NASA TECHNICAL NOTE

N73-24410

NASA TN D-7312

NASA TN D-7312

UNSUPERVISED SPATIAL CLUSTERING WITH SPECTRAL DISCRIMINATION

by Robert R. Jayroe, Jr.

George C. Marshall Space Flight Center
Marshall Space Flight Center, Ala. 35812

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MAY 1973

1. Report No.
NASA TN D-7312

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

Unsupervised Spatial Clustering with Spectral Discrimination

5. Report Date
May 1973

6. Performing Organization Code

7. Author(s)

Robert R. Jayroe, Jr.

8. Performing Organization Report No.
M107

9. Performing Organization Name and Address

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

10. Work Unit No.

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Note

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D. C. 20546

14. Sponsoring Agency Code

15. Supplementary Notes

Prepared by Aero-Astrodynamics Laboratory, Science and Engineering

16. Abstract

The recent development of manned and unmanned space vehicles has brought about an almost unprecedented advance in studies of remotely sensed earth observations. These observations require a multidisciplinary study which includes such fields as agriculture, forestry, geography, demography, cartography, geology, meteorology, hydrology, oceanography, environmental quality, ecology, sensor technology, and interpretation techniques development. With this unprecedented advance comes an unprecedented amount of data. The problem arises of how to analyze and extract information from such large volumes of data in an efficient manner.

The main emphasis of this work is the development of a computer program for extracting features from remotely sensed data presented in digital image form. This computer program requires no human supervision or prejudgment and operates unassisted on the raw digital data.

The presentation of this work also includes a condensed general background on remote sensing of earth features and a short synopsis on some of the most commonly used types of feature extraction techniques. This discussion is followed by a presentation of results obtained from the unsupervised feature extraction computer program along with a description and listing of the computer program.

17. Key Words (Suggested by Author(s))

Earth resources
Remote sensing
Clustering
Classification

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
92

22. Price*
$3.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

ACKNOWLEDGMENT

The author wishes to acknowledge Mr Melvin R. Phillips of the IIT Research Institute, who performed the programing for interfacing the classification programs with the various input-output devices and for integrating these programs into the Earth Resources Processor Computer Program. The author wishes to acknowledge Dr. Harry W. Smedes of the United States Geological Survey, Denver, Colorado, for his many interesting discussions and cooperation and for supplying the raw data, ground truth, and evaluation of results for the Yellowstone Park test site. The author also wishes to acknowledge Dr. David Landgrebe and Mr. Terry Phillips of Purdue University for supplying the data and ground truth for the Purdue Flight Line Cl test site.

FOREWORD

This study was undertaken by Mr. Robert R. Jayroe, Flight Data Statistics Office, Aerospace Environment Division, Aero-Astrodynamics Laboratory, Marshall Space Flight Center. The report provides the mathematical rationale utilized in the development of unsupervised computer routines for extracting features from earth observation data. These routines have been incorporated into an "Earth Resources Data Processor" computer program developed by IIT Research Institute under contract NAS8-26797 and published in NASA CR-61399.

The development of these computer programs provides a method of analysis for processing ERTS and Skylab (EREP) data via a cooperative effort involving the University of Alabama, the Geological Survey of Alabama, Auburn University, and Marshall Space Flight Center.

The author wishes to express his appreciation to Mr Charles Dalton of the Flight Data Statistics Office for his comments and concise review of this report.

UNSUPERVISED SPATIAL CLUSTERING WITH SPECTRAL DISCRIMINATION

I. INTRODUCTION

The objective of this research is the conversion of remotely sensed data into useful information regarding the location and distribution of various classes of identifiable earth observation features. The remotely sensed data are collected from a platform, which may be an airplane, a ship, a balloon, or a satellite. The sensors used generally collect electromagnetic radiation in specified wavelength intervals (multispectral and thermal scanners, other types of radiometers, and multiband photography), echo returns (side-looking aerial radar and sonar), and magnetic field information (magnetometer). The collected data may be analog, digital, or a photographic image, and the data are formatted such that a one-to-one correspondence is preserved between the ground scene and the data, as in an aerial photograph. The data are then analyzed to determine what features can be extracted from the data and to determine the location and distribution of these features. Examples of features could be crop types, diseased crops, bodies of water, water pollution, and land types. Multitudes of other types of features exist. As presently used, there are three main types of feature extraction methods, which will be denoted by human photointerpretation, supervised computer feature extraction, and unsupervised computer feature extraction. The first two feature extraction methods involve human supervision and judgment after the fact, whereas the third method does not permit any change of criterion from datum to datum. The unsupervised computer feature extraction method is based entirely on a logical set of mathematical rules designed to extract the features presented in the remotely sensed data This report discusses the three most commonly used types of feature extraction methods and presents a recently developed computer program for unsupervised feature extraction.

II. SYNOPSIS ON FEATURE EXTRACTION

The most universal method of feature extraction is the classical art of photointerpretation. This art is primarily a process of visual inspection and subjective analysis by a trained human observer, and it has been greatly enhanced by a variety of instruments and machines (stereo viewers, microdensitometers, autoplotters, false color image enhancement, etc). The human's role in this task is to subjectively combine visually acquired inputs relating to spatial properties, color, texture, and temporal phenomena to identify and classify objects in the ground scene [1, 2, 3] In performing this task, the human exercises a sophisticated ability to combine various types of data and draw conclusions as to information subtleties present. As an example of this sophistication, consider the land-use classifications, "pasture" and "improved pasture." Improved pasture is a pasture that is surrounded by a fence, but from an aerial photograph, fences are often difficult to observe. Grazing animals, however, develop the habit of walking along the fences and the photointerpreter merely has to look for a brown path surrounding a green field. The most obvious and serious inadequacy of photointerpretation is the manpower and time needed to analyze large volumes of data resulting from remote sensing on a large-area scale.

Because of this inadequacy, more and more emphasis is being placed on developing computer techniques for analyzing large volumes of remotely sensed data. Present computer techniques are designed to exploit the spectral information of imagery or of spectral scanners for extracting information. One approach to this technique is to use a data bank. The use of a data bank assumes that an object or an area of sufficient dimensions in the ground scene to be considered homogeneous possesses a unique spectral signature. The spectral signatures are produced by recording the returned energy (combined reflected and radiated electromagnetic energy) in each of a number of narrow and discrete wavelength intervals. The data bank would contain prestored or empirically derived statistics on the signatures of known objects or it would contain the ability to model and analytically calculate what a given signature should be.

The difficulty with this technique is that there is appreciable variability in the signatures, which is due to natural temporal changes as well as manmade changes. In practice, this variability is evident even for repeated observations in which great care is taken to maintain constant sun angle, viewing angle, cloud cover, ambient temperature and humidity, instrument calibration, etc. This variability can be attributed to a large degree to a lack of microscopic homogeneity between grossly identical objects or species in two different ground scenes. For example, no two leaves of diseased corn will appear exactly the same, and no two rows of corn will be arrayed exactly the same or be surrounded by the same color of ground exposed between the stalks. Thus, the problem of signature comparison leads to a modeling theory [4].

Another serious drawback to this technique is the amount of computer memory required to store all possible signatures and their variations, and a prohibitive amount of computer time required to compare an unknown spectral signature with all possible signatures stored in the data bank, unless a table look-up procedure can be used [5].

The next approach to be described is probably the most popular and widely used computer feature extraction technique. This approach has been developed extensively at Purdue University [6, 7, 8, 9, 10], and many related techniques [11, 12, 13, 14] and improvements have been derived from this approach. This technique will be called the "supervised classification technique" since it requires human supervision for the selection of training areas, which ultimately determine the number and types of features to be extracted, as well as the accuracy of classification.

To describe this technique, it will be necessary first to describe the data collection and formatting, as well as to develop the mathematical notation for operating on the data. A multispectral scanner is generally used for the data collection The scanner is a monochromator with a detector array for recording the reflected and emitted radiation from the ground in different wavelength intervals. The monochromator is mounted in an aircraft, and the rotating mirror scans the ground scene as the airplane flies, producing an analog signal as shown in Figure 1 If the detector array contains n detectors, then n simultaneous signals or channels of data are recorded for the same ground scene. The analog data are then digitized and recorded on magnetic tape to produce an n-dimensional digital image of the ground scene. Let the algebraic value of the digital number derived from the analog signal be denoted by ${}_kx_{ij}$ where $k = 1,\ldots,n$, the channel number or wavelength interval, $i = 1, 2, 3,\ldots, \ell$, the scan line number or x coordinate of the ground scene, and $j = 1, 2,\ldots, m$, the sample number for scan i or the y coordinate of the ground scene.

Thus, the data set collected contains n channels of data, ℓ scan lines, and m digital samples per scan per channel with an amplitude given by ${}_kx_{ij}$ Each ground-scene coordinate, i and j , of the digital image is normally called a resolution element, and each resolution element is therefore described by an n-dimensional vector called a feature vector, $\vec{x}_{ij}$ The components of the feature vector are $\vec{x}_{ij} = [{}_1x_{ij}, {}_2x_{ij},\ldots {}_nx_{ij}]$

The next step is to produce a boundary map [14] so that it may be compared with an aerial photograph taken of the ground scene at the same time that the multispectral scanner data were collected The boundary map and aerial photograph are shown in Section IV, and a method for producing a boundary map will be discussed in Section III. The photograph allows an observer to select, for example, areas which appear to be crops and then to locate the scan line and column number of the data corresponding to those crops from the boundary map. The data may then be accessed for analysis and used as a training area [15]

For identification of crop species, ground-truth information is required, unless a data bank is available. The ground-truth information could be collected by an observer visiting each field or obtaining the information from a farmer as to the crop type. It is interesting to note that this type of information collection has occasionally been erroneous, and present classification techniques have been sufficiently accurate to point out these errors. With this information, the fields in the aerial photograph can be labeled according to crop type. Training areas are then selected from the boundary map corresponding to particular known fields in the aerial photograph. The desired statistics are calculated from the data in the training area, and a statistical decision rule is used for classifying the remaining data in the digital image. A computer map of the ground scene is then printed out showing the location of all resolution elements that were classified

