CAC Technical Memorandum No. 53

AN ILLIAC IV ALGORITHM FOR CLUSTER ANALYSIS OF 8-CHANNEL MULTI-SPECTRAL IMAGE DATA

by

Martin Ozga

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

May 1975

Reported research was conducted in collaboration with staff of the Laboratory for Applications of Remote Sensing, Purdue University, Lafayette, Indiana 47907.

This work was supported by the National Aeronautics and Space Administration through Grant NGR 14-005-202 in cooperation with the EROS Program of the U. S. Department of Interior. Collaborative support was also provided by the Statistical Reporting Service of the U. S. Department of Agriculture.

1.0 Introduction

This paper is a verbal description of an ILLIAC IV program written in GLYPNIR[1] that performs cluster analysis of 8-channel multispectral image data into spectrally distinct categories. The algorithm closely follows that of the 4-channel cluster analysis program written in ASK to run in 32-bit mode[2] with necessary modifications for 8-channel data and GLYPNIR. Since the theoretical basis for both programs is the same, Section 2. of this paper is repeated from reference (4) via reference (2) with slight changes and included here for completeness.

2.0 The Algorithm

This clustering algorithm was developed at the Laboratory for Application of Remote Sensing (LARS) at Purdue University[3] and implemented on ILLIAC IV at the Center for Advanced Computation at the University of Illinois. This description is taken from the paper "Boundaries in Multispectral Imagery by Clustering", by Arthur G. Wacker and David A. Landgrebe, as it was presented at the 1970 IEEE Symposium on Adaptive Processes (9th), December 1970.[4]

A clustering cell is a rectangular area or window. The vectors associated with each geographical point (in this case, these are eight-dimensional) in the cell are clustered in observation space into a desired number of modes M_m. Thus, a "natural" grouping for these vectors is found in observation space.

The manner in which a set of vectors is clustered is outlined in detail below. The procedure is essentially the ISODATA of Ball and Hall[5] (1965, pp. 1-61) with modifications similar to those suggested by Swain and Fu (1968, pp. 14-19).[6]

Step 1 - Initialization

Let X_1, X_2, ..., X_N be N 8-dimensional vectors from the clustering cell. If the number of modes desired is M_m, the M_m initial mode centers are generated as follows.

The sample mean of the N vectors is computed according to

$$M_j = \frac{1}{N} \sum_{i=1}^{N} X_{ij} \qquad j = 1, 2, \ldots, 8,$$

and the sample variance for each dimension

$$\sigma_j^2 = \frac{1}{N} \sum_{i=1}^{N} (X_{ij} - M_j)^2 \qquad j = 1, 2, \ldots, 8.$$

Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_8)$ and $M = (M_1, M_2, \ldots, M_8)$. Consider the real line intervals $\sigma_i = [M_i - \sigma_i, M_i + \sigma_i]$, $i = 1, 2, \ldots, 8$. The cartesian product $X\sigma_1 \cdot X\sigma_2 \cdot X\sigma_3 \cdot X\sigma_4 \cdot X\sigma_5 \cdot X\sigma_6 \cdot X\sigma_7 \cdot X\sigma_8$ defines a rectangular parallelepiped in the observation space which should contain most of the vectors from the clustering cell. The M_m initial mode centers are chosen to be uniformly spaced along a diagonal of this rectangular parallelepiped. Accordingly, the mode center for the k^{th} mode is:

$$\overleftarrow{M}_k = \overleftarrow{M} + \overleftarrow{\sigma}\left[2 \frac{k-1}{M_m - 1} - 1\right] \qquad k = 1, 2, \ldots, M_m.$$

Initially none of the vectors is assigned to any mode.

Step 2 - Mode Assignment

Determine the Euclidean distance from each vector to each mode center. Assign each vector to the mode with the nearest mode center.

Step 3 - Mode Migration

If Step 2 did not change the assignment of any of the N vectors, go to Step 4. Otherwise, replace the old mode centers by the means of the vector clusters resulting from Step 2, then return to Mode Assignment.

