + 1)/2                                   NS201210
0058                     CONAR = DETAR + (NOPOOL +1) /2                                    NS201220
0059                     WORKBS = CONAR + (NOPOOL+1) /2                                    NS201230
                 C                                                                       *NS201240
                 C       REDUCE THE STATISTICS                                             NS201250
                 C                                                                       *NS201260
0060                     CALL REDSAV(ARRAY(COVAR1),ARRAY(AVAR1),ARRAY(COVAR3),             NS201270
                        * ARRAY(AVAR3), KEPPTS, NOPOOL, NOFEAT, VARSIZ, NOFET3, POLPTR,    NS201280
                        * POLSTK, FETVEC, FETVC3, VARSZ3)                                  NS201290
                 C                                                                       *NS201300
                 C***********************************************************************NS201310
                 C                                                                       *NS201320
                 C       PRINT SUPERVISOR INFORMATION                                     *NS201330
                 C                                                                       *NS201340
                 C***********************************************************************NS201350
                 C                                                                       *NS201360
0061                     CALL GTSERL(SERIAL)                                               NS201370
0062               200   WRITE(PRNTR,1000)                                                 NS201380
0063              1000   FORMAT(//' NONSUPERVISED ECHO (PHASE2) INFORMATION....'/)         NS201390
0064                     WRITE(PRNTR,1001)SERIAL,  NOPOOL, NOFET3, NOFLD3                  NS201400
0065              1001   FORMAT(5X,'CLASSIFICATION STUDY.... ',I7                          NS201410
                        *     /5X,'NO. OF CLASSES................ ',I4                      NS201420
                        *     /5X,'NO. OF CHANNELS............... ',I4                      NS201430
                        *     /5X,'NO. OF FIELDS................. ',I4)                     NS201440
0066                     WRITE(PRNTR,1002) (FETVC3(I), I=1,NOFET3)                         NS201450
0067              1002   FORMAT(5X,'CHANNELS SELECTED ARE........ ',(I17,15(I3,',')))      NS201460
```

FILE NS2INT

```
               C
               C*****************************************************************
               C
               C      WRITE FIRST RECORD ON RESULTS TAPE
               C
               C*****************************************************************
               C
0068                  I0 = 0
0069                  I3 = 3
0070                  PREFIX(1) = 1
0071                  PREFIX(2) = 0
0072                  WRITE(RESULT)PREFIX,RQTAPE,RQFILL,I3,I0,SERIAL,(I0,I=1,7)
0073                  REWIND SDATA
0074                  PREFIX(1) = 2
0075                  WRITE(RESULT)PREFIX,NO,NOFET3,NOFLD3,NOPCCL,(FETVC3(I),I=1,NOFET3)
                     * ,(CSEL(FETVC3(I)),I=1,NOFET3), (FROCAL(1,FETVC3(I)),I=1,NOFET3),
                     * (FROCAL(2,FETVC3(I)),I=1,NOFET3),((POLNAM(I,J),I=1,2),J=1,NOPOOL)
                     * ,((POLPTR(I,J),I=1,2),J=1,NOPOOL),(POLSTK(I),I=1,NC),DATE
0076                  PREFIX(1) = 3
0077           220    READ(SDATA,9220,END=300)CARD
0078           9220   FORMAT(20A4)
0079                  WRITE(RESULT)PREFIX,CARD
0080                  IF(CARD(1) .NE. EOS) GO TO 220
0081           300 .  CONTINUE
               C
               C*****************************************************************
               C
               C      CALL TO THE PROCESSOR SUBROUTINE  N2ECHO
               C
               C*****************************************************************
               C
0082                  FAVBAS = WORKBS + (NOFET3*(NOFET3+2)+1)/2
0083                  FCVBAS = FAVBAS + (VECSIZ + 1)/2
0084                  SAVBAS = FCVBAS + (MTXSIZ + 1)/2
0085                  HLDBAS = SAVBAS + (VECSIZ*NOPOOL + 1)/2
0086                  CHIBAS = HLDBAS + (VECSIZ*7 + 1)/ 2
0087                  DATBS3 = CHIBAS + (NOPOOL + 1) /2
0088                  WRKSIZ = NOFET3*(NOFET3+2)
               C
0089                  CALL N2ECHO(ARRAY(COVAR3), ARRAY(AVAR3), ARRAY(SCALAR),
                     *  ARRAY(DETAR), ARRAY(CONAR), ARRAY(WORKBS), ARRAY(FAVBAS),
                     *  ARRAY(FCVBAS), ARRAY(SAVBAS), ARRAY(HLDBAS), ARRAY(CHIBAS),
                     * ARRAY(DATBS3), DATBS3)
               C
               C*****************************************************************
               C
               C      WRITE THE LAST FILE ON RESULTS TAPE                      *
               C
               C*****************************************************************
               C
0090                  CALL TCPEF(RESULT)
0091                  CALL TCPRF(RESULT)
0092                  K = -1
0093                  FILESV = FILESV + 1
0094                  COPFIL = COPFIL + 1
0095                  PREFIX(1) = 1
0096                  PREFIX(2) = 0
0097                  WRITE(RESULT) PREFIX, RQTAPE,  FILESV, I3, K, (I0, J=1,8)
0098                  CALL TOPEF(RESULT)
0099                  CALL TCPRF(RESULT)
0100                  CALL TCPEF(RESULT)
0101                  CALL TOPDF(RESULT)
0102                  CALL TOPRF(RESULT)
0103                  RETURN
0104                  END
```

LARS Program Abstract ________

MODULE IDENTIFICATION

Module Name: N2ECHO   Function Name: NS2ECHO

Purpose: Performs classification for NS2ECHO and writes results tape.