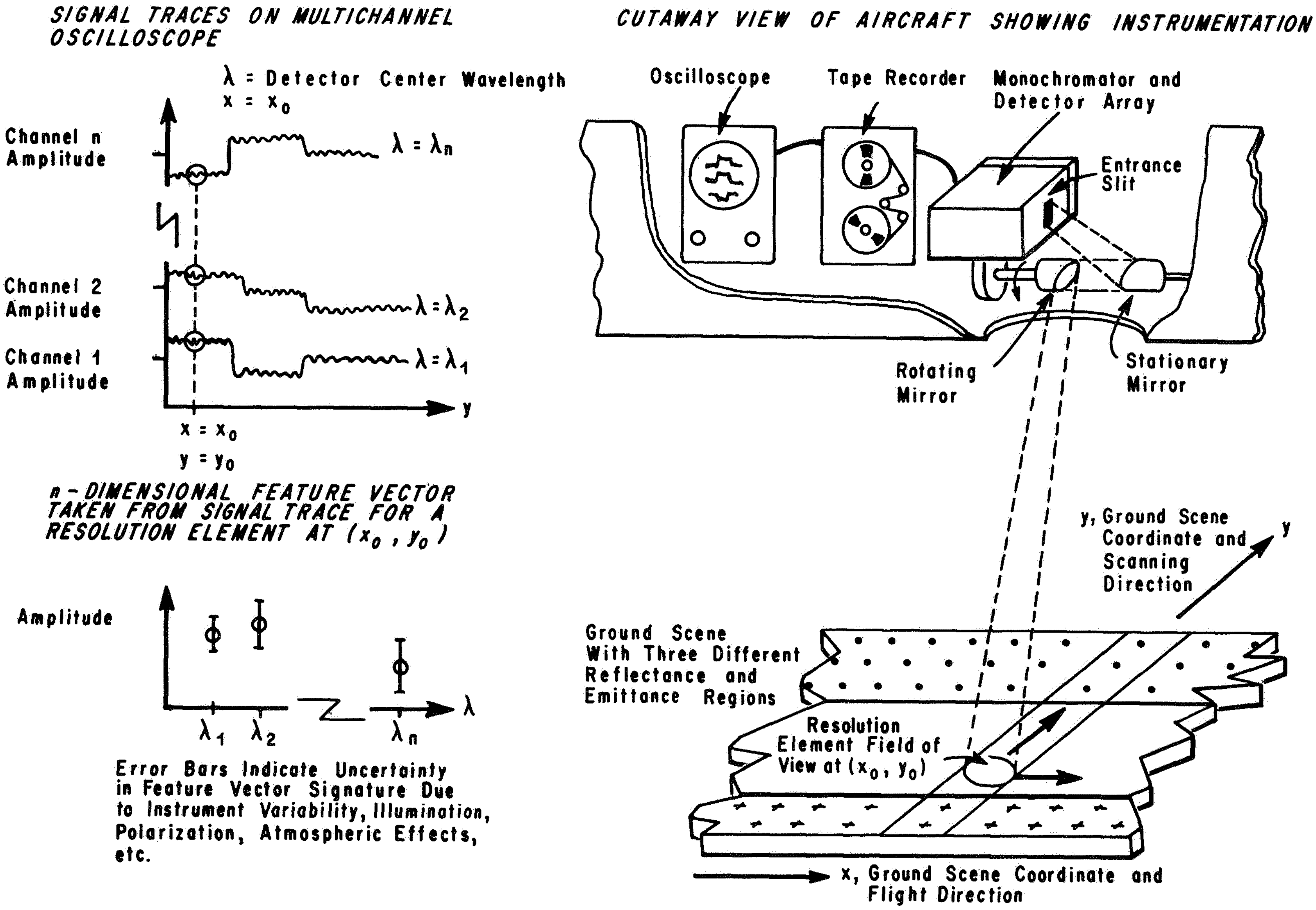
SIGNAL TRACES ON MULTICHANNEL OSCILLOSCOPE
λ = Detector Center Wavelength
x = x_0
Channel n Amplitude
λ = λ_n
Channel 2 Amplitude
λ = λ_2
Channel 1 Amplitude
λ = λ_1
y
x = x_0
y = y_0
n-DIMENSIONAL FEATURE VECTOR TAKEN FROM SIGNAL TRACE FOR A RESOLUTION ELEMENT AT (x_0 , y_0)
Amplitude
λ_1
λ_2
λ_n
λ
Error Bars Indicate Uncertainty in Feature Vector Signature Due to Instrument Variability, Illumination, Polarization, Atmospheric Effects, etc.
CUTAWAY VIEW OF AIRCRAFT SHOWING INSTRUMENTATION
Oscilloscope
Tape Recorder
Monochromator and Detector Array
Entrance Slit
Rotating Mirror
Stationary Mirror
y, Ground Scene Coordinate and Scanning Direction
y
Ground Scene With Three Different Reflectance and Emittance Regions
Resolution Element Field of View at (x_0, y_0)
x, Ground Scene Coordinate and Flight Direction

Figure 1 Data collection platform showing sensor outputs and associated feature vectors.

according to the statistical decision rule as belonging or being similar to the particular training areas selected. The computer map can be compared with the ground-truth information in the aerial photograph for accuracy Accuracies of 80 to 90 percent are not uncommon.

One of the most widely used decision rules is the maximum likelihood ratio technique. The development of this technique depends upon two types of decision errors that can be made in the classification. The first type of error is not assigning a feature vector as belonging to a class when it actually does. The second type of error is assigning a feature vector to a class when in actuality it does not belong. Weights are usually assigned to both types of errors depending on how costly the error is in making the wrong decision. These weights, L_{ij} , are referred to as cost factors and are associated with classifying a feature vector, $\vec{x}$, belonging to class i into class j

Ultimately, the decision procedure must be able to indicate a final choice for each point in the n-dimensional feature space. The feature space, R , must be divided into m mutually exclusive regions, $R_1, R_2, \ldots, R_m$.

Let the probability density function for samples from class i be $f_i(x)$ and the a priori probability of occurence of class i be P_i The probability of misclassifying a sample from class i into class j is

$$P(j|i) = \int_{R_j} f_i(x)dx \qquad , \qquad (1)$$

and the conditional expected cost if the sample is from class i is

$$C_i = \sum_{j=1}^{m} L_{ij}P(j|i) = \sum_{j=1}^{m} L_{ij} \int_{R_j} f_i(x)dx \qquad (2)$$

One useful criterion that is often used is the average cost which is given by

$$C = \sum_{i=1}^{m} P_iC_i = \sum_{j=1}^{m} \int_{R_j} \sum_{i=1}^{m} L_{ij}P_if_i(x)dx \qquad , \qquad (3)$$

and the regions $R_1, \ldots., R_m$ are chosen to minimize the average cost. The average cost is minimized if the integrand

$$\sum_{i=1}^{m} L_{ij} P_i f_i(x)$$

is a minimum, which is equivalent to deciding that a sample x is from class j if

$$\sum_{i=1}^{m} L_{ij} P_i f_i(x) \leqslant \sum_{i=1}^{m} L_{ik} P_i f_i(x) \qquad \text{for all} \quad k \neq j \tag{4}$$

By subtracting

$$\sum_{\substack{i=1 \\ i \neq j,\, k}}^{m} L_{ij} P_i f_i(x)$$

from both sides, the decision rule is obtained. x is in class j if

$$\frac{f_j(x)}{f_k(x)} > \frac{\left(L_{kj} - L_{kk}\right) P_k}{\left(L_{jk} - L_{jj}\right) P_j} \tag{5}$$

Since the cost factors L_{jk} and the a priori probabilities P_j are constants, the decision rule that minimizes the average risk is the ratio of two conditional probability densities or likelihood ratios with a threshold value. The cost factors and a priori probabilities are used only in determining the threshold value. With a simple choice of cost factors $L_{ii} = 0$ for correct classification and $L_{ij} = 1$ with $i = j$ for misclassification and equal a priori probabilities, the decision rule reduces to deciding that x is in class j if

$$f_j(x) > f_k(x) \qquad \text{for all} \quad k \neq j \tag{6}$$

This is generally referred to as the maximum likelihood decision rule.

In actual practice, the cost factors, a priori probabilities, and the probability distributions are not known, but must be estimated from members of each training area. The various classification techniques that have been developed are essentially different methods for estimating the probability distributions.

In most cases, a multivariate Gaussian distribution is assumed for the training area data, and the vector means, $\vec{M}_i$, and covariance matrices, V_i , are calculated from the data in each training area. Because the Gaussian distribution is exponential, the logarithm of the decision rule is used. Thus, decide that $\vec{x}$ belongs to class j for all k = j if

$$\log_e \left|V_j\right| + \left(\vec{x} - \vec{M}_j\right)^T V_j^{-1} \left(\vec{x} - \vec{M}_j\right) \leqslant \log_e \left|V_k\right| + \left(\vec{x} - \vec{M}_k\right)^T V_k^{-1} \left(\vec{x} - \vec{M}_k\right) , \qquad (7)$$

where $\left|V_j\right|$ and V_j^{-1} are the determinant and inverse of the covariance matrix V_j and T denotes the transpose operation of a matrix [16]

The advantage of the supervised technique is that with available ground truth, known areas of particular interest can be used to locate similar areas elsewhere in the data. The disadvantage of the supervised technique is the manual selection of training areas from the data, which, when applied to large data volumes, can become as tedious and as time consuming as the art of photointerpretation. This is especially true of high-altitude multispectral aerial photography and satellite data.

In recent years, the amount of remotely sensed data collected on earth observations has been phenomenal, as evidenced by publications in the literature [17] The future outlook indicates that this data collection will continue to grow, as shown by Earth Resources Technology Satellites 1 and 2, the Skylab Earth Resources Experiment Package, and the Space Shuttle Sortie Laboratory Because the volume of data expected appears to be getting out of hand, much emphasis is being placed on the development of unsupervised techniques for feature extraction These computer techniques are designed to extract features from remotely sensed data without either the assistance and supervision of an observer or the prior benefit of ground-truth information. One promising technique has been developed by Su [18] which uses a sequential variance analysis in combination with an iterative K-mean algorithm approach, and References 19 and 20 give a recent review of the state of the art of other techniques. The usual disadvantages of the unsupervised techniques are that a high degree of classification accuracy is often difficult to obtain and the computer time is relatively extensive. Thus, the specific problem to attack is the development of an unsupervised feature extraction program that classifies with a high degree of accuracy The computer time required can then be more efficiently utilized by optimizing the computer program logic, programing in an efficient machine language, and by using special-purpose computers. The advantage of the unsupervised technique is that no prior information is required before the data analysis. The results obtained with the unsupervised technique are designed to show the location and distribution of the extracted features, but no identification of the features is possible without ground-truth information. Thus, the results of the unsupervised technique can be used for directing ground-truth patrols and the location and amount of ground truth needed can be accurately determined before any ground truth is collected. The adequate determination of where and how much ground truth to collect is an important economic consideration when data are collected on a global or even a regional basis. If data are collected on a seasonal interval, for example, future ground-truth collection can be minimized by monitoring the feature signatures as a function of season. If some of these signatures change significantly from past results, the classification map can be used to direct ground-truth patrols to update only those feature identifications which have changed.