Step 4 - Variance-Covariance Calculation

Upon completion of the above iterative process, clustering of all vectors into M_m categories is complete. To complete the statistical description of a category, its variance and covariance are calculated. Variance is calculated via

$$\sigma_j^2 = \frac{1}{P-1} \sum_{i=1}^{P} (X_{ij} - M_j)^2 \qquad j = 1,2, \ldots, 8$$

and covariances via

$$C_{jk}^2 = \frac{1}{P} \sum_{i=1}^{P} (X_{ij} - M_j)(X_{ik} - M_k) \qquad j = 1,2, \ldots, 8$$
$$k = j, \ldots, 8,$$

where P denotes the number of vectors in each cluster.

3.0 Files

3.1 Input File

The input file, WINDOW, is an eight-channel raw data window file which contains the usual two row (128 ILLIAC word) header followed by the data. Since each channel occupies 8 bits, each point occupies a full 64-bit ILLIAC IV word, containing the data from the corresponding points of the two images used to form the 8-channel data. To determine how much data to process, the program uses the number of long rows (rows of 64 words across the PEs) which is stored in (ILLIAC) word 4 of the header. If the number of points is not divisible by 64, the portion of the last long row not occupied by valid data should be filled in with words with all 64 bits set to 1 as an indication that those words should not be processed. The structure of this and other files is more fully discussed in [7].

3.2 Output Clustered File

The output clustered file, WINDOX, contains a two row header followed by the data. The file type (stored in ILLIAC word 0) is set to 2 to indicate classified data without chi-square values. The clustered value of each point is stored right justified in a 16-bit field. A 16-bit field is used to be compatible with files created by classification programs which store a chi-square value in the upper 8-bits of the field.

3.3 Output Statistics File

The output statistics file, COEFFS, contains the two row header (with the type in ILLIAC word 0 set to 4) followed by the statistics.

The statistics are stored in ILLIAC 32-bit floating point format, two values per word. The statistics are stored by category and consist of the eight mean values followed by the 36 values of the variance-covariance matrix. The variance-covariance matrix is an eight-by-eight symmetric matrix so only 36 values need be stored to completely specify the matrix. The upper triangular portion of the matrix is stored in order by rows. The statistics for two categories are stored in one ILLIAC row with the first category starting in word 0 and the second in word 24. The last 16 words of each row are not used. Thus, we have in each ILLIAC row:

ILLIAC words	Usage
0 - 3	mean values for category i
4 - 21	upper triangular portion of variance-covariance matrix for category i
22 - 23	not used
24 - 27	mean values for category i + 1
28 - 45	upper triangular portion of variance-covariance matrix for category i + 1
46 - 63	not used

Each variance-covariance matrix C is stored in the upper-triangular format

$$C_{11}\ C_{12} \cdots C_{18}\ C_{22}\ C_{23} \cdots C_{28}\ C_{33}\ C_{34} \cdots C_{38} \cdots C_{88}$$

where within a single word the order is inner-outer so that, for example, C_{11} is stored in the inner part of the word and C_{12} in the outer part.