System/Language: CMS/FORTRAN

Author: ________   Date: ________

Latest Revisor: C. A. Pomalaza   Date: 8/21/77

MODULE ABSTRACT

N2ECHO is called by NS2INT to perform the field classification on those fields and singular points identified by NS2ECHO (field extraction phase) and wirtten on an intermediate tape. N2ECHO writes a standard results file to tape.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
1220 Potter Drive
West Lafayette, Indiana 47906

1. Module Usage

N2ECHO

CALL N2ECHO (COVMTX, AVEMTX, SCAPRD, DETCOV, CONST, WORK, MEAN, COV, SAVAVE, HOLD, CHISQR, RDATA, DATBS3)

Input Arguments

COVMTX	R*1 Array of covariances obtained in NS2INT. Dimensioned NTXSIZ*NOPOOL where NOPOOL is the number of classes contained in the reduced STAT DECK and MTXSIZ = VECSIZ * (VECSIZ + 1)/2 where VECSIZ is the number of channels requested for classification (in Phase 1).
AVEMTX	R*4 Array of mean vectors for each class. Dimensioned VECSIZ*NOPOOL.
SCAPRD	R*4 Array of the scalar product of the mean and covariance matrices for each class dimensioned NOPOOL.
DETCOV	R*4 Array of the determinant of the covariance matrix for each class dimensioned NOPOOL.
CONST	R*4 Array of the constant terms used in the computation of the likelihood values dimensioned NOPOOL.
WORK	R*4 Working array for a subroutine cell dimensioned VECSIZ * (VECSIZ + 2).
MEAN	R*4 Array used as a holding buffer for a subroutine cell. Dimensioned VECSIZ.
COV	R*4 Array used as a holding buffer for a subroutine cell. Dimensioned MTXSIZ.
SAVAVE	R*4 Array used for saving the mean from each class of the STAT deck. Dimensioned VECSIZ*NOPOOL.
HOLD	R*4 Array used as a holding buffer for subroutine call. Dimensioned 7*VECSIZ.
CHISQR	R*4 Array used in a subroutine call. Dimensioned NOPOOL.
RDATA	R*4 Area to be used for dynamic allocation. This must be the first unused element of ARRAY in the calling program.

DATBS3 I*4 Number of bytes on ARRAY which one in use (i.e. the number of bytes in ARRAY which precede RDATA).