To maintain current information on current accomplishments in remote sensing, the "Proceedings of the International Symposia on Remote Sensing of the Environment," published by the Willow Run Laboratories, University of Michigan, and the "Annual Earth Resources Aircraft Program Status Review," published by NASA, are highly recommended along with the literature survey listed in Reference 17

Section III discusses a proposed unsupervised computer program for feature extraction.

III. UNSUPERVISED FEATURE EXTRACTION

Feature extraction from remotely sensed data is a very complex process. For this reason, no comprehensive model or explicit external criteria exist for defining and extracting all features. The selection of training areas for feature extraction introduces the bias of human observation and preconceived judgment into the classification criteria. For example, an observer may select a training area containing corn to attempt to classify corn in all other portions of the ground scene, while the data from that training area may only be capable of classifying all sparsely growing, small green plants surrounded by bare soil.

The philosophy behind the development of this classification program is to accept tentatively an internal criterion, i.e., the data itself should naturally suggest the features to be extracted and subsequently identified.

A flow chart of successive stages of the computer program is shown in Figure 2.

The first stage of the program is to produce a boundary map of the data by separating the data into homogeneous and inhomogeneous areas. This is accomplished by

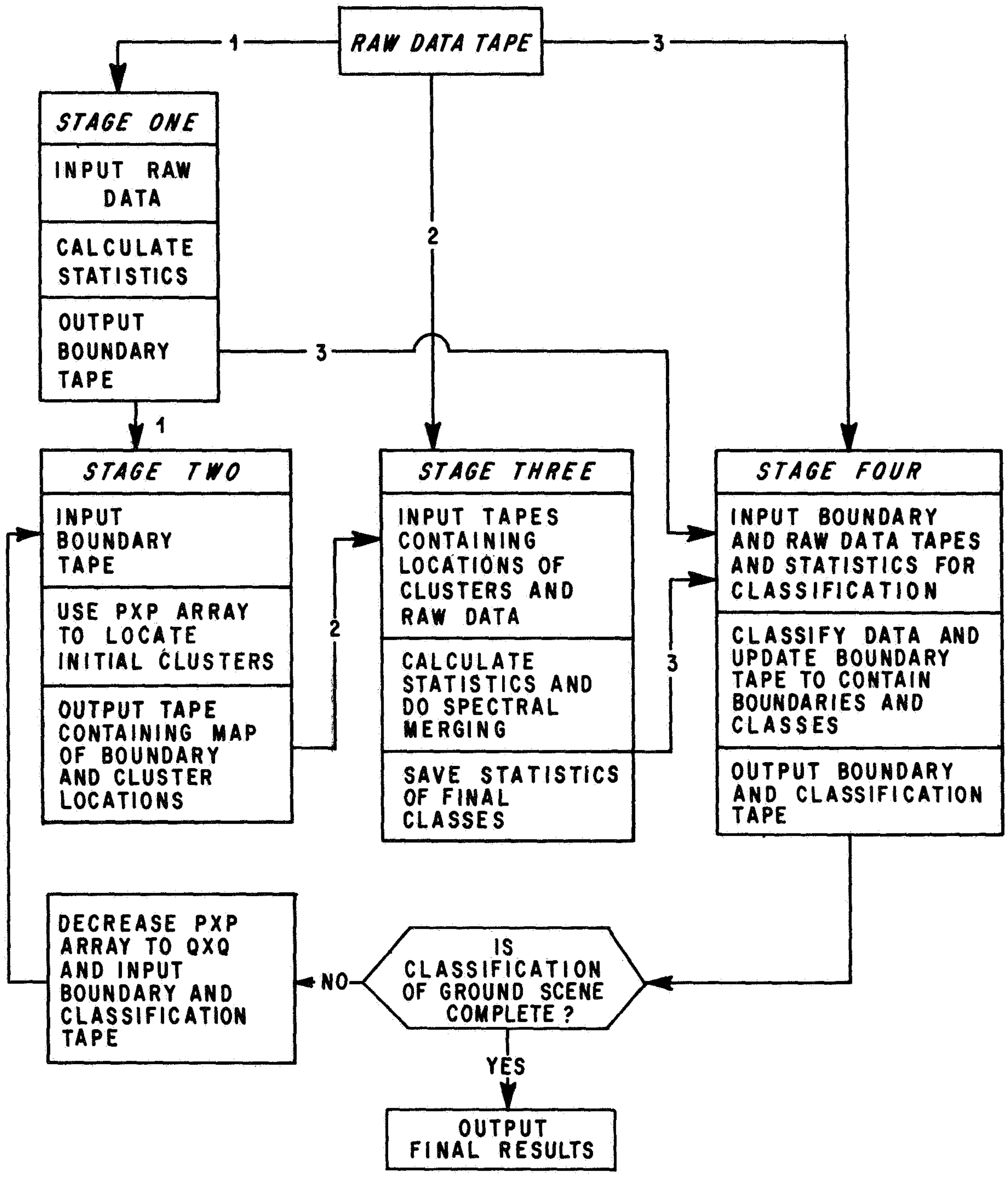
RAW DATA TAPE
1
2
3
STAGE ONE
INPUT RAW DATA
CALCULATE STATISTICS
OUTPUT BOUNDARY TAPE
3
1
STAGE TWO
INPUT BOUNDARY TAPE
USE PXP ARRAY TO LOCATE INITIAL CLUSTERS
OUTPUT TAPE CONTAINING MAP OF BOUNDARY AND CLUSTER LOCATIONS
2
STAGE THREE
INPUT TAPES CONTAINING LOCATIONS OF CLUSTERS AND RAW DATA
CALCULATE STATISTICS AND DO SPECTRAL MERGING
SAVE STATISTICS OF FINAL CLASSES
3
STAGE FOUR
INPUT BOUNDARY AND RAW DATA TAPES AND STATISTICS FOR CLASSIFICATION
CLASSIFY DATA AND UPDATE BOUNDARY TAPE TO CONTAIN BOUNDARIES AND CLASSES
OUTPUT BOUNDARY AND CLASSIFICATION TAPE
DECREASE PXP ARRAY TO QXQ AND INPUT BOUNDARY AND CLASSIFICATION TAPE
NO
IS CLASSIFICATION OF GROUND SCENE COMPLETE ?
YES
OUTPUT FINAL RESULTS

Figure 2. Program logic flow.

computing the average feature vector spectral distance per channel or dimension in the direction of the x and y ground-scene coordinates. The formulas for this calculation are given by

$$s_x = \left[\frac{1}{n}\sum_{k=1}^{n}\left({}_kx_{i,j} - {}_kx_{i-1,j}\right)^2\right]^{1/2} \quad \text{and} \quad s_y = \left[\frac{1}{n}\sum_{k=1}^{n}\left({}_kx_{i,j} - {}_kx_{i,j-1}\right)^2\right]^{1/2} \tag{8}$$

where i and j are the scan line and scan-line column number, respectively, of the data in the digital image and the summation is over n channels of data.

These spectral distances are calculated for each resolution element and stored in a joint probability distribution, $P(s_x, s_y)$. The average feature vector distance per dimension is computed to determine a measure of change present in the data from one resolution element to the next and also to keep the numbers occurring in the calculation in the same range, regardless of the number of dimensions used. The areas of the data where the spectral change, in either the x or y ground-scene coordinate direction, is equal to or less than the average spectral change will be classified as a homogeneous resolution element, otherwise, the resolution element will be classified as a boundary The average distances of the average feature vector spectral distance per channel are computed using the formulas

$$\overline{s_x^2} = \iint s_x^2 \, P(s_x,s_y) \, ds_x ds_y \quad , \tag{9}$$

$$\overline{s_y^2} = \iint s_y^2 \, P(s_x,s_y) \, ds_x ds_y \quad , \tag{10}$$

and

$$\overline{s_x s_y} = \iint s_x s_y \, P(s_x,s_y) \, ds_x ds_y \tag{11}$$

These calculations are used for defining a decision boundary in the joint probability distribution as to what combination of x and y spectral distances is classified as belonging to a homogeneous resolution element or a boundary resolution element. The decision boundary is in the shape of an ellipse which may have principal axes in directions other than s_x and s_y The direction and magnitude of the principal axes are calculated [21] from the eigenvectors and eigenvalues associated with the matrix

$$\begin{bmatrix} \overline{s_x^2} & \overline{s_x s_y} \\ \overline{s_x s_y} & \overline{s_y^2} \end{bmatrix} \qquad (12)$$

The equation of the ellipse in the principal axes coordinate system is

$$\frac{s_x'^2}{\overline{s_x'^2}} + \frac{s_y'^2}{\overline{s_y'^2}} = 1 \quad , \qquad (13)$$

where s_x' and s_y' are the distances s_x and s_y rotated into the principal axes coordinate system by the eigenvector transformation. $\overline{s_x'^2}$ and $\overline{s_y'^2}$ are the eigenvalues or half lengths of the principal axes squared

$$\begin{bmatrix} \frac{1}{\overline{s_x'^2}} & 0 \\ 0 & \frac{1}{\overline{s_y'^2}} \end{bmatrix} \qquad (14)$$

is then rotated back into the s_x and s_y coordinate system by the inverse eigenvector transformation to obtain the equation of the ellipse on the decision boundary in the s_x and s_y coordinate system The equation of the ellipse is given by

$$a s_x^2 + b s_y^2 + c s_x s_y = 1 \quad , \qquad (15)$$

where a, b, and c are determined from rotating equation (14). The decision is to classify a resolution element as being homogeneous if

$$a s_x^2 + b s_y^2 + c s_x s_y \leqslant 1 \qquad (16)$$

A digital image of a boundary map is recorded on magnetic tape for use in the second stage of processing. The magnetic tape utilizes -1 to describe a boundary element and zero for a homogeneous element.

