1-1-1973

Automatic Boundary and Sample Classification of Remotely Sensed Multispectral Data

R. L. Kettig

D. A. Landgrebe

Follow this and additional works at: http://docs.lib.purdue.edu/larstech

http://docs.lib.purdue.edu/larstech/36

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
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
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 0.30-11.70μm. 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 pre­
vious 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

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

Overall performance (1467/1472) = 99.7%

Table 2. Test Field Performance of "Point" Classifier

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

Overall performance (6995/7253) = 96.4%
Table 3. Test Field Performance of BOFIAC

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Pixel Groups</th>
<th>Percentage Correct</th>
<th>Number of Pixel Groups Classified into:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CORN</td>
</tr>
<tr>
<td>Corn</td>
<td>1406</td>
<td>99.0</td>
<td>1392</td>
</tr>
<tr>
<td>Forage</td>
<td>1539</td>
<td>97.5</td>
<td>39</td>
</tr>
<tr>
<td>Soybeans</td>
<td>1572</td>
<td>98.6</td>
<td>4</td>
</tr>
<tr>
<td>Forest</td>
<td>1140</td>
<td>99.6</td>
<td>1</td>
</tr>
<tr>
<td>Water</td>
<td>38</td>
<td>100.0</td>
<td>0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>5695</td>
<td></td>
<td>1436</td>
</tr>
</tbody>
</table>

overall performance (5616/5695) = 98.6%
Table 4. Training Field Performance of the "Point" Classifier

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

**overall performance (3031/3171) = 95.6%**
Table 5. Test Field Performance of "Point" Classifier

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Samples</th>
<th>Percentage Correct</th>
<th>Soybeans</th>
<th>Corn</th>
<th>Oats</th>
<th>Wheat</th>
<th>Red Clover</th>
<th>Alfalfa</th>
<th>Rye</th>
<th>Bare Soil</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soybeans</td>
<td>7171</td>
<td>94.7</td>
<td>6788</td>
<td>178</td>
<td>120</td>
<td>17</td>
<td>2</td>
<td>2</td>
<td>25</td>
<td>39</td>
</tr>
<tr>
<td>Corn</td>
<td>2775</td>
<td>88.7</td>
<td>158</td>
<td>2462</td>
<td>21</td>
<td>1</td>
<td>130</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Oats</td>
<td>1558</td>
<td>84.8</td>
<td>20</td>
<td>8</td>
<td>1321</td>
<td>23</td>
<td>119</td>
<td>17</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>Wheat</td>
<td>2641</td>
<td>97.3</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>2569</td>
<td>0</td>
<td>0</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td>Red Clover</td>
<td>3236</td>
<td>85.8</td>
<td>13</td>
<td>70</td>
<td>174</td>
<td>4</td>
<td>2775</td>
<td>199</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Alfalfa</td>
<td>912</td>
<td>83.2</td>
<td>3</td>
<td>15</td>
<td>61</td>
<td>0</td>
<td>74</td>
<td>759</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rye</td>
<td>621</td>
<td>90.0</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>559</td>
<td>0</td>
</tr>
<tr>
<td>Bare Soil</td>
<td>332</td>
<td>98.5</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>327</td>
</tr>
</tbody>
</table>

|                  | TOTAL             | 6987               | 2733     | 1736 | 2654 | 3100  | 980       | 689     | 367|

overall performance (17560/19246) = 91.2%
Table 6. Test Field Performance of BOFIAC.

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Pixel Groups</th>
<th>Percentage Correct</th>
<th>Number of Pixel Groups Classified into:</th>
<th>Soybeans</th>
<th>Corn</th>
<th>Oats</th>
<th>Wheat</th>
<th>Red Clover</th>
<th>Alfalfa</th>
<th>Rye</th>
<th>Bare Soil</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soybeans</td>
<td>6436</td>
<td>96.2</td>
<td></td>
<td>6189</td>
<td>211</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>Corn</td>
<td>2435</td>
<td>97.8</td>
<td></td>
<td>55</td>
<td>2380</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Oats</td>
<td>1340</td>
<td>96.5</td>
<td></td>
<td>7</td>
<td>0</td>
<td>1293</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Wheat</td>
<td>2250</td>
<td>97.5</td>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2193</td>
<td>0</td>
<td>0</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td>Red Clover</td>
<td>2812</td>
<td>94.3</td>
<td></td>
<td>9</td>
<td>18</td>
<td>80</td>
<td>0</td>
<td>2650</td>
<td>51</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Alfalfa</td>
<td>740</td>
<td>96.0</td>
<td></td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>0</td>
<td>14</td>
<td>710</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rye</td>
<td>572</td>
<td>98.7</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>564</td>
<td>0</td>
</tr>
<tr>
<td>Bare Soil</td>
<td>262</td>
<td>100.0</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>262</td>
</tr>
</tbody>
</table>

TOTAL 16,847  | 6262  | 2610 | 1394 | 2201 | 2707 | 761 | 628 | 284

overall performance (16241/16847) = 96.4%
REFERENCES


Appendix I - Input Data and Output Classification Maps for Flightlines 210 and Cl.
Figure I.A. Input Data for Processing Flightline 210.
Figure I.D. Input Data for Processing Flightline Cl.
Figures I.E. (a and b) Gray Scale Printouts of Flightline Cl.
(c) Classification map for Flightline Cl.
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 PRCLSL 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:

\[
X_{ij} \text{ be the data value of pixel } i \text{ in sample } j
\]

\[
S_j = \frac{\sum_{i=1}^{n_j} X_{ij}}{n_j}
\]
\[ M_j = \frac{1}{n_j} \sum_{i=1}^{n_j} (X_{ij}) = \text{mean value of sample } j \]

\[ Q_j = \sum_{i=1}^{n_j} (X_{ij} - M_j)^2 \]

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

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

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.

\[ D = \frac{1}{2} \ln \left( \frac{1}{2} (S_i^{-1} + S^{-1}) \right) - \frac{1}{2} B, \text{ where} \]

\[ B = (S_i^{-1} M_i + S^{-1} M)^t (S_i^{-1} + S^{-1})^{-1} (S_i^{-1} M_i + S^{-1} M) - M_i^t S_i^{-1} M_i - M^t S^{-1} M \]

where \( M_i, S_i \) and \( M, S \) 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:

\[ D = \frac{1}{2} \ln \left( \frac{1}{2} (S_i + S) \right) + \frac{1}{2} (M_i - M)^t (S_i + S)^{-1} (M_i - M) \]

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.