Output Arguments

Not applicable.

2. Internal Description

1 - N2ECHO calls to LARSYS system routine SMMULT and SAMINV to invert the covariance matrices in COVMTX and produce the determinant in DETCOV needed for classification.

2 - A loop begins where each field statistics matrix (record type 4) on the intermediate tape is classified by LARSYS system routine SAMCLS. The absolute field number (see tape record description) indexes the class.

3 - The records that associates each pixel with its appropriate field are read. The processor classifies each pixel in an homogeneous field by looking up its absolute field number. Singular pixels are classified by a call to CLASS. The results are written line by line following the standard LARSYS format (see RESULTS FILE description in LARSYS System Manual).

4 - The RESULTS FILE is closed and control returns to NS2INT.

3. Input Description

N2ECHO reads field statistics information and pixel data from the intermediate tape via call to LARSYS system routines TOPRD, and GADLIN.

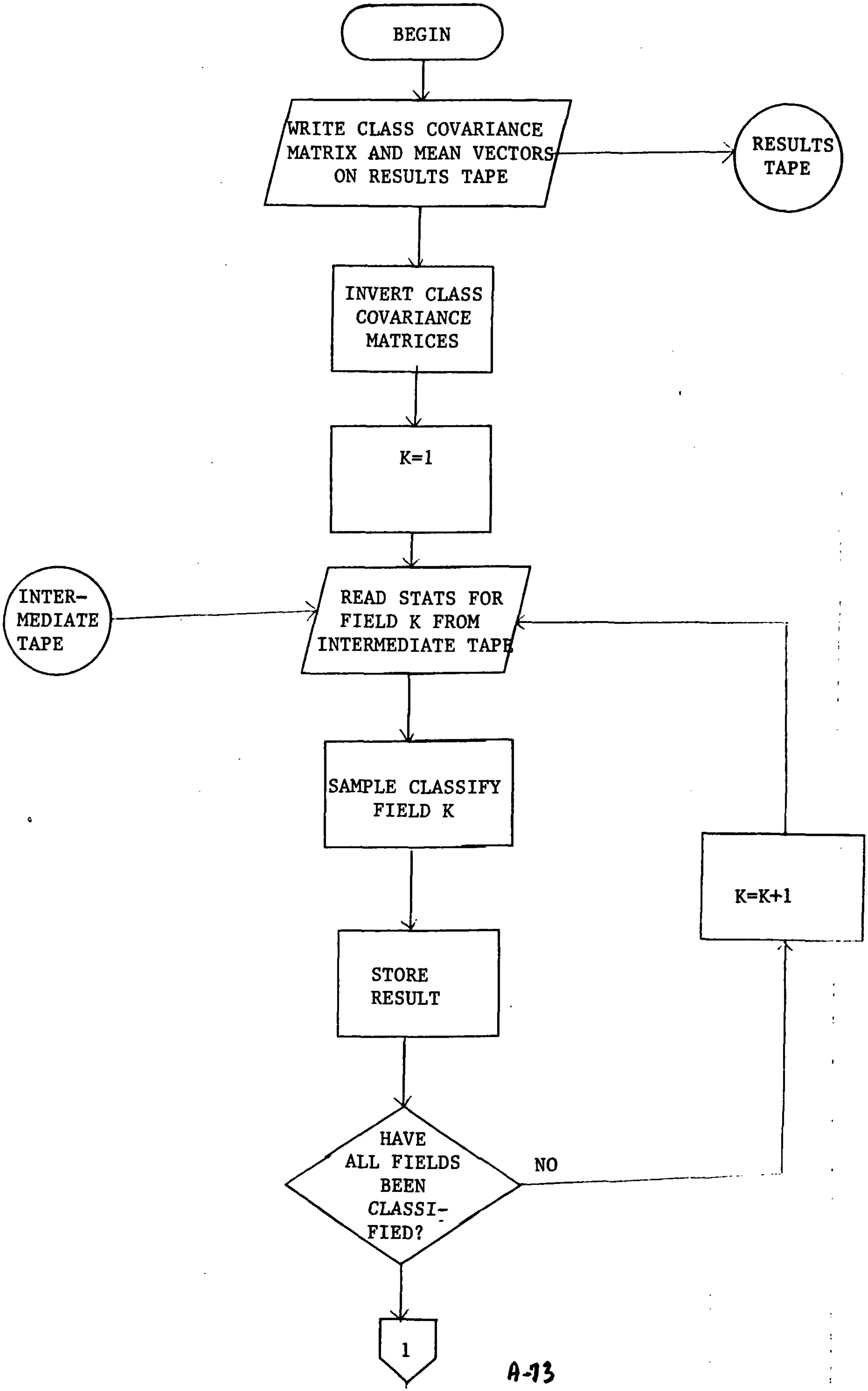
4. Output Description

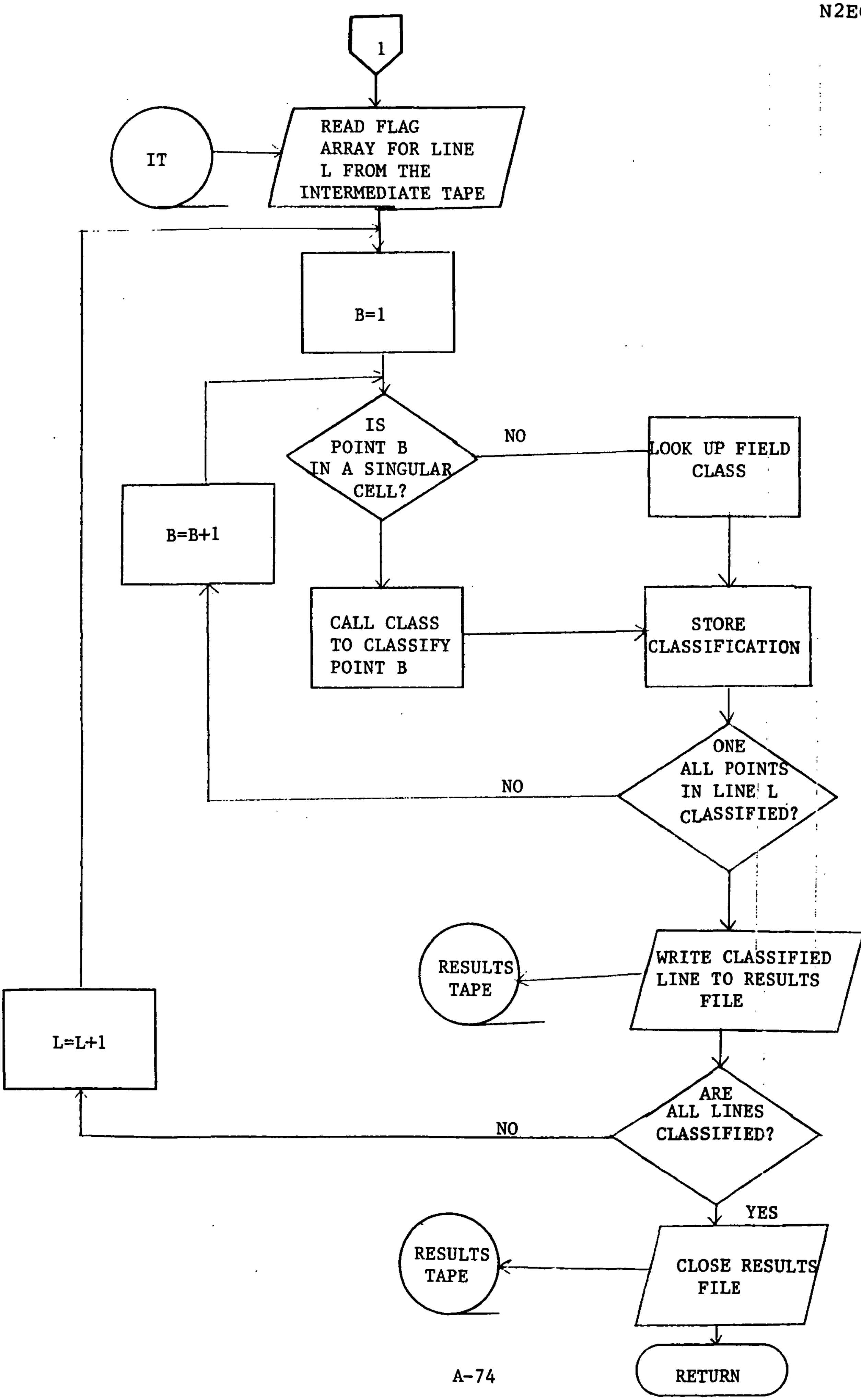
A standard results file is output to tape. See LARSYS System Manual for format description.

5. Supplemental Information

See non-supervised ECHO intermediate tape description. Also Results Tape description on LARSYS System Manual.

6. Flowchart


BEGIN
WRITE CLASS COVARIANCE MATRIX AND MEAN VECTORS ON RESULTS TAPE
RESULTS TAPE
INVERT CLASS COVARIANCE MATRICES
K=1
INTER-MEDIATE TAPE
READ STATS FOR FIELD K FROM INTERMEDIATE TAPE
SAMPLE CLASSIFY FIELD K
K=K+1
STORE RESULT
HAVE ALL FIELDS BEEN CLASSI-FIED?
NO
1


1
IT
READ FLAG ARRAY FOR LINE L FROM THE INTERMEDIATE TAPE
B=1
IS POINT B IN A SINGULAR CELL?
NO
LOOK UP FIELD CLASS
B=B+1
CALL CLASS TO CLASSIFY POINT B
STORE CLASSIFICATION
ONE ALL POINTS IN LINE L CLASSIFIED?
NO
RESULTS TAPE
WRITE CLASSIFIED LINE TO RESULTS FILE
L=L+1
ARE ALL LINES CLASSIFIED?
