LARS Information Note 041773

Automatic Boundary and Sample Classification of Remotely Sensed Multispectral Data

by
R. L. Kettig and
D. A. Landgrebe

The Laboratory for Applications of Remote Sensing

Purdue University, West Lafayette, Indiana

1973

LARS Information Note 041773

AUTOMATIC BOUNDARY FINDING AND SAMPLE CLASSIFICATION OF REMOTELY SENSED MULTISPECTRAL DATA*

by

R. L. Kettig and D. A. Landgrebe

Introduction

Presently, automatic classification of multispectral data images is most commonly effected on a point-by-point basis, as if the data vectors from one resolution element to the next were uncorrelated. In other words, no use is made of the spatial information contained in the scene. This is a useful but suboptimal approach, since in a practical situation strong correlations are certain to exist. "Sample classification" is a name used to refer to those classification schemes in which points are classified in sets (samples) rather than individually. It is assumed that all the points in a set belong to the same class. This assumption is met in applications such as crop species identification, where each field contains just one crop. Before sample classification can be applied to the fields, however, the fields themselves must be located in the data image. This is the role of boundary finding.

* The research reported in this paper is supported by NASA Grant NGL 15-005-112. The authors are with the Laboratory for Applications of Remote Sensing and the Department of Electrical Engineering, Purdue University, W. Lafayette, Indiana 47907.

Background and Scope of the Problem

Wacker and Landgrebe [2] have investigated the potential for the application of sample classification techniques to the problem of crop species identification. Their results indicate that one can expect 5% to 10% better classification accuracy using a sample classifier as opposed to the maximum likelihood point classifier when there is moderate overlap of the class densities in the feature space. The degree of improvement is, of course, highly data-dependent. It is derived from the fact that members of a class tend to cluster spatially as well as spectrally. Or, in other words, the spatial autocorrelation function of the members of a class tends to have a width of many resolution elements.

So far, the question of how to best utilize this spatial information remains unanswered. The first problem is to find regions which are in some sense homogeneous within the data to which the sample classification technique can be applied. Wacker and Landgrebe [2] were able to define fields manually for the purposes of their investigation. This, of course, is unacceptable for general purpose use. A spatial boundary finding algorithm has been developed by Wacker and Landgrebe [3] to automate the process. It uses an unsupervised clustering technique based on the premise that many boundaries are too gradual to be detected by simply comparing neighboring points. Clustering also tends to produce better quality output than the well-known

spatial gradient methods [4], which are inherently "noisy". However, both techniques suffer from the inability to guarantee closed boundaries, which is a significant drawback to their use in sample classification. The IBM Houston Scientific Center, in cooperation with LARS, has developed a closed boundary finding algorithm [5], which assigns every point to some connected field and thereby eliminates the open boundary problem. This algorithm will be discussed more fully in a later section.

Once a closed field has been "found" in the data set, many methods are available for classification. For example, one could simply classify each point in the field individually using a maximum likelihood decision rule and poll the results to determine the field classification. [6] Or one could average the data values over the entire field to obtain an unbiased estimate of the mean vector and then merely classify this mean using a maximum likelihood decision rule. If the field has a sufficient number of points to estimate its probability density in feature space, then one of the more elaborate "minimum distance" decision rules [2] can be employed. The number of data points required can be relatively few if a parametric characterization of the probability density is used. Some of the results reported by Wacker and Landgrebe [2] were obtained in this manner by assuming a Gaussian density

and employing the Jeffreys-Matusita distance to compare field and class densities. This will be discussed in more detail in the next section.

The work reported herein is concerned with the consolidation of the concepts of boundary finding and sample classification toward the goal of automatic sample classification of an entire flightline. It is hypothesized that the results reported by Wacker and Landgrebe for the case of manually defined boundaries will carry over to the case where boundaries are detected by machine. The initial implementation and test of this idea has been accomplished and the results tend to support the hypothesis.

Approach

The closed boundary finding algorithm was selected to form the core of a classifier to be known as BOFIAC (Boundary Finding And Classification). It was chosen for its closed boundary property and its qualitatively good results. The main limitation of the algorithm at this point is processing time. However, no dedicated attempt has been made to streamline the algorithm since it is still at the research evaluation stage. The boundary finder builds fields from small groups of picture elements, called pixel groups, usually (2 elements) x (2 elements). A field begins with one pixel group and expands laterally and down the flightline absorbing more pixel groups until it reaches its natural boundaries. At the boundaries new fields

are begun which expand in the same manner until the entire flightline has been partitioned into homogeneous regions. The algorithm avoids the problem of gradual boundaries mentioned earlier by comparing a candidate pixel group to the entire field for which it is being considered for membership rather than just to a neighboring pixel group. This is accomplished via a "multiple-univariate t-test", which means that a pixel group must satisfy a univariate t-test in each channel in order to be admitted to the field. In addition the variance of the pixel group is tested to ensure that it is composed of a relatively homogeneous set of points. Other tests such as the F-test or Hotelling's multivariate T^2 test could be substituted or added, but these would require more computing time. For the type of data tested so far, the current boundary finder has given satisfactory results.

Once a field has been closed, a classification algorithm is called. This algorithm contains a maximum likelihood Gaussian classifier and a sample classifier. The maximum likelihood classifier is used only if the field contains an insufficient number of points to estimate its probability density in feature space. In that case only the mean vector is classified and the result is assumed to apply to all points in the field. The sample classifier that was chosen is the same one that was used by Wacker and Landgrebe. It was selected for its relative ease of implementation and to provide results comparable to those of Wacker and Landgrebe.

Another subroutine is used to evaluate the classification accuracy. The user specifies certain rectangular test fields within the data set and the class to which each field belongs. The evaluation subroutine examines every pixel group in every test field comparing the classification with the actual class and tabulating the results.

BOFIAC produces a classification map on the line printer as well as tabulated results. The edge points of each field are printed with a user-designated symbol representing the class into which that field has been classified. Internal points are printed as blanks so that the boundary structure will be readily apparent. If two adjacent fields have been assigned to the same class, then the boundary between them is not printed so that they are effectively merged into one field. This results in a cleaner classification map.

Results

To date, BOFIAC has been tested on two flightlines. The data has 12 spectral bands (channels) and was collected by the University of Michigan Scanner. Corn Blight Watch Flightline 210 was overflown at about noon on August 13, 1971 from an altitude of 5,000 feet. The area covered was a 1.4-mile-by-9.7-mile strip of farmland. The classes considered were corn, forage, soybeans, forest and water. A subset of 3 of the 12 available spectral bands was used in the analysis, namely 0.61-0.70μm, 0.72-0.92μm, and 9.30-11.70μm. This subset was chosen because the minimum transformed divergence between

any two training classes was larger for this set than for any other combination of three channels. The transformed divergence is a measure of class separability in feature space [7], but maximizing it will not necessarily guarantee the highest classification accuracy. In other words, some other subset of three channels may give better performance than the ones used here.

Table 1 shows the results obtained using the maximum likelihood Gaussian classifier when the fields used to train the classifier were also used as test fields. One can usually expect high classification accuracy under these conditions, but the 99.7% figure attained here is unusually high. This indicates very little overlap of the training class densities in the feature space. If the training class samples are typical of the classes that they represent (as they should be), then both classifiers should perform well when fields other than training fields are used for test fields. Tables 2 and 3 show that this is indeed the case. The error rate for the maximum likelihood classifier is only 3.6%. And, as expected, BOFIAC reduced this rate significantly to 1.4%.

The other data set on which BOFIAC has been tested is Purdue flightline C1, which was overflown at about noon on June 28, 1966 from an altitude of 2600 feet. The area covered was a 1-mile-by-4.4-mile strip of farmland. A more complex classification was carried out on this flightline involving eight classes as follows: soybeans, corn, oats, wheat,

red clover, alfalfa, rye and bare soil. The spectral bands 0.40-0.44μm, 0.66-0.72μm, and 0.80-1.00μm were selected on the basis of the transformed divergence, as in the previous example. Table 4 shows the results obtained using the maximum likelihood classifier when the fields used to train the classifier were also used as test fields. The 95.6% performance indicates somewhat greater overlap of the class densities in feature space than in the previous example. Tables 5 and 6 show the test field performance of the maximum likelihood classifier and BOFIAC respectively when fields other than training fields were used for test fields. BOFIAC again cut the error rate significantly (from 8.8% to 3.6%) over that of the point classifier.

In conclusion, the consolidation of the concepts of boundary finding and sample classification into one algorithm has been achieved. BOFIAC is producing the type of results that one would expect for the type of data that has been used, and it appears that there is real justification for more extensive testing and refinement of the procedure.

Table 1. Training Field Performance of "Point" Classifier

Group	Number of Samples	Percentage Correct	Number of samples classified into: CORN	FORAGE	SOYBEANS	FOREST	WATER
Corn	379	100.0	379	0	0	0	0
Forage	167	100.0	0	167	0	0	0
Soybeans	793	99.4	1	2	788	2	0
Forest	103	100.0	0	0	0	103	0
Water	30	100.0	0	0	0	0	30
TOTAL	1472		380	169	788	105	30

overall performance (1467/1472) = 99.7%

Table 2. Test Field Performance of "Point" Classifier

Group	Number of Samples	Percentage Correct	Number of samples classified into: CORN	FORAGE	SOYBEANS	FOREST	WATER
Corn	1799	95.4	1716	66	12	5	0
Forage	2014	97.6	32	1965	0	17	0
Soybeans	1920	96.2	21	40	1848	11	0
Forest	1439	96.4	6	17	28	1387	1
Water	81	97.5	0	2	0	0	79
TOTAL	7253		1775	2090	1888	1420	80

overall performance (6995/7253) = 96.4%

Table 3. Test Field Performance of BOFIAC

Group	Number of Pixel Groups	Percentage Correct	Number of Pixel Groups Classified into: CORN	FORAGE	SOYBEANS	FOREST	WATER
Corn	1406	99.0	1392	4	10	0	0
Forage	1539	97.5	39	1500	0	0	0
Soybeans	1572	98.6	4	10	1550	8	0
Forest	1140	99.6	1	2	0	1136	1
Water	38	100.0	0	0	0	0	38
TOTAL	5695		1436	1516	1560	1144	39

overall performance (5616/5695) = 98.6%

Table 4. Training Field Performance of the "Point" Classifier

Group	Number of Samples	Percentage Correct	Number of Samples Classified into: Soybeans	Corn	Oats	Wheat	Red Clover	Alfalfa	Rye	Bare Soil
Soybeans	426	98.1	418	5	0	0	0	0	0	3
Corn	423	96.5	15	408	0	0	0	0	0	0
Oats	423	93.4	0	1	395	1	13	0	13	0
Wheat	697	96.4	0	0	2	672	0	0	23	0
Red Clover	423	96.5	0	1	3	0	408	11	0	0
Alfalfa	259	95.0	0	0	2	0	11	246	0	0
Rye	330	89.1	0	0	15	21	0	0	294	0
Bare Soil	190	100.0	0	0	0	0	0	0	0	190
TOTAL	3171		433	415	417	694	432	257	330	193

overall performance (3031/3171) = 95.6%

Table 5. Test Field Performance of "Point" Classifier

Group	Number of Samples	Percentage Correct	Number of Samples Classified into: Soybeans	Corn	Oats	Wheat	Red Clover	Alfalfa	Rye	Bare Soil
Soybeans	7171	94.7	6788	178	120	17	2	2	25	39
Corn	2775	88.7	158	2462	21	1	130	3	0	0
Oats	1558	84.8	20	8	1321	23	119	17	50	0
Wheat	2641	97.3	0	0	17	2569	0	0	55	0
Red Clover	3236	85.8	13	70	174	4	2775	199	0	1
Alfalfa	912	83.2	3	15	61	0	74	759	0	0
Rye	621	90.0	0	0	22	40	0	0	559	0
Bare Soil	332	98.5	5	0	0	0	0	0	0	327
TOTAL	19,246		6987	2733	1736	2654	3100	980	689	367

overall performance (17560/19246) = 91.2%

Table 6. Test Field Performance of BOFIAC.

Group	Number of Pixel Groups	Percentage Correct	Number of Pixel Groups Classified into: Soybeans	Corn	Oats	Wheat	Red Clover	Alfalfa	Rye	Bare Soil
Soybeans	6436	96.2	6189	211	6	0	3	0	5	22
Corn	2435	97.8	55	2380	0	0	0	0	0	0
Oats	1340	96.5	7	0	1293	0	40	0	0	0
Wheat	2250	97.5	1	0	1	2193	0	0	55	0
Red Clover	2812	94.3	9	18	80	0	2650	51	4	0
Alfalfa	740	96.0	1	1	14	0	14	710	0	0
Rye	572	98.7	0	0	0	8	0	0	564	0
Bare Soil	262	100.0	0	0	0	0	0	0	0	262
TOTAL	16,847		6262	2610	1394	2201	2707	761	628	284

overall performance (16241/16847) = 96.4%

REFERENCES

1. Wacker, A.G., "The Minimum Distance Approach to Classification", Technical Report No. TR-EE71-37, School of Electrical Engineering, Purdue University, Lafayette, Indiana 47906, October 1971. Also LARS Information Note 100771, Laboratory for Applications of Remote Sensing, West Lafayette, Indiana 47906, October 1971.

