E77-10060
NASA CR-
ERIM 109600-69-F

NASA CR-
151000

Final Report

EVALUATION OF ALGORITHMS FOR ESTIMATING WHEAT ACREAGE FROM MULTISPECTRAL SCANNER DATA

WYMAN RICHARDSON AND ALEX P. PENTLAND
Infrared and Optics Division

MAY 1976

(E77-10060) EVALUATION OF ALGORITHMS FOR ESTIMATING WHEAT ACREAGE FROM MULTISPECTRAL SCANNER DATA Final Report, 15 May 1975 - 14 May 1976 (Environmental Research Inst. of Michigan) 100 p HC A05/MF A01 CSCL 02C G3/43

N77-14559
Unclas
00060

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Johnson Space Center
Earth Observations Division
Houston, Texas 77058
Contract No. NAS9-14123, Task 14
Technical Monitor: Dr. A. Potter/TF3

ENVIRONMENTAL
RESEARCH INSTITUTE OF MICHIGAN
FORMERLY WILLOW RUN LABORATORIES, THE UNIVERSITY OF MICHIGAN
BOX 618 • ANN ARBOR • MICHIGAN 48107

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. NASA CR-ERIM 109600-69-F	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

Evaluation of Algorithms for Estimating Wheat Acreage from Multispectral Scanner Data

5. Report Date
May 1976

6. Performing Organization Code

7. Author(s)
Wyman Richardson and Alex P. Pentland

8. Performing Organization Report No.
109600-69-F

9. Performing Organization Name and Address

Environmental Research Institute of Michigan
Infrared & Optics Division
P. O. Box 618
Ann Arbor, Michigan 48107

10. Work Unit No.
Task 14

11. Contract or Grant No.
NAS9-14123

12. Sponsoring Agency Name and Address

National Aeronautics & Space Administration
Johnson Space Center
Houston, Texas 77058

13. Type of Report and Period Covered

Final Technical Report
May 15, 1975 through
May 14, 1976

14. Sponsoring Agency Code

15. Supplementary Notes

The work was performed for the Earth Observations Division.
Dr. Andrew Potter (TF3) was the technical monitor.

16. Abstract

Fourteen different classification agorithms were tested for their ability to estimate the proportion of wheat in an area. For some agorithms, accuracy of classification in field centers was also observed. The data base consisted of ground truth and spring, 1974, Landsat data from 55 sections (1 x 1 mile) from 5 LACIE Intensive Test Sites from Kansas and Texas. The performance measure was the difference between the algorithm's estimate of the proportion of wheat and the proportion determined by the ground truth. Signatures obtained from training fields selected at random from the ground truth were generally representative of the data distribution patterns.

The usual maximum likelihood classifier QRULE, an algorithm substantially equivalent to the LACIE recognition procedure for wheat, served as a standard of comparison for evaluating the performance of the other algorithms. QRULE recognized wheat in the 55 sections with an average absolute error of 6.9% and a wheat bias of 3.6%, an accuracy that did not leave much room for improvement.

LIMMIX, an algorithm that chooses a pure signature when the data point is close enough to a signature mean and otherwise chooses the best mixture of a pair of signatures, reduced the average absolute error to 6.1% and the bias to 1.0%. QRULE run with a null test achieved a similar reduction. Most of the algorithms tested did not improve upon QRULE's estimate of wheat area. A method of estimating area by summing posterior probabilities over all the pixels produced almost the same estimates as the usual pixel-count method.

17. Key Words		18. Distribution Statement
Remote Sensing	Landsat	Initial distribution is listed at the end of this document.
Pattern Recognition	LACIE	
Multispectral Classification		
Classification Algorithm		
Agricultural Surveys		

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	vii + 104	

PREFACE

This report describes part of a comprehensive and continuing program of research concerned with advancing the state-of-the-art in remote sensing of the environment from aircraft and satellites. The research is being carried out for NASA's Lyndon B. Johnson Space Center, Houston, Texas, by the Environmental Research Institute of Michigan (ERIM). The basic objective of this multidisciplinary program is to develop remote sensing as a practical tool to provide the planner and decision-maker with extensive information quickly and economically.

Timely information obtained by remote sensing can be used to predict the production of such important food crops as wheat, and thus allow government to avoid either famine or market oversupply. Other applications of information obtained by remote sensing include forest management, detection and prevention of water pollution and urban land studies. An integral part of obtaining this type of information is the estimation of the proportion of target classes in a scene. Yet the techniques employed in proportion estimation remain limited in many ways. The purpose of this report is to test and evaluate several proportion estimation algorithms which have been developed to overcome the limitations of more conventional algorithms.

The research described here was performed under NASA Contract NAS9-14123, Task 14, and covers the period from 15 May 1975 through 14 May 1976. Dr. Andrew E. Potter has been Technical Monitor. The program was directed by R. R. Legault, Vice-President of ERIM, by J. D. Erickson, Project Director and Head of the Information Systems and Analysis Department, and by R. F. Nalepka, Principal Investigator and Head of the Multispectral Analysis Section. The ERIM number for this report is 109600-69-F.

The experiment that is the subject of this report was initially planned by Harold M. Horwitz with the help of Robert B. Crane and the authors. Richard J. Kauth made helpful technical suggestions and gave editorial assistance. John Lewis contributed to data preparation. The authors gratefully acknowledge the help of all these co-workers.

CONTENTS

FIGURES

TABLES

SUMMARY

Fourteen different classification algorithms were tested for their ability to estimate wheat proportions and correctly discriminate between winter wheat and other pixels. The data base consisted of ground truth and spring, 1974, Landsat data from 55 sections from 5 LACIE Intensive Test Sites in Kansas and the Texas panhandle. In every square mile section, each algorithm's estimate of the proportion of wheat was checked against the true proportion. For some algorithms, accuracy of classification in field centers was also observed.

The reference algorithm, against which all others were evaluated, was QRULE operated in the recognition mode, an algorithm substantially equivalent to the recognition procedure being used as a part of the LACIE (Large Area Crop Inventory Experiment). Wheat and non-wheat training fields were selected at random from the ground truth. Signatures obtained by clustering the points of the training fields appeared to represent well the data distribution patterns in the sites; hence, the tests were of the capabilities of the algorithms given good signatures rather than tests of the AI's ability to select representative fields and properly identify them.

Besides QRULE, the algorithms tested included:

1. LRULE a linear decision rule and ADMAP, an adaptive decision rule based on LRULE. Both rules classify single pixels.
2. several nine-point rules which use data from the 8 neighboring pixels to assist in the classification of the center pixel. They are:
   a. BAYES9, based on the assumption that a pixel probably represents the same material as its neighbor
   b. LIKE9, the nine-point maximum likelihood rule, which amounts to choosing the material with the smallest sum of the 9 multivariate normal exponents

c. PRIOR9, which makes a Bayesian decision on the center pixel based on prior probabilities estimated from neighborhood data values

d. PREF9, which chooses the material with the largest average posterior probability over the 9 pixels

e. VOTE9, which recognizes the material with the largest number of votes (i.e., QRULE decisions) among the 9 pixels

f. AVE9, which averages the data from the 9 pixels and then applies QRULE

3. several mixed pixel rules which estimate the fraction of each pixel belonging to each category. They are:

a. LIMMIX. When the data point is close enough to a signature mean, that pure signature is chosen. Otherwise, the best mixture of a pair of signatures is chosen.

b. LIMMIX B. This is similar to LIMMIX, except that a density is defined for each two-way mixture and a choice is made between pure and mixed densities by maximum likelihood.

c. LIMMIX C. This is the same principle as LIMMIX B except that the two-way mixture density is defined differently.

d. Nine-Point-Mixtures. First, a vote of the 9 pixels is taken as in VOTE9. If either wheat or other gets 8 votes or more, the vote makes the decision. Otherwise the LIMMIX procedure is applied to the center pixel.

4. a cluster mapping decision algorithm. The data of the site are clustered. The clusters are identified as wheat or other, first by the training pixels in the cluster if possible, then by spatial and spectral closeness to identified clusters. The wheat acreage is computed from the total number of pixels in the clusters identified as wheat. Human-aided cluster mapping

and an automatic clustering procedure that relies on spatial closeness to identify unknown clusters were tested.

5. Modifications of QRULE and PRIOR9 to estimate wheat acreage by summing, over all pixels, the posterior probability of wheat. The estimate can be iterated by letting the prior probabilities of a repeated run be the proportion estimates of the previous run.

The algorithms were run without a null test. (A null test is an option to classify a pixel as none of the candidate signatures, and therefore not count it as wheat, when it is further than a given distance from the winning signature.) In addition, QRULE, PRIOR9, LIMMIX and Nine-Point Mixtures were run with a null test and the results compared.

The principal results of the tests are as follows: the good training data enabled QRULE to recognize wheat in the 55 sections with an average absolute error of only 6.9% and a bias in favor of wheat of 3.6%, an accuracy that did not leave much room for improvement. LIMMIX achieved the best no-null-test result, reducing the average absolute error to 6.1% and the bias to 1.0%. Almost identical results were scored by QRULE and PRIOR9 using a null test that decided "none of these" when the chi-square value for the winning signature exceeded 45 (a value considerably higher than the .001 chi-square level of 18.5). A null test made hardly any improvement in the LIMMIX results.

The other mixture algorithms registered smaller improvements over QRULE and had low biases in the 1.4%-1.8% range. None of the remaining algorithms improved on the QRULE absolute error and all but the automatic clustering procedure (whose bias was 1.3) had a bias comparable to QRULE's. Five of the algorithms, LIKE9, AVE9, automatic cluster mapping, ADMAP, and LRULE had noticeably higher average absolute errors of 8.0% or more.

Cluster mapping aided by human judgement did not receive a complete test but the partial results were quite encouraging. Automatic cluster mapping did not fare so well in its initial trial, quite possible because the algorithm did not include the principle of spectral closeness.

The posterior probability method of acreage estimation, with or without iteration, had results very similar to those of the pixel-count method.

Classification accuracy on within-field pixels was measured for QRULE and the nine-point rules. This test showed that deep within the fields, all nine-point rules outperformed QRULE substantially. On near-boundary pixels within the fields, the margin was narrower and LIKE9 and AVE9 were worse. The large proportion of pixels in LANDSAT data either on or adjacent to a boundary shows why most of the nine-point rules performed poorly in the area tests and suggests that their most useful application is to higher resolution data or to areas with larger fields.

When bias was averaged for sections grouped according to the true proportion of wheat, only LIMMIX and QRULE with a null test maintained consistently low levels of bias.

The experimental design had two sound features. The comparison of estimated with true wheat area is a performance measure that realistically refers to the objective of wheat inventory. The use of a section (1x1 mile square) as an experimental unit supplies the replications necessary to draw conclusions, even though allowance has to be made for the dependence of sections within a site. As for the execution of the experiment, the strongest evidence of its correctness is that we can understand and explain most of the results. Taken together, the experimental plan and procedure appear to be able to distinguish between a good and a bad wheat recognition algorithm. They could, therefore, be useful in evaluating new and modified algorithms of current interest.

Because the algorithm LIMMIX that performed best is slow, its use on the type of data in our test is of questionable practicality. In a region of small fields where the performance of QRULE would be expected to break down, LIMMIX could become the algorithm of choice.

2
INTRODUCTION

ERIM has developed, over a period of years under varied sponsorship, numerous algorithms for processing multispectral data to extract earth resource information. The impetus for development has been greatly increased by ERIM's participation in the Large Area Crop Inventory Experiment (LACIE), an experiment to test a prototype application system for the estimation of worldwide wheat acreage, yield, and production. Something in the neighborhood of 20 of the developed algorithms are potentially suitable for wheat inventory. Fourteen of these were tested in the present study and the results are included in this report. Others can be tested in the near future if this is found desirable. Descriptions of the algorithms tested are given in Section 3.

The algorithms classify pure pixels or estimate the fractions of classes included in mixed pixels. They depend for their effectiveness on being furnished with signatures representing the data distributions of the materials present. In our experiment, the signatures were obtained from training fields selected at random from the ground truth, a procedure roughly comparable to that used in the local mode of LACIE, in which training fields are chosen from the test site and identified with credible accuracy by an Analyst Interpreter (AI).

The overall test structure is as follows. The primary performance measure employed during the study is the ability of each algorithm to estimate the proportion of wheat in each experimental unit. This measure and secondary measures are described in Section 5. To evaluate the candidate algorithms, the performance of each is compared with the performance of the usual quadratic classifier QRULE operated in a mode to discriminate between wheat and non-wheat, a rule substantially equivalent to the LACIE classifier.

The elemental experimental unit is a one square mile section. A factor in the tests is the site in which the section is located. There

are 5 of these sites, each having a varied number of usable sections, totalling 55. Because the sections of a site share a common set of signatures and tend to share a data distribution pattern, the experimental units are somewhat dependent. The sites and sections are further discussed in Section 4.

To prepare the data for analysis required a substantial effort. Certain key elements of that effort, such as the method of finding field vertices, are described in Section 4 and in several appendices.

Test results and a detailed discussion of them are contained in Section 5. Overall conclusions and recommendations are given in Section 6.

3
DESCRIPTION OF ALGORITHMS

The decision algorithms tested were of four types: one-point rules, nine-point rules, mixture rules and adaptive processing rules. The one-point rules were QRULE, the usual quadratic decision rule and LRULE, the minimum-risk linear decision rule [9]. The nine-point rules are briefly defined as follows [1]:

BAYES9 is based on the assumption that a pixel probably represents the same material as its neighbor, the degree of dependence specified by a parameter θ between 0 (independence) and 1 (complete dependence). In our tests we used θ values of 0.1, 0.3, and 0.5.

LIKE9, the nine-point maximum likelihood rule, amounts to adding, for each material, the 9 multivariate normal exponents and choosing the material with the smallest sum. It is equivalent to BAYES9 with $\theta = 1$.

PRIOR9 makes a Bayesian decision on the center pixel based on prior probabilities estimated from neighborhood data values. The estimated prior probability of a material is the average, over 9 pixels, of the posterior probability of that material at each pixel.

PREF9 uses as its decision criterion the estimated prior probability just defined for PRIOR9. It is conceptually an improved voting rule that takes account of all the information at each pixel rather than just a vote for the winning material.

VOTE9, applied after QRULE decisions have been made on the 9 pixels, assigns to the center pixel the material most frequently recognized among the 9 pixels.

AVE9 averages the 9 data points and then applies QRULE. To prevent occasional alien points from disturbing the decision rule, the t largest and t smallest data values in each channel are omitted from the average. In our tests, we used t = 1, so the average was taken over the 7 middle data values in each channel.

All the nine-point rules use QRULE at some point. VOTE9 takes a vote among 9 QRULE decisions. AVE9 computes a trimmed mean of 9 data points and then processes with QRULE. The other rules all use the QRULE-computed densities conditional upon each signature as the starting point of their calculations.

Initially, in our testing, we used QRULE in the classification mode. In this mode a decision is made among all the input signatures. If there were 6 wheat and 9 other signatures, for example, then 15 possible decisions could be made. The 6 categories of wheat decisions could then be collected to estimate the wheat acreage. It soon became apparent that the classification mode was not suitable for nine-point rules. VOTE9, for example, might find the vote split among the 6 wheat signatures and still not decide the center pixel is wheat. The other nine-point rules have similar difficulties.

We therefore wrote a version of QRULE that operates in the recognition mode. In this mode, a composite wheat density is obtained by averaging the wheat densities and similarly, a composite non-wheat ("other") density. A maximum likelihood decision is then made between the two composite densities.* The nine-point rules operate on these composite densities without modification. Our test results for QRULE and the nine-point rules were obtained in the recognition mode. Comparison

---

* This rule is substantially the rule used by the Classification and Mensuration Subsystem in LACIE [2].

with classification mode results was made for the Ellis site only. These results confirm the benefit of using the recognition mode for nine-point rules (Table 5), but no preference between the recognition and classification modes of QRULE was indicated. In a future test, the recognition and classification modes will be compared for all sites.

The mixture algorithms tested were LIMMIX, LIMMIX B, LIMMIX C, and Nine-Point Mixtures. They are all modifications of the basic mixture algorithm MIXMAP, which we now describe.

A mixture processing rule does not assume that the signal from the pixel processed represents a single material but rather is a positively-weighted sum of signals from materials represented in the pixel, each weight λ_i being the proportion of material i present in the pixel. A mixture rule estimates these proportions. MIXMAP depends on the simplifying assumption (without which the problem would be intractable) that all signatures have the same covariance matrix. This common covariance matrix is reduced to the identity matrix by a linear transformation of the data point and the signatures. The density in the transformed space can now be measured by the distance from the transformed data point to its transformed mean. All possible proportions of the materials can be represented by the points within the convex hull of the transformed means. The estimation procedure is to find the point in this convex hull nearest to the transformed data point and calculate the estimate of the proportion vector from it. The estimate is a maximum likelihood estimate of $\{\lambda_i\}$ under the assumptions stated and the assumption of normality. References [3] and [4] describe the algorithm in greater detail.

LIMMIX exploits the reasonable assumption that no more than L materials are present simultaneously in a single pixel [5]. For illustration, we suppose that $L = 2$. We choose two threshold values χ_1^2 and χ_2^2. We first make the usual decision among pure materials, taking note of

the chi-square distance χ_p^2 between the data point and the winning mean. If $\chi_p^2 \leq \chi_1^2$, we accept the pure decision without further calculation. Otherwise, we use the MIXMAP procedure to find the best pairwise mixture of materials, computing as a by-product χ_m^2, the distance from the transformed data point to the best two-way mixture. If $\chi_p^2 = \chi_m^2$, it means that the best two-way mixture has turned out to be a pure decision which is accepted if $\chi_p^2 \leq$ a high cutoff level χ_2^2. If $\chi_p^2 > \chi_m^2$, the best mixture is really a mixture and is accepted if $\chi_m^2 < \chi_2^2$. But should the χ_2^2 test fail, the data point is declared an unknown object. LIMMIX has the advantage that the total number of materials is not limited, whereas MIXMAP is subject to the geometrical constraint that the number of materials cannot exceed the number of channels plus one.

LIMMIX B and LIMMIX C are like LIMMIX in that they are decision rules for choosing among pure signatures and mixtures, but they are based on the principle of defining a density for each two-way mixture and then choosing among the pure and mixed densities by weighted maximum likelihood. A detailed description of these algorithms is given in Appendix V.

Nine-Point Mixtures extends the LIMMIX concept by taking advantage of information contained in the adjoining 8 pixels (in the manner of VOTE9, described above) to determine both how many materials and which materials are in a pixel. This procedure is implemented as follows:

A. Make a preliminary pass through the data, classifying each pixel according to the usual quadratic decision rule QRULE.

B. For each pixel, look at it and the adjoining 8 pixels, and count the "votes" (QRULE decisions) on their identity. Pixels may participate in the vote only if their associated chi-square level is less than η_1^2. If at least N_1 of the pixels agree

on identity, the center pixel is classified as this material. In this decision, all the wheat votes are added together and so are all the other votes.