The second stage is concerned with the selection and spatial merging of unknown candidate features based upon the homogeneity of the ground scene, as displayed by the boundary map recorded on the magnetic tape. Because the boundaries on the map are not closed and have open gaps in some cases, the problem is to select homogeneous areas with a mathematical logic that will prevent these areas from containing a mixture of different features. This is accomplished by using a fixed shape p x p resolution element array which moves through the boundary map only in the x or y ground-scene coordinate direction. Initially, a homogeneous area in the boundary map is found which is large enough for the array to fit into. The area covered by the array is designated as belonging to cluster 1 The array is allowed to move in this area until a boundary is encountered and then the direction of movement must be changed. All resolution elements falling within the movement of the fixed array are said to belong to cluster 1, and the zeros previously occurring on the boundary map in these locations are changed to +1 After the array can no longer move and engulf new resolution elements, another location is found which will contain the p x p array All resolution elements fitting into this array will be designated as belonging to cluster 2. The process is repeated until all of the boundary map data have been exhausted Clusters which physically touch on the boundary map are merged and called the same cluster This is called spatial merging, which will be contrasted with spectral merging later After spatial merging, the clusters are renumbered so that the cluster numbers will have a continuous range from 1,...,N The output of this second stage is another map containing the numbers -1, 0, 1, 2,...,N, with -1 indicating boundary resolution elements, 0 indicating homogeneous resolution elements not encountered by the moving fixed shape array, and 1, 2,...,N indicating resolution elements belonging to clusters 1, 2,...., N, respectively The fixed-shape array, if chosen large enough, will not permit the mixing of features because the open gaps in the boundaries will be so small compared to the array size that the array will not be able to pass through the boundary On multispectral scanner data taken at an altitude of 792.5 m (2600 ft), a 10 x 10 array is sufficient to keep different features separate. This also provides a minimum sample size of 100, which is a very adequate sample on which to base statistical calculations. Typically, most multispectral data are collected at higher altitudes so that a 10 x 10 array will, in general, be sufficient to prevent the mixing of features. The magnetic tape containing the boundaries and locations of clusters and the magnetic tape containing the raw data are the inputs to the third stage of processing.

The third stage of processing is concerned with spectral merging of the selected unknown candidate features. The boundary and cluster map tape gives the locations of the raw data on the raw data tape belonging to each cluster The mean feature vectors and covariance matrices, c, are calculated for each cluster using the formulas

$$ {}_{k}\overline{x}_{\ell} = \frac{1}{N_{\ell}} \sum_{i=1}^{N_{\ell}} {}_{k}x_{i\ell} \qquad (17) $$

and

$$ {}_{km}c_{\ell} = \frac{1}{N_{\ell}} \sum_{i=1}^{N_{\ell}} {}_{k}x_{i\ell} \, {}_{m}x_{i\ell} - \left({}_{k}\overline{x}_{\ell}\right) \left({}_{m}\overline{x}_{\ell}\right) \quad , \qquad (18) $$

where x is the algebraic value of the i^{th} sample in the k or m^{th} channel of the ℓ^{th} cluster containing N_{ℓ} samples. The mean value of the samples in the k^{th} channel of the ℓ^{th} cluster is given by ${}_{k}\overline{x}_{\ell}$, while the covariance between channels k and m of the ℓ^{th} cluster is given by ${}_{km}c_{\ell}$ These calculations are used to define decision boundaries with which to physically surround the data belonging to a cluster in n-dimensional space. The most general closed surface that can be used to surround the n-dimensional data is an n-dimensional hyperellipse. The centroid of the cluster ellipse is given by the feature vector mean values ${}_{k}\overline{x}_{\ell}$, while the principal axes direction and magnitude can be determined by the eigenvector transformation as was discussed when equations (12), (13), and (14) were introduced.

In a statistical sense, the eigenvector transformation on the covariance matrix locates the direction of orthogonal principal axes for which the variances are minimized and maximized [22]. The variances are the diagonal elements of the covariance matrix and the off-diagonal elements (covariances) are made zero by the transformation. Thus, the equation of an n-dimensional ellipse in reduced form is obtained for each cluster, and, in general, each cluster will have a different coordinate system. The next step is to derive a decision rule for determining how many clusters actually represent the same feature. This decision is now based entirely upon spectral information rather than the spatial information which was used in equations (12), (13), and (14). The decision rule is that two clusters represent the same feature if the centroids of both clusters are contained in both clusters' ellipses. Although this spectral-merging procedure was derived from physical intuition, a theorem [23] was located which adds mathematical precision. The theorem states that the orthogonal transformation which minimizes the mean-square distance between a set of vectors from the ℓ^{th} cluster, subject to the constraint that the volume of the space is invariant under transformation, is a rotation, E_{ℓ} , followed by a diagonal transformation, W_{ℓ} The rows of the matrix E_{ℓ} are the eigenvectors of the covariance matrix, c_{ℓ} , of the set of vectors, and the elements of W_{ℓ} are those given in equation (19), where ${}_{kk}c_{\ell}{}^{1/2}$ is the standard deviation of the coefficients of the set of vectors in the direction of the k^{th} eigenvector of c_{ℓ}

The diagonal elements of the diagonal transformation, W_{ℓ} , are given by

$$ {}_{jj}w_{\ell} = \left(\prod_{k=1}^{n} {}_{kk}c_{\ell}{}^{1/2} \right)^{1/n} \frac{1}{{}_{jj}c_{\ell}{}^{1/2}} \tag{19}$$

The rationale behind this theorem [23] merits some discussion as applied to spectral merging. This discussion will also apply to spectral classification, which is to be described later

It is desirable to have a measure of similarity between two clusters, S^{-1}, for deciding whether or not two clusters are to be merged. Let $\vec{v}$ be the mean feature vector of one cluster, with components given by equation (17), and let $\left\{ \vec{x}_m \right\}$ be the entire set of feature vectors contained in the other cluster with the subscript m denoting the mth vector of the set. The similarity may be regarded as a mean square spectral distance and should describe the closeness of $\vec{v}$ to the entire set of feature vectors, $\left\{ \vec{x}_m \right\}$ According to the philosophy of Reference 23, the definition of "distance" does not necessarily mean Euclidean distance, but may mean "closeness" in some arbitrary, abstract property of the set $(\vec{x}_m)$ which has yet to be determined. The use of an undetermined distance measure does not alter the definition of similarity, but provides an ordering which similarity lacks. Mathematically the similarity $S^{-1}\,[\vec{v},(\vec{x}_m)]$ of a feature vector $\vec{v}$ and a set of feature vectors $(\vec{x}_m)$ exemplifying a cluster can be written as

$$ S^{-1}\left[\vec{v},\left(\vec{x}_m\right)\right] = \frac{1}{M}\sum_{m=1}^{M} d^2\left(\vec{v},\vec{x}_m\right) \quad , \tag{20}$$

where the distance measure, $d^2(\quad)$, has not yet been specified The inverse of S is used because the smaller the distance, the more the similarity

One possible way to choose the distance measure is to utilize the knowledge that the set $(\vec{x}_m)$ describes one feature, and, therefore, the members of the set should all be very similar or close. Furthermore, the members of the set could be made even more similar by using feature weighting coefficients, ${}_{kk}w_{\ell}$, and minimizing the average intercluster distance of all the members of the set. Thus, the equation to be minimized is

$$\overline{D_\ell^2} = \frac{1}{M_\ell\,(M_\ell - 1)} \sum_{p_\ell = 1}^{M_\ell} \sum_{m_\ell = 1}^{M_\ell} \sum_{k = 1}^{n} {}_{kk}w_\ell^2 \left({}_kx_{m_\ell} - {}_kx_{p_\ell}\right)^2 = \text{minimum}, \qquad (21)$$

where k represents the kth component of the feature vector and m_ℓ and p_ℓ represent the mth and pth feature vector of the set ℓ containing M_ℓ members.

To minimize equation (21), some constraint must be placed on the feature weighting coefficients. Several alternatives are possible, but the most appealing constraint is

$$\prod_{k=1}^{n} {}_{kk}w_\ell = 1 \qquad (22)$$

If the feature weighting coefficients are considered to be the dimensions of a hypercube, then equation (22) is a constant volume constraint. To minimize $\overline{D_\ell^2}$, equation (21) needs to be written in a more convenient form,

$$D_\ell^2 = \frac{M_\ell}{M_\ell - 1} \sum_{k=1}^{n} {}_{kk}w_\ell^2 \left[\frac{1}{M_\ell} \sum_{m_\ell = 1}^{M_\ell} x^2_{m_\ell} + \frac{1}{M_\ell} \sum_{p_\ell = 1}^{M_\ell} {}_kx^2_{p_\ell} - 2 \left(\frac{1}{M_\ell} \sum_{m_\ell = 1}^{M_\ell} {}_kx_{m_\ell} \right) \left(\frac{1}{M_\ell} \sum_{p_\ell = 1}^{M_\ell} {}_kx_{p_\ell} \right) \right] = \frac{2M_\ell}{(M_\ell - 1)} \sum_{k=1}^{n} {}_{kk}w_\ell^2 \left(\overline{{}_kx_\ell^2} - \overline{{}_kx_\ell}^{\,2} \right) \qquad (23)$$

$$= \frac{2M_\ell}{(M_\ell - 1)} \sum_{k=1}^{n} {}_{kk}w_\ell^2\, {}_k\sigma_\ell^2 \quad ,$$

where ${}_k\sigma_\ell^2$ is the variance of the kth dimension of the feature vectors in set ℓ Using the constraint and an undetermined multiplier, the minimizing equation becomes

$${}_{kk}dw_\ell \left({}_{kk}w_\ell\, {}_k\sigma_\ell^2 - \lambda \prod_{j = k}^{n} {}_{jj}w_\ell \right) = 0 \quad \text{for} \quad j = 1, 2, \ldots, n \quad , \qquad (24)$$

which may be rewritten as

$${}_{kk}w_{\ell} = \frac{\sqrt{\lambda}}{{}_{k}\sigma_{\ell}} \tag{25}$$

Using the constraint for determining λ gives

$${}_{kk}w_{\ell} = \left(\prod_{p=1}^{n} {}_{p}\sigma_{\ell} \right)^{1/n} \frac{1}{{}_{k}\sigma_{\ell}} \tag{26}$$

The feature weighting coefficients indicate that if the variance in a particular dimension is small, then the value of the feature can be anticipated with a high degree of accuracy and should be heavily weighed. On the other hand, if the variance is large, little weight should be attached to that feature.