2. Wacker, A.G. and D.A. Landgrebe, "Minimum Distance Classification in Remote Sensing", LARS Print 030772, Laboratory for Applications of Remote Sensing, West Lafayette, Indiana 47906, March 1972.

3. Wacker, A.G., and D.A. Landgrebe, "Boundaries in Multispectral Imagery by Clustering", 1970 IEEE Symposium on Adaptive Processes (9th) Decision and Control, pp. X14.1 - X14.8, December, 1970. Also LARS Information Note 122969, Laboratory for Applications of Remote Sensing, West Lafayette, Indiana 47906, December, 1969.

4. Anuta, P.E., "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques", IEEE Transactions on Geoscience Electronics", Vol. GE-8, Number 4, October, 1970, pp. 353-368.

5. Anuta, P.E., E.M. Rodd, R.E. Jensen, and P.R. Tobias, "Final Report for the LARS/Purdue-IBM Houston Scientific Center Joint Study Program", The Laboratory for Applications of Remote Sensing, Purdue University, Lafayette, Indiana.

6. Huang, T., "Per Field Classifier for Agricultural Applications", LARS Information Note 060569, Laboratory for Applications of Remote Sensing, West Lafayette Indiana 47906, June, 1969.

7. Swain, P.H., T.V. Robertson, A.G. Wacker, "Comparison of the Divergence and B-Distance in Feature Selection", LARS Information Note 020871, Laboratory for Applications of Remote Sensing, West Lafayette, Indiana 47906, February, 1971.

8. Rodd, E.M., "Closed Boundary Field Selection in Multispectral Digital Images", IBM Publication No. 320.2420, IBM Houston Scientific Center, January 14, 1972.

9. Ostle, B., "Statistics in Research", Iowa State University Press, Ames, Iowa , 1963.

Appendix I - Input Data and Output Classification Maps for Flightlines 210 and C1.

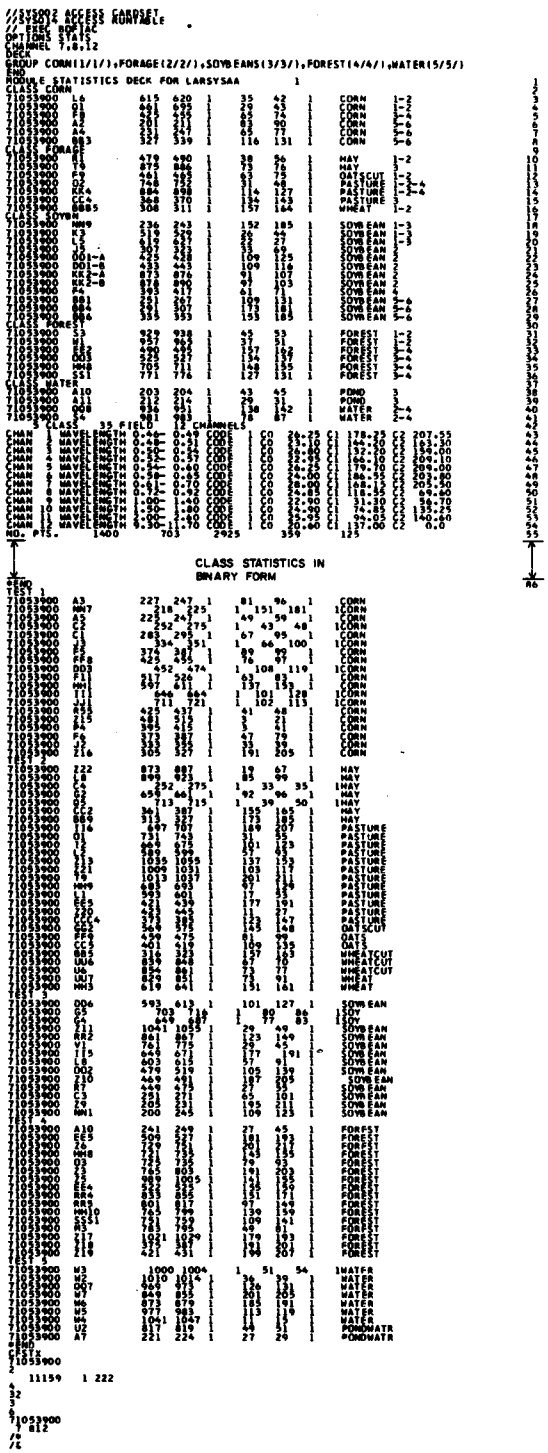


Figure I.A. Input Data for Processing Flightline 210.

0.61 - 0.70µm

0.72 - 0.92µm

9.30 - 11.70µm

Figure I.B. Gray Scale Printouts of Flightline 210.

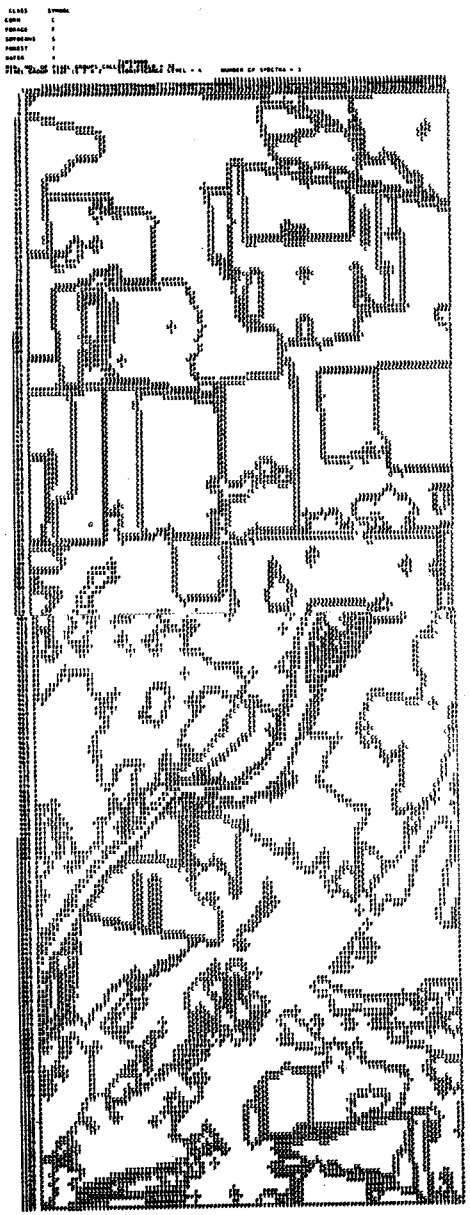


Figure I.C. Classification Map for Flightline 210.

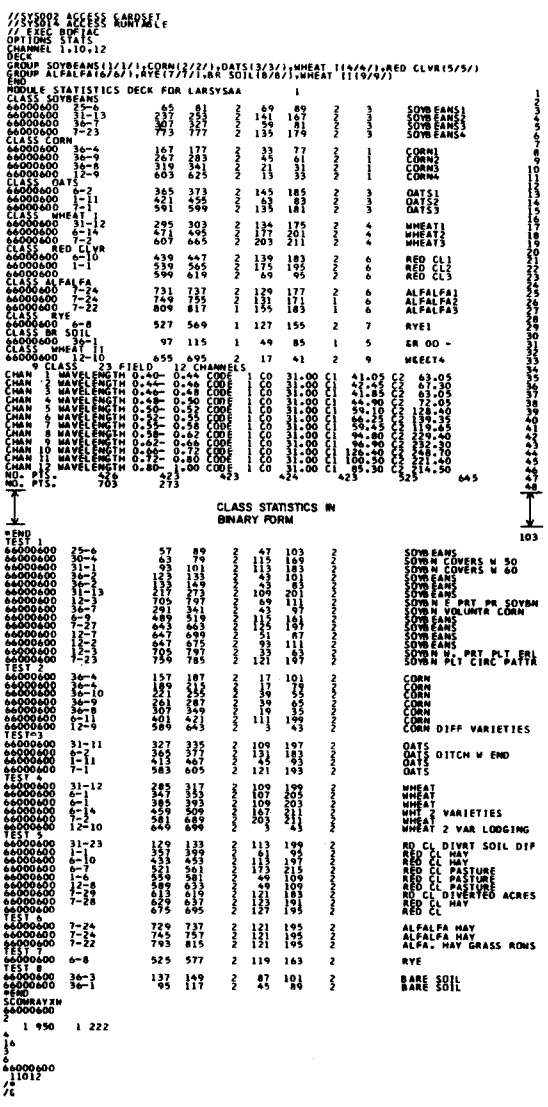
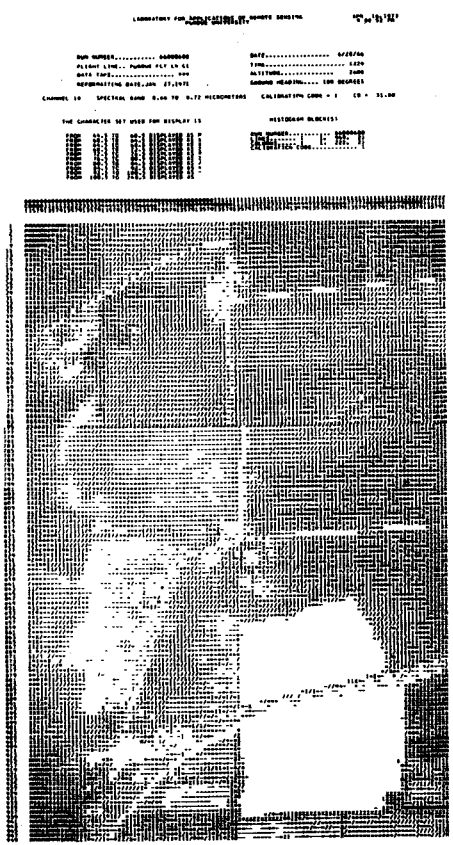
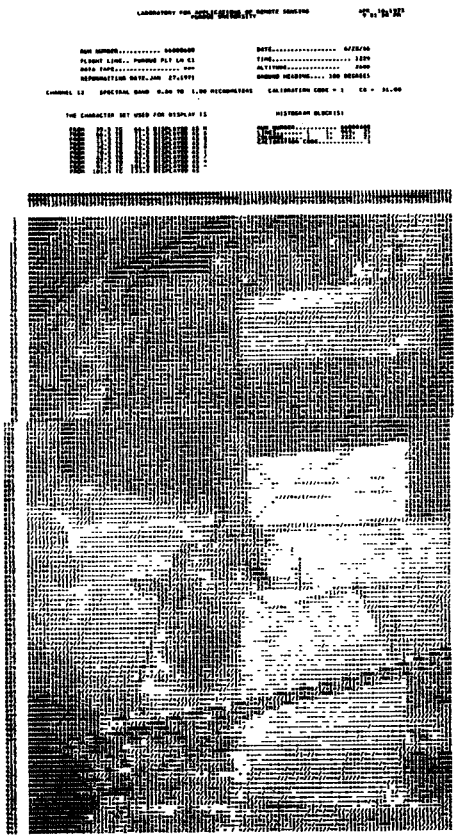


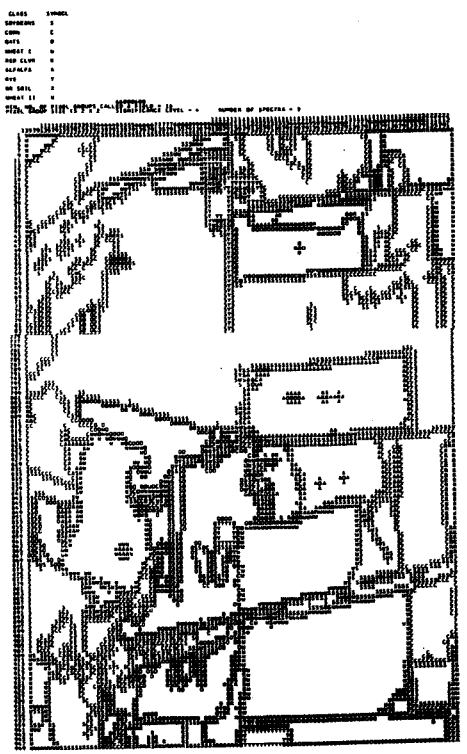
Figure I.D. Input Data for Processing Flightline C1.



0.66 - 0.72µm
(a)



0.80 - 1.00µm
(b)



Classification map
(c)

Figures I.E. (a and b) Gray Scale Printouts of Flightline C1.
(c) Classification map for Flightline C1.

Appendix II - Program Description (refer to program listing in Appendix III)

BOFIAC is organized around a "supervisor" routine which is modeled after the PERFIELD supervisor (SUPER6) found in the LARSYS data processing system at LARS. Subroutines SPACE, REDIF6, SETUP6, and PRCLS1 are system subroutines called by both supervisors to perform some initial tasks such as: reading in data cards, setting up dynamic storage allocation arrays, and computing certain useful quantities. LEARN2 is just another subroutine of this type. LEARN2 is followed by the call to CELLS, which is the boundary finding algorithm. A brief description of CELLS has been given in the main body of this report. A long and highly detailed description has been extracted from reference 8 and can be found in reference 5. In the interest of brevity, only the section on statistical method will be reproduced here because it is fairly important and reasonably brief. The reader interested in the geometry of field building can refer to either of the references or to the subroutine listing itself (Appendix III), which is well documented with "COMMENT" cards.