NO
YES
RESULTS TAPE
CLOSE RESULTS FILE
RETURN

FILE N2ECHO

```
                 C                                                                        *GC000010
                 C       N2ECHO          LARS XXXX                                         GC000020
                 C                       WRITTEN BY C. A. POMALAZA                         GC000030
                 C                                                                        *GC000040
                 C************************************************************************GC000050
                 C                                                                        *GC000060
0001                    SUBROUTINE N2ECHO(COVMTX,AVEMTX,SCAPRD,DETCOV,CONST,WORK,          GC000070
                       & MEAN,COV,SAVAVE,HOLD,CHISQR,RDATA,DATBS3)                         GC000080
                 C                                                                        *GC000090
0002                    IMPLICIT INTEGER*4 (A-Z)                                           GC000100
                 C                                                                        *GC000110
0003                    COMMON /GLCCOM/ BLANK, CARD(20), CHKOUT, COPFIL, CLASSR, CLASSX,   GC000120
                       1    CLUSTX, CONPUT, CPYOUT, CRDRDR, CRDSEQ, DATAPE,                GC000130
                       2    DUPLTP, DUPRUN, ERRMSG, FRPNT,                                 GC000140
                       3    FILESV, FLDBND, HDATA, HEAD(88), ID(200), IMAGEX,              GC000150
                       4    IMARK, KEYBD, MAPTAP, MAXCHA, MAXCLS,                          GC000160
                       5    PAGSIZ, PNCH, POINT, PRESUX, PRNTR, READIN,                    GC000170
                       6    RESTRT, RUNFIL, RUNTAB(10,3),                                  GC000180
                       7    SDATA, SEPARX, SEPTPX, SPARE(10), TEMPAS(30),                  GC000190
                       8    TPSTAT(6), TTFLDX, TYPEWR,                                     GC000200
                       9    TOP, ARRAY(12500)                                              GC000210
0004                    REAL * 8 ARRAY                                                     GC000220
0005                    REAL * 4 FROCAL(5,30)                                              GC000230
0006                    INTEGER * 4 COMENT(16), DATE(5), HED1(16), HED2(16), TIME(5)       GC000240
0007                    LOGICAL * 4 CHKOUT                                                 GC000250
0008                    INTEGER * 2 BLANK2                                                 GC000260
0009                    LOGICAL * 1 BLANK1                                                 GC000270
0010                    EQUIVALENCE (DATSAV,ID(1)), (CURRUN,ID(3)), (FROCAL(1),ID(51)),    GC000280
                       1         (HED1(1),HEAD(8)), (DATE(1),HEAD(26)), (HED2(1),HEAD(39)),GC000290
                       2         (TIME(1),HEAD(58)), (COMENT(1),HEAD(72)),                 GC000300
                       3         (MAPSAV,TPSTAT(1)),                                       GC000310
                       4         (SEPSCR,TPSTAT(2)), (DUPIN,TPSTAT(3)), (DASTAT,TPSTAT(4)),GC000320
                       5         (CDPSER,TPSTAT(5)), (TRADUT,TPSTAT(6)),                   GC000330
                       6         (BLANK,BLANK2,BLANK1)                                     GC000340
                 C                                                                        *GC000350
0011                    COMMON /NS2COM/ CSET(3,30), MTXSIZ, NOFET3, NOPOOL, OFILE, OTAPE,
                       1 POLNAM(2,60), RESULT, ROFILE, ROTAPE,STKPTR, VECSIZ, WRKSIZ,
                       2 CSEL(30), POLPTR(2,60), POLSTK(60), POLNM1(60)
                 C
0012                    REAL*4 CSET
                 C
0013                    INTEGER*4 POLNAM
                 C
0014                    INTEGER*2 CSEL, POLPTR, POLSTK
                 C
0015                    LOGICAL*1 POLNM1
                 C
                 C
0016                    LOGICAL*1 POLNM1
                 C
                 C                                                                        *GC000450
                 C      DEFINITION OF LOCAL VARIABLES                                      GC000460
                 C                                                                        *GC000470
0017                    REAL * 4 SPAR(10),AR(125)                                          GC000480
0018                     REAL* 4 CSET3(3,30)                                               GC000490
0019                    INTEGER* 4 AR1(125)                                                GC000500
0020                    INTEGER* 2 A,B,AR2(250)                                            GC000510
0021                     INTEGER*2 CSEL3(30), BLOCK(4)                                     GC000520
0022                    LOGICAL* 1 LR(8), RS(2)                                            GC000530
0023                    EQUIVALENCE (AR(1),AR1(1), AR2(1))                                 GC000540
0024                    EQUIVALENCE (LR(1), IR(1))                                         GC000550
                 C                                                                        *GC000560
0025                    REAL*4 MEAN(VECSIZ),COV(MTXSIZ)                                    GC000570
0026                    INTEGER*4 REJBUF(20),IR(2)                                         GC000580
0027                    INTEGER*2 BINFLG,NEGONE,IN,FETVEC(30),                             GC000590
                       & FETVC3(30)                                                        GC000600
0028                    EQUIVALENCE (IN,RS(1))                                             GC000610
0029                    INTEGER*4 INFO(17),PREFIX(2)                                       GC000620
0030                    INTEGER*2 BUFFER(2500)                                             GC000630
0031                    INTEGER * 2 SBBUF(22500)                                           GC000640
0032                     REAL*4 RDATA(1)                                                   GC000650
0033                    INTEGER*2 TAPBUF(3000)                                             GC000660
0034                    REAL*4 COVMTX(MTXSIZ,NOPOOL),AVEMTX(VECSIZ,NOPOOL),WORK(WRKSIZ),   GC000670
                       & SCAPRD(NOPOOL),CHISQR(NOPOOL),DETCOV(NOPOOL),CONST(NOPOOL),       GC000680
                       & HOLD(7,VECSIZ), SAVAVE(VECSIZ,NOPOOL), THRTAB(234), PCREJ         GC000690
                 C                                                                        *GC000700
```

FILE N2ECHO

```
                 C
                 C**********************************************************************
                 C
                 C      WRITE COVARIANCE MATRIX AND MEAN VECTOR ON RESULTS TAPE
                 C
                 C**********************************************************************
                 C
0035                    REWIND SDATA
0036                    PREFIX(1)=4
0037                    PREFIX(2)=1
0038                    WRITE(RESULT)PREFIX,COVMTX,AVEMTX
0039                    NEGONE=-1
                 C
                 C**********************************************************************
                 C
                 C       INVERT THE LEADING STAT DECKS ON INTERMEDIATE TAPE
                 C
                 C**********************************************************************
                 C
0040                    DO 510 I=1,NOPOOL
0041                    CALL SAMINV(COVMTX(1,I),NOFET3,DETCOV(I),WORK)
0042                    IF (DETCOV(I).LE.0.) GO TO 520
0043                    DO 789 N = 1, VECSIZ
0044                788 SAVAVE(N,I) = AVEMTX(N,I)
0045                    CALL SMMULT(COVMTX(1,I),AVEMTX(1,I),SCAPRD(I),WORK,NOFET3)
0046                510 CONST(I)=-.5*ALOG((6.2831853**NOFET3)*DETCOV(I))
                 C
                 C      THE CODE BELOW BUILDS A TABLE OF THRESHOLD VALUES
                 C      CORRESPONDING TO FRACTIONS REJECTED FROM 0.1 PERCENT
                 C      TO 99 PERCENT. PCEREJ IS 1.0 -PERCENT REJECTION/100.