Equation (26) is the diagonal transformation discussed in the theorem encompassing equation (19) and is identical to equation (19). As the theorem states, this distance measure can be additionally minimized by applying the eigenvector transformation. It follows by equations (18) and (21) that the similarity criterion [equation (20)] for deciding whether to merge two clusters k and ℓ is given by

$$\begin{aligned} S^{-1}\left[\vec{v}_{k}, \left(\vec{x}_{m_{\ell}}\right)\right] &= \frac{1}{M_{\ell}} \sum_{m_{\ell}=1}^{M_{\ell}} \sum_{p=1}^{n} {}_{pp}w_{\ell}^{2} \left({}_{p}v_{k} - {}_{p}x_{m_{\ell}}\right)^{2} \\ &= \sum_{p=1}^{n} {}_{pp}w_{\ell}^{2} \left[\left({}_{p}v_{k} - {}_{p}\bar{x}_{\ell}\right)^{2} + {}_{pp}c_{\ell}\right] \quad , \end{aligned} \tag{27}$$

where $\vec{v}_{k}$ is the mean feature vector or centroid for cluster k, and $(\vec{x}_{m_{\ell}})$ are the M_{ℓ} feature vectors belonging to cluster ℓ

Substituting equation (19) for ${}_{pp}w_{\ell}^{2}$ gives

$$S^{-1}\left[\vec{v}_{k},\left(\vec{x}_{m_\ell}\right)\right] = \left(\prod_{j=1}^{n} {}_{jj}c_\ell^{\,1/2}\right)^{2/n}\left[\sum_{p=1}^{n}\frac{\left({}_{p}v_{k} - {}_{p}\bar{x}_\ell\right)^{2}}{{}_{pp}c_\ell} + n\right] \tag{28}$$

The decision for merging two clusters will now depend on the threshold value given to the measures of similarity for both clusters. A threshold value can be determined by calculating the average similarity within a cluster, which is given by

$$\begin{aligned}\overline{S^{-1}\left[\left(\vec{x}_{m_\ell}\right),\left(\vec{x}_{m_\ell}\right)\right]} &= \left(\prod_{j=1}^{n} {}_{jj}c_\ell^{\,1/2}\right)^{2/n}\frac{1}{M_\ell}\sum_{M_\ell=1}^{M_\ell}\left[\sum_{p=1}^{n}\frac{\left({}_{p}x_{m_\ell} - {}_{p}\bar{x}_\ell\right)^{2}}{{}_{pp}c_\ell} + n\right. \\ &= \left(\prod_{j=1}^{n} {}_{jj}c_\ell^{\,1/2}\right)^{2/n} 2n\end{aligned} \tag{29}$$

Using equation (29) as a threshold value for equation (28) gives the decision rule, merge clusters k and ℓ if

$$\sum_{p=1}^{n}\frac{\left({}_{p}v_{k} - {}_{p}x_\ell\right)^{2}}{{}_{pp}c_\ell} \leqslant n \tag{30}$$

Notice that equation (30) is the equation of a hyperellipse in the principal axes coordinates, which was earlier derived by physical intuition. Notice also that the threshold value n is independent of cluster and depends only on the dimension of the feature space. Thus, if an elliptical boundary decision rule is used in the principal axis coordinate system, the theorem can be extended to say that the diagonal transformation is not needed and only the eigenvector transformation is needed since the threshold can always be written as some constant times

$$\left(\prod_{j=1}^{n} {}_{jj}c_\ell^{\,1/2}\right)^{2/n}$$

After the initial M set of clusters has been merged into a final N set of clusters, with $N \leq M$, the eigenvector rotation matrix and the equation for the hyperellipse in principal axes coordinates are stored in memory for each final cluster. The clusters are now called classes since each class represents a statistically different feature presented by the data. The regions in feature space R_1, R_2,...., R_N corresponding to each class are distinct since no more merging is possible. This feature extraction information is now ready for use in the last stage of processing.

The final stage of processing is concerned with classifying the data in the digital image of the ground scene and showing the location and distribution of the features. The inputs to this stage of processing are the raw data tape, the statistics for each class, and the boundary tape. The decision rule for classifying a resolution element feature vector $\vec{v}$ into class ℓ is given by

$$\sum_{p=1}^{n} \frac{\left({}_{p}v - {}_{p}\bar{x}_{\ell}\right)^{2}}{2\,{}_{pp}c_{\ell}} \leqslant n \quad , \tag{31}$$

which is the same as equation (30) except for the factor of 2. The factor of 2 is justified based on the following arguments. First, the decision rule for classifying a resolution element can be more lenient than the decision rule for merging two clusters. The errors of mismerging are more pronounced in the classification than the misclassification of individual resolution elements. Secondly, the expression on the left side of the inequality in equation (31) is identical to the argument in the exponential of a disjoint multivariate Gaussian distribution. Thus, a resolution element is not classified as belonging to a class if the n-dimensional exponential n folds. Further justification for using equations (30) and (31) and the threshold values of n and 2n, respectively, is that the terms contained in the summation are chi-square distributed, and a chi-square distribution has a mean value n and variance 2n, where n is the number of degrees of freedom [24].

The data obtained from the raw data tape are classified according to the decision rule in equation (31), and the result of the classification is updated on the boundary tape map. For example, if a resolution element belongs to class 3, a 3 is placed on the boundary tape at the location of the resolution element. If a resolution element is not classified as belonging to any class, no change is made on the boundary tape. If a resolution element can be classified as belonging to several classes, the resolution element is placed in the class which makes the left side of equation (31) a minimum. The boundary tape is now called a classification tape and contains the numbers -1, representing an unclassified boundary resolution element, 0, representing an unclassified homogeneous resolution element, and 1, 2,...., N, representing resolution elements placed in classes 1, 2,...., N, respectively

If the initial size of the p x p array for cluster selection is too large, an incomplete classification of the ground scene will result. The computer program has the capability for using the classification map as a boundary map, treating the classified resolution elements as also being boundaries, decreasing the p x p array to a q x q array, and selecting additional clusters. All previous information obtained on the classes is updated, if appropriate, and the classification map is reclassified using the old unchanged information and the new updated information. This procedure can be repeated as many times as desired, but two classification passes are generally sufficient.

Because of the large amounts of data involved, the output of the classification map is nominally put on microfilm rather than standard computer paper printout. This is evidenced by the fact that the classification map from a 70-mm aerial photograph, using standard computer printout, can easily cover a 37.16-m^2 (400-ft^2) wall.

The results are given in Section IV

IV. RESULTS

Classification maps were obtained from the analysis of two data sets, Purdue's Flight Line Cl and the Yellowstone Park test sites. The advantage of working with these two sets of data is that they have been extensively analyzed by other investigators using different feature extraction techniques. The results of these investigations are mainly available in References 6, 11, and 12 for comparison.

Both data sets were acquired with the same multispectral scanner and contain 12 channels of data. All 12 channels were used in the classification, and the wavelength intervals corresponding to each channel are listed in Table 1 The data sets contain 222 12-dimensional feature vectors per scan, and 901 scans from each set were analyzed. The data are also uncalibrated and, therefore, only contains relative numbers. The similarity of the two data sets drastically comes to an end with the above description.

Flight Line Cl is a very flat agricultural scene approximately 6.5 km long and 1.6 km wide, and the resolution of the data is approximately 6 m. The ground scene mostly contains rectangular patterns of crop acreage which appear homogeneous.

Yellowstone Park, However, is a wilderness area approximately 16 km long and 3.2 km wide with a resolution of approximately 15 m. The wilderness area contains very irregularly shaped patterns and is quite inhomogeneous in some locations. This data set also possesses a considerable dynamic range in the type of terrain and amount of features present. For example, there are mountains and canyons, and the vegetation coverage ranges from dense forest to scattered trees, meadows, and, finally, to bare rocks. The geologic information ranges from a sand and gravel base, abundantly sprinkled with elk droppings in the meadow areas, to rocks covered with moss and lichen, and, finally, to large boulders.

TABLE 1 CHANNEL AND WAVELENGTH CORRESPONDENCE

Channel No.	Wavelength Interval (μm)
1	0.4 - 0.44
2	0.44 - 0.46
3	0.46 - 0.48
4	0.48 – 0.50
5	0.50 – 0.52
6	0.52 - 0.55
7	0.55 - 0.58
8	0.58 - 0.62
9	0.62 – 0.66
10	0.66 - 0.72
11	0.72 - 0.80
12	0.80 – 1.0

CI Flight Line

Figure 3 contains an aerial photograph of the ground scene on the left, a boundary map in the center with the locations of the initial clusters, and a classification map of the ground scene using statistics from these clusters. The aerial photograph also contains a list of symbols that identifies the contents of the ground scene The list of symbols and their identification are given in Table 2. Examination of the results obtained from the multispectral scanner data reveals a problem that does not occur in multiband photographic data The aircraft acquiring the scanner data was not quite able to fly a straight line and had occasional yaw problems. The scanner data are roll compensated, however, as is the case with most scanners.

The computer maps shown in Figure 3 demonstrate the first two-out-of-three intermediate outputs that can be obtained from the computer program and are considerably scaled down to show that the pattern in the data has a definite resemblance to that of the aerial photograph. Hence, it is not possible or necessary at this stage to be able to read the symbols on the computer maps.

The boundary map contains 79 different cluster locations after spatial merging. A 10 x 10 array was used to select these clusters. The first cluster is at the top of the map just to the right of the road, while the second cluster is to the right and just below cluster 1 Cluster 3 is the top left cluster in a cornfield, while cluster 4 is the first cluster