C. If the two materials with the largest number of votes each have N_2 or more votes, then the pixel is assumed to be a mixture of these two materials and the proportion is estimated by the proportion of votes.

D. If tests B and C fail, the LIMMIX procedure is applied to the center pixel. If its chi-square level is less than or equal to η_2^2, accept the QRULE decision. If the chi-square level is greater than η_2^2, find the best two-way mixture and accept it if its chi-square level is less than η_3^2. Otherwise, declare the point alien.

The accuracy with which LIMMIX estimates wheat area depends on the choice of the processing parameters χ_1^2 and χ_2^2 just as the accuracy of Nine-Point Mixtures is dependent on the choice of N_1, N_2, η_1^2, η_2^2, and η_3^2. Objective methods of training these parameters by making one pass through the data are described in Appendix VI. The principal technique is to make a prior estimate m of the proportion of mixture pixels in the scene, a relatively stable figure, and then adjust the parameters so that the proportion of mixture decisions agrees with m.*

Cluster mapping uses a clustering algorithm to classify the data rather than merely to provide signatures for some other classifier. The classification is accomplished as follows:

1. Cluster the entire region to be classified, marking each pixel with the number of the cluster to which it belongs.

---

* This procedure for training parameters replaces a non-objective procedure defined and used in the lastest contract quarterly progress report.

2. Map the cluster numbers of the pixels.
3. Identify as many clusters as possible by observing which clusters appear in the training fields.
4. Continue to identify clusters by observing which ones are spatially in the midst of clusters already assigned to wheat or other.
5. Give each remaining cluster the identity of the identified clusters that are nearest spectrally. The determination of spectrally nearest clusters is a calculation on the cluster signatures.
6. Estimate the wheat area from the total number of pixels in the clusters identified as wheat.

The cluster mapping procedure was originally developed to expand the ground truth furnished by the AI. It was carried out with the aid of a human interpreter in steps 3, 4, and 5 with results as shown in Table 1. The accuracy of the technique suggested that it be used as a classifier in its own right.

As a classifier, cluster mapping would enjoy three advantages over conventional classification techniques:

1. Cluster mapping is less sensitive to ground truth errors than are conventional techniques. This is because cluster mapping forms its own estimate of the spectral classes in the scene. The identity of these classes is then decided by majority rule, e.g., if cluster 10 occurs more often in wheat fields than other fields, then cluster 10 is called wheat. Thus, as long as a large majority of the ground truth pixels are correctly identified, no errors are made. Conventional techniques, on the other hand, can make large classification errors from small ground truth errors.
2. Cluster mapping requires less extensive ground truth because every cluster need not be represented in the ground truth.

TABLE 1. ACCURACY OF THE CLUSTER-MAPPING PROCEDURE WHEN IT INCLUDES HUMAN JUDGMENT

Site	Estimated Wheat %	Actual Wheat %	Rms Error for Sections	Mean Absolute Error for Sections	Number of Sections
Ellis	50.4	45.8	4.6	4.6	4
Deaf Smith	33.1	33.3	6.0	5.3	4
Randall	45.5	47.2	3.3	3.1	5
Finney	27.2	20.1	2.7	2.1	9
Saline	74.1	70.5	4.7	3.8	4

COMPARISON WITH QRULE OVER 26 SECTIONS:

	Bias (Mean Algebraic Error)	Median Absolute Error	Mean Absolute Error	Rms Error
Cluster mapping	1.9	3.2	4.3	5.8
QRULE	2.4	4.3	4.9	5.9

Those that fail to appear in the training fields may very well be correctly identified by spectral or spatial closeness.

3. In the cluster mapping technique, classification of pixels is done before human intervention (such as providing ground truth areas). Cluster mapping is, therefore, uniquely suited to applications such as on-board satellite data processing where human interaction is both difficult and expensive.

The cluster mapping procedure would be efficient and repeatable if the human interpreter could be replaced by computer logic. This hope together with the advantages of the procedure suggest that cluster mapping is worthy of considerable developmental effort.

Our initial attempt to automate cluster mapping is a processing module called TRAIN that uses spatial information to identify unaffiliated clusters. The algorithm is described in detail as follows;

1. Examine the training areas. A cluster occurring in one or more training areas is called "wheat" if twice as many of the cluster's pixels appear in wheat fields as other. In order that this vote be representative, it must satisfy the condition that the cluster account for a least 2% of the training area. An analogous rule identifies the cluster as "other". If the cluster is not identified, it is called "unknown".
2. For each unknown cluster, look at each pixel and each of its four nearest neighbors. Keep a count of the number of wheat neighbors, the number of the other neighbors and the number of unknown neighbors. The exception to this rule is that if three or more of the four neighbors belong to the cluster in question, then no neighbors are counted for that pixel, so that when a pixel is on the edge of a field we will not try to identify it by its neighbors.
3. Look at each unknown cluster in turn and identify it as wheat if it passes the following two tests

$$\frac{\text{number of wheat neighbors}}{\text{number of other neighbors}} > \text{factor 1}$$

$$\frac{\text{number of wheat neighbors + number of other neighbors}}{\text{number of unknown neighbors}} > \text{factor 2}$$

   The tests for identifying it as other are analogous. Factor 1 and factor 2 are initially 1.9.
4. Every time a cluster is identified by step 3, the number of unknown and the number of wheat or other neighbors changes, so a cluster that failed the tests of step 3 previously may later pass them. Therefore, steps 2 and 3 are applied repeatedly until there is no change in cluster identification.

5. Reduce factor 1 by 0.3 and factor 2 by 0.5 and repeat steps 2-5. Stop the iteration when factor 2 becomes less than zero and call all the remaining unknown clusters "other".

To explain the algorithm, we have made it appear that it is necessary to go through the data many times. Actually, we keep a matrix of association frequencies and go through the data only once.

Although improvements to this rule spring to mind, such as the joint use of spectral and spatial measures of closeness, time limitations have restricted us to the implementation and testing of the rule just described.

The final procedure tested was our adaptive processing algorithm ADMAP. Adaptive processing updates the mean vectors of the crop signatures based on decisions made by a classifier and on the values of the individual data vectors which are classified. The approach is based on the following idea. Suppose a sequence of observations (data vectors), z_j, z_{j+1}, ... were all recognized as material class A by the classifier, but that these observations tended to cluster to one side of the current estimate of the mean, μ_A, of that material class. This would provide us with some evidence that the mean of the material class A had shifted. A decision-directed adaptive classifier is one which automatically adjusts the value of μ_A so as to bring it closer to the current observations which were classified as material A.

We would like our decision-directed adaptive classifier to take account of some additional considerations. The amount by which we allow a signature to be modified in any particular updating cycle may be different in different spectral channels. Also, a particular crop may not be observed for some time, and during that time the true mean of that crop, along with the means of other crops, may shift. Hence we would like to be able to adapt all signatures based upon the observations and classifications of one or a few of them.

In practice, resolution elements often overlap two or more different crop types, producing an observation far from the mean of any particular crop class. We would like to avoid using these observations as well as "wild" observations from any other cause.

The Kalman filter (an account of which is given as Appendix I) combines these considerations into one systematic approach. ADMAP carries out the Kalman filter with a few additional modifications. These include the ability to weight a pixel by a confidence factor based on the χ^2 value associated with that pixel's classification, in order to exclude 'wild' pixels and mixture pixels; the ability to make use of ground truth information where available in the scene; and the ability to update after each scan line or portion of a scan line, (rather than after each point), to increase efficiency.

ADMAP has a parameter θ_1 that determines how much weight to give the new data value in updating the mean. It thus determines how rapidly ADMAP adjusts the means. In our tests, we used values of $\theta_1 = 10^{-5}$, 10^{-6} and 10^{-7}, to produce faster, medium or slower adjustment, respectively.

All of the rules were tested without a null test. (A null test is an additional stipulation that if the pixel is not within a given distance of the winning signature it is classified as none of the candidate signatures and is therefore counted as not wheat.) QRULE, PRIOR9, LIMMIX and Nine-Point Mixtures were run with and without a null test and the results compared. The null test for LIMMIX is the χ_2^2 test. To turn off the null test, χ_2^2 is set to a very large number. The η_3^2 test plays an analogous role in the Nine-Point Mixtures algorithm. The null test for QRULE and PRIOR9 is to decide null if the chi-square value of the winning signature is greater than a given test value.

Modifications of QRULE and PRIOR9 to estimate wheat acreage by summing posterior probabilities were programmed and tested. The procedure is described as follows.

The maximum likelihood decision rule for recognizing wheat is to compute for each pixel two density functions $P(X|W)$ and $P(X|O)$ of the pixel data vector X. $P(X|W)$ is the density, also called "likelihood", of X given that the true distribution is wheat and $P(X|O)$, the density of X given that the distribution is "other". $P(X|W)$ may be a composite wheat density, that is, a weighted sum of normal densities, each representing a different variety or condition of wheat, and $P(X|O)$ is likely to be similarly formed. The pixel is decided to be wheat if $P(X|W)$ is greater than $P(X|O)$.

By the use of Bayes' formula, we can turn the densities around and compute $P(W|X)$, the probability that the pixel is wheat and $P(O|X)$, the probability that it is other:

$$P(W|X) = \frac{P(W)P(X|W)}{P(W)P(X|W) + P(O)P(X|O)}$$

$$P(O|X) = \frac{P(O)P(X|O)}{P(W)P(X|W) + P(O)P(X|O)}$$

where $P(W)$ is the prior probability of wheat and $P(O)$ is the prior probability of other. $P(W)$ and $P(O)$ are defined to add to 1.

The "posterior probabilities" $P(W|X)$ and $P(O|X)$ add to 1 as probabilities should. A justification for the maximum likelihood estimate is that it is equivalent to choosing the material with the largest posterior probability. The rule is most commonly applied with equal prior probabilities, but likelihoods are sometimes weighted by unequal priors.

As an alternative to the usual method of wheat acreage estimation, which classifies each pixel as all wheat or all other and then counts the number of wheat pixels in the area, M. Rassbach has proposed [8] that we allot to wheat the expected amount of wheat in each pixel, which is $P(W|X)$, and then sum these individual expected values to

obtain the expected amount of wheat in the area. The estimated proportion of wheat is this value divided by the number of pixels.

PRIOR9 is a weighted maximum likelihood rule like QRULE, except that the weights (prior probabilities) are derived from neighborhood data values rather than set once and for all at the start of the run. Thus, $P(W|X)$ is as previously described, except that it has the PRIOR9 weights.

QRULE in the posterior probability mode was programmed to run with an option of iteration. The user sets the prior probabilities P(W) and P(O) (equal priors is the default case) and then the program iterates a prescribed number of times, the priors for each iteration being the proportions estimated by the previous iteration*. Although this violates the concept of prior probability, we were tempted by the thought that if the wheat proportion came out 10%, for example, then 10% and 90% would be better probabilities to use in the decision rule than 50% and 50%. We wanted to see, at least, what the result of this iteration would be. The iteration concept does not apply to PRIOR9.

*The iteration concept is a special case of the University of Houston Maximum Likelihood Estimate procedure [12] and was independently proposed by H. M. Horwitz of ERIM in February, 1975.

4
DESCRIPTION OF THE TEST SET

The data base used for the tests described here consists of ground truth and unitemporal data from 55 sections from 5 LACIE Intensive Test Sites. The data base preparation included:

1. checking the data from bad lines and removing the effects of striping (see Appendix III)
2. locating digitally the vertices of the field boundaries
3. selecting training fields by a random procedure
4. clustering the points of the training fields and combining the clusters into a manageable number of signatures
5. computing the ground truth percentage of wheat acreage in each section of each site (see Appendix II).

The sites and the sections within the sites were chosen to provide a variety of conditions for comparing the performance of data-processing algorithms but to eliminate gross sources of error that would render such comparison meaningless. The presence of any of the following sources of error was considered serious enough to justify deleting a section from the test set:

1. Misleading Ground Truth - In several cases there are fields which are described as wheat in the ground truth but which are known to have been severely damaged or destroyed by natural causes such as hail, drought, or insects previous to the data collection.
2. Data Errors, such as bad or repeated lines - These phenomena could seriously affect the results of this test, because of the small size of the experimental unit, but can be adequately compensated for over large areas.
3. Clouds - It is difficult to define the boundaries of a cloud with the precision necessary for this experiment.

As a result of this selection procedure, the following sites were included in the test set:

Ellis	Kansas	12 June 74	9 Sections
Deaf Smith	Texas	27 May 74	6 Sections
Randall	Texas	27 May 74	7 Sections
Finney	Kansas	26 May 74	24 Sections
Saline	Kansas	6 May 74	9 Sections

To use the field information for defining signatures and measuring performance it was necessary to obtain accurate line and point coordinates of each field corner. For simplicity and accuracy, we used a digitizer on a photographic image of the site to obtain field vertices in one set of coordinates and then transformed them into line and point coordinates which were kept as continuous measurements rather than integers. The transformation was obtained by a second order regression on field corners identifiable on both the photographic image and the line printer maps of the site.

AI designations of training fields were not available, so a random selection procedure was used. Wheat training fields were chosen from among all the wheat fields in the test site containing at least 5 field-center pixels. (A field-center pixel is one whose center is at least 1.5 pixel-widths from the boundary of the field.) The fields were chosen at random, one at a time, until there were at least 5 fields containing together at least 200 field-center pixels. These requirements were subordinate to the restriction that the number of wheat training fields should not exceed half the number of eligible wheat fields and that the number of wheat training pixels should not exceed half the number of field-center wheat pixels. The non-wheat ("other") training fields were chosen at random among all the eligible "other" fields until at least 10 fields and 300 pixels were chosen subject to the previous restriction. In all, 6.7% of the pixels in the test areas were chosen as training pixels.

Wheat signatures were obtained by clustering all the field-center pixels of the wheat training fields and then using the program GROUP (described in Appendix IV) to combine the clusters into the smallest number of signatures possible without adversely affecting classification accuracy. "Other" signatures were obtained analogously.

5
TEST RESULTS AND DISCUSSION

Our principal performance measure for evaluating the algorithms defined in Section 3 is the difference between estimated and true wheat area in each of the 55 sections. In addition, we compare the performance of QRULE and the nine-point rules on field interiors by counting the number of within-field pixels misclassified. These two sets of results are reported in Sections 5.1 and 5.2. Detailed discussion follows in Section 5.3.

In an attempt to discern general tendencies in the results, we have averaged the results over 55 sections and used Student's t test to measure their significance. But the assumption of independent samples, on which the t test is based, fails because in each of the five sites the sections have a common selection of training sets and tend to share a data distribution pattern. This dependence increases the standard deviation of the mean, effectively cutting down the number of degrees of freedom, so that significance cannot be proved by such a t test. But a result not significant at 54 degrees of freedom will be even less significant when the dependence is taken into account. Thus, the reported significance of the t test is a bound on the possible significance of the result.

5.1 FIELD INTERIOR RESULTS

The within-field pixels were identified by locating the vertices of each field in floating point coordinates and using a subroutine (POLYGN) that accepts for processing only those pixels whose centers are more than a specified minimum distance ("inset") within the polygonal boundary of the field.

Two collections of within-field pixels were used in the tests, one with an inset of 1.5 and one with an inset of 0.5. We feel confident that the within-field pixels with a 1.5 inset are really inside the

intended fields, but do not claim the perfection of field location that would guarantee that every pixel with a 0.5 inset be totally within the intended field. There are 19,880 0.5-inset pixels and 5394 1.5-inset pixels, showing that a large proportion of the 0.5 group are adjacent to a field boundary.

The performance of nine-point rules in field interiors is of particular interest because they are designed to take advantage of neighborhood homogeneity. The comparison of their performance to that of QRULE on interior pixels, on near-boundary pixels, and on all pixels shows whether they are performing as intended, and if not, where the problems might be. The superiority of rules designed to adapt better to boundaries is tested.

Tables 2 and 3 show the result of testing QRULE and the nine-point rules in the recognition mode on within-field pixels with a 0.5 inset and a 1.5 inset, respectively. By subtracting the 1.5-inset misclassifications from the 0.5-inset misclassifications, a misclassification rate for pixels adjacent to a field boundary is obtained. This rate for the various rules and sites is given in Table 4. Thus, Tables 3 and 4 give rates for two separate classes of pixels: the interior pixels and the adjacent-to-boundary pixels, respectively. Table 5 compares the performance of the classification and recognition modes of QRULE and the nine-point rules for the Ellis site.

5.2 RESULTS OF WHEAT AREA ESTIMATION

All the decision algorithms tested in this report are compared as wheat area estimators over the 55 sections. The estimate is obtained for each section by dividing the number of pixels recognized as wheat by the total number of pixels in the section. The measure of performance is the difference between the estimate and the true proportion of wheat (measured by adding the areas of the ground truth wheat fields and dividing by the area of the section).

A summary of these results is given as Table 6*. The differences which we will also call "errors", are expressed as percent, i.e., the differences of the two proportions times 100. The first column is the bias, namely the average of the signed differences (errors). If the positive differences just cancel out the negative differences, the bias would be zero. The second column is a bound on the statistical significance of the bias found by calculating

$$t = \frac{\text{bias } \sqrt{55}}{\text{standard deviation of the differences}}$$

and looking up t in a table of the t distribution at 54 degrees of freedom. The smaller the number in Column 2, the more significant is the bias. The number would be the probability of getting a bias this large by chance alone if all 55 sections were independent samples. Because of dependence among the sections, the real probability is a larger number.

Columns 3, 4, and 5 are three measures of the average absolute error: the median, the mean, and the root mean square (rms), respectively. The pattern of errors (shown in Figure 1) is that most are quite small -- 8% or less -- but there are a few quite large ones where the algorithms really missed. The median is a figure not affected by changes in the large errors. It thus indicates how the rules are doing on sections with small errors. The rms error gives most of its weight

---

* The results of testing LIMMIX and Nine-Point Mixtures given in the latest contract quarterly progress report should be disregarded because of errors in the implementation of the decision algorithms.