The Statistical Method Used in Boundary Finding (adapted from reference 5)

In this discussion, each point of a digital picture is called a pixel (picture element), and a small group of points is called a pixel group. A field is a collection of connected pixel groups which have been found to be statistically

similar. The t-test [9] is used to compare a pixel group, which has not yet been assigned to a field, with a neighboring field. Three assumptions are implicit: (1) the populations of all fields are normal, (2) the populations of all fields have the same variance, and (3) the spectral components are uncorrelated from channel to channel. When these assumptions are violated, the t-test may be sub-optimal, but it often gives completely satisfactory results. For example, when flightline 210 was processed by BOFIAC using channels 7, 8, and 12, the average magnitude of the inter-channel correlation coefficient (averaged over all classes and all pairs of channels) was 0.53. Yet the test field classification accuracy was 98.6%.

Call the unassigned pixel group "Sample 1" and the field "Sample 2". The goal is to compute a value of t in each channel to compare against a critical value to determine if the two samples are statistically similar (i.e. to determine if the pixel group is part of the field). The "critical value" is determined from a look-up table based upon the number of points in each sample (n_1 and n_2) and a significance level specified by the user. A two-tailed test is used with the number of degrees of freedom equal to $(n_1-1)+(n_2-1)$.

Consider any particular single channel, and let:

X_{ij} be the data value of pixel i in sample j

$$S_j = \sum_{i=1}^{n_j} X_{ij}$$

$M_j = \frac{1}{n_j}(S_j)$ = mean value of sample j

$$Q_j = \sum_{i=1}^{n_j} (X_{ij})^2$$

$$V_j = \sum_{i=1}^{n_j} (X_{ij}-M_j)^2 = Q_j - n_j(M_j)^2 = Q_j - \frac{1}{n_j} S_j^2$$

$$V = \frac{V_1 + V_2}{(n_1-1)+(n_2-1)}$$ = pooled estimate of variance

Then t is defined by : $$t = \frac{M_1 - M_2}{\sqrt{V(\frac{1}{n_1} + \frac{1}{n_2})}}$$

Note that if the decision is made to incorporate sample 1 into sample 2, then it is a simple matter to update the statistics of sample 2 as follows:

$n_2' = n_2 + n_1$

$S_2' = S_2 + S_1$

$M_2' = \frac{1}{n_2'} S_2'$

$Q_2' = Q_2 + Q_1$

$V_2' = Q_2' - n_2'(M_2')^2$

where primes are used to denote the updated statistics.

In addition to passing the t-test in all channels, the following must be true in all channels before sample 1 is added to sample 2: $\sqrt{n_1 V_1} < .15\ S_1$ and $\sqrt{n_2 V_2} < .15\ S_2$. This ensures that every pixel group added to the field is itself a relatively homogeneous set of points. The constant 0.15 was derived empirically.

The Classification Algorithm

The classifier (CLASS1) is called from within CELLS everytime a field is closed. If the number of points in the field falls below a user specified threshold, a maximum likelihood Gaussian classifier (subroutine MLGC) is used to classify the field as discussed in the main body of this report. Otherwise the following distance measure is computed for each class and the class is chosen for which this distance is minimum.

$$D = \frac{1}{2} \cdot \ell n \frac{\left|\frac{1}{2}\,(S_i^{-1}+S^{-1})\right|}{|S_i^{-1}S^{-1}|^{1/2}} - \frac{1}{4}\,B, \text{ where}$$

$$B=(S_i^{-1}\underline{M}_i+S^{-1}\underline{M})^t(S_i^{-1}+S^{-1})^{-1}(S_i^{-1}\underline{M}_i+S^{-1}\underline{M})-\underline{M}_i{}^tS_i^{-1}\underline{M}_i-\underline{M}^tS^{-1}\underline{M}$$

where $\underline{M}_i$, S_i and $\underline{M}$, S are the mean vector and covariance matrix for the ith class and for the field that is to be classified. This expression is equivalent to the Bhattacharyya distance [2] which is closely related to the Jeffreys-Matusita distance. It is of interest to note that the Bhattacharyya distance can also be expressed in the following form which is simpler and faster to compute:

$$D = \frac{1}{2}\,\ell n \frac{\left|\frac{1}{2}\,(S_i+S)\right|}{|S_iS|^{1/2}} + \frac{1}{4}\quad(\underline{M}_i-\underline{M})^t(S_i+S)^{-1}(\underline{M}_i-\underline{M})$$

The original algorithm was coded using the first expression. It could be reprogrammed to compute the simpler expression

instead, but the overall time savings would be insignificant. This is a result of the fact that the entire operation is currently limited by the time required to find the boundaries, not the classification time.

The Evaluation Procedure

The test field performance of BOFIAC is evaluated in the subroutine SUMARY which is called from the subroutine PRINT which is responsible for printing the classification map. SUMARY finds those test fields which lie wholly or partially within the area whose classification map is about to be printed and effectively "shrinks" their boundaries to force alignment with the map boundaries and/or pixel group boundaries. The classification of each pixel group within the reduced test fields is then compared with the field class and the results are tabulated.

The remainder of this appendix will be devoted to the control parameters required by BOFIAC and a discussion of the dynamic storage arrays used in the algorithm. These sections are included mainly for the benefit of anyone desiring to use BOFIAC to process data or trying to modify the algorithm to suit their own needs.

Control Parameters

1. For the sake of expediency, the BOFIAC supervisor uses the same subroutines as the PERFIELD supervisor for reading in training field statistics and test field information. This means that standard $PERFIELD control cards must appear in the data deck, but it does not imply that BOFIAC has the same options and capabilities as PERFIELD.

The recommended deck set-up (example) is as follows:

```
OPTIONS STATS
CHANNEL 6, 10, 12
DECK
GROUP  CORN (1/1/), FORAGE (2/2/), SOYBEANS (3/3/)
GROUP  FOREST (4/4/), WATER (5/5/)
END
{Standard statistics deck from $STAT processor.}
TEST 1
{Area coordinate cards for test fields from group 1.}
TEST 2
{Area coordinates cards for test fields from group 2.}
                    .
                    .
                    .
                    .
*END
```

The OPTIONS card may be omitted if one does not care to see a summary of the training field statistics. The GROUP card(s) is used only to coordinate the various training and test classes, not for subclass grouping. For example, OATS (6/13/) would mean that the test fields labeled TEST 6 correspond to the 13th training class in the statistics deck and both are oats.

2. The next data card is read in by subroutine LEARN2 using the format 80A1. It contains the symbols, ordered

according to group number, that are to be used to represent the various classes on the classification map.

The remaining control parameters are read in by a subroutine in CELLS, namely GETPRM.

a) An alphabetic "run identifier" is read into the array AFS via the format 4A4. This identifier will be printed at the head of the output.

b) The program variable PXGR is the number of pixels on the side of a (square) pixel group. It is read next via the format I1. Normally this parameter is '2'.

c) The parameters R1, R2, W1, and W2 are read in next via the format 4I4. They indicate the rows and columns of the picture to be processed in the form: first row, last row, first column, last column. If the specified rows and columns do not come out on a pixel group boundary, the program will delete sufficient rows and/or columns to come out even. The maximum number of columns which can be processed is 125 x PXGR. There is no restriction on the number of rows.

d) The significance level to be used in the t-test is read into the variable SIGLEV via the format I1. The acceptable values are '1' through '4'. A value of '4' will make it easiest for pixel groups to be added to a field and will thus give the largest fields.

e) MFS is the minimum number of pixel groups a field must have in order to engage the sample classifier instead of the maximum likelihood classifier. For PXGR=2, values of '16' and '32' have given consistently good results in the past. The format is I2.

f) NCHAN is the number of spectral bands to be used in the picture analysis. Normally this has been '3'. The format is I1.

g) OUT is the FORTRAN logical data set number to which results are to be written. Normally OUT=6. The format is I1.

h) RUNNUM is the run number of the data set to be processed. It is read in via format I8.

i) The last card defines the NCHAN channels to be used. The channel numbers are read in via the format 4I2; the maximum value of NCHAN is '4'.

Dynamic Core Allocation

Dynamic storage is an efficient way of allocating core for variable length arrays such as those used in BOFIAC. The array "DYNAM" is dimensioned 1, but is located in memory below the main program, subroutines, and other common blocks. This makes a large block of unallocated core available for use in DYNAM. DYNAM is partitioned into "sub-arrays" according to exactly how much space is needed by each. The length and address of each sub-array is computed based upon the number of features to be used

in the processing (NOFET3 or NCHAN) and the number of classes considered (NOPOOL). For example, COVAR4 is the starting address, in DYNAM, of the training class covariance matrices (or inverse matrices as the case may be). Each matrix is symmetric and is therefore stored in a compact form which requires only NOFET3 x (NOFET3+1)/2 REAL*4 words of memory. There are NOPOOL such matrices, so the total number of words allocated for the DYNAM (COVAR4) array is NOPOOL x NOFET3 x (NOFET3 + 1)/2. This is followed immediately by the DYNAM (AVAR4) array which contains the training class mean vectors (at least temporarily). The total number of REAL*4 words allocated for the DYNAM (AVAR4) array is NOPOOL * NOFET3. Similarly DETBS4, DOTBAS, CONBAS, and AVEBAS are base addresses of other sub-arrays in DYNAM.

DYNAM is defined as an array of REAL*4 words, but in some instances it would be more convenient to have an array of REAL*8 or INTEGER*4 words. For this purpose ARRAY and TABLE1 are equivalenced to DYNAM. This of course assigns three different names and types to the same core location, so a greater than normal degree of caution is required. For example, note that when COVAR4=2 x COVAR3-1 the core location referenced by DYNAM (COVAR4), ARRAY (COVAR3), and TABLE1 (COVAR4) is the same in all three cases, but the interpretation of the contents of that location is different.

Appendix III - FORTRAN Listing of BOFIAC

The program listed here operates under the IBM System 360/Model 44 programming system 44PS. Also included are the required LINK EDIT commands. Only the modules designated with an "L" in the LINK EDIT commands are listed here. The modules designated with an "R" are part of the LARSYS software system and listings can be found in the LARSYS documentation.

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0001

0001              COMMON/ARRAY6/TOP,TOP1,ARRAY
0002              INTEGER*4      TOP,TOP1
0003              REAL*8         ARRAY(1)
0004              COMMON/PERCOM/GRPNAM,NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,NOFLDS,
                 *               NOGRPS,NOPOOL,POLNAM,STKPTR,TTITLE,VARSIZ,VARSZ3,
                 *               CSET,CSET3,
                 *               CSEL,CSEL3,FETVEC,FETVC3, GRPSTK,POLPTR,POLSTK,
                 *               STATKY,
                 *               CSET1,POLNM1
0005              INTEGER*4      GRPNAM(2,60),NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,
                 *               NOFLDS,NOGRPS,NOPOOL,POLNAM(2,60),STKPTR,
                 *               TTITLE(30),VARSIZ,VARSZ3
0006              REAL*4         CSET(3,30),CSET3(3,30)
0007              INTEGER*2      CSEL(30),CSEL3(30),FETVEC(30),FETVC3(30),
                 *               GRPSTK(60),POLPTR(2,60),POLSTK(60),STATKY
0008              LOGICAL*1      CSET1(3,30),POLNM1(60)
0009              COMMON/CLASS/  COVAR4,AVAR4,DETBS4,DOTBAS,CONBAS,AVEBAS
0010              INTEGER*4      COVAR4,AVAR4,DETBS4,DOTBAS,CONBAS,AVEBAS
0011              INTEGER*4      COVAR3,AVAR3,DETBS3,DECK,IEND,ITIME,OTIME,CORF,
                 *               TABLE1(1),N
0012              REAL*4         TIME,DYNAM(1)
0013              EQUIVALENCE    (ARRAY,DYNAM),(ARRAY,TABLE1)
0014              COMMON/GLOCCM/CONPUT,DATAPE,FILESV,HEAD,ID,IMARK,MAPSAV,MAPTAP,
                 *               MAXCHA,MAXCLS,PAGSIZ,READIN,RUNTAB,SAVTAP
0015              INTEGER*4      CONPUT,DATAPE,FILESV,HEAD(79),ID(200),IMARK,MAPSAV,
                 *               MAPTAP,MAXCHA,MAXCLS,PAGSIZ,READIN,RUNTAB(10,3),
                 *               SAVTAP
0016              COMMON/PRTCOM/FLDBAS,TABBAS
0017              INTEGER*4      FLDBAS,TABBAS
0018              CONPUT=1
0019              READIN=5
0020              CALL SPACE(ARRAY,TOP)
0021              CALL REDIF6(IEND,.TRUE.,DECK)
0022              CALL SETUP6(ARRAY,TOP,COVAR3,AVAR3,FLDBAS,DETBS3,DECK)
0023              N=NOPOOL*NOFET3
0024              CORE=TOP-8*DETBS3-4*(3*NOPOOL+N+NOPOOL*NOPOOL)
0025              WRITE(6,2) CORE
0026            2 FORMAT(' UNUSED CORE IN ARRAY = ',I7)
0027              IF(CORE.LE.0) STOP 1
0028              CALL PRCLS1(ARRAY(COVAR3),ARRAY(AVAR3),ARRAY(DETBS3),ARRAY)
0029              COVAR4=2*COVAR3-1
0030              AVAR4=2*AVAR3-1
0031              DETBS4=2*DETBS3-1
0032              DOTBAS=DETBS4+NOPOOL
0033              CONBAS=DOTBAS+NOPOOL
0034              AVEBAS=CONBAS+NOPOOL
0035              TABBAS=AVEBAS+N
0036              DO 5 I=1,NOPOOL
0037            5 DYNAM(CONBAS+I-1)=-.5*ALOG(DYNAM(DETBS4+I-1))
0038              DO 6 I=1,N
0039            6 DYNAM(AVEBAS+I-1)=DYNAM(AVAR4+I-1)
0040              CALL LEARN2(DYNAM(COVAR4),DYNAM(AVAR4),DYNAM(DOTBAS),DYNAM)
0041              CALL CELLS
0042              WRITE(6,4) ((GRPNAM(1,I),GRPNAM(2,I)),I=1,NOPOOL)
0043            4 FORMAT('1TEST FIELD RESULTS'//10X,'NO. OF    PCT.'/' CLASS    PIX GP
                 *S    CORCT ',26A4)
0044              TABLE1(2)=0
0045              TABLE1(3)=0
0046              DO 10 I=1,NOPOOL
0047              TABLE1(4)=0
0048              DO 11 J=1,NOPOOL
0049           11 TABLE1(4)=TABLE1(4)+TABLE1(TABBAS+I-1+(J-1)*NOPOOL)
0050              TABLE1(5)=TABLE1(TABBAS+(I-1)*(1+NOPOOL))
0051              DYNAM(1)=100.
0052              IF(TABLE1(4).NE.0) DYNAM(1)=FLOAT(TABLE1(5)*100)/TABLE1(4)
0053              TABLE1(2)=TABLE1(2)+TABLE1(5)
0054              TABLE1(3)=TABLE1(3)+TABLE1(4)
0055           10 WRITE(6,3) GRPNAM(1,I),GRPNAM(2,I),TABLE1(4),DYNAM,(TABLE1(TABBAS+
                 *I-1+(J-1)*NOPOOL),J=1,NOPOOL)
0056            3 FORMAT('0',2A4,I6,2X,F7.1,1X,I5,12I8)
0057              DYNAM(1)=FLOAT(TABLE1(2)*100)/TABLE1(3)
0058              WRITE(6,7) DYNAM
0059            7 FORMAT(//'0OVERALL TEST FIELD PERFORMANCE = ',F5.1,'% CORRECT')
0060              STOP
0061              END
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0001