                 C
0047                    PCREJ = 1.0
0048                    DO 511 I=1, 234
0049                    PCREJ = PCREJ - .001
0050                    IF(I .GT. 150) PCREJ = PCREJ - 0.009
0051                    CALL THRESC(PCREJ, THRTAB(I), VECSIZ, ERROR)
0052                    IF(ERROR .NE. 0) CALL ERPRNT(377, 'STOP')
0053             511    CONTINUE
0054                    GO TO 535
0055                520 CALL ERPRNT(160,'GO')
0056                    WRITE(PRNTR,9222)
0057               9222 FORMAT('SINGULAR CLASS')
0058                    STOP
                 C
                 C**********************************************************************
                 C
                 C      CLASSIFY EACH FIELD
                 C
                 C**********************************************************************
                 C
0059             535    CALL TCPRD(12,68,ERROR,INFO,2,WLF)
0060                    CALL TOPRD(12,16,ERROR,SPAR,2,WLF)
0061                    CLLWTH = SPAR(1)
0062                    LCOUNT=8+(VECSIZ+MTXSIZ)*4
0063                    L1=ID(4)
0064                    DO 20 J=1,L1
0065                    CALL TCPRD(12,LCOUNT,ERROR,AR(1),2,WLF)
0066                    A=AR2(1)
0067                    B=AR2(2)
0068                    D=AR1(2)
0069                    DO 11 K2=1,VECSIZ
0070                 11 MEAN(K2)=AR(2+K2)
0071                    DO 12 K2=1,MTXSIZ
0072                 12 COV(K2)=AR(2+VECSIZ+K2)
0073                    CALL SAMCLS(CLASS1,CHISQR,A1,COVMTX,AVEMTX,SCAPRD,CONST,COV,MEAN,
                      & D,VECSIZ,MTXSIZ,NOPOOL,.FALSE.)
0074                    SOBUF(J)=CLASS1
0075                 20 CONTINUE
                 C
                 C**********************************************************************
                 C
                 C      WRITE RECORD 5 ON RESULTS FILE
                 C
                 C**********************************************************************
                 C
0076                    PREFIX(1)=5
```

FILE N2ECHO

```
0077              PREFIX(2)=1
0078              PTO=(INFO(3)-INFO(7)+1)/INFO(9)
0079              LINQ=(INFO(5)-INFO(4)+1)/INFO(6)
0080              WRITE(RESULT)PREFIX,PTQ,LINQ,INFO,ID,((CSET(I,J),I=1,3),J=1,30)
            C********************************************************************
            C                                                                   *
            C     CHECK FOR CORE OVERFLOW                                       *
            C                                                                   *
            C********************************************************************
            C                                                                   *
0081              NSR = PTQ + 6
0082              NSD = ID(6)
0083              RBASE = NSR*VECSIZ + 1
0084              LB = (NSD*VECSIZ + 3)/4 + RBASE
0085              IF (MOD(LB,2) .EQ. 1) LB = LB + 1
0086              BASE = DATBS3*8 + LB*4
0087              CORE = TOP- BASE
0088              IF(CORE .GT. 0) GO TO 650
0089              CORE = -CORE
0090              CALL ERPRNT(376, 'GO')
0091              WRITE(TYPEWR, 9600) CORE
0092              WRITE(PRNTR, 9600) CORE
0093        9600  FORMAT(10X, I6, 'BYTES')
0094              CALL RTMAIN
            C                                                                   *
            C                                                                   *
            C********************************************************************
            C                                                                   *
            C     ENTER MAIN LOOP OF CLASSIFICATION                             *
            C                                                                   *
            C********************************************************************
            C                                                                   *
0095        650   DO 600 I = 1,30
0096              CSEL3(I) = 0
0097              DO 600 J = 1,3
0098        600   CSET3(J,I) =  -50000.