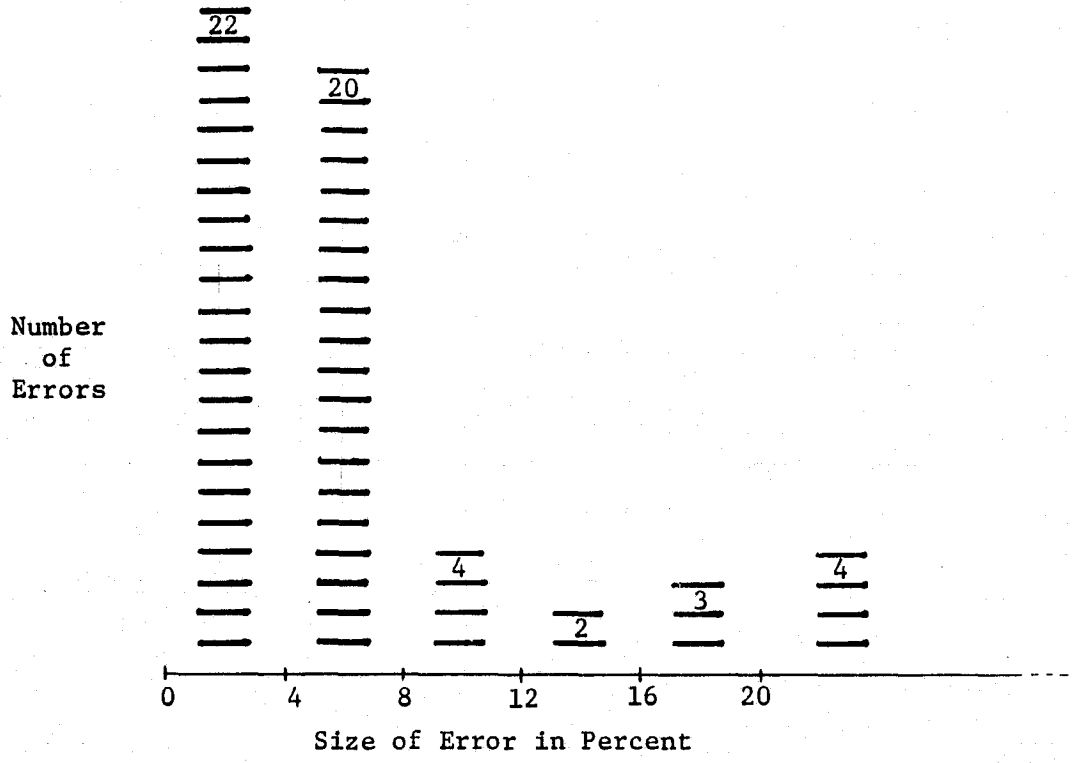

FIGURE 1. DISTRIBUTION OF ERRORS MADE BY THE QRULE ALGORITHM OVER 55 SECTIONS

to the large errors. An error of 30%, for example, gets 100 times as much weight in the rms error as an error of 3%**. The mean absolute error goes to neither extreme, giving significant weight to both the large and small errors. In describing the performance of decision algorithms, we talk mostly about the bias and the mean absolute error.

The next column, the mean improvement over QRULE, is calculated as follows. For each section, the difference between the absolute error

** The rms error is like a standard deviation (i.e., square root of the variance) of the errors except that the deviations are from zero rather than the mean. The standard deviations about the mean are so similar to the rms errors that they are not included in the table.

of QRULE and that of the algorithm in question is recorded as the algorithm's "improvement" for that section. A positive difference means that the algorithm has a smaller absolute error than QRULE, indicating superiority over QRULE, while a negative difference indicates inferiority. Column 6 is the mean of these improvements. It can also be calculated by subtracting the mean absolute error of the algorithm from that of QRULE. Column 8 is the standard deviation of the improvements. A small figure in column 8 together with a small average improvement (column 6) would show that the algorithm in question produces about the same wheat estimates as QRULE; a large figure in column 8 would show that it behaves differently. Column 7 is a bound on the statistical significance of the mean improvement, a figure obtained from a t value as in Column 2.

Table 7 gives the result of testing LIMMIX B and LIMMIX C with a variety of parameter settings. The table contains the same measures as Table 6. Two parameters are varied:

1. m is the prior probability that a pixel represents a mixture (see Appendix V). In setting the weights for a Bayesian decision among the pure and mixed signatures, the pure signatures divide up equally the prior weight 1 - m of a pure pixel and the two-way mixture signatures divide up equally the prior weight m of a mixed pixel. A larger m results in a greater emphasis on mixed signatures; a smaller m, on pure signatures.

2. The Bayesian decision between the best mixed density and the best pure density is carried out by choosing the lesser of the two quantities

$$\chi_p^2 + \text{constant}_p \quad \text{and} \quad \chi_m^2 + \text{constant}_m$$

where χ_p^2 and χ_m^2 are the chi-square values of the best pure and the best mixture densities, respectively. Changing the value m has the effect of changing constant_m in this comparison.

Another way of tipping the balance for or against mixtures is by multiplying χ_m^2 by the constant γ (see Appendix VI). A $\gamma > 1$ de-emphasizes mixtures, a $\gamma < 1$ emphasizes them. γ has little effect on data points close to the mixture line segment between the two pure means where χ_m^2 is very small, but plays an increasingly important role as the data point departs from the line segment. The purpose of defining γ was to control the behavior of points that were not well represented by either pure or mixture signatures. A $\gamma > 1$ would tend to steer such points to a pure signature rather than to some possibly inappropriate mixture.

Table 8 gives the bias for all the rules, first in all the sections (repeating Table 6) and then in three groups of sections having different ranges of the true proportion of wheat:

1. a "low wheat" group of 17 sections with less than 30% wheat
2. a "middle wheat" group of 26 sections with 30%-50% wheat
3. a "high wheat" group of 12 sections with more than 50% wheat.

The purpose is to see whether some rules have a bias depending on the true proportion of wheat. Table 11 gives the bias for every rule in each site.

In Tables 6 and 8, the results for LIMMIX B and LIMMIX C with $m = 0.4$ and $\gamma = 1$ are reported. $m = 0.4$ was chosen because it was estimated that in a typical Kansas site, about 40% of the pixels represent mixtures [6].

Table 9 gives the result of running QRULE, PRIOR9, LIMMIX and Nine-Point Mixtures with and without a null test. QRULE was run with null tests of 45, 35, and 25. By this we mean that when the chi-square value of the winning signature is greater than 45, say, the pixel is decided to be none of the given signatures, implying that it is not wheat. PRIOR9 was run with an unintentional null test of 45, due to an error in the code, and later compared to a corrected version with the

null test turned off. LIMMIX and Nine-Point Mixtures were run with null tests determined for each site by the 98th percentile in a histogram of the chi-square value. It was never set lower than the .001 chi-square value of 18.5 nor higher than 51.0. The settings were Ellis, 24.8; Deaf Smith, 51.0; Randall, 18.5; Finney, 51.0; and Saline, 18.5. Results for the LIMMIX B and LIMMIX C algorithms, which contain no null test, are included for comparison.

Test results for the posterior probability method of estimating acreage are reported in Table 10. QRULE with 0, 1 and 2 iterations and PRIOR9 were the algorithms to which the method was applied. Pixel-count results for these algorithms are included for comparison.

TABLE 2. PERFORMANCE OF QRULE AND NINE-POINT RULES IN THE RECOGNITION MODE ON WITHIN-FIELD PIXELS WITH AN INSET OF 0.5 OR MORE. THE PERFORMANCE MEASURE IS THE PERCENT OF PIXELS MISCLASSIFIED

RULE	TOTAL	ELLIS	DEAF SMITH	RANDALL	FINNEY	SALINE
QRULE	8.09	3.86	19.23	0.96	8.07	12.60
BAYES9(.1)	7.68	3.41	18.47	0.93	7.70	11.89
BAYES9(.3)	7.37	3.13	17.60	0.96	7.63	10.77
BAYES9(.5)	7.26	3.09	17.05	0.77	7.55	10.77
LIKE9	7.92	4.01	19.89	0.54	7.32	14.05
PRIOR9	7.39	3.37	17.96	.95	7.24	11.85
PREF9	6.35	4.01	13.14	0.61	6.18	11.18
VOTE9	7.02	3.65	18.17	0.80	6.52	11.40
AVE9	8.70	4.32	18.46	0.80	8.34	16.80

TABLE 3. PERFORMANCE OF QRULE AND THE NINE-POINT RULES IN THE RECOGNITION MODE ON INTERIOR PIXELS (WITHIN-FIELD PIXELS WITH AN INSET OF 1.5 OR MORE). THE PERFORMANCE MEASURE IS THE PERCENT OF PIXELS MISCLASSIFIED

RULE	TOTAL	ELLIS	DEAF SMITH	RANDALL	FINNEY	SALINE
QRULE	6.14	3.68	15.47	0.91	7.22	7.05
BAYES9(.1)	5.78	2.91	14.38	0.91	6.94	6.70
BAYES9(.3)	5.51	1.84	13.94	0.91	7.03	5.29
BAYES9(.5)	5.38	1.69	14.16	0.76	6.87	5.11
LIKE9	3.93	1.23	13.95	0.38	4.20	4.94
PRIOR9	4.92	2.91	13.29	.94	5.36	6.68
PREF9	2.61	0.61	8.07	0.46	2.63	4.76
VOTE9	4.15	0.76	14.82	0.53	4.59	4.76
AVE9	4.71	1.38	15.25	0.38	4.79	8.46

TABLE 4. PERFORMANCE OF QRULE AND THE NINE-POINT RULES IN THE RECOGNITION MODE ON NEAR-BOUNDARY PIXELS (WITHIN-FIELD PIXELS WITH AN INSET OF 0.5 TO 1.5). THE PERFORMANCE MEASURE IS THE PERCENT OF PIXELS MISCLASSIFIED

RULE	TOTAL	ELLIS	DEAF SMITH	RANDALL	FINNEY	SALINE
QRULE	8.82	3.92	20.37	1.03	8.37	14.31
BAYES9(.1)	8.38	3.56	19.71	0.97	7.97	13.49
BAYES9(.3)	8.06	3.51	18.72	0.86	7.84	12.46
BAYES9(.5)	7.96	3.51	17.92	0.80	7.80	12.51
LIKE9	9.40	4.84	21.69	0.69	8.42	16.87
PRIOR9	8.31	3.51	19.38	.97	7.90	13.44
PREF9	7.75	5.02	14.68	0.74	7.44	13.17
VOTE9	8.09	4.52	19.18	1.03	7.20	13.44
AVE9	10.18	5.20	19.44	1.14	9.61	19.37

TABLE 5. COMPARISON OF THE PERFORMANCE OF THE CLASSIFICATION AND RECOGNITION MODES OF QRULE AND THE NINE-POINT RULES FOR THE ELLIS SITE. THE PERFORMANCE MEASURE IS THE PERCENT MISCLASSIFIED

	INSET ≥ 1.5		INSET ≥ .5	
RULE	CLASSIFY	RECOGNIZE	CLASSIFY	RECOGNIZE
QRULE	3.52	3.68	3.83	3.86
BAYES9(.1)	3.37	2.91	3.83	3.41
BAYES9(.3)	2.91	1.84	3.83	3.13
BAYES9(.5)	3.07	1.69	3.76	3.09
LIKE9	2.45	1.23	5.20	4.01
PRIOR9	3.52	2.91	3.76	3.31
PREF9	1.53	0.61	3.90	4.01
VOTE9	2.14	0.76	3.97	3.65
AVE9	3.06	1.38	4.92	4.32

TABLE 6. COMPARISON OF DECISION ALGORITHMS (THE MEASURE IS THE ESTIMATED MINUS THE TRUE PERCENT WHEAT OVER 55 SECTIONS)

	Bias (Mean Algebraic Error)	Bound on Significance of Bias	Median Absolute Error	Mean Absolute Error	Root-Mean-Square-Error	Mean Improvement Over QRULE	Bound on Significance of Improvement	Stand. Dev. of Improvement
QRULE	3.6	0.007	4.6	6.9	10.4	---	---	---
BAYES9 (0.1)	3.5	0.012	4.7	6.9	10.5	0	1.0	0.8
BAYES9 (0.3)	3.4	0.02	4.1	7.0	10.9	-0.1	0.6	1.5
BAYES9 (0.5)	3.3	0.02	4.3	7.0	11.0	-0.2	0.6	2.0
LIKE9	3.9	0.025	5.8	8.8	13.0	-1.9	0.007	5.0
PRIOR9	3.3	0.02	5.2	6.9	10.5	0	0.9	1.3
PREF9	3.0	0.04	4.4	7.2	11.2	-0.3	0.6	4.3
VOTE9	3.0	0.06	4.1	7.5	12.0	-0.6	0.20	3.2
AVE9	4.7	0.001	5.9	8.3	11.3	-1.4	0.001	2.2
LIMMIX (0.4)	1.0	0.4	3.8	6.1	9.2	0.8	0.20	4.3
LIMMIX B (0.4)	1.8	0.17	4.2	6.7	10.2	0.2	0.8	4.7
LIMMIX C (0.4)	1.4	0.3	3.8	6.6	10.0	0.3	0.7	5.3
9-PT MIX	1.9	0.2	3.4	6.4	10.5	0.5	0.3	4.0
Automatic Cluster Mapping	1.3	0.4	4.7	8.0	12.5	-1.1	0.11	5.1
ADMAP (10^{-5})	4.7	0.005	5.3	8.7	13.0	-1.8	0.04	6.3
ADMAP (10^{-6})	5.4	0.001	5.6	8.3	12.7	-1.4	0.11	6.3
ADMAP (10^{-7})	5.2	0.001	4.7	8.1	12.4	-1.2	0.17	6.0
LRULE	5.3	0.001	4.8	8.1	12.5	-1.2	0.14	6.0

TABLE 7. COMPARISON OF LIMMIX PROCEDURES FOR VARIOUS PARAMETER SETTINGS (THE MEASURE IS THE ESTIMATED MINUS THE TRUE PERCENT WHEAT OVER 55 SECTIONS)

	Y	Bias (Mean Algebraic Error)	Bound on Significance of Bias	Median Absolute Error	Mean Absolute Error	Root-Mean-Square-Error	Mean Improvement Over QRULE	Bound on Significance of Improvement	Stand. Dev. of Improvement
QRULE	---	3.6	0.007	4.6	6.9	10.4	---	---	---
LIMMIX, m = 0.4	---	1.0	0.4	3.8	6.1	9.2	0.8	0.20	4.3
9-PT MIX	---	1.9	0.2	3.4	6.4	10.5	0.5	0.3	4.0
LIMMIX B, m = 0.25	0.8	3.4	0.012	4.8	6.9	10.1	0	1.0	4.1
	0.9	3.3	0.015	4.2	6.6	10.4	0.3	0.6	3.3
	1.0	2.9	0.04	3.5	6.5	10.6	0.4	0.4	3.2
	1.2	3.0	0.04	3.6	6.6	10.7	0.3	0.5	3.1
	1.4	3.2	0.02	3.6	6.7	10.7	0.1	0.7	2.8
LIMMIX B, m = 0.4	0.8	3.5	0.012	4.8	7.0	10.4	-0.1	0.9	3.8
	0.9	2.1	0.11	4.4	6.6	9.8	0.3	0.6	4.5
	1.0	1.8	0.17	4.2	6.7	10.2	0.2	0.8	4.7
	1.2	2.4	0.09	4.0	6.6	10.5	0.3	0.6	3.7
	1.4	2.7	0.05	3.4	6.6	10.6	0.3	0.6	3.2
LIMMIX C, m = 0.25	0.8	3.1	0.02	4.7	6.8	9.9	0.1	0.9	4.4
	0.9	2.0	0.14	4.2	6.6	9.9	0.3	0.7	4.6
	1.0	1.9	0.17	4.4	6.8	10.3	0.1	0.9	4.7
	1.2	1.6	0.23	4.0	6.7	10.3	0.2	0.8	4.7
	1.4	2.0	0.14	4.2	6.7	10.2	0.2	0.8	4.6
LIMMIX C, m = 0.4	0.8	2.8	0.009	4.9	6.6	9.5	0.2	0.8	5.5
	0.9	1.5	0.025	4.0	6.5	9.6	0.4	0.6	5.3
	1.0	1.4	0.23	3.8	6.6	10.0	0.3	0.7	5.3
	1.2	1.9	0.3	4.3	6.8	10.1	0.1	0.8	5.2
	1.4	1.6	0.17	3.9	6.7	10.1	0.2	0.8	5.0

TABLE 8. HOW DECISION ALGORITHM BIAS VARIES WITH THE TRUE PERCENT WHEAT (BIAS IS THE MEAN OF THE ESTIMATED MINUS THE TRUE PERCENT WHEAT)

	Total Bias	Bias for Sections with < 30% Wheat "Low Wheat"	Bias for Sections with 30-50% Wheat "Middle Wheat"	Bias for Sections with > 50% Wheat "High Wheat"
No. of Sections	55	17	26	12
QRULE	3.6	4.1	3.9	2.5
QRULE posterior	2.7	3.3	2.7	1.9
QRULE null 45	1.0	0.5	1.2	1.5
QRULE null 35	0.1	-0.3	0.1	0.7
QRULE null 25	-1.9	-2.0	-2.3	-1.1
BAYES9 (0.1)	3.5	3.7	3.7	2.7
BAYES9 (0.3)	3.4	3.4	3.5	3.1
BAYES9 (0.5)	3.3	3.2	3.3	3.6
LIKE9	3.9	2.9	2.6	8.0
PRIOR9	3.3	3.4	3.5	2.7
PREF9	3.0	2.1	2.5	5.4
VOTE9	3.0	1.4	3.1	4.9
AVE9	4.7	5.3	4.3	5.0
LIMMIX (0.4)	1.0	-0.3	2.2	0.2
LIMMIX B (0.4)	1.8	-0.1	3.3	1.4
LIMMIX C (0.4)	1.4	-0.4	2.9	0.5
9-PT MIX	1.9	-0.3	3.2	1.9
Automatic Cluster Mapping	1.3	-1.9	2.4	3.4
ADMAP (10^{-5})	4.7	5.1	4.8	4.0
ADMAP (10^{-6})	5.4	6.3	5.4	4.3
ADMAP (10^{-7})	5.2	5.9	5.1	4.5
LRULE	5.3	6.0	5.2	4.4

TABLE 9. COMPARISON OF DECISION ALGORITHMS WITH AND WITHOUT A NULL TEST. (THE MEASURE IS THE ESTIMATED MINUS THE TRUE PERCENT WHEAT OVER 55 SECTIONS.)

Rule	Bias (Mean Algebraic Error)	Bound on Significance of Bias	Median Absolute Error	Mean Absolute Error	Root-Mean-Square-Error	Mean Improvement Over QRULE	Bound on Significance of Improvement	Stand. Dev. of Improvement
QRULE	3.6	0.007	4.6	6.9	10.4	---	---	---
QRULE null 45	1.0	0.4	4.0	6.0	9.7	0.9	0.09	4.0
QRULE null 35	0.1	0.9	4.0	6.2	9.9	0.7	0.3	4.8
QRULE null 25	-1.9	0.17	5.2	7.0	10.3	-0.1	0.9	6.4
PRIOR9	3.3	0.02	5.2	6.9	10.5	-0.0	0.9	1.3
PRIOR9 null 45	0.9	0.5	3.6	6.0	9.9	0.9	0.14	4.3
LIMMIX	1.0	0.4	3.8	6.1	9.2	0.8	0.20	4.3
LIMMIX null	0.9	0.5	3.4	6.0	9.1	0.9	0.14	4.4
9-PT MIX	1.9	0.2	3.4	6.4	10.5	0.5	0.3	4.0
9-PT MIX null	1.8	0.2	3.3	6.3	10.4	0.6	0.23	3.8
LIMMIX B	1.8	0.17	4.2	6.7	10.2	0.2	0.8	4.7
LIMMIX C	1.4	0.3	3.8	6.6	10.0	0.3	0.7	5.3

TABLE 10. COMPARISON OF THE PIXEL-COUNT METHOD OF ACREAGE ESTIMATION WITH THE METHOD OF SUMMING POSTERIOR PROBABILITIES. (THE MEASURE IS THE ESTIMATED MINUS THE TRUE PERCENT WHEAT OVER 55 SECTIONS.)