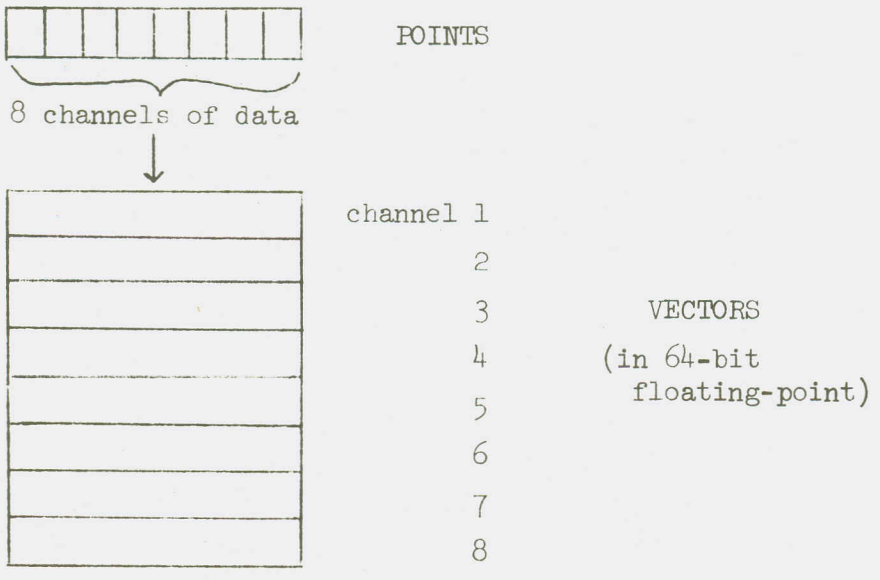
4.0 Internal Data Structure

Since many iterations are made through the data, the entire contents of the input file WINDOW are read into memory before processing begins. This places an upper limit of 10112 pixels on the input file. Also, since GLYPNIR implements only 64-bit operations on the ILLIAC IV,

all values which are input to or output from computations are stored as 64-bit floating-point or 64-bit integer values.

4.1 The Raw Data Points

The raw data points are initially read into the PE VECTOR POINTS, but for computation, the values for each channel are converted to 64-bit floating-point and stored down the same PE (as the original point) in VECTORS. Thus, for PE i we have:



4.2 The Mean Values (mode centers)

To make the computations more parallel by decreasing routing among the PEs, all mean values for all categories are stored in all PEs. The values for all eight channels in each category are stored contiguously down each PE (and in the same position in each PE) in the PE VECTOR MODECENTER.

4.3 Categorized Points

During each iteration, the category to which each point is currently assigned is stored in the PE VECTOR PRESENTCAT as a 64-bit integer. The value stored is one less than that which appears in the

final output file WINDOX. Since the raw data points in their original packed format are no longer needed, PRESENTCAT occupies the same area of memory as POINTS.

5.0 Program Execution

5.1 Subroutine READANDCONVERT

Subroutine READANDCONVERT is called to read the input file. The first page of the input file is read. The file type is checked and if not equal to one, execution is terminated with an error indication. The number of (ILLIAC) long rows is checked. If that is too large, execution is terminated with an error indication. The current maximum is 158. The remainder of the file is read. Each point is converted to eight floating-point numbers which are stored in the same PE as described in Section 4.1.

5.2 Subroutine INITIALMODECENTERS

Subroutine INITIALMODECENTERS is called to compute the initial mode centers as per Step 1 of the theoretical description given above in Section 2. The sample mean is computed by first adding, in parallel, the values of the data in each PE, summing across PEs to get the same value in each PE, and dividing by the number of points. The sample variance is computed similarly by getting the sums of the differences squared between each point and each sample mean in each PE, summing across the PEs, and dividing by the number of points. The initial mode centers are then computed along the diagonal of the parallelepiped as per the theoretical description.

5.3 Subroutine MIGRATE

Subroutine MIGRATE is the inner loop of the program. On each iteration, new mode centers are computed and each point is clustered to the nearest mode center. The process continues until an iteration occurs in which no point changes category. During each iteration, the square of the Euclidean distance is computed between each point and each mode

center. The square is used since we are interested only in relative distances and wish to avoid taking the square root to save time. Each point is placed in the category for which this distance is the smallest. If this category is different from the category to which the point was previously assigned, a (CU) Boolean flag CHANGED is set to TRUE, forcing another iteration. If CHANGED is TRUE at the end of this process, new mode centers are computed. First, the number of points in each PE in each category is computed, then these are summed across PEs to get the total number of points in each category in all PEs. The new mode center for each category is computed as the mean (in each channel) of all points in that category. The next iteration is then performed if required.