0001              SUBROUTINE LEARN2(COVMTX,AVEMTX,DOTAB,WORK)
0002              COMMON/PERCOM/GRPNAM,NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,NOFLDS,
                 *               NOGRPS,NOPOOL,POLNAM,STKPTR,TTITLE,VARSIZ,VARSZ3,
                 *               CSET,CSET3,
                 *               CSEL,CSEL3,FETVEC,FETVC3, GRPSTK,POLPTR,POLSTK,
                 *               STATKY,
                 *               CSET1,POLNM1
0003              INTEGER*4      GRPNAM(2,60),NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,
                 *               NOFLDS,NOGRPS,NOPOOL,POLNAM(2,60),STKPTR,
                 *               TTITLE(30),VARSIZ,VARSZ3
0004              REAL*4         CSET(3,30),CSET3(3,30)
0005              INTEGER*2      CSEL(30),CSEL3(30),FETVEC(30),FETVC3(30),
                 *               GRPSTK(60),POLPTR(2,60),PCLSTK(60),STATKY
0006              LOGICAL*1      CSET1(3,30),POLNM1(60)
0007              COMMON/NUMBER/NUMBER
0008              INTEGER*2      NUMBER(48)
0009              REAL*4         COVMTX(VARSZ3,1),AVEMTX(NOFET3,1),DOTAB(1),WORK(1)
0010              READ(5,3) (NUMBER(I),I=1,NOPOOL)
0011            3 FORMAT(80A1)
0012              WRITE(6,2)
0013            2 FORMAT('1 CLASS      SYMBOL')
0014              DO 100 I=1,NOPOOL
0015              WRITE(6,1) GRPNAM(1,I),GRPNAM(2,I),NUMBER(I)
0016            1 FORMAT('0',2A4,4X,A1)
0017          100 CALL PFMULT(COVMTX(1,I),AVEMTX(1,I),DOTAB(I),WORK,NOFET3)
0018              RETURN
0019              END
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0001

0001              SUBROUTINE CELLS
            C     MAIN ROUTINE FOR FIELD SELECTION PROGRAM.
            C     MAIN CONTROLS THE  GEOMETRY OF FIELD BUILDING.
0002              IMPLICIT  INTEGER*2  (E-Z)
0003              INTEGER*4   OUT, IN, JNR, NS
0004              LOGICAL*1   SAME, ASIN, ENOUGH, CLOSE, OPEN
0005              COMMON   AFTOT(250,4), AFS(250,4), ASUM(256,4), ASQ(256,4), AT(4),
                 1    AFLD(250), APGR1, AST, APGR, OUT, IN, JNR, NS
0006              COMMON /INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),
                 1    FSR(250), LASTR(130), SP(130,100), PKGR, PXGR1, NCHAN,
                 2    ROWS, W1, W2, WL2, WIDE, WFS, JW, JS, JR, RL1, IFLD, SIGLEV,
                 3    IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT, JLINE
0007              COMMON /RESULT/ SAME, ASIN(256), ENOUGH(250), CLOSE(250),OPEN(250)
0008              INTEGER*4 ID, RUNNUM
0009              COMMON /LARS/ ID(200), RUNNUM, RCSEL(30),
                 1    COV(256,6), AFCOV(250,6)
0010              CALL GETPRM
0011              IFLD = 1
0012              DO 37 I = 1,250
0013              DO 30 J = 1,4
0014              AFTOT(I,J) = 0.0
0015           30 AFS(I,J) = 0.0
0016              DO 33 J = 1,6
0017           33 AFCOV(I,J) = 0.0
0018              OPEN(I) = .FALSE.
0019              CLOSE(I) = .TRUE.
0020           37 AFLD(I) = 0.0
0021              OLDRW(W1) = 1
0022              RWFLD(W1) = 1
0023              FSR(1) = 1
0024              JR = R1+PXGR
            C
            C     THIS CALL READS OVER ROWS OF THE PICTURE PRECEEDING THE FIRST ROW
            C     TO BE ANALYZED.
0025                CALL ROOVER
            C
0026           39 WL2 = W2-PXGR1
0027              CALL PRINTO
            CV    INITIALIZATION IS NOW COMPLETE.
            C
            C     STARTING ON FIRST ROW.
0028              CALL PGROW
            C     DO FIRST ROW OF PIXEL GROUPS.
            C
0029              WL3 = WL2-PXGR
0030              JW = W1
            C     SUBROUTINE SET ASSUMES THAT THE COLUMN OF THE NEW FIELD IS IN
            C     JW.
0031              CALL SET
0032              OLDRW(W1) = 1
0033              DO 100 JW = W1,WL3,PXGR
0034              JS = JW+PXGR
0035              CALL STU
            C     CALL TO STU HERE ACTUALLY COMPARES TO A GROUP BESIDE. THIS
            C     OCCURRS BECAUSE IN THE FIRST ROW OLDRW IS JUSED TO STORE THE
            C     CURRENT ROW.
0036              IF (SAME) GO TO 45
            C     PIXEL GROUP TO RIGHT IS DIFFERENT SO START A NEW FIELD.
0037              IFLD = IFLD+1
0038              NEXT = NEXT +1
0039              FSR(IFLD) = 1
0040           45 CALL ADD(IFLD,JS)
            C     ONE MORE PIXEL GROUP IN FIELD IFLD.
0041              RWFLD(JS) = IFLD
            C     BOTH RWFLD AND OLDRW ARE LOADED BECAUSE PRINT WILL USE RWFLD
            C     AND OLDRW IS NEEDED WHEN THE NEXT ROW IS STARTED.
0042          100 OLDRW(JS) = IFLD
0043              CALL PRINT
            C     PLACE RESULTS OF FIRST ROW IN PRINT ARRAY VIA CALL TO PRINT.
            C
0044              IR1 = R1+PXGR
0045              IR2 = R2-PXGR1
0046              JNR = 1
            C     INITIALIZATION OF DO 250 LOOP WHICH WILL PROCESS ALL
            C     REMAINING ROWS.
0047              DO 250 JR = IR1,IR2,PXGR
0048              JNR = JNR +1
0049              CALL PGROW
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0002