0099              K = 0
0100              DO 610 I = 1 ,30
0101              IF(CSEL(I) .EQ. 0) GO TO 610
0102              K = K + 1
0103              CSEL3(K) = CSEL(I)
0104              DO 620 J = 1,3
0105              CSET3(J,K) = FRQCAL(J+2, I)
0106        620   CONTINUE
0107        610   CONTINUE
            C                                                                   *
            C                                                                   *
0108              BLOCK(2) = 1
0109              BLOCK(3) = PTQ
0110              BLOCK(4) = 1
0111              PTS = PTQ
0112              PTS6 = PTS + 6
0113              JPTS=PTS/CELWTH
0114              BUF=9*JPTS
0115          690 COUNT=VECSIZ*(PTS6)+4
0116              DO 702 KM=1,LINQ
0117               BLOCK(1) = KM
0118               CALL GADLIN(BLOCK(1), CSEL3, CSET3, ID, 12, VECSIZ, NSR,
                 *          RDATA(RBASE), RDATA(1), ROLL, ERROR)
0119              CALL TCPRO(12,BUF,ERROR,BUFFER,2,WLF)
0120              RDPTR=1
0121              BUFPTR=1
0122              OUTPTR=3
0123              TAPBUF(1) = 0
0124              TAPBUF(2)=KM+INFO(4)-1
0125              DO 703 K9=1,VECSIZ
0126              DO 703 K10=1,6
0127        703   FOLD(K10+1, K9) = RDATA( PTS +(K9-1)*PTS6 + K10)
0128              DO 700 KCO=1,JPTS
0129              IF (BUFFER(BUFPTR).EQ.NEGONE) GO TO 720
0130              DO 710 K2=1,CELWTH
0131              BUFNUM=BUFFER(BUFPTR+1)
0132              TAPBUF(OUTPTR)=SQBUF(BUFNUM)
0133          710 OUTPTR=OUTPTR+1
0134              RDPTR=RDPTR+CELWTH
0135              BUFPTR=BUFPTR+4
```

FILE N2ECHO

```
0136           GO TO 700
0137       720 DO 740 K7=1,CELWTH
0138           DO 730 K3=1,VECSIZ
0139       730 HOLD(1,K3)=RDATA(RDPTR + (K3-1)*PTS6 )
0140       777 CALL CLASS(HOLD,VECSIZ,1,NOPOOL,SAVAVE,COVMTX,IR(1),IR(2),CONST,
              & THRTAB)
0141           RS(2)=LR(2)
0142           RS(1)=LR(6)
0143           TAPBUF(OUTPTR)=IN
0144           OUTPTR=OUTPTR+1
0145       740 RDPTR=RDPTR+1
0146           BUFPTR=BUFPTR+4
0147       700 CONTINUE
0148           PREFIX(1)=6
0149           PREFIX(2)=1
0150           PT=PTS+2
0151           WRITE(11)PREFIX,(TAPBUF(J),J=1,PT)
0152       692 CONTINUE
0153           TAPBUF(1)=0
0154           TAPBUF(2) = 0
0155           PREFIX(1)=7
0156           WRITE(RESULT)PREFIX,(TAPBUF(J),J=1, PT)
0157           PTQ = 0
0158           PREFIX(1) = 8
0159           LINQ = 0
0160           WRITE(RESULT)PREFIX,PTQ,LINQ,INFO,ID,CSET
0161           RETURN
0162           END
```

APPENDIX B

Ancillary Data Files Required by the Nonsupervised ECHO Processor

Nonsupervised Field Extraction Processor Disk Files

The NS1ECHO processor writes two disk files during execution. The first file BUFFER FILE is written as a sequential file by Subroutine NSECHO to store field annexation information row by row during the processing. BUFFER FILE is read by RDWRTE during the writing of the intermediate tape. The second file is also written by NSECHO to store the field statistics. This file is used by RDWRTE to compute the data values on the Intermediate Results Tape if the 'MAP' options has been selected (see control card description for NS1ECHO).

BUFFER FILE

This file contains one record of length m bytes for every line processed by NSECHO. The length m equals four times the number of cells/line. The format of the record is I*2 with the following structure.

BUFFER(2i+1) = -1 if cell i is not homogeneous.
The relative field number of cell i if cell i is homogeneous.

BUFFER(2i+2) = -1 if cell i is not homogeneous.
Absolute statistics record number (in the STAT SCRATCH disk file) if cell i is homogeneous.

STAT SCRATCH

This file contains one record of 2+NC+NC*(NC+1)/2 words for every field opened by NSECHO where NC is the number of channels considered by NSECHO. Each record has the following structure:

STAT(1)	I*2 Relative field number.
STAT(2)	I*2 Absolute sequence number of the field.
STAT(3)	I*4 Number of pixels in the field used to compute these statistics.
STAT(3+I)	R*4 Means for i^{th} channel if field STAT(2).
STAT(3+NC+1) until STAT(3+NC+NC*(NC+1)/2)	R*4 Correlation matrix of the field.

The BUFFER FILE is written sequentially using unformatted fortran IO. STAT SCRATCH is written using system support routine DEFINE FILE and unformatted fortran IO.

INTERMEDIATE TAPE FILE

This file is written on tape by Subroutine RDWRTE in the Nonsupervised ECHO processor's Field Extraction Phase and becomes the primary input to NS2ECHO (the Classification Phase). The Nonsupervised ECHO Intermediate Tape File is composed of six different types of data records. One each of record types 1, 2, and 3 are written on each Intermediate Tape File, one record type 4 is written for each set of field statistics recorded in the STAT SCRATCH file (one covariance matrix and a vector of channel-means for each field isolated by NSECHO), and one type 5 followed by a type 6 for every line processed in the Nonsupervised ECHO Field Extraction Phase.

Record Type 1 (ID Record)

This is similar to the conventional 800 Byte LARSYS ID record described in the LARSYS SYSTEM MANUAL 1 . This records structure is as follows:

Bytes	Format	Size	Description
1-4	I*4	1 word	Intermediate tape number
5-8	I*4	1 word	Intermediate tape file number
9-12	I*4	1 word	Intermediate run number
13-16	I*4	1 word	Number of fields isolated by the field extraction phase
17-20	I*4	1 word	Number of data channels
21-24	I*4	1 word	Number of data samples per channel per line
25-40	Alpha	4 words	Flightline Identification (16 characters)
41-44	I*4	1 word	Month data was taken
45-48	I*4	1 word	Day data was taken
49-52	I*4	1 word	Year data was taken
53-56	Alpha	1 word	Time data was taken
57-60	I*4	1 word	Altitude of aircraft
61-64	I*4	1 word	Ground heading of aircraft
65-76	Alpha	2 words	Date data run was generated (12 characters)
77-80	I*4	1 word	Number of lines in this run

Bytes	Format	Size	Description
81-82	I*2	½ word	Number of the first channel used by NSECHO
83-84	I*2	½ word	Calibration code of the first channel used by NSECHO
85-200	I*2		For each channel used repeat information of the half words in bytes 81-82 and 83-84. The remaining bytes are equal to 0.