5.4 Subroutine VARCOVAR

Subroutine VARCOVAR calculates the variance-covariance matrix for each category, converts it to 32-bit floating-point, and packs it in the form described in Section 3.3. Since two categories are packed into one row of ILLIAC memory, the values for two categories are computed at the same time. The usual method of computing the values for all points in each PE and then summing across PEs is used. Two elements of the matrix (one for each category) are computed at a time and then these are packed in the appropriate places in the row being created.

5.5 Subroutine PACKANDWRITE

Subroutine PACKANDWRITE packs the categorized points and writes the categorized file (WINDOX) and the statistics file (COEFFS) to disk. The packing involves collapsing four rows into one with 16 words for each unpacked row. Therefore, 16 iterations are needed through the loop to completely pack one row. In each iteration, 4 PEs (one for each unpacked row) are enabled, the unpacked rows are routed so that all points to go in that PE get into it, and they are packed by shifting and ORing. The mode pattern is then shifted on to the right for the next iteration.

5.6 The Main Program

The main program consists of calls to the subroutines described in Sections 5.1 through 5.5. Each subroutine is called once and the division into subroutines is merely to introduce modularity. The extra time spent in the calling of the subroutines is extremely small in comparison with the total execution time.

6.0 Operation

6.1 Files

The input (raw data) file is WINDOW. The output clustered window is WINDOX, and the output statistics file is COEFFS. WINDOW and WINDOX may occupy the same area on disk since WINDOW is completely read in before WINDOX is written. Of course, WINDOX and COEFFS must occupy different areas. Due to a peculiarity in implementing timing with GLYPNIR, a one page file LINE, occupying an area distinct from the other files, is required.

6.2 Completion Codes and Program Termination

Completion codes are found by looking at the value of TRO in the POF. As with all such register printouts, the value is in hexadecimal. If the last digit is "A", the run was successful. Any digits preceding the "A" indicate the number of iterations. Thus if the value of TRO is 56A, there were 86 (decimal) iterations.

If the three characters preceding the last are "BAD" some sort of file error occurred. The last digit will be the error number. Thus a value of "BAD4" indicates error 4. Error numbers currently are:

0 Unable to read first page of file WINDOW

1 Bad file type (not 0 or 1) for WINDOW

2 Bad number of ILLIAC rows (greater than 158)

3 Unable to read remainder of file WINDOW (after first ILLIAC page)

4 Unable to write file WINDOX

5 Unable to write file COEFFS

In certain rare instances, other errors may occur which cause immediate termination, so that the value of TRO will not be as specified above. The cause of the error can usually be determined by examining the POF. The most common of such errors is not allocating enough ILLIAC disk space for a file.

6.3 Timing

Elapsed time is calculated for the entire run. The time is available in register IIA, in tenths of milliseconds. A timing printout may also be obtained by retrieving the file LINE from ILLIAC disk at the completion of the run.

References

1. Layman, T. and Baer, D., "GLYPNIR Reference Manual", ILLIAC IV Document No. 263, (ILLIAC IV Project, University of Illinois), December 1972.

2. Thomas, J., "An ILLIAC IV Algorithm for Cluster Analysis of ERTS-1 Data", CAC Technical Memorandum No. 17, (Center for Advanced Computation, University of Illinois), May 1974.

3. Swain, P. H., "Pattern Recognition: A Basis for Remote Sensing Data Analysis", LARS Information Note 11572, (The Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana), 1972.

4. Wacker, A. G. and Landgrebe, D. A., "Boundaries in Multispectral Imagery by Clustering", IEEE Symposium on Adaptive Processes (9th), 1970.

5. Ball, G. H. and Hall, D. J., "ISODATA, A Novel Method of Data Analysis and Pattern Classification", (Stanford Research Institute, Menlo Park, California), 1965.

6. Swain, P. H. and Fu, K. S., "On the Application of Nonparametric Techniques to Crop Classification Problems", National Electronics Conference Proceedings, 24, 1968, pp. 14-19.

7. Thomas, J., "ERTS-ILLIAC Data File Formats", CAC Technical Memorandum No. 19, (Center for Advanced Computation, University of Illinois), May 1974.