Rule	Bias (Mean Algebraic Error	Bound on Significance of Bias	Median Absolute Error	Mean Absolute Error	Root-Mean-Square-Error	Mean Improvement Over QRULE	Bound on Significance of Improvement	Stand. Dev. of Improvement
QRULE:								
pixel count	3.6	0.007	4.6	6.9	10.4	---	---	---
posterior with								
0 iterations	3.6	0.007	4.5	6.5	9.8	0.3	0.11	1.6
1 iteration	3.6	0.007	4.6	7.0	10.6	-0.2	0.6	2.1
2 iterations	3.6	0.007	4.6	7.1	10.8	-0.2	0.4	2.2
PRIOR9:								
pixel count	3.3	0.02	5.2	6.9	10.5	-0.0	0.9	1.3
posterior	3.6	0.007	4.9	6.9	10.4	0.0	1.0	1.4

TABLE 11. HOW ALGORITHM BIAS VARIES WITH SITE. (BIAS IS THE MEAN OF THE ESTIMATED MINUS THE TRUE PERCENT WHEAT.)

	Ellis	Deaf Smith	Randall	Finney	Saline	Total
QRULE	1.0	4.9	-0.5	2.0	12.9	3.6
QRULE null 45	0.9	-0.9	-2.9	-1.8	12.9	1.0
QRULE null 35	0.9	-2.7	-5.1	-2.6	12.8	0.1
QRULE null 25	0.5	-5.2	-9.8	-4.7	11.1	-1.9
QRULE posterior (0)	0.5	1.7	-0.7	1.2	12.1	3.6
QRULE posterior (1)	0.7	1.2	-0.8	0.6	14.2	3.6
QRULE posterior (2)	0.7	1.1	-0.8	0.5	14.4	3.6
BAYES9 (.1)	1.1	4.1	-0.6	1.6	13.6	3.5
BAYES9 (.3)	1.2	2.6	-0.5	1.3	14.8	3.4
BAYES9 (.5)	1.2	1.6	-0.4	1.1	15.5	3.3
LIKE9	0.6	-4.5	-0.7	1.6	22.5	3.9
PRIOR9	1.2	3.7	-0.5	1.3	13.5	3.3
PREF9	3.2	-0.8	-2.2	-0.1	17.7	3.0
VOTE9	0.8	2.4	-0.8	-0.5	17.7	3.0
AVE9	2.5	4.8	0	3.2	14.9	4.7
LIMMIX (0.4)	1.6	9.0	-1.4	-3.4	8.9	1.0
LIMMIX B (0.4)	1.7	13.4	-1.3	-3.5	10.8	1.8
LIMMIX C (0.4)	1.1	14.7	-0.9	-4.0	9.1	1.4
Nine-Point Mixtures	1.3	8.6	-1.3	-2.8	12.9	1.9
Automatic Cluster Mapping	2.3	2.9	-2.5	-4.9	18.7	1.3
ADMAP (10^{-5})	0.2	1.4	-1.1	3.6	18.8	4.7
ADMAP (10^{-6})	0.3	2.3	-0.6	5.1	18.1	5.4
ADMAP (10^{-7})	0.3	2.4	-0.3	4.6	17.9	5.2
LRULE	0.4	2.4	-0.1	4.6	17.9	5.3

5.3 DISCUSSION OF WITHIN-FIELD AND AREA ESTIMATION RESULTS

Looking at Table 3, we see that all the nine-point rules do better than QRULE on the interior pixels of the fields (those with an inset of 1.5 or more). On the near boundary pixels (Table 4) AVE9 and LIKE9 do uniformly worse than QRULE. These two rules include in their decision calculations data from all 9 pixels as if all 9 came from the same distribution. Thus these two rules do well on interior pixels where this assumption is true and not so well on the near-boundary pixels where it isn't. Unfortunately, even among the within-field pixels, only 27% are also interior pixels; among all the pixels in the site, this percentage drops to about 16%. Thus AVE9 and LIKE9, which do well on only 16% of the pixels in a region and poorly on the rest, are noticeably worse than QRULE as wheat area estimators (Table 6).

VOTE9 and PREF9 do better near boundaries because they can have three of the 9 pixels outside the field and still have a quorum to vote correctly. They both do consistently better than QRULE on within-field pixels. PREF9, a voting rule that uses more information than VOTE9, outperforms VOTE9 on the interior pixels and is slightly superior on the near-boundary pixels and on area estimation. (The latter difference is not statistically significant.) But neither rule quite measures up to QRULE as an area estimator, although the difference is not statistically significant.

BAYES9 and PRIOR9 are designed to be effective in boundary areas, and thereby, be more useful in Landsat data processing. Although they both score better than QRULE on interior and near-boundary pixels, their area estimation results (Table 6) are no better than QRULE's. Their improvement (Column 6 of Table 6) of zero is the top score for the nine-point rules. Of course, BAYES9 (.1) is defined to be similar in effect to QRULE because its parameter assigns small weight to the dependence between pixels. This similarity is shown by its small standard deviation

of improvement (0.8%). The analogous figure for PRIOR9 (1.3%) illustrates that it, too, is in effect similar to QRULE because of the important role played by the center pixel.

All the nine-point rules do better, as expected, in the recognition mode than in the classification mode (Table 5). The test on Ellis data does not indicate which mode is best for QRULE. If future tests do not indicate a superiority of the recognition mode of QRULE, the classification mode would be faster and therefore preferable.

As we can see from Tables 6 and 7, the mixture algorithms show a slight, but consistent improvement over QRULE as an area estimator (i.e., have a lower mean absolute error as indicated by plus values in the improvement column). None of the improvements come close to statistical significance but the fact that they are with one exception positive indicates a trend toward improvement and shows that LIMMIX B and LIMMIX C are relatively insensitive to their parameter settings.

Nine-Point Mixtures also show an improvement over QRULE but not as much as LIMMIX. We had thought that by not estimating a mixture for pixels complying with a neighborhood consensus, the algorithm would decide more accurately than LIMMIX. Replacing the VOTE9 technique used in Nine-Point Mixtures by another nine-point algorithm such as PREF9 or a gradient method [1] might lead to an improvement in results. Of course, the difference of 0.3 in the mean absolute error of LIMMIX and Nine-Point Mixtures could easily have been the result of chance alone.

The algorithm for classifying by automatic cluster mapping has a larger mean absolute error than QRULE's (see Table 6). The median error is about equal to QRULE's and the rms error is 2.1% greater, showing that automatic cluster mapping does as well as QRULE on the small errors but gets poorer results overall by making some pretty bad mistakes. A comparison with the more favorable results for human-aided cluster mapping (Table 1, Section 3) indicates that our initial attempt at automatic cluster mapping would benefit from further development.

The best linear LRULE does not perform as well as QRULE (Table 6). Previous comparisons of LRULE and QRULE [7] indicated that QRULE did better than LRULE on training fields but no better on test fields. QRULE can be shown by Bayesian decision theory to outperform, on the training set, any rule such as LRULE that uses only the limited information that QRULE does. Although only 6.7% of the test pixels were chosen for training, about 25% of the test pixels are contained in training fields. (This difference in percentages is accounted for by the 1.5 inset requirement for training pixels and the ratio of 3.7 between the number of 0.5-inset and 1.5-inset pixels within fields.) We would expect QRULE's superiority on the training pixels to extend to all the training field pixels, because they are so similar, and thus explain QRULE's lower error rate in the present test.

Another possible source of strength for QRULE is that it was run in the recognition mode, while LRULE is confined by its formulation to run in the classification mode. We don't know yet which mode of running QRULE produces the most accurate area estimates.

The median absolute error is approximately the same for QRULE and LRULE, showing that LRULE's poorer performance reflects a few big errors rather than a general inferiority. A look at the individual section results confirm this conclusion. Of 5 sections with a difference in estimates of 10% or more, four favor QRULE.

The adaptive processing algorithm ADMAP is based on LRULE rather than QRULE in order to make it run faster. Consequently, it includes LRULE's inferiority to QRULE in the present test results. But even if we compare ADMAP with LRULE, we observe a trend toward poorer performance at higher adaptation rates. The reason why adaptation does so little good in this test is that each site is peppered with training fields and hence there is nothing to be gained by adapting. We would expect ADMAP to be useful if the signatures were extended from another site or time and weren't quite right, or if we were processing a large area with gradually changing signatures.

The results in Table 10 indicate that using the posterior probability method of estimating acreage makes very little difference. For PRIOR9, the two methods of estimating acreage have nearly identical results. The standard deviation of the difference between the two methods (a figure not given in the table) is 0.4%, showing a consistently close agreement between the two methods over the sections. A histogram of the posterior probabilities for all the pixels in the 55 sections showed that the posterior probability of wheat was greater than 99.5% in 36.3% of the pixels and less than 0.5% in another 53%. Only 10.7% of the pixels had posterior probabilities between 0.5% and 99.5%. So for PRIOR9, the two methods of estimating acreage are, for all practical purposes, the same.

Attached to QRULE, the posterior probability method with no iterations scores a slight improvement of 0.3% over the pixel-count method, loses half a per cent on one iteration and reaches convergence on one iteration. Convergence is shown by a mean difference of zero between the two iterations and a standard deviation of 0.2%, figures not given in the table.

The null test results in Table 9 show that a null test level of 45 improves the mean absolute error of QRULE as much as does any algorithm. The improvement is largely maintained for a test level of 35 but drops to zero when the level is lowered to 25. The null test version of PRIOR9 mirrors the result for QRULE. 45 is a rather high level for the chi-square value, which reaches the 0.001 significance level at 18.5. The high level cuts out pixels that are wildly different from wheat, but preserves the identity of wheat pixels that might be coming from a wheat distribution similar, but not identical, to the training set distributions.

The null test makes little difference in the performance of LIMMIX and Nine-Point Mixtures. The improvement over QRULE in bias and absolute error remains nearly the same. No doubt these mixture rules classify as mixtures many pixels that would fail a null test in a non-mixture rule.

The bias (defined as the mean of the signed differences between estimated and true wheat proportions) is of critical importance in a large-scale survey because an unbiased decision algorithm increases in accuracy as it is averaged over many samples but a biased one does not. But the bias results are difficult to interpret because the interaction between the training set choice and the data distribution pattern is a primary source of bias (Table 11). In three sections of Saline, for example, large areas of river bottom grass not represented in the training sets masquerade as wheat in the multispectral recognition, thereby introducing a considerable wheat bias. Again, all but four large wheat fields in Finney are irrigated and none of these four are represented in the wheat training sets. Consequently, they are recognized as other, introducing bias towards other. These are just two examples that we know about; there may be other such interactions.

With this qualification in mind, we consider the bias results. We first consider the overall bias results in Tables 6 and 7. We note that QRULE, LRULE, ADMAP, and the nine-point rules have a significant wheat bias while QRULE with a null test, the automatic cluster mapping rule and the mixture rules do not.

Although the parameter settings of LIMMIX B and LIMMIX C have very little effect on the improvement over QRULE, they do appear to affect the bias (Table 7). The smallest bias occurs at $m = 0.4$ and $\gamma = 1.0$, confirming theoretical expectations. It is with these parameter values that LIMMIX B and LIMMIX C results are reported in Tables 6 and 8. LIMMIX C has a consistently smaller bias for equivalent parameter settings than LIMMIX B. LIMMIX has the smallest bias of all the mixture algorithms and shares with QRULE (null 45) the distinction of having the smallest bias of all the algorithms tested.

We next consider trends in the bias related to the true proportion of wheat (Table 8). Four main trends are apparent:

1. An overall positive wheat bias most noticeable in QRULE without a null test, LRULE, ADMAP and the nine-point rules.
2. A decreasing bias trend (i.e., a tendency for high bias in low-wheat sections and low bias in high-wheat sections). This trend is apparent in QRULE, LRULE, ADMAP, BAYES9(.1) and PRIOR9.
3. An increasing bias trend (i.e., a tendency for low bias in low-wheat and high bias in high-wheat sections) observable in the nine-point rules LIKE9, PREF9 and VOTE9, in the automatic clustering rule and in QRULE with a null test.
4. A high-center bias trend (i.e., a tendency for significant bias only in the middle-wheat group) for the mixture rules.

The reduction of bias from 3.6 to 1.0 by imposing a high null test on QRULE suggests that the overall wheat bias in QRULE and related rules (such as QRULE, posterior mode, BAYES9(.1), PRIOR9, LRULE and ADMAP) is mostly accounted for by the identification as wheat some of the wildly non-wheat pixels when the null test is not operating.

The second trend, the decreasing bias for QRULE and related rules, can be explained in the same way. We would expect more wildly non-wheat pixels in a low-wheat than a high-wheat section. Hence the wild-pixel bias is greater in the low wheat sections. When the wild-pixel bias is removed by a null test, both the overall wheat bias and the decreasing trend disappear.

One might try to explain the decreasing trend by the fact that QRULE, LRULE and ADMAP are run with equal priors, and we would expect to see, on the low-wheat sections, that a rule with priors that overestimate wheat would itself overestimate wheat. On the high-wheat sections, the rule would have the opposite tendency. But because PRIOR9, a rule that sets its own priors on the basis of neighborhood data values, also exhibits such a trend, this explanation is of doubtful validity.

The third trend, the increasing bias for nine-point rules LIKE9, PREF9 and VOTE9, has an explanation that is most easily applied to VOTE9. When wheat fields are small and scarce, there is a shortage of neighboring wheat votes to bolster up an otherwise reasonable wheat decision. Thus, the scales are tipped against wheat -- the bias decreases. When wheat neighbors are plentiful, there is a greater tendency to decide wheat -- the bias is larger. The other nine-point rules tend to behave like VOTE9; they decide on wheat if the evidence from the center pixel is bolstered by neighboring data values.

The automatic cluster mapping rule exhibits the same trend and for a similar reason. Unknown clusters are identified by the identity of their neighbors. A shortage of wheat neighbors cuts down the wheat estimate and a plentiful supply builds it up. Thus, the bias of the cluster mapping procedure, although small overall, is seen in Table 8 to increase with the amount of wheat present.

Our inferences about bias trends should be tempered by the fact that the groupings by percentage wheat are not independent of the choice of site. Fourteen of the 17 sections in the < 30% group are from Finney and half the 12 sections in the > 50% group are from Saline. It is, therefore, quite possible that trends that appear to relate decision algorithm bias to the percent of wheat present are really the result of the interaction of training set choices with data distribution patterns in the sites.

6
CONCLUSIONS AND RECOMMENDATIONS

Nearly all reasonably-conceived decision algorithms seem to perform well on data from a single pass in the growing season when they have good local signatures. The average absolute error for the 14 algorithms tested ranged from 6.1% to 8.8% and the wheat bias from -1.9% to 5.4%.

A properly-chosen null test can lower the bias of QRULE and reduce the average absolute error. In our test using four-channel data and good local signatures, a chi-square level in the range 35 to 45 defined the null test that best improved performance. QRULE's bias of 3.6% was reduced to 1.0% and 0.1% for levels 45 and 35, respectively, and its absolute error of 6.9% reduced to 6.0% and 6.2%.

The nine-point rules outperformed QRULE on field interiors but were no better, and in some instances noticeably worse than QRULE as wheat area estimators. The nine-point rules PRIOR9 and BAYES9, designed to be effective in the boundary areas so plentiful in Landsat data, scored the best of the nine-point rules by equalling QRULE's performance. They might be helpful in areas with larger field sizes or in processing future satellite data having a higher resolution than Landsat data.

The mixture rules led by LIMMIX maintained a slight, but consistent improvement over QRULE in the test. Compared with QRULE's overall bias of 3.6% and mean absolute error of 6.9%, the comparable figures for the mixture rules ranged from 1.0% (LIMMIX) to 1.9% and from 6.1% (LIMMIX) to 6.7%, respectively.

The posterior probability method of acreage estimation, with or without iteration, is very similar in result to the usual pixel-count method.

Our initial attempt at automatic cluster mapping did not fulfill the promise of human-aided cluster mapping. Further development, especially the incorporation of the principle of spectral closeness in identifying unknown clusters, would be likely to improve the results.

The test results reported here apply to circumstances similar to those of the test: clean data, one good Landsat pass in the growing season, good local signatures, a wheat-producing area like the sites in Kansas and Texas with similar field sizes. It would be difficult to extrapolate these results to other conditions, particularly to poorly-registered multitemporal data. The results do indicate the relative strengths of the decision algorithms when there are few pixels per field and many mixtures. But it is not clear that the order of rule performance would be maintained with less representative signatures. ADMAP, for example, is designed to adjust to such circumstances and cluster mapping should prove to be more insensitive to ground truth errors than the other rules.

The experimental design had two sound features. The comparison of estimated with true wheat errors is a performance measure that realistically refers to the objective of wheat inventory. The use of a section (1x1 mile square) as an experimental unit supplies the replications necessary to draw conclusions, even though allowance has to be made for the dependence of sections within a site.

As for the execution of the experiment, the strongest evidence of its correctness is that we can understand and explain most of the results. For example, we note that the best-scoring rules on Landsat data, where near-boundary pixels are plentiful, are mixture rules designed to make a sensible decision on boundary pixels. The two nine-point rules likely to be most inapplicable to Landsat data (AVE9 and LIKE9 because of their assumption of neighborhood homogeneity) had the poorest scores. The cluster mapping procedure scored best when both of its basic principles of nearness were employed.

Taken together, the experimental plan and procedure appear to be able to distinguish between a good and a bad wheat recognition algorithm. They could, therefore, be useful in evaluating new and modified algorithms and in so doing, speed up the cycle of testing and development.

The pairwise mixture algorithm LIMMIX, the algorithm that performed best in our test, is many times slower than QRULE. For the type of data in our study, the improvement in accuracy would probably be considered too small to be worth the extra time. For a region of small fields where the performance of QRULE would be expected to break down, LIMMIX could become the algorithm of choice.

APPENDIX I

DESCRIPTION OF THE KALMAN FILTER

The Kalman filter is an iterative filter, especially useful for digital computation, that produces an estimate of a time sequence of state vectors from a corresponding time sequence of measurement vectors. In the simplest application, 5 elements must be defined. These are: (1) the state vector, (2) the measurement vector, (3) an observation matrix relating the state vector to the measurement vector (assuming no measurement noise) by a linear transformation, (4) a covariance matrix describing additive noise in the measurement, and (5) a covariance matrix describing the statistics of the successive differences in the state vector.

In order to apply the Kalman filter to remote sensing data, we must make an association between the elements of the Kalman filter and elements of the classifier. This can be done in a number of ways, one of which is now described.

Assume that the most important statistics to update are the components of the mean vector of each material class, and that we will update after each single observation. Then we make the following identifications.