            C     READ IN NEXT ROW OF PIXEL GROUPS.
0050              DO 160 JW = W1,WL2,PXGR
0051              ASIN(JW) = .FALSE.
            C     INITIALIZATION OF PIXEL GROUPS AS ASSIGNED.
0052              JS = JW
0053              CALL STU
            C     CHEKC AGAINST GROUP ABOVE.
0054              IF (.NOT. SAME) GO TO 130
            C
            C     PUT GROUP INTO SAME FIELD AS GROUP ABOVE.
0055              ASIN(JW) = .TRUE.
0056              NFLD = OLDRW(JW)
0057              RWFLD(JW) = NFLD
0058              CALL ADD(NFLD,JW)
0059              IF (JW .EQ. W1) GO TO 160
            C
            C     CHECK BACK ACROSS FOR UNASSIGNED GROUP.
            C     IF ONE GROUP BACK IS ALREADY ASSIGNED, GO TO NEXT GROUP.
0060              JS = JW-PXGR
0061          120 IF (ASIN(JS)) GO TO 160
0062              CALL STR
            C     COMPARE AGAINST ONE GROUP BACK. IF IT IS DIFFERENT,
            C     TRANSFER TO PROCESSING OF NEXT GROUP.
0063              IF (.NOT. SAME) GO TO 160
0064              ASIN(JS) = .TRUE.
0065              NFLD = RWFLD(JW)
0066              RWFLD(JS) = NFLD
0067              CALL ADD(NFLD,JS)
0068              JS = JS -PXGR
            C
            C     CHECK TO SEE IF BACK AT BEGINNING OF ROW.
0069              IF (JS .GE. W1) GO TO 120
0070              GO TO 160
0071          130 IF (JW .EQ. W1) GO TO 160
            C     SINCE THIS GROUP DIFFERS FROM THE ONE ABOVE, ASK IF THE ONE
            C     TO ITS LEFT IS ASSIGNED AND IF SO COMPARE TO IT.
            C
0072              IF (ASIN(JW-PXGR)) GO TO 135
0073              IF (JW .EQ. WL2) GO TO 140
            C     THE ABOVE IF PROVIDES FOR SPECIAL TREATMENT AT THE END OF A ROW.
0074              GO TO 160
0075          135 JS = JW
0076              CALL STR1
0077              IF (SAME) GO TO 137
0078              IF (JW .EQ. WL2) GO TO 140
            C     THE IF ASKS IF END OF ROW.
0079              GO TO 160
0080          137 ASIN(JW) = .TRUE.
0081              NFLD = RWFLD(JW-PXGR)
0082              RWFLD(JW) = NFLD
0083              CALL ADD(NFLD,JW)
0084              GO TO 160
0085          140 CALL SET
0086              FSR(IFLD) = JLINE+1
            C     SPECIAL CODE FOR NEW FIELD AT END OF ROW.
0087              ASIN(JW) = .TRUE.
0088              RWFLD(JW) = IFLD
0089              IF (ASIN(JW-PXGR)) GO TO 160
            C     IF ONE BACK IS ASSIGNED NO NEED TO COMPARE IT.
            C
0090              JS = JW-PXGR
0091          150 CALL STR
0092              IF (.NOT. SAME) GO TO 160
            C     LOOP TO CHECK BACK ACROSS.
0093              ASIN(JS) = .TRUE.
0094              RWFLD(JS) = IFLD
0095              CALL ADD(IFLD,JS)
0096              JS = JS-PXGR
0097              IF (ASIN(JS)) GO TO 160
0098              IF (JS .GE. W1) GO TO 150
0099          160 CONTINUE
            C
            C     FOLLOWING IS GOING BACK THROUGH ROW TO PICK UP ANY GROUPS
            C     NOT YET ASSIGNED AND CALLING THEM NEW FIELDS.
0100              JW = WL3
0101          170 IF (ASIN(JW)) GO TO 190
0102              ASIN(JW) = .TRUE.
0103              CALL SET
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3, LEVEL 4   DATE  73106                          PAGE  0003
0104              FSR(IFLD) = JLINE+1                                                   CEL01600
0105              RWFLD(JW) = IFLD                                                      CEL01610
0106        180   JW = JW-PXGR                                                          CEL01620
0107              IF (JW .LT. W1) GO TO 200                                             CEL01630
           C      IF AT START OF ROW THEN ARE FINISHED.                                 CEL01640
           C                                                                            CEL01650
0108              IF (ASIN(JW)) GO TO 190                                               CEL01660
0109              JS = JW                                                               CEL01670
0110              CALL STR                                                              CEL01680
           C      CHECK AGAINST ONE BEHIND IF IT IS NOT ASSIGNED.                       CEL01690
0111              IF (.NOT. SAME) GO TO 170                                             CEL01700
           C      IF A NEW FIELD GO BACK AND ASSIGN.                                    CEL01710
           C                                                                            CEL01720
0112              ASIN(JW) = .TRUE.                                                     CEL01730
0113              RWFLD(JW) = IFLD                                                      CEL01740
0114              CALL ADD(IFLD,JW)                                                     CEL01750
           C      NOW GO CHECK NEXT GROUP BACK ACROSS.                                  CEL01760
0115              GO TO 180                                                             CEL01770
0116        190   JW = JW-PXGR                                                          CEL01780
0117              IF (JW .GE. W1) GO TO 170                                             CEL01790
0118        200   CALL PRINT                                                            CEL01800
0119              DO 220 I = W1,WL2,PXGR                                                CEL01810
0120              OPEN(OLDRW(I)) = .TRUE.                                               CEL01820
0121              CLOSE(RWFLD(I)) = .FALSE.                                             CEL01830
0122        220   OLDRW(I) = RWFLD(I)                                                   CEL01840
           C      RWFLD IS MOVED TO OLDRW FOR COMPARISONS FROM NEXT ROW.                CEL01850
           C      ALSO OPEN NOW INDICATES WHICH VECTORS OF AFLD ARE OPEN ON             CEL01860
           C      LAST ROW AND CLOSE INDICATES WHICH VECTORS ARE CLOSED                 CEL01870
           C      ON CURRENT ROW.                                                       CEL01880
           C                                                                            CEL01890
0123              DO 230 ICLOSE = 1,250                                                 CEL01900
0124              IF (OPEN(ICLOSE) .AND. CLOSE(ICLOSE)) CALL CLSTK                      CEL01910
0125              OPEN(ICLOSE) = .FALSE.                                                CEL01920
0126        230   CLOSE(ICLOSE) = .TRUE.                                                CEL01930
           C      FOR ALL FIELDS JUST CLOSED, CLASSIFICATION IS PERFORMED               CEL01940
           C      AND THE VECTOR OF AFLD RETURNED TO THE STACK.                         CEL01950
           C     THEN OPEN AND CLOSE ARE INITIALIZED FOR THE NEXT ROW.                  CEL01960
           C                                                                            CEL01970
           C      PRINT1 IS CALLED TO PRINT RESULTS OF 50 ROWS.                         CEL01980
           C      PRINT1 IS CALLED AFTER THE FRIST 100 ROWS HAVE BEEN PROCESSED         CEL01990
           C      TO PRINT THE FIRST FIFTY ROWS AND THEN AFTER EACH FIFTY ROWS          CEL02000
           C      HAVE BEEN PROCESSED TO PRINT AN ADDITIONAL 50 ROWS.                   CEL02010
0127              IF (JNR .GE. 100 .AND. MOD(JNR,50) .EQ. 0) CALL PRINT1                CEL02020
0128        250   CONTINUE                                                              CEL02030
0129              JR = IR2+1                                                            CEL02040
0130              CALL PRINT1                                                           CEL02050
           C      THIS CALL TO PRINT1 WILL PRINT RESULTS OF ALL REMAINING               CEL02060
           C      ROWS.                                                                 CEL02070
0131              RETURN
0132              END                                                                   CEL02140
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3, LEVEL 4   DATE  73106                          PAGE  0002
0062              CALL GADRUN (RUNNUM,11,ID,ERRGAN)                                     GET00780
0063              IF(ERRGAN.EQ.0) GO TO 335
0064              WRITE(15,5010) ERRGAN
0065       5010   FORMAT(' GADRUN ERROR ',I2)
0066              PAUSE ' EL BOMBO '
0067        335   CONTINUE
0068              DO 340 I = 1,30                                                       GET00790
0069        340   RCSEL(I) = 0                                                          GET00800
0070              IF (IN .EQ. 15) WRITE (15,350)
0071        350   FORMAT (' WHICH CHANNELS  (4I2)')                                     GET00820
0072              READ (IN,360)(PIX(I),I=1,NCHAN )
0073        360   FORMAT (4I2)                                                          GET00840
0074              DO 370 I = 1,NCHAN
0075        370   RCSEL(PIX(I)) = 1
0076              RETURN                                                                GET00850
0077              END                                                                   GET00860
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3, LEVEL 4   DATE  73106                          PAGE  0001
0001              SUBROUTINE GETPRM                                                     GET00010
           C      GETPRM READS IN VARIAOUS PARAMETERS. THE FIRST IS THE INPUT           GET00020
           C      UNIT NUMBER FROM WHICH THE REMAINING PARAMETERS ARE                   GET00030
           C      TO BE READ.                                                           GET00040
           C      THEN COMES THE TITLE, PIXEL GROUP SIZE, ROWS AND                      GET00050
           C      COLUMNS TO BE PROCESSED, SIGNIFICANCE LEVEL, MINIMUM FIELD            GET00060
           C      SIZE, NUMBER OF SPECTRA AND THE UNIT NUMBER                           GET00070
           C      TO WHICH THE OUTPUT IS TO BE WRITTEN. OTHER INITIALAZATION            GET00080
           C      FUNCTIONS ARE ALSO PERFORMED.                                         GET00090
0002              IMPLICIT  INTEGER*2  (E-Z)                                            GET00100
0003              INTEGER*4  OUT, IN, JNR, NS                                           GET00110
0004              LOGICAL*1  SAME, ASIN, ENOUGH, CLOSE, OPEN                            GET00120
0005              COMMON  AFTOT(250,4), AFS(250,4), ASUM(256,4), ASQ(256,4), AT(4),     GET00130
                 1     AFLD(250), APGR1, AST, APGR, OUT, IN, JNR, NS                    GET00140
0006              COMMON /INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),          GET00150
                 1     FSR(250), LASTR(130), SP(130,100), PXGR, PXGR1, NCHAN ,          GET00160
                 2     ROWS, W1, W2, WL2, WIDE, MFS, JW, JS, JR, RL1, IFLD, SIGLEV,     GET00170
                 3     IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT, JLINE        GET00180
0007              COMMON /RESULT/ SAME, ASIN(256), ENOUGH(250), CLOSE(250),OPEN(250)    GET00190
0008              INTEGER*4 ID,RUNNUM,ERRGAN                                            GET00200
0009              COMMON /LARS/ ID(200),RUNNUM,RCSEL(30)                                GET00210
0010              DATA ATT /'T' /                                                       GET00220
           C      WRITE (15,5)                                                          GET00230
           C      READ (15,65) IN                                                       GET00250
0011          5   FORMAT (' INPUT ')                                                    GET00240
0012              IN=5
0013              IF (IN .EQ. 15) WRITE (15,6)                                          GET00260
0014          6   FORMAT (' RUN IDENTIFIER ')                                           GET00270
0015              READ (IN,7) (AFS(I,1),I=1,4)                                          GET00280
0016          7   FORMAT (4A4)                                                          GET00290
0017              IF (IN .EQ. 15) WRITE (15,10)                                         GET00300
0018         10   FORMAT (' GROUP SSZE ')                                               GET00310
0019              READ (IN,15) PXGR                                                     GET00320
0020         15   FORMAT (I1)                                                           GET00330
0021              IF (IN .EQ. 15) WRITE (15,30)                                         GET00340
0022         30   FORMAT (' ROWS & COLS 4I4')                                           GET00350
0023              READ (IN,35) R1,R2,W1,W2                                              GET00360
0024         35   FORMAT (4I4)                                                          GET00370
0025              IF (IN .EQ. 15) WRITE (15,40)                                         GET00380
0026         40   FORMAT (' SIGLEV ')                                                   GET00390
0027              READ (IN,120) SIGLEV                                                  GET00400
0028              AST = ATT                                                             GET00410
           C                                                                            GET00420
0029              IF (IN .EQ. 15) WRITE (15,60)                                         GET00430
0030         60   FORMAT (' MIN SIZE ')                                                 GET00440
0031              READ (IN,65) MFS                                                      GET00450
0032         65   FORMAT (I2)                                                           GET00460
0033              R2 = R2-MOD(R2-R1+1,PXGR+0)                                           GET00470
0034              W2 = W2-MOD(W2-W1+1,PXGR+0)                                           GET00480
           C      THE USE OF PXGR+0  IN THE ABOVE STATEMENTS AVOIDS                     GET00490
           C      AN ERROR FOR ILLEGAL TYPE.                                            GET00500
0035              WIDE = W2-W1+1                                                        GET00510
0036              ROWS = R2-R1+1                                                        GET00520
0037              PXGR1 = PXGR-1                                                        GET00530
0038              RL1 = MFS-PXGR1                                                       GET00540
0039              IDF = 7*PXGR**2-2                                                     GET00550
0040              APGR = PXGR**2                                                        GET00560
0041              APGR1 = 1.0/APGR                                                      GET00570
0042              IF (IN .EQ. 15) WRITE (15,80)                                         GET00580
0043         80   FORMAT (' NUMBER OF SPECTRA ')                                        GET00590
0044              READ (IN,120) NCHAN                                                   GET00600
0045              IF (IN .EQ. 15) WRITE (15,110)                                        GET00610
0046        110   FORMAT(' OUTPUT ')                                                    GET00620
0047              READ (IN,120) OUT                                                     GET00630
0048        120   FORMAT (I1)                                                           GET00640
0049              WRITE (OUT,130) (AFS(1,I),I=1,4)                                      GET00650
0050        130   FORMAT (30X4A4)                                                       GET00660
0051              DO 300 I = 1,250                                                      GET00670
0052        300   STACK(I) = I                                                          GET00680
0053              NEXT = 1                                                              GET00690
0054              PWIDE = WIDE/PXGR                                                     GET00700
0055              DO 310 I = 1,130                                                      GET00710
0056        310   LASTR(I) = 0                                                          GET00720
0057              JLINE = 0                                                             GET00730
0058              IF (IN .EQ. 15) WRITE (15,320)
0059        320   FORMAT (' RUN NUMBER    (I8)')
0060              READ (IN,330) RUNNUM
0061        330   FORMAT (I8)
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3, LEVEL 4   DATE  73106                          PAGE  0001
0001              SUBROUTINE PGROW                                                      PGR00010
           C      READS IN A NEW ROW OF PIXEL GROUPS AND COMPUTES THE SUMS AND SUMS     PGR00020
           C      OF SQUARES OF VALUES OF EACH PIXEL GROUP.                             PGR00030
0002              IMPLICIT  INTEGER*2  (E-Z)                                            PGR00040
0003              INTEGER*4  OUT, IN, JNR, NS                                           PGR00050
0004              LOGICAL*1  SAME, ASIN, ENOUGH, CLOSE, OPEN                            PGR00060
0005              COMMON  AFTOT(250,4), AFS(250,4), ASUM(256,4), ASQ(256,4), AT(4),     PGR00070
                 1     AFLD(250), APGR1, AST, APGR, OUT, IN, JNR, NS                    PGR00080
0006              COMMON /INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),          PGR00090
                 1     FSR(250), LASTR(130), SP(130,100), PXGR, PXGR1, NCHAN ,          PGR00100
                 2     ROWS, W1, W2, WL2, WIDE, MFS, JW, JS, JR, RL1, IFLD, SIGLEV,     PGR00110
                 3     IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT               PGR00120
0007              COMMON /RESULT/ SAME, ASIN(256), ENOUGH(250), CLOSE(250),OPEN(250)    PGR00130
0008              LOGICAL*1 DATA(1280,4), DUML                                          PGR00140
0009              LOGICAL*1 IDATA(2740)                                                 PGR00150
0010              INTEGER*4 ID,LNID,ERRRV,RUNNUM                                        PGR00160
0011              COMMON /LARS/ ID(200),RUNNUM,RCSEL(30),COV(256,6),                    PGR00170
                 1     AFCOV(250,6)                                                     PGR00180
0012              DIMENSION PIXEL(768,4)                                                PGR00190
0013              L1 = 1                                                                PGR00200
0014              L2 = WIDE                                                             PGR00210
0015              W0 = W1-1                                                             PGR00220
0016              DO 30 I = 1,PXGR                                                      PGR00230
0017              CALL TOPRV(I1,ID(6),ERRRV,IDATA,2,ID(5),RCSEL,LNID)                   PGR00240
0018              IF (ERRRV .EQ. 1) STOP 10                                             PGR00250
0019              IF (ERRRV .NE. 0) WRITE (15,15) ERRRV                                 PGR00260
0020         15   FORMAT (' ERROR ',I8)                                                 PGR00270
0021              DO 20 NS =1,NCHAN                                                     PGR00280
0022              K = (NS-1)*ID(6)                                                      PGR00290
0023              KK = L1                                                               PGR00300
0024              DO 20 J = W1,W2                                                       PGR00310
0025              DATA(KK,NS) =IDATA(K+J)                                               PGR00320
0026         20   KK = KK+1                                                             PGR00330
0027              L1 = L2+1                                                             PGR00340
0028         30   L2 = L2+WIDE                                                          PGR00350
0029              WORDS =WIDE*PXGR                                                      PGR00360
0030              DO 100 NS = 1,NCHAN                                                   PGR00370
0031              CALL SHIFTE(PIXEL(1,NS),DATA(1,NS),WORDS)                             PGR00380
0032              L = -PXGR                                                             PGR00390
0033              DO 50 JW = W1,WL2,PXGR                                                PGR00400
0034              L = L+PXGR                                                            PGR00410
0035              ASUM(JW,NS) = 0.0                                                     PGR00420
0036              ASQ(JW,NS) = 0.0                                                      PGR00430
0037              KK = L                                                                PGR00440
0038              DO 50 I = 1,PXGR                                                      PGR00450
0039              DO 40 J = 1,PXGR                                                      PGR00460
0040              ASUM(JW,NS) = ASUM(JW,NS)+PIXEL(KK+J,NS)
0041         40   ASQ(JW,NS) = ASQ(JW,NS) + PIXEL(KK+J,NS)**2
0042         50   KK = KK+WIDE                                                          PGR00490
0043        100   CONTINUE                                                              PGR00500
0044              NS1=NCHAN-1                                                           PGR00510
0045              L1 = 0                                                                PGR00520
0046              DO 110 NS = 1,NS1                                                     PGR00530
0047              L2 = NS+1                                                             PGR00540
0048              DO 110 NSI = L2,NCHAN                                                 PGR00550
0049              L1 = L1+1                                                             PGR00560
0050              L = -PXGR                                                             PGR00570
0051              DO 110 JW = W1,WL2,PXGR                                               PGR00580
0052              L = L+PXGR                                                            PGR00590
0053              COV(JW,L1) = 0.0                                                      PGR00600
0054              KK = L                                                                PGR00610
0055              DO 110 I = 1,PXGR                                                     PGR00620
0056              DO 105 J = 1,PXGR                                                     PGR00630
0057        105   COV(JW,L1) = COV(JW,L1) + PIXEL(KK+J,NS)*PIXEL(KK+J,NSI)              PGR00640
0058        110   KK = KK+WIDE                                                          PGR00650
           C      THE DO 110 LOOPS COMPUTE THE COVARIANCE MATRIX FOR EACH PIXEL GROUPGR00660
           C      ON THIS ROW                                                           PGR00670
0059              RETURN                                                                PGR00680
0060              ENTRY RDOVER                                                          PGR00690
           C       ENTRY RDOVER READS OVER ROWS OF THE PICTURE WHICH ARE NOT            PGR00700
           C      TO BE PROCESSED.                                                      PGR00710
0061              IF (R1 .EQ. 1) RETURN                                                 PGR00720
           C      IF R1 IS 1, THEN NO ROWS NEED TC BE READ UVER.                        PGR00730
0062              II = R1-1                                                             PGR00740
0063              DO 120 I = 1,II                                                       PGR00750
0064        120   CALL TOPRV(I1,ID(6),ERRRV,IDATA,2,ID(5),RCSEL,LNID)                   PGR00760
0065              RETURN                                                                PGR00770
0066              END                                                                   PGR00780
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0001