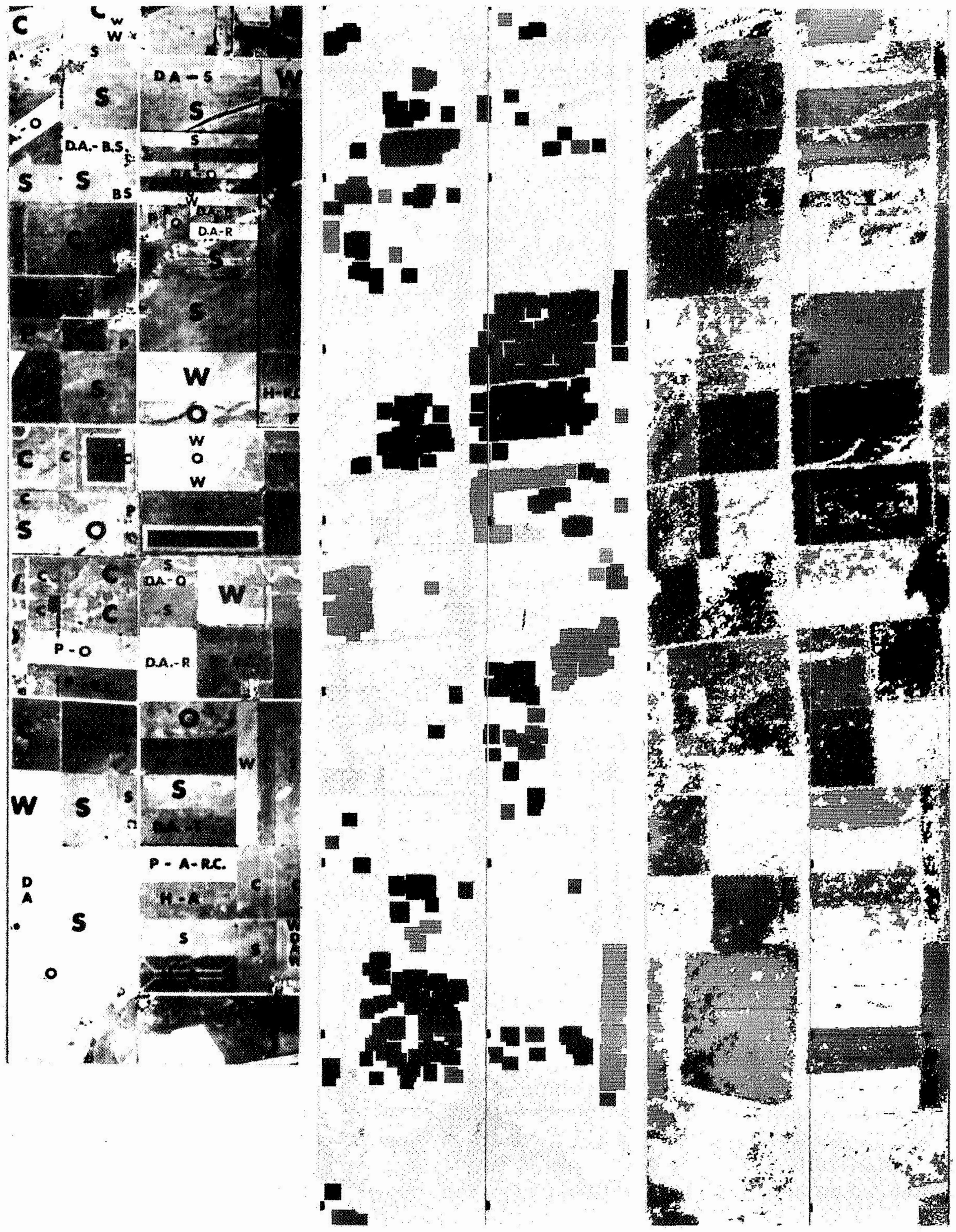

Figure 3. Flight Line Cl – Aerial photograph, cluster selection, and initial classification.

TABLE 2. GROUND TRUTH INFORMATION

Symbol	Description
A	Alfalfa
C	Corn
H	Hay
O	Oats
P	Pasture
R	Rye
S	Soybeans
T	Timothy
W	Wheat
B.S.	Bare Soil
D.A.	Diverted Acres
R.C.	Red Clover

in the soyfield below the wheatfield to the left on the road. Cluster 5 is the first cluster directly below clusters 1 and 2 and is also located in a soyfield Cluster 6 is in the same soyfield as cluster 4 and is just below cluster 4, while cluster 8 is just to the right of cluster 6. The other clusters follow using the same computer pattern logic. Only 43 single character output symbols are available to name the 79 clusters, and, therefore, the symbol usage is recycled starting with cluster 44. Tables 3 and 4 give the original cluster numbers, ground-truth information, a description of the spectral merging process, and the final class number. Remember that when several clusters are spectrally merged, the number given to the merged cluster is the smallest cluster number of the clusters involved in the merging, and some of the remaining clusters not involved in the merging may have their cluster numbers changed to keep the numbering of the clusters in a consecutive order. This is illustrated in the spectral multiple merging columns of Tables 3 and 4. If no ground truth is available for a cluster, the letter U is used to indicate that it is unidentified. Examination of Table 3 multiple merge number 1 reveals that the statistics of clusters 1 and 2 were merged together and the result was called cluster 1 Cluster 3 did not merge with cluster 1, but was renamed cluster 2 so that consecutive numbering would be preserved. Cluster 4 was renamed cluster 3 because it would not merge with clusters 1 and 2, and cluster 5 was renamed cluster 4 because it would not merge with clusters 1, 2, or 3 Cluster 6 would merge with cluster 3 and the statistics of cluster 3 were updated to include the statistics of cluster 6. Finally, cluster 23 was able to merge with clusters 8, 9, 11, and 12, and the statistics of cluster 8 were updated to include the statistics of clusters 9, 11, 12, and 23. The renaming of the clusters resulting from this multiple merging is shown in multiple merge column number 2, and this column continues until another multiple merge is encountered. The classification map resulting from the merging and first classification pass is shown on the right side of Figure 3. This

TABLE 3. MERGING PROCEDURE FOR FIRST 40 CLUSTERS

Cluster No.	Identification	Spectral Multiple Merge No.				Final Class No.
		1	2	3	4	
1	U	1	1	1	1	1
2	U	1	1	1	1	1
3	C	2	2	2	2	2
4	S	3	3	3	3	3
5	S	4	4	4	4	4
6	S	3	3	3	3	3
7	S	4	4	4	4	4
8	S	3	3	3	3	3
9	S	4	4	4	4	4
10	S	5	5	5	5	5
11	DA-BS	6	6	6	6	6
12	C	7	7	7	7	7
13	C	7	7	7	7	7
14	C	7	7	7	7	7
15	U	8	8	8	8	8
16	S	9	8	8	8	8
17	S	3	3	3	3	3
18	BS	6	6	6	6	6
19	S	10	9	3	3	3
20	U	8	8	8	8	8
21	C	11	8	8	8	8
22	C	12	8	8	8	8
23	C	8, 9, 11, 12	8	8	8	8
24	C		10	9	9	9
25	S		7	7	7	7
26	S		4	4	4	4
27	S		7	7	7	7
28	W		11	10	10	10
29	S		3,9	3	3	3
30	C			11	11	11
31	O			12	12	12
32	O			12	12	12
33	C			13	13	13
34	W			10	10	10
35	W			10	10	10
36	O			12	12	12
37	U			14	14	14
38	W			10	10	10
39	C			15	3	3
40	W			10	10	10

TABLE 4. MERGING PROCEDURE FOR CLUSTERS 41 THROUGH 79

Cluster No.	Identification	Spectral Multiple Merge No.				Final Class No.
		4	5	6	7	
41	C	14	14	14	14	14
42	U	15	14	14	14	14
43	S, C	3	3	3	3	3
44	U	14, 15	14	14	14	14
45	O		15	15	15	15
46	W		10	10	10	10
47	S		16	16	16	16
48	U		16	16	16	16
49	DA-R		17	17	17	17
50	DA-R		18	17	17	17
51	P-O		19	18	18	18
52	DA-R		17, 18	17	17	17
53	DA-R			17	17	17
54	DA-R			17	17	17
55	O			19	19	19
56	O			19	19	19
57	C			8	8	8
58	C			20	20	20
59	C			8	8	8
60	S			21	21	21
61	S			22	21	21
62	S			21, 22	21	21
63	S				21	21
64	S				21	21
65	C				7	7
66	S				22	22
67	S				7	7
68	S				5	5
69	S				5	5
70	S				5	5
71	S				3	3
72	S				5	5
73	S				23	23
74	U				1	1
75	S				22	22
76	S				23	23
77	S				24	24
78	U				3	3
79	U				3	3

map contains 24 classes, or features, as indicated by this highest number in the last column of Table 4. Because the classification was incomplete, additional clusters were selected by a 6 x 6 array starting with cluster number 25, and using the classification map as input for the cluster selection rather than the boundary map. The merging procedure is listed in the last column of Table 5 since no multiple merges were encountered. The computer program only allows for 43 classes and, therefore, refused to accept any additional data after cluster 48, as indicated in Table 5. Table 6 lists the initial cluster numbers of all clusters that are located in the same field for both classification passes. This provides an additional check to determine whether the merging was conducted properly and to assist in locating the first classification pass clusters shown in Figure 3 if desired. The reason that one field may have several clusters is that boundary points within a field may not permit spatial merging of these clusters. Spectral merging is used to overcome this problem.

TABLE 5 MERGING PROCEDURE FOR CLUSTERS 25 THROUGH 52

Cluster No.	Identification	Final Class No.
25	WATER	25
26	C	26
27	C	26
28	DA-RC	27
29	S	28
30	S	29
31	DA-RC	30
32	DA-RC	31
33	P	32
34	DA-RC	33
35	C	34
36	DA-RC	35
37	DA-RC	30
38	P	36
39	C	37
40	C	37
41	C	38
42	H-RC	30
43	H-RC	30
44	C	39
45	O	40
46	O	41
47	O	42
48	U	43
49	RC	
50	RC	
51	O	
52	RC	

TABLE 6. MULTIPLE CLUSTER FIELDS

Classification Pass 1-10x10 Array		Classification Pass 2-6x6 Array	
Initial Cluster No.	Identification	Initial Cluster No.	Identification
1,2	U	26,27	C
4,6,8	S	31,32,34,36,37	DA-RC
5,7,9	S	39,40	C
12,13,14	C	42,43	H-RC
17,19	S	45,46,47,51	O
21,22,23,24	C		
25,27	S		
31,32	O		
34,35,38,40	W		
49,50,52,53,54	DA-R		
55,56	O		
57,58,59	C		
60,63,64	S		
66,73,75,76	S		
67,77	S		
68,69,70,71,72	S		

Table 7 lists the final class number, computer symbol printout, and a brief description of the class or feature based upon the available ground truth.

A user may now desire to interpret the results for his specific needs, which, for example, may be crop identification. Table 8 was prepared for this example and for examining the final results. Classes 1, 7, D, and / are not listed in the table because a specific crop name or feature could not be attached. Notice that classes 7, D, and / occur at the edges of the computer map.

Figures 4 through 15 are the final classification results for flight line Cl, and Figures 4 through 9 correspond to the left side of the aerial photograph, while Figures 10 through 15 correspond to the right side.

In'Figure 4, water is represented by the letter O and wheat by the number 0. The 0's have a slightly rectangular shape as compared to the letter O. Figures 7 through 9 start to show several areas that were not classified. This is because the maximum of 43 classes was obtained in the area of Figures 7 and 13, and the remaining unclassified resolution elements were significantly different from the 43 previously obtained features.