1. The mean vectors of each material are combined into a single vector identified as the state vector, x_t. The initial condition, x_o, is given by the initial training data for each crop.

2. The observed data vector is identified as the measurement, z_t.

3. The classified output (a recognition vector) is used to produce a matrix, H_t, of zeros and ones (a spotting function) which selects the correct components of the state vector to provide a relationship between the state vector and the noise-free measurement.

4. The covariance matrices of all the signatures are averaged. This is identified as an average estimate of the measurement noise covariance, R, as required for the Kalman filter.

5. An augmented matrix if formed by replicating and scaling the matrix R. This augmented matrix is identified as the covariance Q of the successive differences in the state vector. Covariance Q is assumed to be some simple function θ of R, and this assumption results in significant savings in computation time, since matrix inversions are not required for each update, and the computer memory requirements are minimal.

With these assumptions the Kalman filter equations become:

$$\hat{x}_t = \hat{x}_{t-1} + K_t(z_t - H_t\hat{x}_{t-1}) \tag{1}$$

where: $\hat{x}_t$ is the estimate of state vector x_t

K_t is the Kalman filter and minimizes $E((\hat{x}_t - x_t)(\hat{x}_t - x_t)^t)$

It is shown in [9] that

$$K_t = P'_t H_t^t [H_t P_t H_t^t + R_t]^{-1} \tag{2}$$

where

$$P_t = P'_t - K_t H_t P'_t \tag{3}$$

$$P'_t = P_{t-1} + Q_{t-1} \tag{4}$$

and P_o is chosen to reflect one's confidence in the accuracy of the starting signatures.

This expression for K_t further simplifies to:

$$K_t = \frac{\phi_{tM}}{\phi_{tMM} + 1} \otimes I_n \tag{5}$$

where

$$\phi_1 = \theta \tag{6}$$

M_t = a column vector with a 1 in position K, and zeros elsewhere (a spotting function)

$$\phi_{t+1} = \phi_t + \frac{\phi_{tM}\phi_{tM}^t}{\phi_{tMM} + 1} + \theta \tag{7}$$

$$\phi_{tM} = \phi_t M_t \tag{8}$$

$$\phi_{tMM} = M_t^t \phi_t M_t \tag{9}$$

and θ is assumed to have the form:

$$\theta = \theta_1 \begin{pmatrix} 1 & \theta_2 \cdots \\ \theta_2 & 1 \\ \vdots & \end{pmatrix}$$

where θ_1, θ_2 are scalars; θ_2 is in the range $0 \leq \theta_2 < 1$, because θ_2 is the amount of correlation in the variations in signature means, and θ_1 is closely related to the updating rate.

Further details about the Kalman filter are contained in Reference [9].

APPENDIX II

MEASUREMENT OF THE TRUE PROPORTION OF WHEAT IN EACH SECTION

J. Lewis

The principal standard for comparing wheat recognition algorithms is the true proportion of wheat in each section of each site. This appendix describes the way we determined this figure.

The ground truth information we were furnished for each site contained the crop type and acreage of each numbered field. Some areas such as small fields, houses and roads were not listed in the ground truth information. Therefore, the proportion of wheat in a section would not have been accurately measured by dividing the acreage of the wheat fields by the acreage of all the fields. Neither would dividing the wheat acreage by 640 have been sufficiently accurate because of variation in the area of a section.

Instead, our procedure was to compute the area of each section and the whole site by a program that accepts the continuous line and point coordinates of the corners as input and computes the area of the section/site in pixel units correct to one one-thousandth of a pixel. The pixel area of each section includes the interior roads but not the surrounding roads. The same remark applies to the whole site. Thus, the difference between the area of the site and the sum of the areas of the sections measures the area of the roads between the sections. From this information we can find the area of one half of a road running around the section and add it to the previously-computed area of the section. The wheat acreage converted to pixel units is then divided by this augmented section area to obtain the true proportion of wheat in the section.

APPENDIX III

THE DETECTION AND CORRECTION OF STRIPING IN LANDSAT DATA

W. Richardson and J. Lewis

III.1 INTRODUCTION

Four-channel Landsat data comes from 24 detectors: four for line 1, four for line 2, and so on up to line 6. Then they begin over on line 7. When the detectors do not have uniform gain settings, we can see on the graymap of some channels a striping that has a period of 6 lines. The POINT module STRIPE was written to detect such striping and the module UNBAND to correct for it.

The output of STRIPE consists of 6 tables:

1. A 6 x 4 table of detector means, associating each of four means with one of the first 6 lines of the rectangle processed.
2. A 6 x 4 table of detector standard deviations showing whether any one detector has such a variable performance that the associated data would be of doubtful utility. (This test was used in CITARS [10]).
3. A listing of the four channel means. Each mean is the sum of all data values divided by the number of pixels in that channel.
4. A listing of the four channel standard deviations, computed by the formula:

$$\sqrt{\frac{\sum_{\substack{\text{in that}\\ \text{channel}}} (\text{data value})^2}{\text{no. data values in channel}} - (\text{channel mean})^2}$$

5. A 6 x 4 table of differences between detector mean and channel mean, showing whether any detectors are significantly out of

line with the others. To decide whether correction is necessary, we compare the figures in this table with the corresponding channel standard deviations.

6. A 6 x 4 table of recommended additive corrections to equalize the detectors.

The correction vector is either punched onto two cards which are read by the module UNBAND that carries out the correction or is transmitted directly to UNBAND, depending on whether the operator wishes to look at the STRIPE output before correcting or would prefer to carry out the correction automatically.

III.2 HOW THE CORRECTIONS ARE CALCULATED

The corrections are obtained for each channel separately. We start with a central value C, such as the channel mean. We compute the differences between the 6 detector means in that channel and C. Then we compute the integer correction that puts each corrected detector mean as close to zero as possible. For example

Detector Mean -C	Correction
.1	0
2.7	-3
.7	-1
- .7	1
-1.8	2
-1.1	1

It is not enough to do this for the channel mean alone. The following example shows a possible set of 6 differences from the channel mean, the correction that would be imposed and a better correction that is possible.

Detector Mean - Channel Mean	Automatic Correction	Better Correction
0.6	-1	0
0.6	-1	0
0.6	-1	0
-0.6	1	1
-0.6	1	1
-0.6	1	1

The better correction puts all 6 detector means within 0.2 of each other while the automatic correction keeps them at a distance of 0.8. The better correction would have been obtained if we had started with the channel mean + 0.4 rather than the channel mean.

The best correction is obtained by applying the central value procedure to a range of central values on either side of the channel mean C:

$$C,\ C + 0.1,\ C - 0.1,\ C + 0.2,\ C - 0.2,\ \ldots,\ C - 0.5$$

For each such central value, a correction vector is generated and the variance of the corrected detector means is computed. The central value producing minimum variance is considered optimal and the corrections calculated from that central value are accepted as the recommended corrections.

If the step size had been infinitely small rather than 0.1, then the procedure just described can be proved to yield corrections optimal in the sense of minimum variance of the corrected detector means.

Proof.

We first show that there is an optimal correction vector. Let X be the largest absolute difference between a detector mean and the channel mean. Let S be the class of correction vectors each of whose elements is smaller in absolute magnitude than $12X + 12$. Let d be any correction

vector outside the class S. We will show that there is an element of S with variance $\leqq$ that of d. Subtract the first element of d from every element of d. The variance of d is unchanged and the first element of d is now 0. If all the elements of the new d are smaller in magnitude than 12X + 12, then the new d is in S and has the same variance as the old d. There remains the case that elements of the new d are greater in magnitude than 12X + 12. Suppose that the second element of d > 12X + 12. Then both the first and second terms will make a contribution to the variance of at least

$$\left(\frac{12X + 12}{2}\right)^2 \Big/ 6 \;=\; (6X + 6)^2/6 \;=\; 6(X + 1)^2$$

This is greater than the variance of the correction generated by C. Thus the minimum over the finite set S is the minimum over the set of all correction vectors.

Let $(e_1, e_2, \ldots, e_6)$ be an optimal correction vector; i.e., the corrected detector means $(g_1, \ldots, g_6)$ have minimum variance. Let G be the mean of $g_1, \ldots, g_6$. Then $|g_i - G| \leqq .5$ for all i. Otherwise there exists i such that an integer could be added or subtracted from g, to bring it closer to G. Let G' be the mean of $g_1, \ldots, g_6$ with the improved g_i. Then

$$\text{old } \textstyle\sum (g_i - G)^2 \;>\; \text{new } \sum (g_i - G)^2 \;>\; \text{new } \sum (g_i - G')^2.$$

The latter inequality holds because of the theorem that the sum of squared differences of a set of numbers from a fixed value is minimal when that fixed value is the mean of the set.

We have shown that the correction vector producing minimum variance is the vector generated by a central value G. The same minimum variance is obtained for the vector generated by

$$\ldots\ G-2,\ G-1,\ G,\ G+1,\ G+2,\ \ldots$$

Hence one of these equivalent central values lies in the interval C $\pm$ 0.5. Q.E.D.

III.3 SOME EXAMPLES OF CORRECTION FOR STRIPING

Module STRIPE has been run on the Deaf Smith, Randall, Finney, Saline and Ellis intensive study site data with the results that are given in Table III-1.

TABLE III-1.

STRIPING AT FIVE SITES AND ITS CORRECTION

Deaf Smith 27 May 74

Detector Mean - Channel Mean				Recommended Correction			
0.6	0.3	2.7	0.4	-1	0	-3	0
0.4	0.7	0.7	1.7	-1	0	-1	-2
0.4	0.1	-0.7	0.3	-1	0	1	-1
-0.4	-0.3	-1.8	-1.0	0	1	2	1
-0.7	-0.7	-1.1	-0.6	0	1	1	0
-0.3	-0.2	0.1	-0.1	0	1	0	0

Randall 27 May 74

0.7	0.7	2.2	-0.7	0	-1	-2	1
0	0.6	0.3	1.5	0	-1	0	-1
0.1	-0.9	-1.3	0.3	0	1	2	0
-0.8	-0.9	-1.2	-0.6	1	1	2	1
-0.5	-0.2	-0.5	-0.5	1	0	1	1
0.4	0.7	0.4	-0.1	0	-1	0	0

TABLE III-1. (CONT.)

Detector Mean - Channel Mean				Recommended Correction			
			Finney 26 May 74				
0.4	0.2	1.5	-0.6	0	0	-1	0
0.3	0.1	0.5	1.4	0	0	0	-2
0.1	-0.2	-0.6	0.4	0	0	1	-1
-0.5	-0.4	-0.7	-0.5	1	0	1	0
-0.3	0.2	-0.4	-0.6	1	0	1	0
0	0.1	-0.4	-0.2	0	0	1	0
			Saline 6 May 74				
0.3	0.1	1.6	-0.6	0	0	-1	1
0.1	-0.2	0.4	1.6	0	0	0	-1
0.1	-0.3	-0.4	0.6	0	0	1	0
-0.5	-0.2	-1.0	-0.5	1	0	1	1
-0.1	0.5	-0.8	-0.8	0	-1	1	1
0	0.1	0.1	-0.3	0	0	0	1
			Ellis 12 June 74				
0.6	0.4	0.8	-0.4				
0.6	0.6	0.7	1.5				
0.2	-0.2	-0.6	0.4	not computed			
-0.5	-0.7	-0.7	-0.6				
-0.7	0	-0.6	-0.7				
-0.2	-0.1	0.4	-0.1				

In each case, the run was made for the rectangle enclosing the Intensive Test Site that had been graymapped in all four channels. A larger area was not used because without the graymaps we couldn't be sure that there weren't some bad lines that would distort the estimates of the detector means. Deaf Smith had such a line (848) near the top of the site and we ran STRIPE starting at the line after the bad one. Finney was missing line 727 and 728 near the bottom of the site, so we ran STRIPE from the top of the site to 726.

The line sets have been cyclically permuted for all sites but Randall to make the detector biases correspond. The correspondence was achieved by putting in the top two line sets a positive bias in channel 3 on the first line set and a large positive bias in channel 4 on the next. The correspondence between Randall and Deaf Smith, which are contained in the same ERTS frame, was verified by observing that the corresponding lines differed by a multiple of 6. When we compare the mean difference table for Randall and Deaf Smith, we note that the large differences correspond but that some of the small differences do not, showing that field-to-field variation among the line sets accounts for some small differences in the detector means but not the big ones.

For the present study we ran UNBAND with the recommended corrections on the Deaf Smith, Randall, Finney and Saline tapes. Deaf Smith and Randall were corrected because they had large detector biases. In Deaf Smith, for example, the range of detector bias in channel 3 is 4.5, quite large compared to the standard deviation of 7.5 in that channel. And striping is apparent in the graymap of Randall, channel 4. Finney and Saline were corrected because very little processing had been done on them and it was no loss of effort to bring the detectors into line first. Ellis was not corrected because much processing had been done on it and the biases were not excessive.

APPENDIX IV

NUMBER OF SIGNATURES NECESSARY FOR ACCURATE CLASSIFICATION*

W. Richardson, A. Pentland, R. Crane and H. Horwitz

IV.1 INTRODUCTION

Computer processing of multispectral scanner data as a means for measuring the earth's resources depends for its success on the definition of spectral classes, i.e., signatures, corresponding to materials to be recognized and backgrounds in the scene. Clustering techniques for defining these classes have been used with success, but have left unresolved the question of how many signatures to define. When classes are too few, they are so broad they overlap, resulting in unnecessarily large classification errors, while too many classes increase classification costs and cause difficulty in matching spectral classes with materials in the scene.**

A procedure at ERIM is to cluster the points into small spectral classes by a processing module CLUSTR and then to combine the clusters into larger signatures by a program GROUP. CLUSTR uses a relatively simple algorithm because it is applied to every data point. The number of small clusters it produces is an upper bound on the number of significant modes in the data space. GROUP, working on the set of clusters, much fewer in number than the data points, can take time to be careful.

---

* This appendix is to be presented as a paper at the Symposium on Machine Processing of Remotely Sensed Data, Purdue University, June 1976.

** When clustering is unsupervised, the difficulty of identifying spectral classes increases with the number of classes and with the smallness of the classes. When clustering is supervised and recognition is extended from training to test areas, test classes may appear between training modes and thus be recognized better by broader signatures.

It uses covariance information and before each step of combining a pair of clusters, considers all possible pairs in the light of certain criteria. At the end of a run of GROUP, the analyst has a choice of sets of combined signatures, each set being the best choice given the number of signatures. He also is provided tables and graphs to help decide how many signatures to use.

IV.2 DESCRIPTION OF A TECHNIQUE FOR DETERMINING THE NUMBER OF SIGNATURES

Our procedure for reducing the number of signatures combines signatures within categories. In principle, the procedure can be applied to any number of categories from one on up. The present implementation, program GROUP, requires two, which we name for definiteness "wheat" and "other". Both categories are treated the same way.

The procedure is summarized by the following steps:

A. Compute for each pair of signatures (clusters) within each category up to five measures of intersignature distance.
   1. Distance based on a combined covariance matrix.
   2. Determinant of the combined covariance matrix.
   3. Trace of the combined covariance matrix.
   4. Probability of misclassification between the pair.
   5. Increase in the probability of misclassification between categories (we describe these measures more fully below).

B. For each distance criterion selected, rank every pair of signatures and then combine the pair with the smallest weighted sum of ranks. Punch or otherwise save this combined signature.

C. Compute descriptive statistics such as the following:
   1. The average pairwise probability of misclassification between categories.
   2. The maximum determinant scaled to compare with distance measurement.
   3. The maximum trace scaled to compare with distance measurement.

D. Compute the observed probability of misclassification by classifying the training data from which the signatures were extracted. The classification uses the current set of signatures.

E. Repeat steps A - D until only one signature per category remains.

F. Display the statistics computed in C and D in a table and graphs.

From these displays, the user decides how many signatures are right for the multispectral recognition problem being attacked. The procedure has minimized the use of qualitative judgement by selecting from the myriad of possible signature combinations* a few likely candidates and providing information to aid in the qualitative choice among the few. When the user has made his choice, he assembles the chosen set of signatures from among those saved.

The input to the program GROUP is a number of "wheat" and "other" signatures. Each signature is in the form of a mean vector and a covariance matrix, parameters that are assumed to specify a multivariate normal distribution of data vectors from the material the signature represents. Signatures computed from fewer than 5 points are not accepted by the program.

The program provides 5 criteria for combining groups. Any of these criteria or any subset of them may be used. If two or more criteria are chosen, then the possible pairs of signatures to be combined are ranked according to each criterion and the pair with the smallest weighted sum of ranks is chosen. In that way the pair of signatures combined is the one most generally in harmony with the criteria selected. The 5 criteria are as follows:

1. An average covariance matrix A_W for the wheat signatures and one A_O for the other are calculated. The pair of signatures combined is the one with the smallest squared distance,

---

* There are 769,129 different signature combinations of 7 wheat and 7 other initial signatures.

$$(\mu_2 - \mu_1)^T A_W^{-1} (\mu_2 - \mu_1)$$

or

$$(\mu_2 - \mu_1)^T A_O^{-1} (\mu_2 - \mu_1)$$

depending on whether the pair is wheat or other. It is essentially the square of the usual distance between the means but with the scale modified by the inverse of the average covariance matrix.

2. The determinant of the combined covariance matrix. The combined covariance matrix of the training set is the covariance matrix of the union of the two sets except that each set may be given an arbitrary weight. If the weights are proportional to the number of pixels used in calculating the signature, then the combined signature is identical to the signature calculated from all points of the two sets. If the two sets have circular signatures far apart, for example, the combined covariance matrix is long and thin whereas the average covariance matrix is circular. The determinant is the product of the eigenvalues, in other words the product of the variances in the axial directions of the ellipsoidal distribution. The bigger the determinant, the more spread out the distribution.
3. The trace of the combined covariance matrix. The trace is the sum of the diagonal elements, namely the variances, and is also the sum of the eigenvalues. It is invariant under a rotation of the space. Like the determinant, it is a measure of how spread out the combined distribution is.
4. The squared Mahalanobis distance

$$D_{ij}^2 = (\mu_j - \mu_i)^T \left(\frac{R_i + R_j}{2} \right)^{-1} (\mu_j - \mu_i)$$

This is the same distance as criterion 1. except that the covariance matrix modifying the distance is the average of the two covariance matrices of the pair rather than the average of all the covariance matrices in the category. The difficulty with this criterion is that the more spread out a signature is, the smaller is its distance to any other signature. The criterion thus tends to encourage large variances rather than to hold them down. This criterion is included in the program largely by tradition. Our former method of combining signatures was to make a table of the probability of misclassification (p. of m.) defined for each pair of signatures as

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{1}{2}D_{ij}} e^{-\frac{t^2}{2}} dt \qquad (1)$$

and then to group the signatures intuitively as suggested by the table. Expression (1) is an estimate of the probability of deciding on signature j, given that the distribution is really represented by signature i or vice versa -- an estimate that becomes exact [11] if the covariance matrices of signature i and signature j are both equal to $(R_i + R_j)/2$.