      SUBROUTINE ADD (INTO,FROM)
C     ADD ADDS A NEW PIXEL GROUP INTO AFTOT AND AFS.
C     INTO IS THE ROW OF AFTOT AND AFS CONTAINING
C     THE FIELD TO BE ADDED INTO. FROM CONTAINS THE
C     ELEMENT OF ASUM AND ASQ CONTAINING THE CURRENT PIXEL
C     GROUP WHICH IS BEING ADDED.
      IMPLICIT  INTEGER*2  (E-Z)
      INTEGER*4  OUT, IN, JNR, NS
      LOGICAL*1  SAME, ASIN, ENOUGH, CLOSE, OPEN
      COMMON  AFTOT(250,4), AFS(250,4), ASUM(256,4), ASQ(256,4), AT(4),
     1     AFLD(250), APGR1, AST, APGR, OUT, IN, JNR, NS
      COMMON /INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),
     1     FSR(250), LASTR(130), SP(130,100), PXGR, PXGR1, NCHAN,
     2     ROWS, W1, W2, WL2, WIDE, MFS, JW, JS, JR, RL1, IFLD, SIGLEV,
     3     IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT
      COMMON /RESULT/ SAME, ASIN(256), ENOUGH(250), CLOSE(250),OPEN(250)
      INTEGER*4 ID, RUNNUN
      COMMON /LARS/ ID(200), RUNNUM, RCSEL(30),COV(256,6),
     1     AFCOV(250,6)
      NC=(NCHAN**2-NCHAN)/2
      AFLD(INTO) = AFLD(INTO) + 1.0
      DO 50 NS = 1,NCHAN
      AFTOT(INTO,NS) = AFTOT(INTO,NS) + ASUM(FROM,NS)
   50 AFS(INTO,NS) = AFS(INTO,NS)+ASQ(FROM,NS)
      DO 60 LI = 1,NC
   60 AFCOV(INTO,LI) = AFCOV(INTO,LI) + COV(FROM,LI)
C     THE DO 60 LOOPS ADD THE COVEARIANCE MATRIX FOR
C     THE GROUP INTO THAT OF THE FIELD
      RETURN
      ENTRY SET
C     WHEN A NEW FIELD IS STARTED, A NUMBER IS PULLED FROM
C     THE STACK OF AVAILABLE VECTORS OF AFTOT AND AFS
C     (AND ELEMENT OF AFLD). THIS VECTOR IS USED FOR THE NEW FIELD.
      IFLD = STACK(NEXT)
      NEXT = NEXT +1
      NC = (NCHAN **2-NCHAN )/2
      AFLD(IFLD) = 1.0
      DO 100 NS = 1,NCHAN
      AFTOT(IFLD,NS) = ASUM(JW,NS)
  100 AFS(IFLD,NS) = ASQ(JW,NS)
      DO 110 LI = 1,NC
  110 AFCOV(IFLD,LI) = COV(JW,LI)
      RETURN
      ENTRY CLSTK
C     ENTRY CLSTK CLOSES OUT A ELEMENT OF AFLD BY RETURNING
C     IT TO STACK, CLASSIFYING THE FIELD AND ZEROING
C     AFLD AND THE ROW OF AFTOT AND AFS USED FOR THE CLOSED FIELD.
      NEXT = NEXT -1
      STACK(NEXT) = ICLOSE
      CALL CLASS1
      DO 200 J = 1,4
      AFTOT(ICLOSE,J) = 0.0
  200 AFS(ICLOSE,J) = 0.0
      DO 210 J = 1,6
  210 AFCOV(ICLOSE,J) = 0.0
      AFLD(ICLOSE) = 0.0
      RETURN
      END
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0001

      SUBROUTINE STU
C     THE STATISTICAL SUBROUTINE FOR T-TEST. RETURNS SAME = TRUE IF
C     TWO GROUPS BEING COMPARED ARE THE SAME.
C     ENTRY STU COMPARES A PIXEL GROUP WITH THE FIELD CONTAINING
C     THE PIXEL GROUP IMMEDIATELY ABOVE  THE CURRENT PIXEL GROUP..
C
C     J CONTAINS THE ELEMENT OF AFLD REPRESENTING SAMPLE 2 IN THE
C     PAPER. JS IS THE INDEX OF THE CURRENT PIXEL GROUP. NOTE
C     THAT JS IS PASSED TO THIS SUBROUTINE VIA COMMON WHEREAS
C     J IS COMPUTED DIFFERENTLY ACCORDING TO WHICH
C     ENTRY POINT IS USED.
C
      IMPLICIT  INTEGER*2  (E-Z)
      INTEGER*4  OUT, IN, JNR, NS
      LOGICAL*1  SAME, ASIN, ENOUGH, CLOSE, OPEN
      COMMON  AFTOT(250,4), AFS(250,4), ASUM(256,4), ASQ(256,4), AT(4),
     1     AFLD(250), APGR1, AST, APGR, OUT, IN, JNR, NS
      COMMON /INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),
     1     FSR(250), LASTR(130), SP(130,100), PXGR, PXGR1, NCHAN ,
     2     ROWS, W1, W2, WL2, WIDE, MFS, JW, JS, JR, RL1, IFLD, SIGLEV,
     3     IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT
      COMMON /RESULT/ SAME, ASIN(256), ENOUGH(250), CLOSE(250),OPEN(250)
      DIMENSION ATSTAT(34,4)
      DATA ATSTAT /12.706, 4.303, 3.182, 2.776, 2.571, 2.447, 2.365,
     1     2.306, 2.262, 2.228, 2.201, 2.179, 2.160, 2.145, 2.131, 2.120,
     2     2.110, 2.101, 2.093, 2.086, 2.080, 2.074, 2.069, 2.064, 2.060,
     3     2.056, 2.052, 2.048, 2.045, 2.042, 2.021, 2.000, 1.980, 1.960,
     4     31.821, 6.965, 4.541, 3.747, 3.365, 3.143, 2.998, 2.896,
     5     2.821, 2.764, 2.718, 2.681, 2.650, 2.624, 2.602, 2.583,
     6     2.567, 2.552, 2.539, 2.528, 2.518, 2.508, 2.500, 2.492,
     7     2.485, 2.479, 2.473, 2.467, 2.462, 2.457, 2.423, 2.390,
     8     2.358, 2.326,
     9     63.657, 9.925, 5.841, 4.604, 4.032, 3.707, 3.499, 3.355,
     A     3.250,  3.169, 3.106, 3.055, 3.012, 2.977, 2.947, 2.921,
     B     2.898,  2.878, 2.861, 2.845, 2.831, 2.819, 2.807, 2.797,
     C     2.787,  2.779, 2.771, 2.763, 2.756, 2.750, 2.704, 2.660,
     D     2.617,  2.576 ,
     E     636.619, 31.598, 12.941, 8.610, 6.859, 5.959, 5.405, 5.041,
     F     4.781, 4.587, 4.437, 4.318, 4.221, 4.140, 4.073, 4.015,
     G     3.965, 3.922, 3.883, 3.850, 3.819, 3.792, 3.767, 3.745, 3.725,
     H     3.707, 3.690, 3.674, 3.659, 3.646, 3.551, 3.460, 3.373, 3.291/
      J = OLDRW(JW)
      GO TO 10
      ENTRY STR
C     ENTRY STR CCMPARES A GROUP WITH THE FIELD CONTAINING
C     THE PIXEL GROUP TO THE RIGHT OF THE CURRENT GROUP.
      J = RWFLD(JS+PXGR)
      GO TO 10
      ENTRY STR1
C     ENTRY STR1 COMPARES A FILED CONTAINING A GIVEN PIXEL GROUP
C     TO THE GOURP IMMEDIATELY TO ITS RIGHT.
      J = RWFLD(JS-PXGR)
   10 AN1 = AFLD(J)*APGR
C     AN1 IS NUMBER OF PIXELS IN FIELD CURRENT PIXEL GROUP IS BEING
C     COMPAREO TO.
      BNUMB = AN1+APGR-2.0
      DO 25 NS = 1,NCHAN
      AV1 = ASUM(JS,NS)/APGR
      AV2 = AFTOT(J,NS)/AN1
C     AV1 AND AV2 ARE THE MEANS OF THE SAMPLES.
      D1 = ASQ(JS,NS)-ASUM(JS,NS)**2/APGR
      D2 = AFS(J,NS)-AFTOT(J,NS)**2/AN1
C     D1 AND D2 ARE THE VARIANCES OF THE TWO SAMPLES. IN THE
C     FOLLOWING LOGIC THE STANDARD DEVIATIONS OF THE SAMPLES
C     ARE COMPARED TO THE MEANS. IF EITHER STANDARD DEVIATION
C     IS LARGER THAN 15% OF THE MEAN, SAME = .FALSE. IS RETURNED.
      IF (D1 .EQ. 0.0) GO TO 12
      IF (SQRT(D1* APGR)) .GT. .15*AV1) GO TO 15
   12 IF (D2 .EQ. 0.0) GO TO 18
      IF (SQRT(D2/AN1) .LE. .15*AV2) GO TO 18
   15 SAME  = .FALSE.
      RETURN
   18 ASP = (D1+D2)/BNUMB
      IF (ASP .NE. 0.0) GO TO 20
C     ASP IS THE SP**2 IN THE PAPER
      AT(NS) = 0.0
      GO TO 25
   20 AT(NS) = (AV1-AV2)/SQRT(ASP*(1.0/AN1+APGR))
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0002

   25 CONTINUE
   30 IDT = BNUMB
      IF (IDT .GT. 30) IDT = 30
      IF (IDF .LE. 30) GO TO 40
      IF (IDF .LE. 40) GO TO 39
      IF (IDF .LE. 60) GO TO 38
      IF (IDF .LE. 120) GO TO 37
      IDT = IDT+1
   37 IDT = IDT+1
   38 IDT = IDT+1
   39 IDT = IDT+1
C     THE ABOVE LOGIC GETS THE CORRECT INDEX FOR THE NUMBER OF DEGREES
C     OF FREEDOM.
   40 SAME = .TRUE.
      DO 60 NS = 1,NCHAN
   60 IF (ABS(AT(NS)) .GT. ATSTAT(IDT,SIGLEV)) SAME = .FALSE.
C     IF THE VALUE OF T (AT(NS)) IS GREATER THAN THE CRITERION VALUE
C     FOR THE T FOR ANY SPECRA, THEN THE NULL HYPOTHESIS THAT
C     THE TWO SAMPLES ARE THE SAME IS REJECTED.
      RETURN
      END
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                PAGE  0001