201-203	R*4	1 word	Lower limit in Micrometers of the first spectral band on the original MIST tape
205-208	R*4	1 word	Upper limit in Micrometers of the first spectral band on the original MIST tape
209-212	R*4	1 word	The suggested value of "CO" calibration pulse for the first spectral band
213-216	R*4	1 word	The suggested value of "C1" calibration pulse for the first spectral band
217-220	R*4	1 word	The suggested value of "C2" calibration pulse for the first spectral band
221-800	R*3		Repeat of words in bytes 200 to 220 for the channels on the original MIST tape. The remaining bytes are set to 0.

Record Type 2

This record is 17 fullwords long. It describes the area which has been processed by the Field Extraction Phase of the Nonsupervised ECHO processor to produce the Intermediate Tape File.

Bytes	Format	Description
1-4	I*4	MIST run number of processed area
5-12	Alpha	Field designation on field description card
13-16	I*4	Beginning line number
17-20	I*4	Last line number
21-24	I*4	Line interval
25-28	I*4	First column number

Bytes	Format	Description
29-32	I*4	Last column number
33-36	I*4	Column interval
37-68	Alpha	Information from columns 51-80 on the field description card

Record Type 3

This record is 33 words long and stores the parameters used in the Nonsupervised Field Extraction Phase.

Bytes	Format	Parameters
1-4	R*4	Cell width
5-8	R*4	Variance test threshold
9-12	R*4	Mean test threshold
13-16	R*4	Homogeneity test threshold for the first channel used
17-132	R*4	Homogeneity test thresholds for the remaining channels

Record Type 4

This record is the same as on the STAT SCRATCH file. There are m records of type 4, where m is the number of fields isolated by the Field Extraction Phase of the Nonsupervised ECHO processor. Each record has 2+NC+NC*(NC+1)/2 words where NC is the number of channels used by NSECHO. The structure of each record is:

Size	Format	Contents
1-2	I*2	Relative field number
3-4	I*2	Absolute sequence number of the field
5-8	I*4	Number of pixels in the field with this statistics
9-(NC+2)*4	R*4	Mean value for the 1^{st}, 2^{nd}, . . . NC^{th} channel used by NSECHO
(NC+2)*4+1 - (NC+2)*4+NC*(NC+1)*2	R*4	Correlation matrix of the field

Record Type 5

This is similar to a standard data line in a LARSYS MIST tape. If the MAP option was specified, the original data has been altered so that for the pixels identified as falling in an object, the channel mean of the object replaces the raw data value. If the MAP option is not active, the raw data is copied unaltered to the intermediate tape.

Each data record will contain one scan line of data from ID(5) (see ID Record) channels. The first halfword (2 bytes) of the record will be the line number. The second halfword (2 bytes) will be the roll parameter (which is a number indicating relative position of the roll of the aircraft for this line of data). If the roll parameter is -32,767, the data for the given line does not exist. If the roll parameter has not been calculated, it will be set to 32,767. The fifth byte will be the first sample from the requested channel. The sixth byte will be the second sample from the first requested channel, and so on through ID(6) samples and ID(5) channels. A Type 5 record will be ID(5)*ID(6)+4 bytes long.

All data for each channel is from the field of view of the scanner except the last six bytes. The last six are calibration data in the order of appearance.

1. C_0 "0" or dark level
2. VC_0 Variance of C_0
3. C_1 Calibration source C_1
4. VC_1 Variance of C_1

5. C_2 Calibration source C_2

6. VC_2 Variance of C_2

where C_i - Calibration value i and VC_i - calculated variance of calibration value i

During the reformatting process a record may be had due to tape or other errors. When this happens, the data roll parameter and calibration points will be set to zero. On good data records all data and calibration values will be in the range of 0 to 255 (bit form) with no sign included in the eight bits. A data value of 0 to 255 means that the data point was cut off during the digitization process. Data values then range between 0 and 255 with 0 indicating low relative irradiance and 255 indicating high relative irradiance.