5. The average pairwise wheat-other p. of m. For each wheat-other pair, the Mahalanobis distance D is computed and from that the p. of m. as in criterion 4. The criterion is a weighted average of these pairwise p. of m.'s. The wheat signatures start out with weights α_i that add to 1 and the other signatures with weights β_j that add to 1. The weights are initially equal but may be set in the control input. When two signatures are combined, their weights are added. The average pairwise wheat-other p. of m. is

$$\sum_{\text{wheat } i} \sum_{\text{other } j} \alpha_i \beta_j \text{ p. of m. } (i,j)$$

This number is printed at every step of the program and is one of the ways the user decides when the combining has gone far enough.

There is a case to be made for using only criterion 5. for combining. After all, is not the ultimate goal to minimize the probability of misclassification? The reason the distance criteria are also included is because experience shows that the training data seldom fully represent the data to be processed. If two distant signatures are combined because such a combination does not adversely affect the p. of m. of the training data, the combination might swallow up competing signatures in the test data. The safest plan is to use one or more distance criteria along with criterion 5. so that the two signatures to be combined will be a good choice both from the standpoint of distance and p. of m.

The criteria can be weighted so that the p. of m. criterion 5. gets half the weight and the distance criteria divide the other half. At the end of the run, a summary table is printed, each row of which corresponds to the number of signatures, so that the rows go from two to the original number of signatures. The columns refer to the criteria for the signature that was combined at that step and to other useful information. Digital plots of any requested columns of the table are given. The columns of the table we have found most useful are

1. Criterion 5., the average pairwise wheat-other p. of m.
2. The (2n)th root of the maximum covariance determinant. The determinant is the product of the eigenvalues. Hence, the nth root of the determinant is the geometric mean of the eigenvalues. An eigenvalue is the variance of the distribution in the direction of an axis of the ellipsoid. The variance is a squared quantity. Its square root, the standard deviation, is in units of

Euclidean distance. Thus the (2n)th root of the covariance determinant is an average standard deviation of the distribution, a measure of how spread out the distribution is. The maximum of these values shows how spread out the combined signatures are getting.

3. The square root of 1/n(maximum covariance trace). The trace of a covariance matrix is the sum of the diagonal terms (the variances) and is also the sum of the eigenvalues. Thus the trace/n is the arithmetic mean of the eigenvalues, an average variance, and its square root is therefore an average standard deviation of the distribution. It is also a measure of how spread out the distribution is. The only difference between this measure and the previous one is that the arithmetic rather than the geometric mean of the eigenvalues is taken.
4. The average pairwise p. of m. (as in column one) multiplied by one half the number of signatures in the set. The purpose of the multiplication is to make the average pairwise p. of m. more closely approximate the overall p. of m. Suppose for example there are three "other" signatures and one wheat signature. There are three wheat-other pairwise p. of m.'s, $p(W_1O_1)$, $p(W_1O_2)$, and $p(W_1O_3)$. Prob{other|wheat} is more closely approximated by $p(W_1O_1) + p(W_1O_2) + p(W_1O_3)$ than by 1/3 this amount. But prob{wheat|other} = $1/3\ p(W_1O_1) + 1/3\ p(W_1O_2) + 1/3\ p(W_1O_3)$ because the probability of choosing O_1 is 1/3 and the subsequent probability of deciding on wheat is $p(W_1O_1)$ and similarly for O_2 and O_3. Thus, the average of prob{other|wheat} and prob{wheat|other} is approximated by

$$\frac{2}{3}\left[p(W_1O_1) + p(W_1O_2) + p(W_1O_3)\right]$$

which is the average pairwise p. of m. times one half the num-

ber of signatures in the set. The figure we have calculated is an overestimate of the p. of m. just as the average pairwise p. of m. is an underestimate so columns one and four bound the true theoretical p. of m. between categories.

5. The observed p. of m. calculated by classifying the training points using the current set of signatures. This empirical measure of performance of the signature set complements the theoretical measures.

IV.3 APPLICATION OF THE TECHNIQUE

This process of clustering and GROUPing has been carried out on Landsat MSS data drawn from 5 agricultural sites in Kansas and Texas. For each site, training fields were selected at random and then divided into the two categories "wheat" and "other". CLUSTR was then run in a supervised mode to provide several signatures (clusters) for each category, and these signatures were used as input to GROUP. The statistics produced by GROUP as the number of signatures was reduced to one per category were displayed in digital plots such as those in Figures IV-1 through IV-6.

The first four figures typify the plots of maximum determinant, maximum trace, average pairwise p. of m. and this last measure multiplied by one-half the number of signatures. These measures tend to behave as expected, decreasing rapidly at first as the number of signatures increases and then flattening out. The typical backward slant of the curve for pairwise p. of m. times factor (Figure IV-4) probably indicates that the factor overcompensates in its task of making pairwise p. of m. a better estimate of the overall p. of m. Possibly a factor half as large would be a good compromise between the two bounds.

The observed p. of m. on occasion follows the pattern of the other measures (Figure IV-5) but when the number of points misclassified is small, the observed p. of m. jumps about randomly. Figure IV-6 shows a case where a maximum of 8 points were misclassified. These misclassified

points may reflect the unpredictable behavior of clusters too small to be accepted by GROUP or weakness in the original definition of the clusters.

IV.4 CONCLUSIONS

Starting with either field-by-field signatures or clusters, the question of how many and which signatures to use is often decided by guesswork. The GROUP procedure attempts to solve this problem by providing the analyst with the most likely sets of combined signatures and the information needed to choose from among them.

The rule used by GROUP in choosing which signatures to combine is constructed according to two principles: first, signatures chosen to be combined should be as close to each other as possible; second, the combining of these signatures should keep the probability of misclassification between categories as small as possible. GROUP then provides the analyst with sufficient information about its combining activities to allow him to choose from among the sets of signatures the one set which he believes represents the best compromise between cost and classification accuracy.

The GROUP procedure may also be used for investigating both practical and theoretical questions. Some of the investigations which might profitably employ GROUP include the relationship between theoretical and empirical measures of the probability of misclassification, the robustness of various schemes for signature selection; and the number of signatures normally needed to maintain accurate classification.

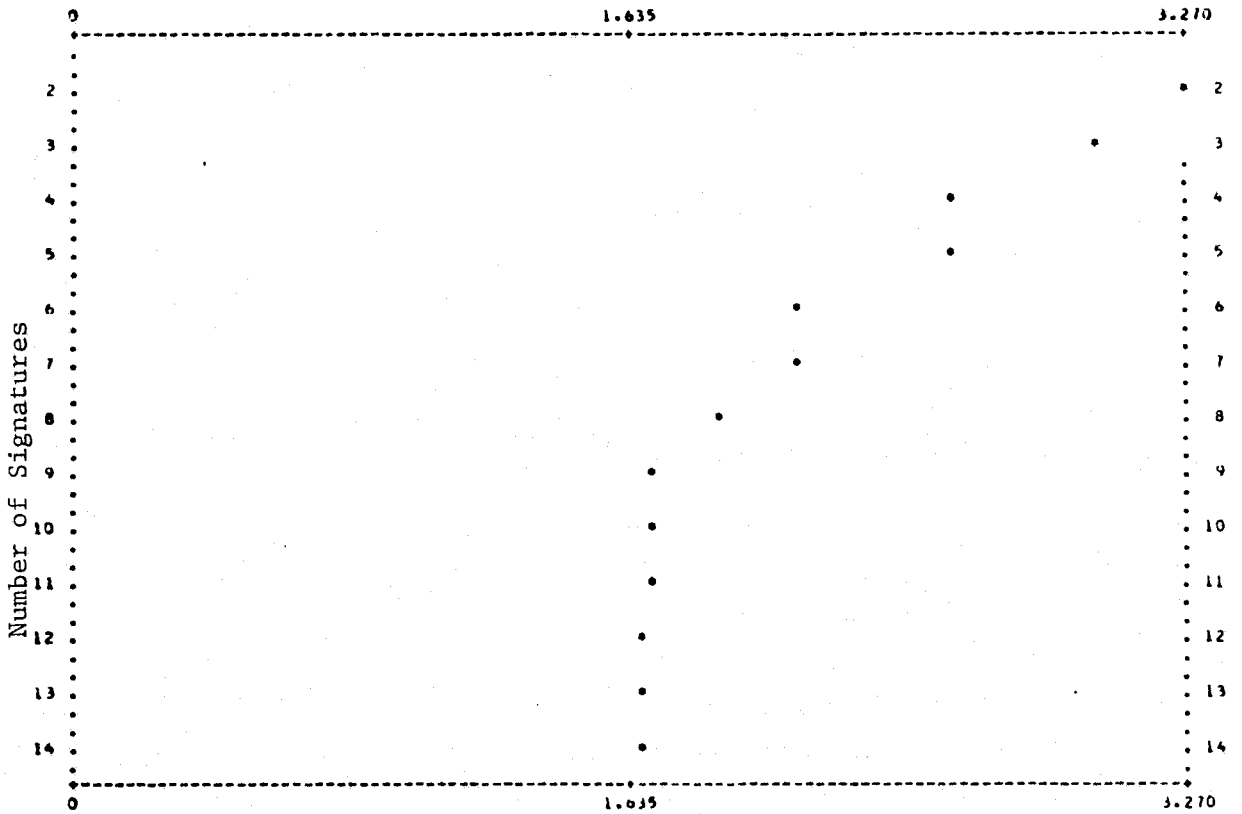

FIGURE IV-1. MAXIMUM DETERMINANT (SALINE SITE)

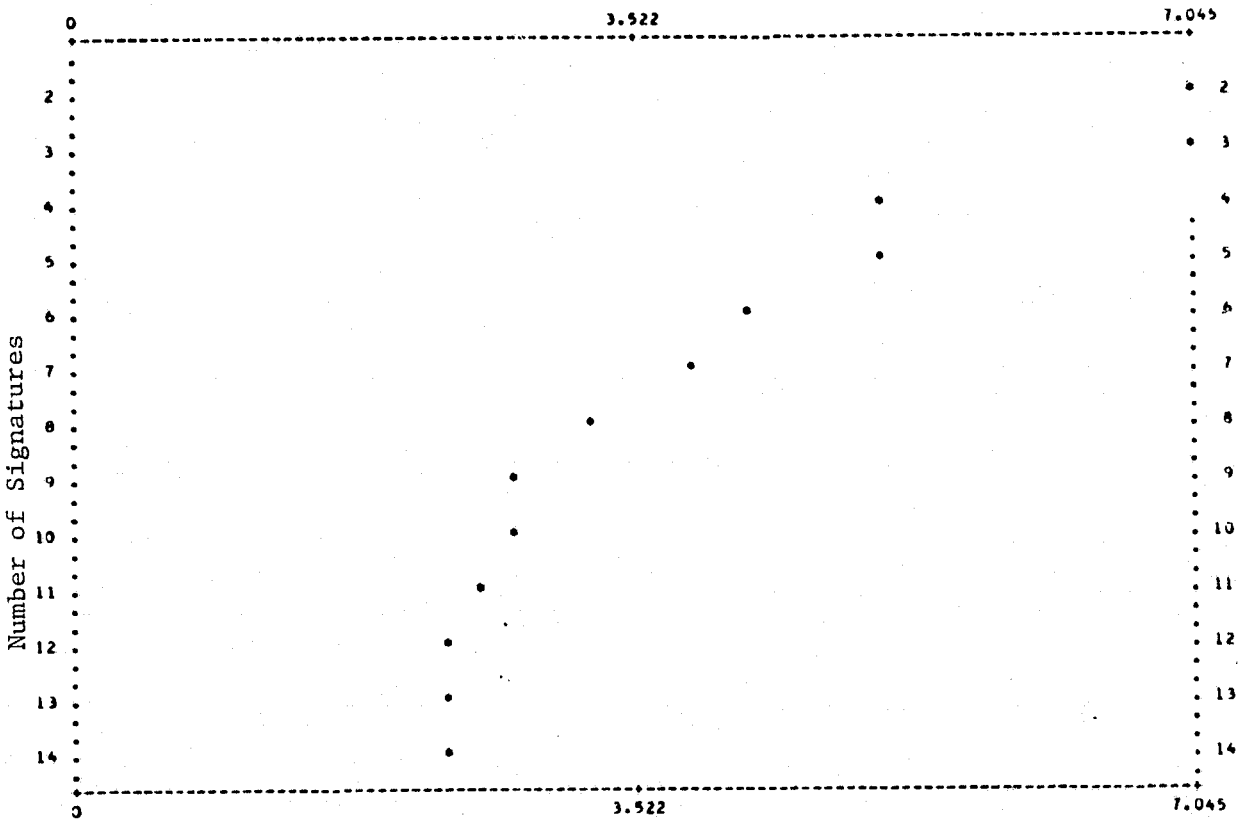

FIGURE IV-2. MAXIMUM TRACE (SALINE SITE)

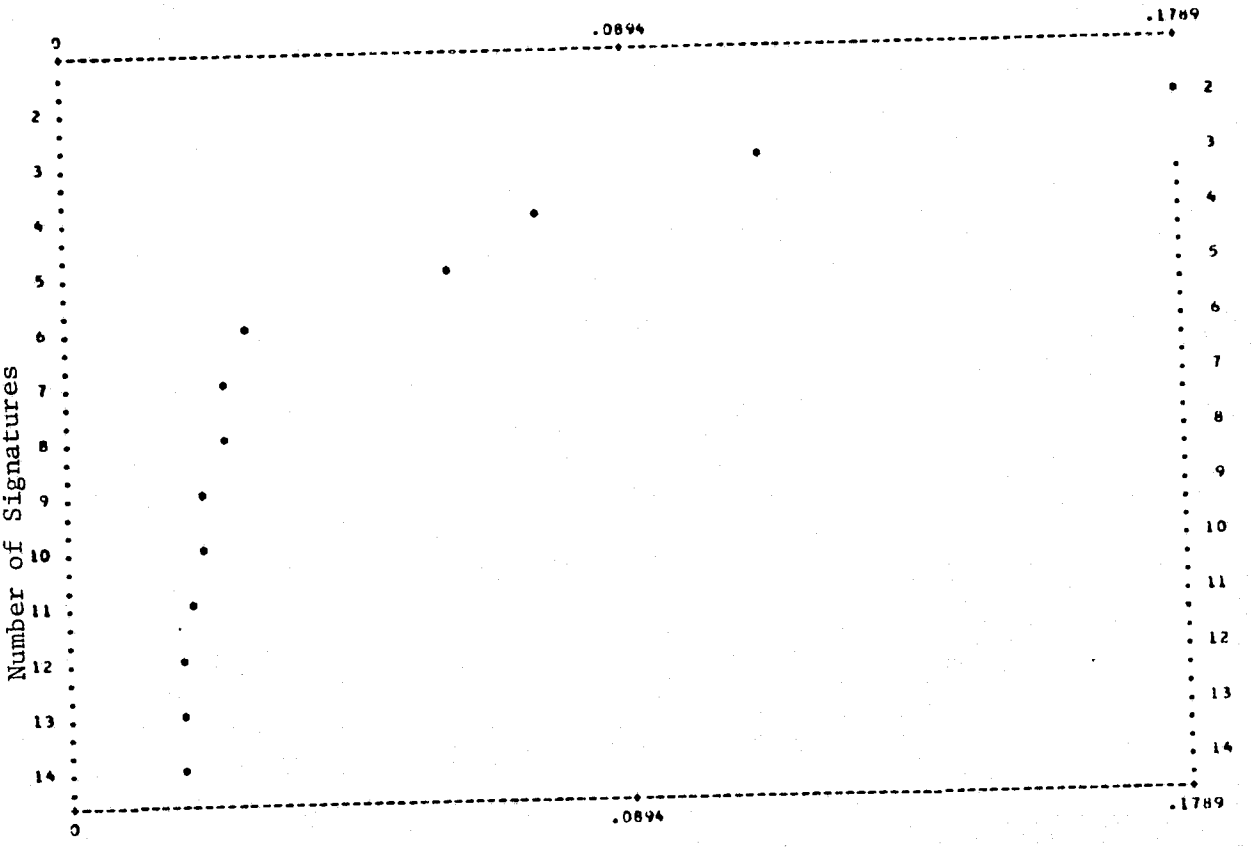

FIGURE IV-3. AVERAGE PAIRWISE PROBABILITY OF MISCLASSIFICATION (SALINE SITE)

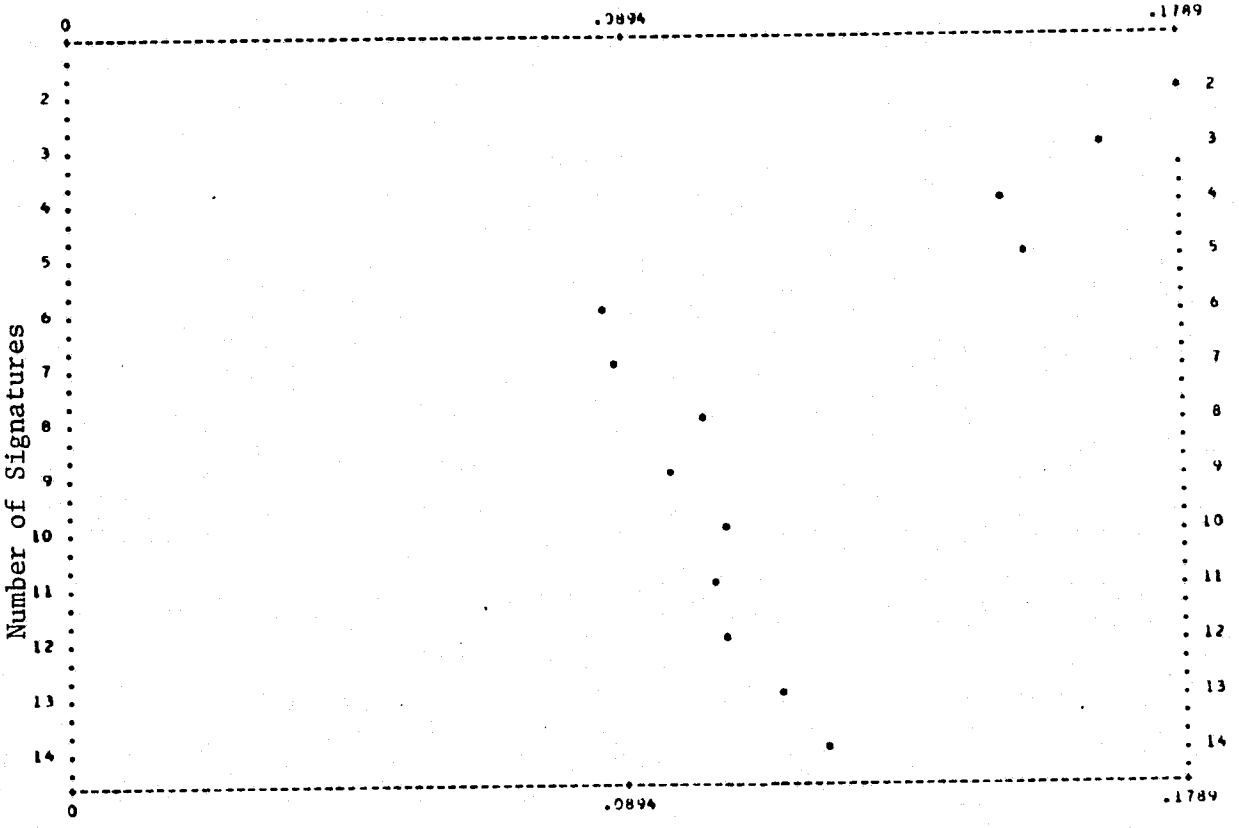

FIGURE IV-4. PAIRWISE PROBABILITY OF MISCLASSIFICATION TIMES FACTOR (SALINE SITE)