TABLE 7 FEATURE SYMBOL AND DESCRIPTION

Class No.	Computer Symbol	Brief Description and Comments
-1		Unclassified boundary resolution element
0		Unclassified nonboundary resolution element
1	1	Unidentified - Classified as corn in Reference 6
2	2	Corn
3	3	Mixture - 84% soy, 8% corn,and 8% unidentified
4	4	Soy beans
5	5	Soy beans
6	6	Bare soil
7	7	Mixture - 51% soy and 49% corn
8	8	Mixture - 73% corn and 27% soy
9	9	Corn
10	0	Wheat
11	A	Corn
12	B	Oats
13	C	Corn
14	D	Undecided - Probably corn
15	E	Oats
16	F	Probably all soy - 89% soy and 11% unidentified
17	G	Diverted acres and rye
18	H	Pasture and Oats
19	I	Oats
20	J	Corn
21	K	Soy beans
22	L	Soy beans
23	M	Soy beans
24	N	Soy beans
25	O	Water
26	P	Corn
27	Q	Diverted acres and red clover
28	R	Soy beans
29	S	Soy beans
30	T	Diverted acres and red clover
31	U	Diverted acres and red clover
32	V	Pasture
33	W	Diverted acres and red clover
34	X	Corn
35	Y	Diverted acres and red clover
36	Z	Pasture
37	=	Corn
38	$	Corn
39	'	Corn
40	,	Oats
41	(	Oats
42	*	Oats
43	/	Unidentified

TABLE 8. USER INTERPRETATION OF RESULTS

Category	Computer Map Symbol
Corn	2,8,9,A,C,J,P,X,=,$,,
Soy	3,4,5,F,K,L,M,N,R,S
Bare soil	6
Wheat	0
Oats	B,E,H,I,',(,*
Diverted acres – Rye	G
Diverted acres – Red Clover	Q,T,U,W,Y
Pasture	V,Z
Water	0

According to Reference 6, the majority of misclassification and nonclassification can be attributed to weed growth and low-lying areas within different fields. This probably accounts for the appearance of boundaries in the fields presented in these results, e.g., Figure 10 contains a horizontally presented wheatfield near the bottom with a misclassification of oats in the middle. This misclassification is caused by the presence of a low-lying area. The boundaries in the diverted acres-red clover field just above this wheatfield are caused by the presence of a small sand dune.

It may be of interest to note that there are usually more features representing a row crop such as corn and soy and that the non-row crop fields such as wheat and oats are more homogeneously classified per field. This result is probably because of the sensor having to average not only over the canopy structure of a row crop but, in addition, averaging over a percentage of bare soil observable between the plants. The type of soil could also vary from location to location.

Yellowstone Park

Figure 16 contains a video reprint from channel 9 of the ground scene on the left, a boundary map in the center with the location of the initial clusters, and a first pass classification map with an additional selection of clusters.

Figure 4. Final Cl classification map – Section 1.

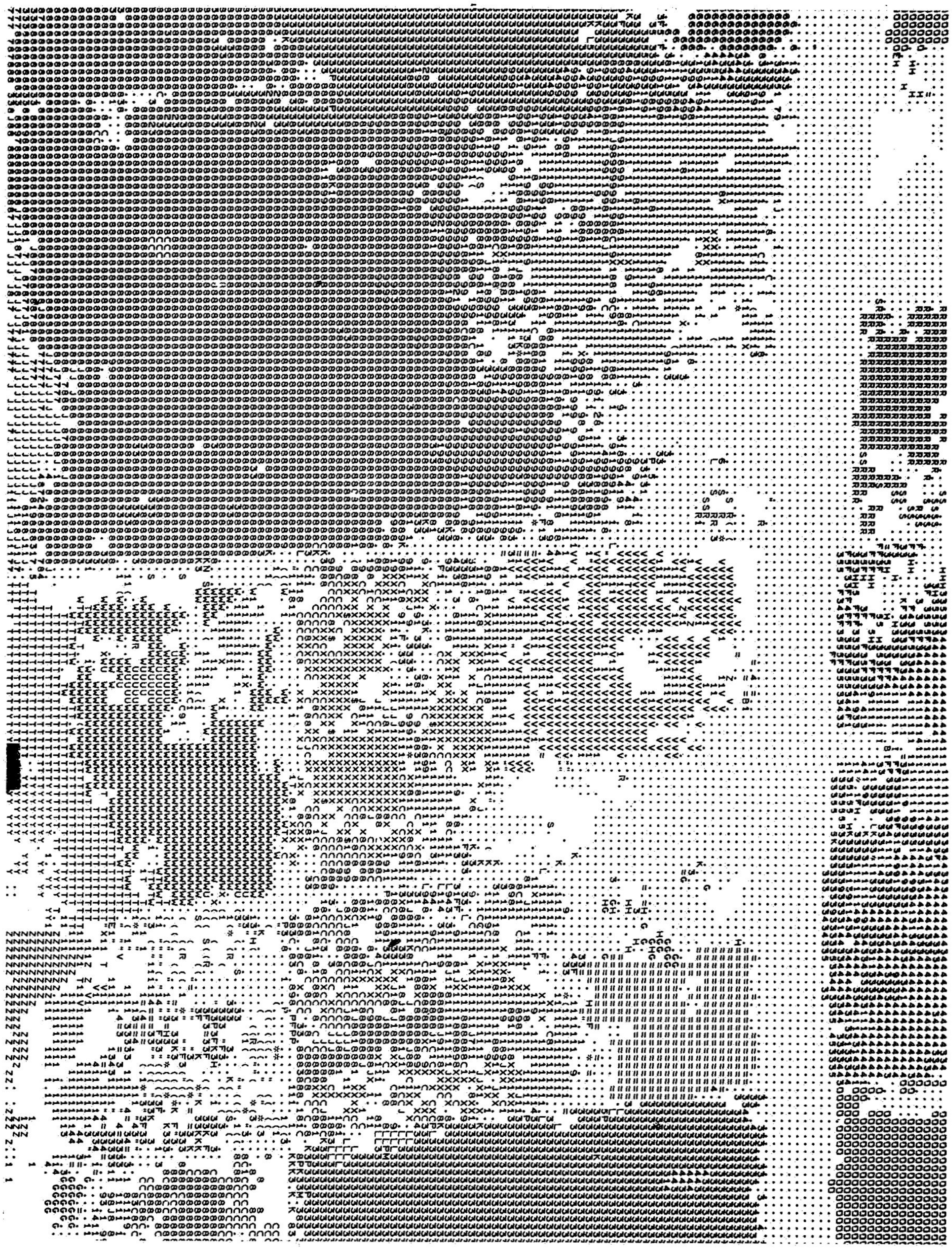

Figure 5. Final Cl classification map – Section 2.

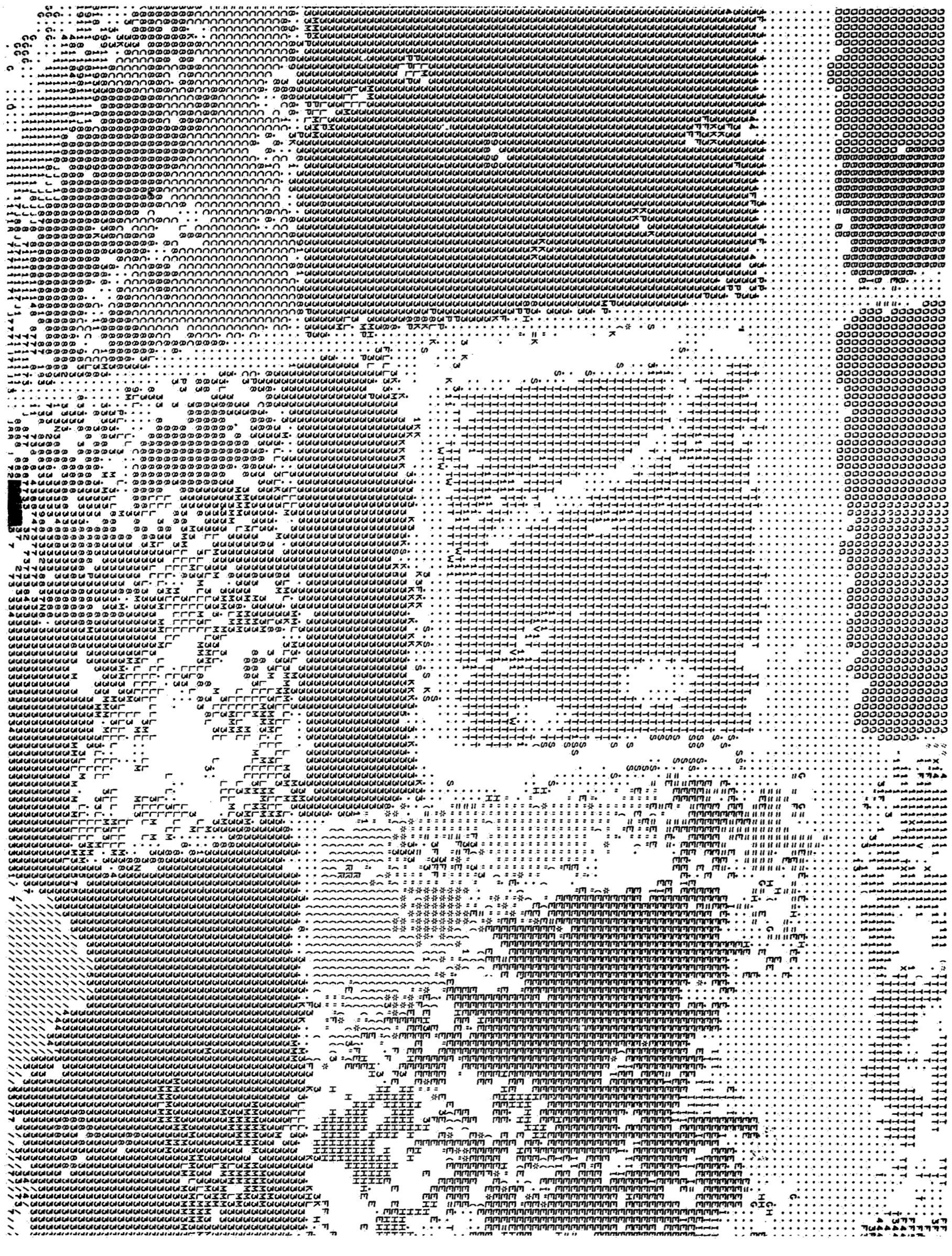

Figure 6. Final Cl classification map – Section 3.