      SUBROUTINE CLASS1
      IMPLICIT INTEGER*2 (E-Z)
      INTEGER*4  OUT,IN,JNR,NS
      LOGICAL*1  SAME,ASIN,ENOUGH,CLOSE,OPEN
      COMMON/PERCOM/GRPNAM,NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,NOFLDS,
     *                NOGRPS,NOPOOL,POLNAM,STKPTR,TTITLE,VARSIZ,VARSZ3,
     *                CSET,CSET3,
     *                CSEL,CSEL3,FETVEC,FETVC3, GRPSTK,POLPTR,POLSTK,
     *                STATKY,
     *                CSET1,POLNM1
      INTEGER*4       GRPNAM(2,60),NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,
     *                NOFLDS,NOGRPS,NOPOOL,POLNAM(2,60),STKPTR,
     *                TTITLE(30),VARSIZ,VARSZ3
      REAL*4          CSET(3,30),CSET3(3,30)
      INTEGER*2       CSEL(30),CSEL3(30),FETVEC(30),FETVC3(30),
     *                GRPSTK(60),POLPTR(2,60),POLSTK(60),STATKY
      LOGICAL*1       CSET1(3,30),POLNM1(60)
      COMMON/CLASS/   COVAR4,AVAR4,DETBS4,DOTBAS,CONBAS,AVEBAS
      INTEGER*4       COVAR4,AVAR4,DETBS4,DOTBAS,CONBAS,AVEBAS
      COMMON    AFTOT(250,4), AFS(250,4),  ASUM(256,4),  ASQ(256,4),
     *AT(4),AFLD(250), APGR1, AST, APGR, OUT, IN, JNR,  NS
      COMMON/INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),
     *     FSR(250), LASTR(130), SP(130,100), PXGR, PXGR1, NCHAN,
     *     ROWS, W1, W2, WL2, WIDE, MFS, JW, JS, JR, RL1, IFLD, SIGLEV,
     *     IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT,  JLINE
      COMMON/RESULT/ SAME, ASIN(256), ENOUGH(250), CLOSE(250),OPEN(250)
      LOGICAL*1 NXTRW
      INTEGER*4 ID,RUNNUM
      COMMON/LARS/ ID(200), RUNNUM, RCSEL(30), COV(256,6),AFCOV(250,6)
      COMMON/ARRAY6/TOP,TOP1,ARRAY
      INTEGER*4       TOP,TOP1
      REAL*8          ARRAY(1)
      REAL*4          DYNAM(1)
      EQUIVALENCE     (ARRAY,DYNAM)
      REAL*4          FCOV(10),FMEAN(4), FDET,FDOT,SUM(10),WORK(4),COEFF,
     *                VAL,ABDET,RHO(60)
      INTEGER*4       KK
      VAL=APGR*AFLD(ICLOSE)
      DO 199 I=1,NOFET3
  199 FMEAN(I)=AFTOT(ICLOSE,I)/VAL
      AMFS=MFS
      IF(AFLD(ICLOSE).GE.AMFS) GO TO 120
      CALL MLGC(NOFET3,NOPOOL,DYNAM(CCNBAS),DYNAM(COVAR4),DYNAM(AVEBAS),
     *FMEAN,PCLASS,VAL)
      PCLASS=PCLASS+256
      GO TO 150
  120 KK=0
      DO 281 I=1,NOFET3
      DO 281 J=1,I
      KK=KK+1
      IF(J.EQ.I) GO TO 282
      FCOV(KK)=(AFCOV(ICLOSE,I-(J*(J+1))/2+(J-1)*NOFET3)-FMEAN(I)*AFTOT(
     *ICLOSE,J))/(VAL-1.)
      GO TO 281
  282 FCOV(KK)=(AFS(ICLOSE,J)-FMEAN(I)*AFTOT(ICLCSE,J))/(VAL-1.)
  281 CONTINUE
      CALL PFINV(FCOV,NOFET3,FDET,DYNAM)
      IF(FDET.LE.0.) STOP 2
      CALL PFMULT(FCOV,FMEAN,FDOT,DYNAM,NOFET3)
      DO 330 I=1,NOPOOL
      DO 310 J=1,VARSZ3
  310 SUM(J)=FCOV(J)+DYNAM(COVAR4+(I-1)*VARSZ3+J-1)
      COEFF=(2.**(NOFET3/2.))/((DYNAM(DETBS4+I-1)*FDET)**.25)
      DO 320 J=1,NOFET3
  320 WORK(J)=FMEAN(J)+DYNAM(AVAR4+(I-1)*NOFET3+J-1)
      CALL PFINV(SUM,NOFET3,ABDET,DYNAM)
      IF(ABDET.LE.0.) STOP 3
      CALL PFMULT(SUM,WORK,VAL,DYNAM,NOFET3)
      VAL=.25*(VAL-DYNAM(DOTBAS+I-1)-FDOT)
      RHO(I)=ALOG(COEFF/SQRT(ABDET))+VAL
  330 CONTINUE
      VAL=RHO(1)
      PCLASS=257
      DO 340 I=2,NOPOOL
      IF(RHO(I).LE.VAL) GO TO 340
      VAL=RHO(I)
      PCLASS=256+I
  340 CONTINUE
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                    PAGE  0002

0064          150 J=FSR(ICLOSE)
0065          270 NXTRW=.FALSE.
0066              DO 280 I=1,PWIDE
0067              IF(SP(I,J).NE.ICLOSE) GO TO 280
0068              SP(I,J)=PCLASS
0069              NXTRW=.TRUE.
0070          280 CONTINUE
0071              J=J+1
0072              IF(NXTRW.AND.J.LE.JLINE) GO TO 270
0073          350 CONTINUE
0074              RETURN
0075              END
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                    PAGE  0001

0001              SUBROUTINE MLGC(NF,NC,CON,COR,AVE,DATA,IC,TMAX)
0002              INTEGER*4      NF,NC
0003              INTEGER*2      IC
0004              REAL*4          CON(1),COR(1),AVE(1),DATA(1),TMAX
0005              REAL*4          Y,X,FAC
0006              INTEGER*4       NCF,JK,I,J,J1,K,KI
0007              NCF=-NF
0008              JK=0
0009              TMAX=-1.0E60
0010              DO 50 I=1,NC
0011              NCF=NCF+NF
0012              Y=CON(I)
0013              DO 30 J=1,NF
0014              J1=NCF+J
0015              X=DATA(J)-AVE(J1)
0016              DO 20 K=1,J
0017              JK=JK+1
0018              KI=NCF+K
0019              FAC=X*COR(JK)*(DATA(K)-AVE(KI))
0020           20 Y=Y-FAC
0021           30 Y=Y+.5*FAC
0022              IF(Y.LE.TMAX) GO TO 50
0023              TMAX=Y
0024              IC=I
0025           50 CONTINUE
0026              RETURN
0027              END
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   UATE  73106                    PAGE  0001

0001              SUBROUTINE PRINT                                                     PRI00010
0002              IMPLICIT  INTEGER*2  (E-Z)                                           PRI00020
0003              INTEGER*4  OUT, IN, JNR, NS                                          PRI00030
0004              LOGICAL*1  SAME, ASIN, ENOUGH, CLOSE, OPEN                           PRI00040
0005              COMMON/PRTCCM/FLDBAS,TABBAS
0006              INTEGER*4      FLDBAS,TABBAS
0007              COMMON/ARRAY6/TOP,TOP1,ARRAY
0008              INTEGER*4      TOP,TOP1
0009              REAL*8         ARRAY(1)
0010              REAL*4         DYNAM(1)
0011              EQUIVALENCE    (ARRAY,DYNAM)
0012              COMMON   AFTOT(250,4), AFS(250,4), ASUM(256,4), ASQ(256,4), AT(4),  PRI00050
                 1      AFLD(250), APGR1, AST, APGR, OUT, IN, JNR, NS                 PRI00060
0013              COMMON /INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),        PRI00070
                 1      FSR(250), LASTR(130), SP(13C,100), PXGR, PXGR1, NCHAN         PRI00080
                 2      ROWS, W1, W2, WL2, WIDE, MFS, JW, JS, JR, RL1, IFLD, SIGLEV,  PRI00090
                 3      IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT, JLINE     PRI00100
0014              COMMON /RESULT/ SAME, ASIN(256), ENOUGH(250), CLOSE(250),OPFN(250)PRI00110
0015              COMMON/NUMBER/NUMBER
0016              DIMENSION NUMBER(48), PRNT(130), HEAD(10), P1(130), P2(130)         PRI00120
0017              EQUIVALENCE (ASUM(1),P1(1)), (ASQ(1),P2(1))                         PRI00130
0018              LOGICAL*1 FIRST                                                      PRI00140
0019              DATA HEAD / '0', '1', '2', '3', '4', '5', '6',                      PRI00150
                 1     '7', '8', '9' /                                                 PRI00160
0020              DATA EMPTY / '  ' /                                                  PRI00170
            C                                                                          PRI00250
            C     ENTRY TO PRINT MERELY STORES RESULTS OF THE CURRENT ROW INTO         PRI00260
            C     THE PRINT ARRAY, SP.                                                 PRI00270
0021              JLINE = JLINE+1                                                      PRI00280
0022              I = 0                                                                PRI00290
0023              DO 100 J = W1,WL2,PXGR                                               PRI00300
0024              I = I+1                                                              PRI00310
0025          100 SP(I,JLINE) = RWFLD(J)                                               PRI00320
0026              RETURN                                                               PRI00330
0027              ENTRY PRINT1                                                         PRI00340
            C     ENTRY PRINT1 ACTUALLY PRINTS RESULTS OF A NUMBER OF ROWS.            PRI00350
            C     BEFORE PRINTING, ANY FIELDS WHICH ARE OPEN ON THE LAST               PRI00360
            C     ROW TO BE PRINTED ARE CLASSIFIEC (NOT CLOSED).                       PRI00370
0028              JLAST = MOD(JNR,50)+50                                               PRI00380
0029              IF (JNR .LT. 50) JLAST = JNR                                         PRI00390
0030              J = 0                                                                PRI00400
0031              DO 110 JW = W1,WL2,PXGR                                              PRI00410
0032              J = J+1                                                              PRI00420
0033              IF (SP(J,JLAST) .GE. 256) GO TO 110                                  PRI00430
            C     WHEN A FIELD IS CLOSED, SP IS SET TO THE FIELD IDENTIFICATION        PRI00440
            C     INDEX PLUS 256. THUS ONLY THOSE FIELDS WITH ENTRIES IN SP            PRI00450
            C     OF LESS THAN 256 ARE STILL OPEN.                                     PRI00460
0034              ICLOSE = RWFLD(JW)                                                   PRI00470
            C     SUBROUTINE CLASS ASSUMES THAT ICLOSE CONTAINS THE ELEMENT OF         PRI00480
            C     AFLD TO BE CLASSIFIED.                                               PRI00490
0035              CALL CLASS1                                                          PRI00500
0036              FSR(ICLOSE) = 101                                                    PRI00510
0037          110 CONTINUE                                                             PRI00520
            C     FSR SHOULD BE EQUAL TO THE FIRST ROW OF A FIELD WHICH                PRI00530
            C     NEEDS TO HAVE ITS ENTRIES IN THE SP ARRAY PROCESSED.                 PRI00540
            C     SINCE THOSE ROWS OF THIS FIELD DOWN TO ROW JR                        PRI00550
            C     ARE ABOUT TO BE PRINTED, THE NEXT ROW OF THIS FIELD                  PRI00560
            C     WHICH WILL HAVE ACTIVE ENTRIES IN SP IS ROW JR+PXGR.                 PRI00570
            C                                                                          PRI00580
            C     ALL FIELDS UNCLASSIFIED IN THE AREA TO BE PRINTED ARE                PRI00590
            C     CLASSIFIED.                                                          PRI00600
            C                                                                          PRI00610
0038              CALL SUMARY (ARRAY(FLDBAS),DYNAM(TABBAS),JRLINE,JLAST)
0039              DO 200 PLINE = 1,JLAST                                               PRI00620
0040              JRLINE = JRLINE + PXGR                                               PRI00630
0041              DO 160 J = 1,PWIDE                                                   PRI00640
0042              SPK = SP(J,PLINE)                                                    PRI00650
0043              IF (J .EQ. 1 .OR. J .EQ. PWIDE .OR. JR .EQ. IR2+1 .AND.              PRI00660
                 1    PLINE .EQ. JLAST .OR. FIRST .AND. PLINE .EQ. 1) GO TO 150        PRI00670
0044              PRNT(J) = EMPTY                                                      PRI00680
0045              IF (SPK .EQ. SP(J,PLINE+1) .AND. SPK .EQ. SP(J,PLINE-1)              PRI00690
                 1     .AND. SPK .EQ. SP(J+1,PLINE) .AND. SPK .EQ. SP(J-1,PLINF))      PRI00700
                 2     GO TO 160                                                       PRI00710
0046          150 PRNT(J) = NUMBER(SPK-256)                                            PRI00720
0047          160 CONTINUE                                                             PRI00730
            C     THE DO 160 LOOP SETS UP A LINE TO BE PRINTED.                        PRI00740
            C     THE VECTOR PRNT CONTAINS THE TEXT OF THE LINE TO BE PRINTED.         PRI00750
            C     THE LINE CONTAINS BLANKS EXCEPT FOR BOUNDARIES OF FIELDS.            PRI00760
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                    PAGE  0002