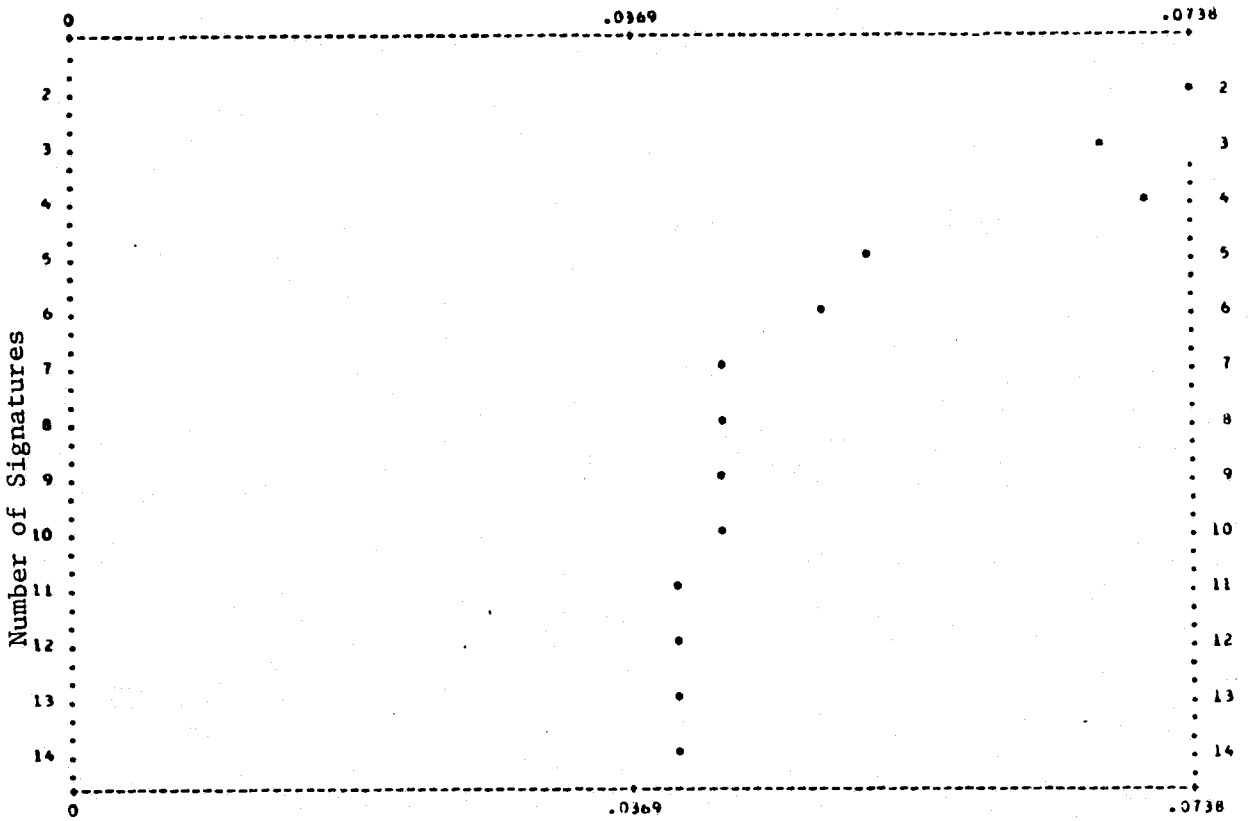

FIGURE IV-5. OBSERVED PROBABILITY OF MISCLASSIFICATION (SALINE SITE)

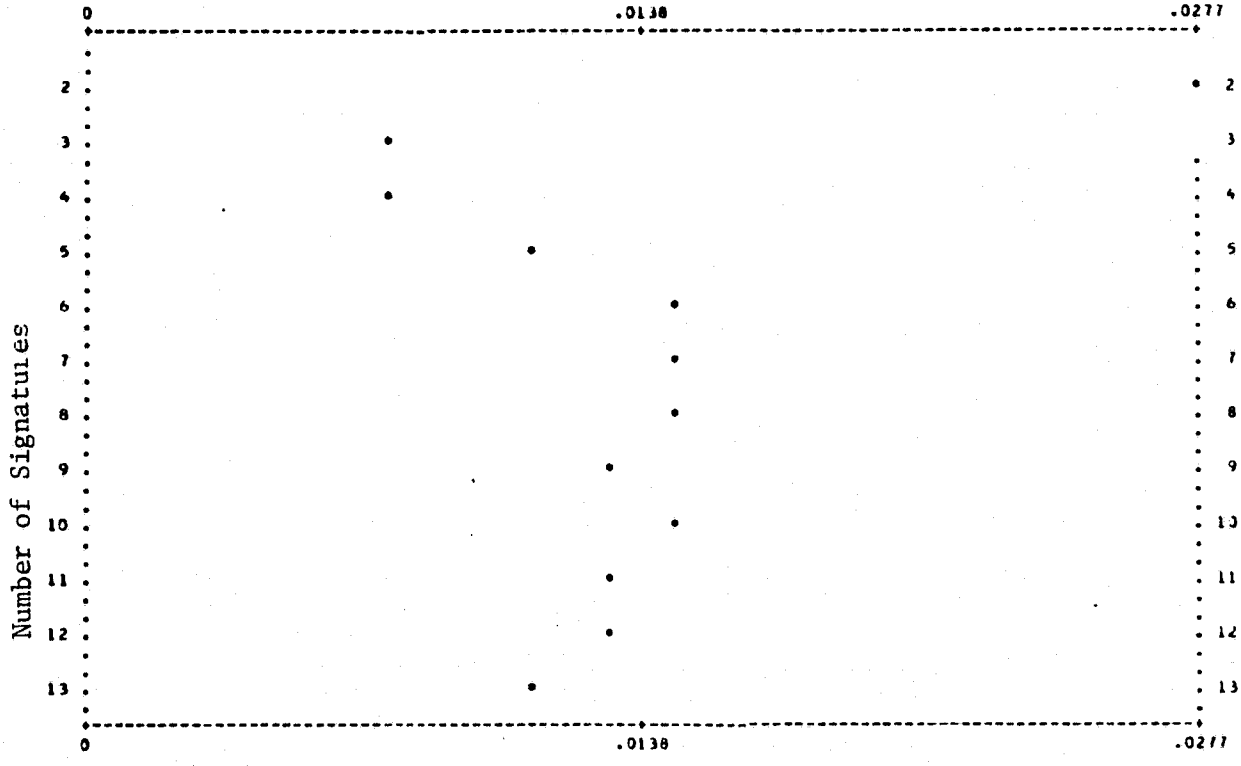

FIGURE IV-6. OBSERVED PROBABILITY OF MISCLASSIFICATION (FINNEY SITE)

APPENDIX V

BAYESIAN FORMULATION OF A TWO-AT-A-TIME MIXTURE ALGORITHM

W. Richardson, R. Kauth and A. Pentland

V.1 INTRODUCTION

The algorithm LIMMIX [5] for processing multispectral scanner data decides that a pixel represents a pure signature if the chi-square value χ_p^2 of the winning signature is $\leq$ a constant χ_1^2. Otherwise it considers all two-way mixtures of signatures and computes the proportion estimate λ and the chi-square value χ_m^2 of the winning mixture. If $\chi_p^2 \leq \chi_m^2$ and $\chi_p^2 \leq \chi_2^2$, then again it is decided that the pixel represents a pure signature. If $\chi_m^2 < \chi_p^2$ and $\chi_m^2 \leq \chi_2^2$, it is decided that the pixel represents the winning mixture with proportion λ. If all of these conditions fail, it is decided that the pixel represents an alien object.

The LIMMIX procedure is arbitrary in some respects. When $\chi_p^2 \leq \chi_1^2$, all possibility that the data point might be a mixture is ruled out, yet there is no reason why such mixtures might not occur. Similarly, when $\chi_p^2 > \chi_1^2$, mixtures are favored except in the event that the best mixture has a proportion estimate of one. To replace the element of arbitrariness by decision-theoretic principles, we propose two procedures that define a density for each two-way mixture and then choose among the pure and mixed densities by a Bayesian rule, i.e., weighted maximum likelihood.

The plan for defining a two-way mixture density is:

1. assume that the two materials to be mixed have the same covariance matrix which is estimated by the average of the two given covariance matrices.
2. make a transformation of the means and the data point reducing the common covariance matrix to the identity.
3. define the mixture density in transformed space.

4. divide this density by a constant to transform it back to the original space.

Specifically, suppose we are defining a density for the mixture of two signatures with means A and B and covariance matrices R_A and R_B. Let R be $\frac{1}{2}(R_A + R_B)$. The density in the original space is

$$f(x) = \frac{K}{\sqrt{|R|}} e^{-\frac{1}{2}(x-\mu)^T R^{-1}(x-\mu)}$$

where μ = A or B. Let $R^{-1} = C^T C$. Let $y = CX$. It is easily shown that the covariance matrix of y is CRC^T which = I, the identity matrix. The expected value of $y = C\mu$ which we call μ'. The density of y is

$$g(y) = K\, e^{-\frac{1}{2}(y-\mu')^T(y-\mu')}$$

The χ^2 value is the same in both spaces:

$$(y-\mu')^T(y-\mu') = (x-\mu)^T R^{-1}(x-\mu)$$

but the densities differ by a constant

$$f(x) = g(y)/\sqrt{|R|}$$

V.2 HOW LIMMIX DEFINES PURE AND MIXTURE DENSITIES

LIMMIX follows the above general plan for creating mixture densities. Specifically, LIMMIX finds the point z on the line segment between the transformed means (hereafter called "the segment") nearest to the transformed data point y. The estimate of the proportions of the mixture are the proportions into which z divides the segment. The multivariate normal density g(y) is then computed with $\mu' = z$ and divided by $\sqrt{|R|}$ to transform it back to the original space. The two-way mixture with the largest such density is then selected. Actually, $-2 \ln f(x)$ is computed rather than f(x) and the smallest of these values

is chosen. The density in this form comes out $\chi_m^2 + \ln |R|$ where χ_m^2 is the squared distance $\Sigma(y_i - z_i)^2$ from y to z. An analogous value χ_p^2 is calculated for the pure signatures:

$$\chi_p^2 = (x - \nu)^T S^{-1}(x - \nu)$$

where ν is the signature mean and S is its covariance matrix, and the pure signature with the largest density, i.e., the smallest $\chi_p^2 + \ln |S|$ is chosen. The rule for deciding between pure and mixed densities was given in the previous section.

V.3 LIMMIX B, -- A NEW TWO-WAY MIXTURE ALGORITHM

The mixture density used in LIMMIX is conditioned upon the mean being at a certain point z. A mixture density which could be compared to the pure densities would be conditional only on the fact that a pixel represents a mixture of two materials A and B, regardless of proportion. It would be defined for each data point x and integrate to 1 over the data space. The first of our two proposals for defining such a mixture density is to make the LIMMIX density integrate to 1 by dividing by a constant v, which is, in fact, the integral of the present LIMMIX quasi-density (f(x), defined earlier) over the whole space. We will call this procedure LIMMIX B.

We will now calculate v. The quasi-density g(y) was defined by supposing that z, the point on the segment nearest to the transformed data point y, is the mean of a standard normal distribution. Divide the space into three regions by passing planes through the transformed means perpendicular to the segment. The volume of the two end regions adds to 1 because each is half a standard n-variate normal distribution, where n is the number of channels. To obtain the volume of the middle region, we integrate it on a plane perpendicular to the segment. It is the integral of a standard n-variate normal distribution over n-1 dimensions. It is no loss of generality to assume the segment is in

the direction of the n^{th} axis because any rotation of the space will preserve the unit covariance matrices. Hence $(y_n - z_n)^2 = 0$.

Thus the integral we want is

$$\int_0^D \int_{\substack{n-1 \\ space}} \frac{1}{(2\pi)^{n/2}} e^{-\frac{1}{2}\sum_1^{n-1}(y_i - z_i)^2} dy_1 \ldots dy_{n-1}\, dy_n$$

$$= \int_0^D (2\pi)^{-\frac{1}{2}} \int_{\substack{n-1 \\ space}} \frac{1}{(2\pi)^{\frac{n-1}{2}}} e^{-\frac{1}{2}\sum_1^{n-1}(y_i - z_i)^2} dy_1 \ldots dy_{n-1}\, dy_n$$

The inner integral is 1 because it is the integral of an n-1 variate standard normal density over n-1 dimensions. Thus the volume of the cylinder is

$$\frac{D}{\sqrt{2\pi}}$$

where D is the length of the segment. Hence

$$v = 1 + \frac{D}{\sqrt{2\pi}}$$

To make the quasi-density in transformed space a real density that integrates to 1, we divide it by v. When we divide this density in transformed space by $\sqrt{|R|}$ we have a real density for the mixture in the original data space:

$$\frac{g(y)}{v\sqrt{|R|}} = \frac{f(x)}{v}$$

We now make a Bayesian decision among the pure and mixed densities, i.e., we give each density a prior weight and choose the density with

the greatest weighted maximum likelihood. We estimate a parameter m representing the proportion of mixed pixels in the scene. We assign the mixed densities together a weight of m and pure densities a weight of 1-m. We assume that each of the s pure densities is equally likely and each of the s(s - 1)/2 mixed densities is equally likely. Then each pure density has a prior weight of

$$\frac{1 - m}{s}$$

and each mixed density a weight of

$$\frac{m}{s(s - 1)/2}$$

So, in theory, we compute the pure densities {h} and the mixed quasi-densities {f} and choose the density corresponding to the biggest of

$$\left\{\frac{1 - m}{s} h\right\} \quad \text{and} \quad \left\{\frac{2m}{vs(s - 1)} f\right\}$$

i.e., the biggest of

$$\{h\} \quad \text{and} \quad \left\{\frac{2m}{v(1 - m)(s - 1)} f\right\}$$

i.e., the smallest of

$$\{-2 \ln h\} \quad \text{and} \quad \left\{-2 \ln \frac{2m}{v(1 - m)(s - 1)} - 2 \ln f\right\}.$$

Let

$$Q(D) = -2 \ln \frac{2m}{v(1 - m)(s - 1)}$$

$\{-2 \ln h\}$ is $\{\chi_p^2 + \ln|S|\}$ where χ_p^2 is the χ^2 value for the pure distribution. $\{-2 \ln f\}$ is $\{\chi_m^2 + \ln|R|\}$ where χ_m^2 is the χ^2 value for the mixed distribution. Hence, we choose the pure signature or mixture corresponding to the smallest among

$$\{\chi_p^2 + \ln|S|\} \text{ and } \{\chi_m^2 + \ln|R| + Q(D)\}$$

The programming differences between LIMMIX and LIMMIX B are minimal because LIMMIX computes and compares $\{\chi_p^2 + \ln|S|\}$ and $\{\chi_m^2 + \ln|R|\}$. LIMMIX adds Q(D) to $\chi_m^2 + \ln|R|$ and chooses the pure or mixed signature corresponding to the smaller of the two winning values. The only parameter to be estimated is m and it is relatively stable for similar scenes.

LIMMIX B computes D conveniently as follows. The transformed means are CA and CB. Hence,

$$\begin{aligned} D^2 &= (CB - CA)^T (CB - CA) \\ &= (B - A)^T C^T C(B - A) \\ &= (B - A)^T R^{-1} (B - A) \end{aligned}$$

Thus

$$D = \sqrt{(B - A)^T R^{-1} (B - A)}$$

the Mahalanobis distance between the means.

V.4 LIMMIX C -- A NEW BAYESIAN TWO-WAY MIXTURE ALGORITHM

Our first method, LIMMIX B, for defining a mixture density was suggested by previous mixture estimation practices. Our second method, which we will call LIMMIX C, is derived logically from a Bayesian assumption that the parameter α defining the mixture $(1 - \alpha)A + \alpha B$ has a rectangular distribution between 0 to 1.

The joint density of y and α in transformed space is

$$\begin{aligned} g(\alpha, y) &= K\, e^{-\frac{1}{2}\sum_{i=1}^{n} [y_i - (1 - \alpha)A_i - \alpha B_i]^2} && \text{if } 0 \leq \alpha \leq 1 \\ &= 0 && \text{otherwise} \end{aligned}$$

where A_i and B_i are now the coordinates of the transformed means.

To get the mixture density at y, call it g(y), we integrate out the α:

$$g(y) = K \int_0^1 e^{-\frac{1}{2}\sum_{i=1}^{n} [y_i - (1-\alpha)A_i - \alpha B_i]^2} \, d\alpha$$

This integral appears formidable in n-space but it can be simplified by rotating and translating the space so that A is at 0, B is at D on the y_1 axis (where D is the distance between A and B) and y is on the y_1, y_2 plane. The covariance matrices remain the identity matrix under this second transformation. To get the coordinates (y_1, y_2) of the new y, drop a perpendicular from the old y to the line A,B. Let the foot, z, of the perpendicular be represented by $(1 - \theta)A + \theta B$ and let the distance from the old y to z be χ. The coordinates of the new y are

$$y_1 = \theta D$$

$$y_2 = \chi$$

Now

$$g(y) = K \int_0^1 e^{-\frac{1}{2}[(y_1 - \alpha D)^2 + y_2^2]} \, d\alpha$$

where $K = (2\pi)^{-\frac{n}{2}}$. K can be omitted because it multiplies every density, pure and mixed.

Let

$$\beta = \alpha D - y_1 \qquad d\alpha = d\beta/D$$

$$g(y) = e^{-\frac{y_2^2}{2}} \frac{\sqrt{2\pi}}{D} \int_{-y_1}^{D-y_1} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\beta^2} d\beta$$

$$= e^{-\frac{\chi_m^2}{2}} \frac{\sqrt{2\pi}}{D} [\Phi((1 - \theta)D) - \Phi(-\theta D)]$$

where Φ is the cumulative normal integral.

Having defined the mixture density in transformed space, we now proceed as before to obtain the density in the original data space by dividing* by $\sqrt{|R|}$, to weight the pure and mixed densities by

$$\frac{1 - m}{s} \quad \text{and} \quad \frac{m}{s(s - 1)/2}$$

respectively, and to choose among the pure and mixed signatures the one with the largest weighted density.