Record Type 6

Identical to a record in BUFFER FILE, i.e. it has m words for each line, processed, where me is the number of cells per line. The structure of each record is:

Bytes	Format	Contents
1-2	I*2	-1 if the cell is singular, otherwise relative field number of the cell
3-4	I*2	-1 if the cell is singular, otherwise number of the statistics record of the field the cell belongs to
5-2m	I*2	Similar to the first two halfwords describing the nature of the remaining (m-1) cells

Final Report Distribution List

NAS9-14970

NAME	NUMBER OF COPIES
NASA/Johnson Space Center Houston, Texas 77058	
ATTN: J. D. Erickson/SF3	(1)
ATTN: M. C. Trichel/SF3	(1)
ATTN: L. F. Childs/SF	(1)
ATTN: K. J. Demel/SF5	(1)
ATTN: F. Weber/SF5	(1)
ATTN: G. O. Boatwright/SF3	(1)
ATTN: K. Baker/SF4	(1)
ATTN: H. G. DeVezin, Jr./FM8	(1)
ATTN: R. P. Heydorn/SF3	(1)
ATTN: M. C. McEwen/SF3	(1)
ATTN: D. H. Hay/SF12	(1)
ATTN: D. L. Amsbury/SF5	(1)
ATTN: J. G. Garcia/SF3	(1)
ATTN: F. G. Hall/SF2	(1)
ATTN: B. L. Carroll/C09	(1)
ATTN: E. Laity/SF121	(2)
ATTN: R. Shirkey/JM6	(4)
ATTN: J. T. Wheeler/AT3	(1)
ATTN: G. E. Graybeal/SF4	(2)
ATTN: I. D. Browne/SF3	(5)
IBM Corporation FSD Mail Code 56 1322 State Park Drive Houston, Texas 77058	
ATTN: R. E. Oliver	(1)
Department of Mathematics Texas A&M University College Station, Texas 77843	
ATTN: L. F. Guseman, Jr.	(1)
ERIM P. O. Box 8618 Ann Arbor, Michigan 48107	
ATTN: R. F. Nalepka	(1)
ATTN: W. A. Malila	(1)
ATTN: R. C. Cicone	(1)
Kansas State University Department of Statistics, Calvin 19 Statistical Lab Manhattan, Kansas 66506	
ATTN: A. M. Feyerherm	(1)

NAME	NUMBER OF COPIES
U. S. Department of Interior Geological Survey GSA Building, Room 5213 Washington, D. C. 20242	
ATTN: Mr. W. A. Fischer	(1)
NASA Wallops Wallops Station, Virginia 23337	
ATTN: Mr. James Bettle	(1)
U. S. Department of Interior EROS Office Washington, D. C. 20242	
ATTN: Dr. Raymond W. Fary	(1)
U. S. Department of Interior EROS Office Washington, D. C. 20242	
ATTN: Mr. William Hemphill	(1)
NAVOCEANO, Code 7001 Bay St. Louis, MS 39520	
ATTN: Mr. J. W. Sherman, III	(1)
University of Texas at Dallas Box 688 Richardson, Texas 75080	
ATTN: Dr. Patrick L. Odell	(1)
Department of Mathematics University of Houston Houston, Texas 77004	
ATTN: Dr. Henry Decell	(1)
U. S. Department of Agriculture Statistical Reporting Service Room 4833, South Bldg. Washington, D. C. 20250	
ATTN: W. H. Wigton	(1)
Goddard Space Flight Center National Aeronautics & Space Administration Greenbelt, Maryland 20771	
ATTN: Mr. W. Alford, 563	(1)
ATTN: Dr. J. Barker, 923	(1)

NAME	NUMBER OF COPIES
EROS Data Center U. S. Department of Interior Sioux Falls, South Dakota 57198	
ATTN: Mr. G. Thorley	(1)
U. S. Department of Agriculture Soil & Water Conservation Research Division P. O. Box 267 Weslaco, Texas 78596	
ATTN: Dr. Craig Wiegand	(1)
U. S. Department of Interior USGA National Center Mail Stop 115 Geography Program Reston, Virginia 22092	
ATTN: Dr. James R. Anderson	(1)
Director, Remote Sensing Institute South Dakota State University Agriculture Engineering Building Brookings, South Dakota 57006	
ATTN: Mr. Victor I. Myers	(1)
U. S. Department of Agriculture Forest Service 240 W. Prospect Street Fort Collins, Colorado 80521	
ATTN: Dr. Richard Driscoll	(1)
University of California School of Forestry Berkeley, California 94720	
ATTN: Dr. Robert Colwell	(1)
Environmental Remote Sensing Applications Laboratory Oregon State University Corvallis, Oregon 97331	
ATTN: Dr. Barry J. Schrumpf	(1)
U. S. Department of Interior Director, EROS Program Washington, D. C. 20242	
ATTN: Mr. J. M. Denoyer	(1)

NAME	NUMBER OF COPIES
Remote Sensing Laboratory 129 Mulford Hall University of California Berkeley, California 94720 ATTN: C. M. Hay	(1)
Department of Mathematics University of Houston Houston, Texas 77004 ATTN: H. P. Decell, Jr.	(1)
NASA Lyndon B. Johnson Space Center Public Affairs Office, Code AP Houston, Texas 77058	(1)
National Aeronautics and Space Administration Scientific and Technical Information Facility Code KS Washington, D. C. 20546	(1)
Department of Watershed Sciences Colorado State University Fort Collins, Colorado 80521 ATTN: Dr. James A. Smith	(1)
NASA/Johnson Space Center Earth Resources Program Office Office of the Program Manager Houston, Texas 77058	(1)
NASA/Johnson Space Center Earth Resources Program Office Program Analysis & Planning Office Houston, Texas 77058 ATTN: Dr. O. Glenn Smith/HD	(1)
NASA/Johnson Space Center Earth Resources Program Office Systems Analysis and Integration Office Houston, Texas 77058 ATTN: Mr. Richard A. Moke/HC ATTN: Mr. M. Jay Harnage, Jr./HC	(1) (1)
Earth Resources Laboratory, GS Mississippi Test Facility Bay St. Louis, Mississippi 39520 ATTN: Mr. D. W. Mooneyhan	(1)

NAME	NUMBER OF COPIES
Lewis Research Center National Aeronautics & Space Administration 21000 Brookpark Road Cleveland, Ohio 44135	
ATTN: Dr. Herman Mark	(1)
John F. Kennedy Space Center National Aeronautics & Space Administration Kennedy Space Center, Florida 32899	
ATTN: Mr. J. P. Claybourne/AA-STA	(1)
Texas A&M University Institute of Statistics College Station, Texas 77843	
ATTN: Dr. H. O. Hartley	(1)
Code 168-427 Jet Propulsion Laboratory 4800 Oak Grove Drive Pasadena, California 91103	
ATTN: Mr. Fred Billingsley	(1)
NASA Headquarters Washington, D. C. 20546	
ATTN: Mr. W. Stoney/ER	(1)
ATTN: Mr. Leonard Jaffee/ER	(1)
ATTN: Mr. M. Molloy/ERR	(1)
ATTN: Mr. James R. Morrison/ERR	(1)
ATTN: Ms. Ruth Whitman/ERR	(1)
Texas A&M University Remote Sensing Center College Station, Texas 77843	
ATTN: Mr. J. C. Harlan	(1)
USGS National Center Mail Stop 115 Geography Program Reston, Virginia 22092	
ATTN: James Wray	(1)
Canada Centre For Remote Sensing 2464 Sheffield Road Ottawa, Canada K1A 0Y7	
ATTN: Dr. David Goodenough	(1)