            C     ON BOUNDARIES THE FIELD IDENTIFICATION CHARACTER IS PRINTED.         PRI00770
0048          200 WRITE (OUT,210) JRLINE, (PRNT(J),J=1,PWIDE)                          PRI00780
0049          210 FORMAT (' ',I4,1X, 114A1)                                            PRI00790
0050              IF (JR .EQ. IR2+1) RETURN                                            PRI00800
            C     THIS IF WILL CAUSE A RETURN WITHN THE LAST CALL TO                   PRI00810
            C     IS BEING EXECUTED. (JR WAS SET TO IR2+1 IN MAIN JUST                 PRI00820
            C     BEFORE THE LAST CALL TO PRINT1.)                                     PRI00830
0051              DO 300 I = 1,PWIDE                                                   PRI00840
0052              LASTR(I) = SP(I,50)                                                  PRI00850
            C     LASTR IS THE LAST ROW FROM THIS PRINT. IT IS ARRANGED                PRI00860
            C     IN COMMON AS THE ZEROE'TH VECTOR OF SP SO THAT                       PRI00870
            C     IT IS AVAILABLE FOR COMPARISONS THE NEXT TIME                        PRI00880
            C     PRINT1 IS CALLED.                                                    PRI00890
            C                                                                          PRI00900
0053              DO 300 J = 1,50                                                      PRI00910
0054          300 SP(I,J) = SP(I,J+50)                                                 PRI00920
            C     UNPRINTED ROWS OF RESULTS ARE  MOVED TO THE FIRST 50                 PRI00930
            C     ROWS OF THE SP PRINT ARRAY.                                          PRI0C940
0055              DO 310 J = 1,250                                                     PRI00950
0056          310 FSR(J) = FSR(J) -50                                                  PRI00960
            C     ELEMENTS OF FSR MUST CORRESPOND TO THE NEW PLACEMENT OF ROWS         PRI0C970
            C     IN SP. THUS THE DO 310 LOOP ALTERS FSR.                              PRI0C980
0057              FIRST = .FALSE.                                                      PRI00990
            C     FIRST = .FLASE. INDICATES TO FUTURE CALLS TO PRINT1                  PRI01000
            C     THAT PRINT1 HAS BEEN CALLED PREVIOUSLY. THE DIFFERENCE               PRI01010
            C     IN THE FIRST CALL IS THAT THE FIRST LINE IS A BOUNDARY.              PRI01020
0058              JLINE = 50                                                           PRI01030
0059              RETURN                                                               PRI01040
0060              ENTRY PRINT0                                                         PRI01050
            C      ENTRY PRINT0 PRINTS THE HEADING AND INITIALIZES JLINE               PRI01060
            C     AND FIRST.                                                           PRI01070
0061              WRITE (OUT,420) MFS,PXGR, PXGR, SIGLEV, NCHAN                        PRI01080
0062          420 FORMAT (' MIN. NO. OF PIXEL GROUPS CALLED A FIELD =',                PRI01090
                 1     ,I3/' PIXEL GROUP SIZE IS ',I1,' X ',I1,                        PRI01100
                 2         SIGNIFICANCE LEVEL = ',I1,                                  PRI01110
                 3     '     NUMBER OF SPECTRA = ',I1/)                                PRI01120
0063              J = 0                                                                PRI01130
0064              DO 430 JW = W1,WL2,PXGR                                              PRI01140
0065              J = J+1                                                              PRI01150
0066              PRNT(J) = EMPTY                                                      PRI01160
0067              P1(J) = EMPTY                                                        PRI01170
0068              IF (JW .GT. 99) PRNT(J) = HEAD(MOD(JW/100,100)+1)                    PRI01180
0069              IF (JW .GT. 9) P1(J) = HEAD(MOD(JW/10,10)+1)                         PRI01190
0070          430 P2(J) = HEAD(MOD(JW+0,10)+1)                                         PRI01200
0071              WRITE (OUT,440) (PRNT(J),J=1,PWIDE)                                  PRI01210
0072              WRITE (OUT,440) (P1(J),J=1,PWIDE)                                    PRI01220
0073              WRITE (OUT,440) (P2(J),J=1,PWIDE)                                    PRI01230
0074          440 FORMAT (' ',4X,114A1)                                                PRI01240
0075              JRLINE = -PXGR+1                                                     PRI01250
0076              FIRST = .TRUE.                                                       PRI01260
0077              RETURN                                                               PRI01270
0078              END                                                                  PRI01280
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  73106                                PAGE  0001

0001              SUBROUTINE SUMARY (FLDARY,TABLE,JRLINE,JLAST)
0002              IMPLICIT INTEGER*2 (A-Z)
0003              COMMON /INT2/ PIX(1280), RWFLD(256), OLDRW(256), STACK(250),        PR100070
                1     FSR(250), LASTR(130), SP(130,100), PXGR, PXGR1, NCHAN,          PR100080
                2     ROWS, W1, W2, WL2, WIDE, MFS, JW, JS, JR, RL1, IFLD, SIGLEV,    PR100090
                3     IDF, R1, R2, IR1, IR2, ICLOSE, PCLASS, PWIDE, NEXT, JLINE       PR100100
0004              COMMON/PERCOM/GRPNAM,NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,NOFLDS,
               *                 NOGRPS,NOPOOL,POLNAM,STKPTR,TTITLE,VARSIZ,VARSZ3,
               *                 CSET,CSET3,
               *                 CSEL,CSEL3,FETVEC,FETVC3, GRPSTK,POLPTR,POLSTK,
               *                 STATKY,
               *                 CSET1,POLNM1
0005              INTEGER*4      GRPNAM(2,60),NOCLS,NOFEAT,NOFET3,NOFLD,NOFLD3,
               *                 NOFLDS,NOGRPS,NOPCOL,POLNAM(2,60),STKPTR,
               *                 TTITLE(30),VARSIZ,VARSZ3
0006              REAL*4         CSET(3,30),CSET3(3,30)
0007              INTEGER*2      CSEL(30),CSEL3(30),FETVEC(30),FETVC3(30),
               *                 GRPSTK(60),POLPTR(2,60),POLSTK(60),STATKY
0008              LOGICAL*1      CSET1(3,30),POLNM1(60)
0009              INTEGER*4      FLDARY(10,NOFLDS),TABLE(NCPCCL,1)
0010              DATA FLAG/0/
0011              IF(FLAG.EQ.1) GO TO 2
0012              DO 9 I=1,NOPOOL
0013              DO 9 J=1,NOPOOL
0014            9 TABLE(I,J)=0
0015              FLAG=1
0016            2 TP=JRLINE+PXGR
0017              BP=JRLINE+JLAST*PXGR
0018              DO 1 N=1,NOFLDS
0019              TF=FLDARY(2,N)
0020              BF=FLDARY(3,N)
0021              LF=FLDARY(5,N)
0022              RF=FLDARY(6,N)
0023              IF(TF.GT.BP.OR.BF.LT.TP+PXGR1.OR.LF.GT.WL2.OR.RF.LT.W1+PXGR1)
               *GO TO 1
0024              IF(LF.LE.W1) GO TO 3
0025              R=MOD(LF-W1+0,PXGR+0)
0026              IF(R.NE.0) LF=LF+PXGR-R
0027              GO TO 4
0028            3 LF=W1
0029            4 IF(TF.LE.TP) GO TO 5
0030              R=MOD(TF-TP+0,PXGR+0)
0031              IF(R.NE.0) TF=TF+PXGR-R
0032              GO TO 6
0033            5 TF=TP
0034            6 IF(BF.GT.BP) BF=BP
0035              IF(RF.GT.WL2) RF=WL2
0036              NI=(BF-TF)/PXGR
0037              NJ=(RF-LF)/PXGR
0038              IF(NI.EQ.0.OR.NJ.EQ.0) GO TO 1
0039              TF=(TF-JRLINE)/PXGR
0040              LF=(LF-W1)/PXGR+1
0041              BF=TF+NI-1
0042              RF=LF+NJ-1
0043              R=FLDARY(8,N)
0044              DO 8 I=TF,BF
0045              DO 8 J=LF,RF
0046              SPK=SP(J,I)-256
0047            8 TABLE(R,SPK)=TABLE(R,SPK)+1
0048            1 CONTINUE
0049              RETURN
0050              END
```

```
73/106                                                          BOFIAC   LINKAGE EDIT

LIST         PHASE BOFIAC,S,NOAUTO
LIST         INCLUDE PERCOM,R
LIST         INCLUDE AAGLOCOM,R
LIST         INCLUDE BOFIAC,L
LIST         INCLUDE LEARN2,L
LIST         INCLUDE CELLS,L
LIST         INCLUDE GETPRM,L
LIST         INCLUDE PGROW,L
LIST         INCLUDE STU,L
LIST         INCLUDE ADD,L
LIST         INCLUDE CLASS1,L
LIST         INCLUDE MLGC,L
LIST         INCLUDE PRINT,L
LIST         INCLUDE SUMARY,L
LIST         INCLUDE SHIFTE,L
LIST         INCLUDE SPACE,R
LIST         INCLUDE IBCOM=,R
LIST         INCLUDE STAT,R
LIST         INCLUDE REDSTA,R
LIST         INCLUDE PFINV,R
LIST         INCLUDE PFMULT,R
LIST         INCLUDE TOPRY,R
LIST         INCLUDE FRXPR=,R
LIST         INCLUDE ALOG,R
LIST         INCLUDE SQRT,R
LIST         INCLUDE GADRUN,R
LIST         INCLUDE BUFORG=,R
LIST         INCLUDE USEROPT,R
LIST         INCLUDE UNITAB=,R
LIST         INCLUDE CTLWRD,R
LIST         INCLUDE LARS12,R
LIST         INCLUDE MINV,R
LIST         INCLUDE EXP,R
LIST         INCLUDE GTUNIT,R
LIST         INCLUDE MTAPE,R
LIST         INCLUDE IBCD,R
LIST         INCLUDE REDIF6,R
LIST         INCLUDE SETUP6,R
LIST         INCLUDE PRCLS1,R
LIST         INCLUDE GETIME,R
LIST         INCLUDE TSTREQ,R
LIST         INCLUDE ICAPF,R
LIST         INCLUDE POLSCN,R
LIST         INCLUDE CHANEL,R
LIST         INCLUDE CLSCHK,R
LIST         INCLUDE REDSAV,R
LIST         INCLUDE RDFLDS,R
LIST         INCLUDE WRTTRN,R
LIST         INCLUDE WRTMTX,R
LIST         INCLUDE DECCLS,R
LIST         INCLUDE IVAL,R
LIST         INCLUDE LAREAD,R
LIST         INCLUDE FRXPI=,R
LIST         INCLUDE ARRAY6,R
LIST          ENTRY                                                                   CREATED
```

```
                                                                                         73/106  PAGE    2

FLGS  L.CTR    OBJCODE   ADDR   STMT      SOURCE STATEMENT

                                   1  *        SHIFT IS A FORTRAN CALLED ROUTINE TO CONVERT PIXEL DATA TO       XX 00030
                                   2  *        PROPER INTEGER*2 FORMAT. THE CALLING SEQUENCE IS                 XX 00040
                                   3  *        CALL SHIFT (PIX,DATA,WORDS) WHERF PIX IS AN I*2 VECTOR WHERE     XX 00050
                                   4  *        THE CONVERTED DATA IS TO BE PLACED. DATA IS A LOGICAL VECTOR     XX 00060
                                   5  *        WHERE THE DATA COMES IN ONE NUMBER PER BYTE, AND WORDS IS        XX 00070
                                   6  *        THE NUMBER OF NUMBERS TO BE CONVFRTED. WORDS IS AN I*2           XX 00080
                                   7  *        VARIABLE.                                                        XX 00090
      000000                       8  SHIFTE   CSECT                                                            XX 00100
      000000   90EC 000C  0000C    9           STM   14,12,12(13)                                               XX 00110
      000004   05C0               10           BALR  12,0                                                       XX 00120
                          00006   11           USING *,12                                                       XX 00130
      000006   5821 0004  00004   12           L     2,4(1)   ADDRESS OF DATA                                   XX 00140
      00000A   5831 0008  00008   13           L     3,8(1)   ADDRESS OF WORDS                                  XX 00150
      00000E   5811 0000  00000   14           L     1,0(1)   ADDRESS OF PIX                                    XX 00160
                                  15  *                                                                     XX 00170
      000012   1B55               16           SR    5,5   ZERO BXLE COUNTER                                    XX 00180
      000014   1BAA               17           SR    10,10  ZERO WORK REGISTER                                  XX 00190
      000016   5860 C03A  00040   18           L     6,=F'1'    INCREMENT FOR BXLE                              XX 00200
      00001A   4873 0000  00000   19           LH    7,0(3)   LOAD WORDS                                        XX 00210
      00001E   5B70 C03A  00040   20           S     7,=F'1'    GET END OF LOOP CONDITION                       XX 00220
                                  21  *                                                                     XX 00230
      000022   43A2 0000  00000   22  LOOP     IC    10,0(2)   LOAD DATA BYTE                                   XX 00240
      000026   40A1 0000  00000   23           STH   10,0(1)   STORE INTO OUTPUT (PIX)                          XX 00250
      00002A   4A20 C03E  00044   24           AH    2,=H'1'   ADD TO DATA ADDRESS                              XX 00260
      00002E   4A10 C040  00046   25           AH    1,=H'2'  ADD TO STORAGE ADDRESS                            XX 00270
                                  26  *                                                                     XX 00280
      000032   8756 C01C  00022   27           BXLE  5,6,LOOP                                                   XX 00290
                                  28  *                                                                     XX 00300
      000036   982C D01C  0001C   29           LM    2,12,28(13)                                                XX 00310
      00003A   92FF D00C  0000C   30           MVI   12(13),X'FF'                                               XX 00320
      00003E   07FE               31           BR    14                                                         XX 00330
                              +   32   LTORG
      000040   00000001       +   33   DC   F'1'
      000044   0001           +   34   DC   H'1'
      000046   0002           +   35   DC   H'2'
                          00000   36           END   SHIFTE                                                     XX 00340
```