If the winning density is mixed, we take as the estimate of the proportion of the mixture, not the maximum likelihood estimate of α as before, but the expected value $\hat{\alpha}$ of α given y.

$$\hat{\alpha} = \mathcal{E}(\alpha|y) = \int_0^1 \alpha\, g(\alpha|y) d\alpha$$

where $g(\alpha|y)$ is the density of α given y

---

* The second transformation that lined up A, B and y with the y_1 and y_2 axes was a rotation, i.e. an orthogonal transformation having a determinant of 1, and thus doesn't multiply the density by a factor.

$$g(\alpha|y) = \frac{g(\alpha,y)}{g(y)}$$

$$= \frac{K\, e^{-\frac{1}{2}\left[(y_1 - \alpha D)^2 + y_2^2\right]}}{K\, e^{-\frac{y_2^2}{2}} \frac{\sqrt{2\pi}}{D} \left[\Phi((1-\theta)D) - \Phi(-\theta D)\right]}$$

$$= \frac{e^{-\frac{1}{2}(y_1 - \alpha D)^2}}{\frac{\sqrt{2\pi}}{D} \left[\Phi((1-\theta)D) - \Phi(-\theta D)\right]}$$

$$\hat{\alpha} = \int_0^1 \alpha\, e^{-\frac{1}{2}(y_1 - \alpha D)^2} \, d\alpha / \text{denominator}$$

Let

$$\beta = \alpha D - y_1, \qquad d\alpha = d\beta/D, \qquad \alpha = (\beta + y_1)/D.$$

$$\hat{\alpha} = \frac{\int_{-y_1}^{D-y_1} \left(\frac{\beta + y_1}{D}\right) \frac{1}{D}\, e^{-\frac{1}{2}\beta^2} \, d\beta}{\frac{\sqrt{2}}{D} \left[\Phi((1-\theta)D - \Phi(-\theta D)\right]}$$

In the numerator, two terms can be integrated separately.

$$\hat{\alpha} = \frac{\frac{\sqrt{2\pi}}{D} \frac{y_1}{D} \int_{-y_1}^{D-y_1} \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}\beta^2} \, d\beta + \frac{\sqrt{2\pi}}{D^2} \int_{-y_1}^{D-y_1} \beta \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2}\beta^2} \, d\beta}{\frac{\sqrt{2\pi}}{D} \left[\Phi((1-\theta)D) - \Phi(-\theta D)\right]}$$

Now $\int x\,\phi(x)$ is $-\phi$, where ϕ is the normal integrand so

$$\hat{\alpha} = \frac{y_1}{D} + \frac{1}{D}\,\frac{-\phi(D - y_1) - -\phi(-y_1)}{\Phi((1 - \theta)D) - \Phi(-y_1)}$$

$$= \theta - \frac{1}{D}\,\frac{\phi((1 - \theta)D) - \phi(-\theta D)}{\Phi((1 - \theta)D) - \Phi(-\theta D)}$$

$\hat{\alpha}$ is a function of θ and D. Table V-1 gives some representative values. It can easily be shown that $\hat{\alpha}$ is a symmetrical function of θ in the sense that $\hat{\alpha}(1 - \theta) = 1 - \hat{\alpha}(\theta)$ by using the identities $\phi(x) = \phi(-x)$ and $\Phi(x) = 1 - \Phi(-x)$. It can be shown that $\hat{\alpha} \to 0$ as $\theta \to -\infty$ and $\hat{\alpha} \to 1$ as $\theta \to \infty$ by using the asymptotic relationships

$$1 - \Phi(x) \sim \frac{\phi(x)}{x}$$

$$1 - \Phi(x) \sim \frac{\phi(x)}{x}\left(1 - \frac{1}{x^2 + 3}\right)$$

which, as $x \to \infty$, have errors that go to zero like $1/x^2$ and $1/x^6$, respectively.

Although LIMMIX C appears to require lengthy computations for each pixel, the precalculation of two tables can speed it up almost to the pace of LIMMIX B. As with LIMMIX B, we write the density in chi square form by applying the operation -2 ln

$$-2 \ln g(y) = \chi_m^2 - 2 \ln \left\{\frac{\sqrt{2\pi}}{D}\,[\Phi(D - D\theta) - \Phi(-D\theta)]\right\}$$

and add on the terms

$$+ \ln |R| - 2 \ln \frac{2m}{(1 - m)(s - 1)}$$

To convert to the data space and include the prior weights.

TABLE V-1. THE LIMMIX C MIXTURE ESTIMATE $\hat{\alpha}$ AS A FUNCTION OF THE LIMMIX MIXTURE ESTIMATE θ AND THE DISTANCE D BETWEEN THE TRANSFORMED MEANS (THE TABLE IS SYMMETRICAL FOR $\theta > 0.5$)

	θ =	0.0	0.1	0.2	0.3	0.4	0.5
D =	1	0.46	0.47	0.48	0.48	0.49	0.50
	2	0.36	0.39	0.41	0.44	0.47	0.50
	3	0.26	0.30	0.34	0.39	0.45	0.50
	4	0.20	0.24	0.29	0.35	0.42	0.50
	5	0.16	0.20	0.26	0.33	0.41	0.50
	7	0.11	0.16	0.22	0.31	0.40	0.50
	10	0.08	0.13	0.21	0.30	0.40	0.50

The curly bracket term is a function only of D and θ. We can precompute a table of this term with $s(s-1)/2$ rows, one for each possible value of D, and 11 columns for $\theta = 0.0, 0.1, 0.2, \ldots, 1.0$. In applying the table, we defer to the pure signature, i.e. throw out the mixture, if $\theta \overset{<}{=} 0$ or $\overset{>}{=} 1$. This decision would have been made the slow way, too, unless m were unusually large. When $0 < \theta < 1$ we compute the second term by linear interpolation. The second table is of $\hat{\alpha}$ as a function of D and θ like Table V-1. Its construction and use is analogous.

V.5 COMPARISON OF TWO-WAY MIXTURE ALGORITHMS

When $0 < \theta < 1$, the densities defined by LIMMIX B and LIMMIX C are asymptotically equal as $D \to \infty$.

Proof: The densities for LIMMIX B and LIMMIX C are, respectively,

$$\frac{1}{1 + \frac{D}{\sqrt{2\pi}}} \, e^{-\chi_m^2/2}$$

and

$$\frac{\sqrt{2\pi}}{D} \; e^{-\chi_m{}^2/2} \; [\Phi(D - D\theta) - \Phi(-D\theta)]$$

Two quantities are asymptotic if their quotient → 1.

$$\frac{\text{LIMMIX C density}}{\text{LIMMIX B density}} = \left(1 + \frac{D}{\sqrt{2\pi}}\right) \frac{\sqrt{2\pi}}{D} \; [\Phi(D - D\theta) - \Phi(-D\theta)]$$

$$= \left(\frac{\sqrt{2\pi}}{D} + 1\right) [\Phi(D - D\theta) - \Phi(-D\theta)]$$

The first factor → 1 as $D \to \infty$. $\Phi(D - D\theta) \to 1$ as $D \to \infty$ because $\theta < 1$. $\Phi(-D\theta) \to 0$ as $D \to \infty$ because $\theta > 0$. Thus the second factor of the ratio of densities → 1 as $D \to \infty$. Q.E.D.

The ratio of the LIMMIX C to the LIMMIX B density → 0 if θ is < 0 or > 1 because the square bracket factor → 0 in that case. This case is not normally of practical significance because unless the prior estimate of the proportion m of mixture pixels is extremely high, the pure signature B will outweigh the (A,B) mixture signature $(1 - m)/s$ to $\frac{m}{s(s - 1)/2}$, and hence, will always prevail when $\theta > 1$. Similarly, the pure signature A will prevail when $\theta < 0$.

When $\theta = 1$ or 0, the LIMMIX C density is asymptotic to one half the LIMMIX B density, showing that LIMMIX C has a greater tendency to defer to pure signatures near the pure means than does LIMMIX B.

Of these two-at-a-time mixture algorithms we have described, LIMMIX C has the soundest theoretical justification because it rests on only two assumptions:

1. that a relatively stable proportion of the pixels in a scene are two-way mixtures
2. that among the mixture pixels, the mixture proportion α has a rectangular distribution between 0 and 1.

LIMMIX B is sounder theoretically than LIMMIX because it is asymptotic to LIMMIX C as $D \to \infty$ and because the mixture density it uses is a true density in the sense that its integral over the data space is 1. LIMMIX C takes a little longer to compute than the other two.

APPENDIX VI

TRAINING THE PARAMETERS OF THE LIMMIX PROCEDURES

VI.1 INTRODUCTION

A training procedure is useful if it is objective and efficient. Toward this goal, the following procedures for training mixture algorithm parameters are directed.

VI.2 TRAINING LIMMIX PARAMETERS

For LIMMIX, the value of χ_2^2 can be set one of these ways:

1. From a table of χ^2 distribution, we can find a value (such as 18.465 for four channels) which contains 99.9% of all pixels belonging to the distribution in question and set χ_2^2 equal to this value.
2. Experience in looking at maps of processed multispectral scanner data using different rejection thresholds (i.e., χ^2 cutoff levels like χ_2^2) may indicate that a higher value of χ_2^2 (such as 30 for 4-channel data) is most likely to separate the alien pixels from the true members of the training distributions.
3. A value can be set for χ_2^2 that results in designating as alien a certain given percentage, such as 2% of the pixels.

Two of these methods could be combined, by, for example, setting χ_2^2 at 18.465 or the 2% point, whichever is higher.

χ_1^2 can be set to produce a desired percentage of mixture decisions such as the estimated percentage of mixture pixels in the scene. The latter number can be estimated by geometry from a distribution of field sizes and shapes or by using a program such as POLYGN at ERIM that counts the number of pixels that are within a polygon and at least a given distance from the boundary. One would expect such a percentage to remain relatively stable from scene to scene and one might, with practice, estimate it pretty closely at a glance.

The suggested method of setting χ_1^2 and method 3. of setting χ_2^2 can be carried out in one pass through the data by keeping three histograms:

1. of χ_p^2 for all pixels
2. of χ_p^2 for those pixels that have $\chi_p^2 \leq \chi_m^2$
3. of χ_m^2 for those pixels that have $\chi_p^2 > \chi_m^2$.

At the end of the run, the histograms are converted to relative frequencies by dividing by the number of pixels processed. The relative frequencies of the first histogram add to 1 because the histogram records every pixel processed. The same conclusion does not apply to the second and third histograms, but their relative frequencies, all put together, do add to 1. For each possible value of χ_2^2, we can compute the percentage of pixels that such a χ_2^2 would have made alien by adding the relative frequencies of the second distribution for intervals $> \chi_2^2$ to those of the third distribution for intervals $> \chi_2^2$. This can be done by the program and presented as a table showing the percent of alien pixels implied by each possible choice of χ_2^2. The percentage of pure decisions implied by a given choice of χ_1^2 can be found by adding the relative frequencies of the first distribution for all intervals $\leq \chi_1^2$ and adding to that the relative frequencies of the second distribution for intervals $> \chi_1^2$ but $\leq \chi_2^2$. The percentage of mixtures is one minus the sum of the percentage of pure and alien. One can thus find the value of χ_2^2 that will produce a desired proportion of alien decisions and a value of χ_1^2 that will produce a desired proportion of mixture decisions.

VI.3 TRAINING LIMMIX B AND LIMMIX C PARAMETERS

The parameter m of LIMMIX B, an estimate of the percentage of mixed pixels in the scene, is used to give proper weight to the collection of mixture densities. The percentage of mixture decisions made by LIMMIX B will not, in general, equal m because a Bayesian rule

follows the principle of minimizing expected loss rather than holding results to a fixed percentage.

To draw an analogy, an equally-weighted Bayesian decision between two densities, A and B, will not in general result in equal errors (i.e., the probability of A given B will not equal the probability of B given A) because the decision rule follows the principle that the sum of the two errors must be a minimum. If the principle is followed that the two errors are equal (a "minimax" rule), the weights will, in general, be unequal.

It is not clear whether it is better to set the parameter m equal to the estimated percentage M of mixed pixels and let the algorithm find as many mixtures as it will, or whether set m in such a way as to produce M percent mixed decisions. If the user chooses the latter course, he can find that value of m by compiling a histogram during one pass through the data. Let the constant term of LIMMIX B

$$Q(D) = -2 \ln \frac{2m}{v(1 - m)(s - 1)}$$

be written

$$-2 \ln \frac{2}{v(s - 1)} \quad -2 \ln \frac{m}{1 - m}$$

or W(D) + Y(m) for short. We histogram the value of Y(m) that would make the two sides equal, namely

$$\chi_p^2 + \ln|S| - \chi_m^2 - \ln|R| - W(D)$$

After the run, we put the histogram in the form of a cumulative percentage from the top down and find the percentile Y_o corresponding to the desired percentage M of mixtures. We then find the m for which

$$-2 \ln \frac{m}{1 - m} = Y_o$$

which comes out, after algebra,

$$m = \frac{1}{e^{Y_o/2} + 1}$$

The procedure for setting the parameter m in LIMMIX C is analogous. We isolate the m term, which is the same as in LIMMIX B, and histogram the value of that term that would make the two densities, in -2 ln form, equal. In other words, we histogram the difference of those densities with the $-2 \ln \frac{m}{1 - m}$ term missing. We then find the value of m from the histogram as before.

We have experimented with another way of modifying the LIMMIX B and C procedures to produce agreement between the percentage of mixture decisions and the percentage of mixture pixels in the scene, namely, to multiply the mixture χ^2 by a parameter γ. As before, it is not necessary to run the rule again and again with different values of γ until the desired percentage of mixture decisions is produced. We need only make one pass through the data keeping a histogram of the value of γ required to produce equality between the best mixture and best pure density. For LIMMIX B, this value of γ is the solution of

$$\gamma \chi_m^2 + \ln|R| + Q(D) = \chi_p^2 + \ln|S|$$

which is

$$\gamma = \frac{\chi_p^2 + \ln|S| - \ln|R| - Q(D)}{\chi_m^2}$$

At the end of the run we convert the histogram to a percentage distribution, cumulate it from the top down and set γ equal to the percentile corresponding to the desired percentage of mixture decisions. An analogous procedure applies to LIMMIX C.

VI.4 TRAINING NINE-POINT MIXTURE PARAMETERS

As a first step in training the Nine-Point-Mixture parameters, we made a histogram of the number of winning votes (i.e., QRULE decisions) among the 9 pixels surrounding and including the pixel being processed. Votes for all the wheat signatures were added together to make one wheat vote and similarly for other. The results for the 5 sites are given as Table VI-1.

TABLE VI-1. CUMULATIVE HISTOGRAM OF THE WINNING VOTES FOR WHEAT OR OTHER AMONG ALL 9-POINT NEIGHBORHOODS OF PIXELS IN FIVE SITES

	$\geq$ 6 votes	$\geq$ 7 votes	$\geq$ 8 votes	$\geq$ 9 votes
Ellis	89.8%	72.1	61.0	48.8
Deaf Smith	87.5	72.2	58.6	40.8
Randall	92.7	79.7	70.9	61.4
Finney	90.8	76.6	65.4	52.5
Saline	88.6	75.8	61.7	42.5

After looking at this table, we selected 8 as the number N_1 of votes required to make a consensus decision. Sixty percent of the pixels in each site had this majority which is about all the pixels within homogeneous areas that one would expect to find. Also, 8 is a good consensus because either the center pixel is among the 8 and well-imbedded within them or else it is an island among the 8 and probably incorrectly classified.

The number N_2 of votes that two signatures must get to arrive at a split-vote mixture decision we set at four. It cannot be more than four and if it is less, we would have the problem of what to do with three 3-vote totals. The inference that the center pixel is a mixture of two 3-vote signatures is weaker than the same inference for 4-vote signatures.

We set η_1^2 to a high number like 51 that would exclude no points truly associated with the training distributions but would screen out extraneous points.

η_2^2 can be set equal to the χ_1^2 of LIMMIX or it could be set more systematically by compiling a frequency distribution of χ_p^2 for all pixels with < 8 winning votes. A thorough job of setting this parameter and the one corresponding to χ_2^2 would require compiling three histograms as in LIMMIX. The issue is clouded by the likelihood that some of the 8-vote pixels represent a mixture of two signatures of the same category, making it difficult to estimate the percentage of mixture decisions.

C-2

REFERENCES

1. W. Richardson and J. M. Gleason, Multispectral Processing Based on Groups of Resolution Elements, Technical Report 109600-18-F, Environmental Research Institute of Michigan, Ann Arbor, Michigan, May 1975.

2. Large Area Crop Inventory Experiment, Classification and Mensuration Subsystem (CAMS) Requirements (Level III), Report Number LACIE-00200, Vol. II, NASA Lyndon B. Johnson Space Center, Houston, Texas, 16 December 1974, pp. 32-34.

3. R. F. Nalepka and P. D. Hyde, Estimating Crop Acreage From Space-Simulated Multispectral Scanner Data, Technical Report 31650-148-T, Environmental Research Institute of Michigan, Ann Arbor, Michigan, August 1973.

4. H. M. Horwitz, P. D. Hyde and W. Richardson, Improvements in Estimating Proportions of Objects from Multispectral Data, Technical Report 190100-25-T, Environmental Research Institute of Michigan, Ann Arbor, Michigan, April 1974.

5. H. M. Horwitz, J. T. Lewis and A. P. Pentland, Estimating Proportions of Objects from Multispectral Scanner Data, Technical Report No. 109600-13-F, Environmental Research Institute of Michigan, Ann Arbor, Michigan, May 1975, p. 98 ff.

6. W. A. Malila, R. C. Cicone and J. M. Gleason, Wheat Signature Modeling and Analysis for Improved Training Statistics, Technical Report 109600-66-F, Environmental Research Institute of Michigan, Ann Arbor, Michigan, May 1976.

7. R. B. Crane, W. Richardson, R. H. Hieber and W. A. Malila, A Study of Techniques for Processing Multispectral Scanner Data, Technical Report 31650-155-T, Environmental Research Institute of Michigan, Ann Arbor, Michigan, September 1973.

8. Proposed by M. Rassbach at a meeting on signature extension at NASA Lyndon B. Johnson Space Center, Houston, Texas, 13 March 1975.

9. R. B. Crane and J. F. Reyer, Adaptive Processing for LANDSAT Data, Technical Report 109600-14-F, Environmental Research Institute of Michigan, Ann Arbor, Michigan, May 1975.

10. W. A. Malila, D. P. Rice and R. C. Cicone, Final Report on the CITARS Effort, Environmental Research Institute of Michigan, Technical Report 109600-12-F, Environmental Research Institute of Michigan, April 1975.

11. W. A. Malila, R. B. Crane and W. Richardson, Discrimination Techniques Employing Both Reflective and Thermal Multispectral Signals, Technical Report 31650-75-T, Environmental Research Institute of Michigan, Ann Arbor, Michigan, 1973, p. 42, Eq. 9,

12. Charles Peters, William Coberly, The Numerical Evaluation of the Maximum Likelihood Estimate of Mixture Proportions, Annual Report, University of Texas at Dallas, JSC-09703, May 1975, pp. 83-93.