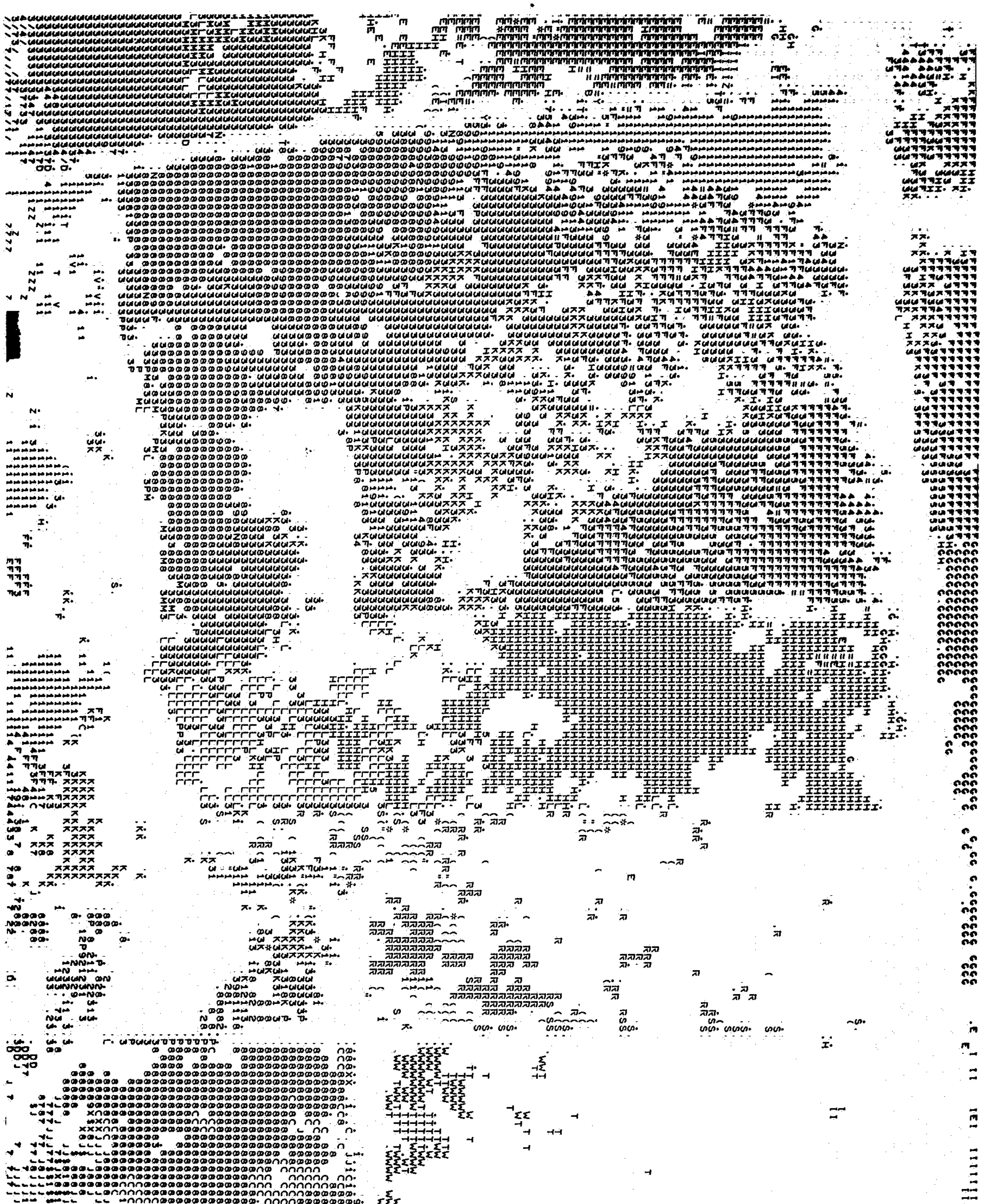

Figure 7. Final Cl classification map – Section 4.

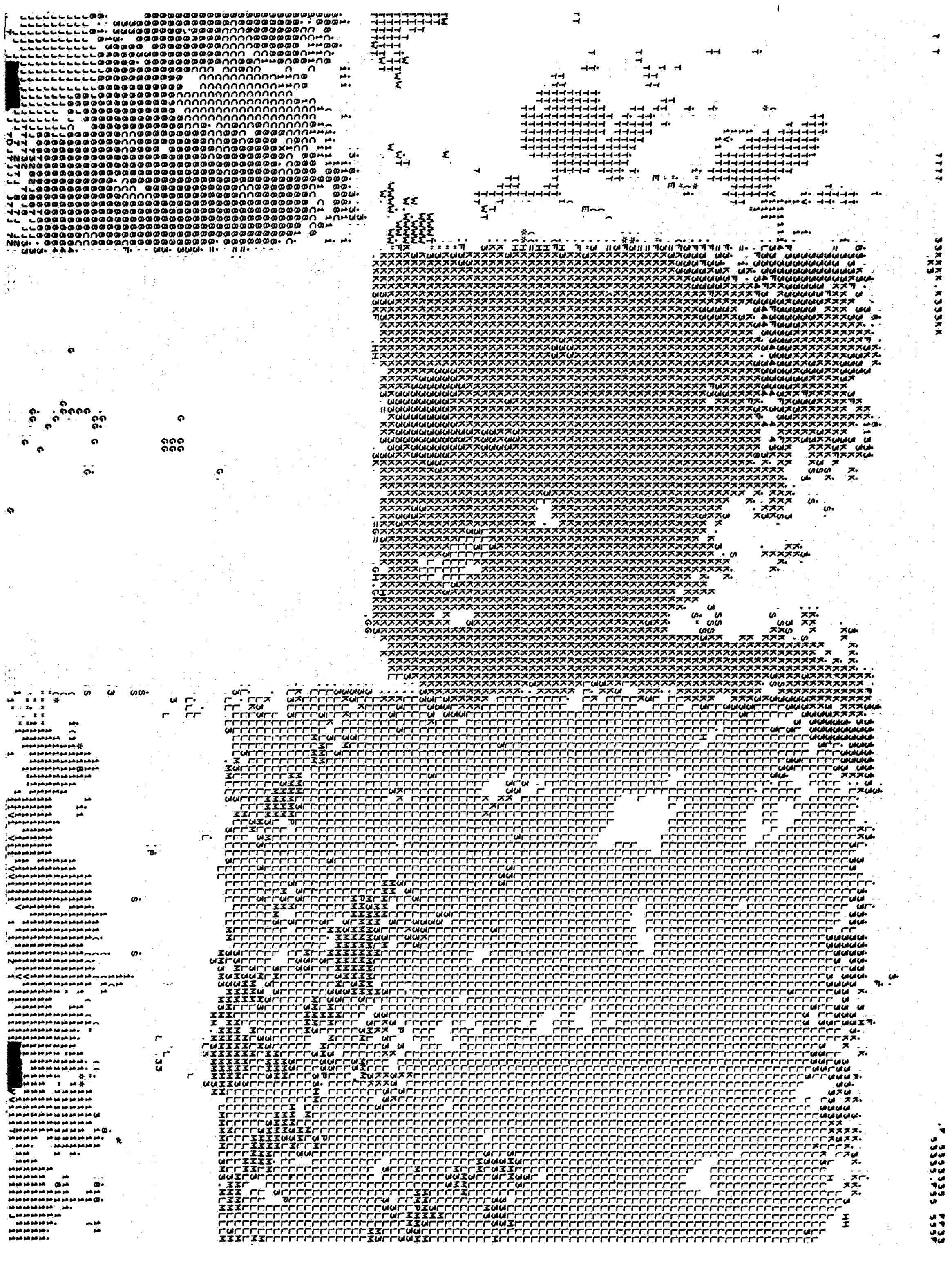

Figure 8. Final Cl classification map – Section 5

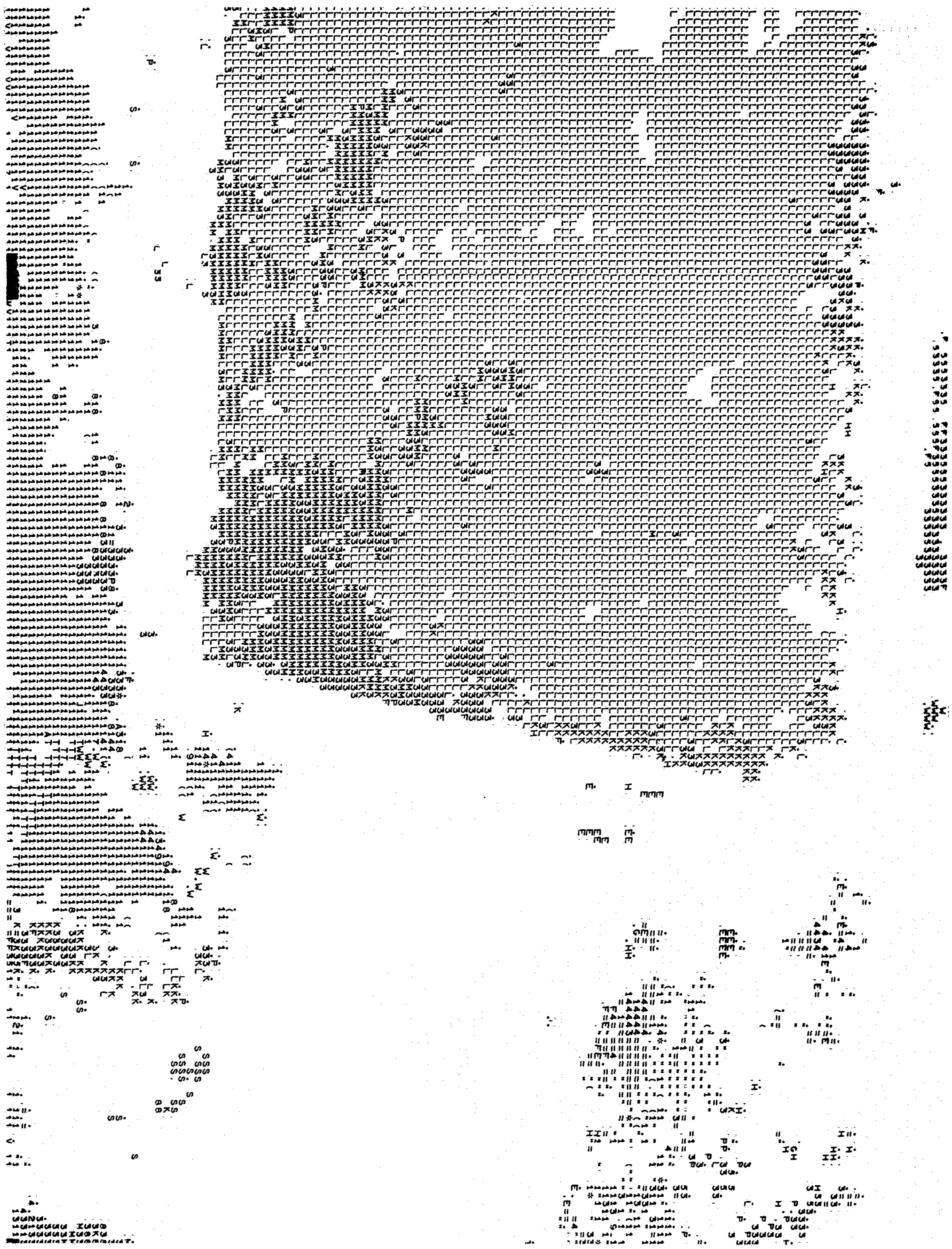

Figure 9. Final Cl Classification map – Section 6.

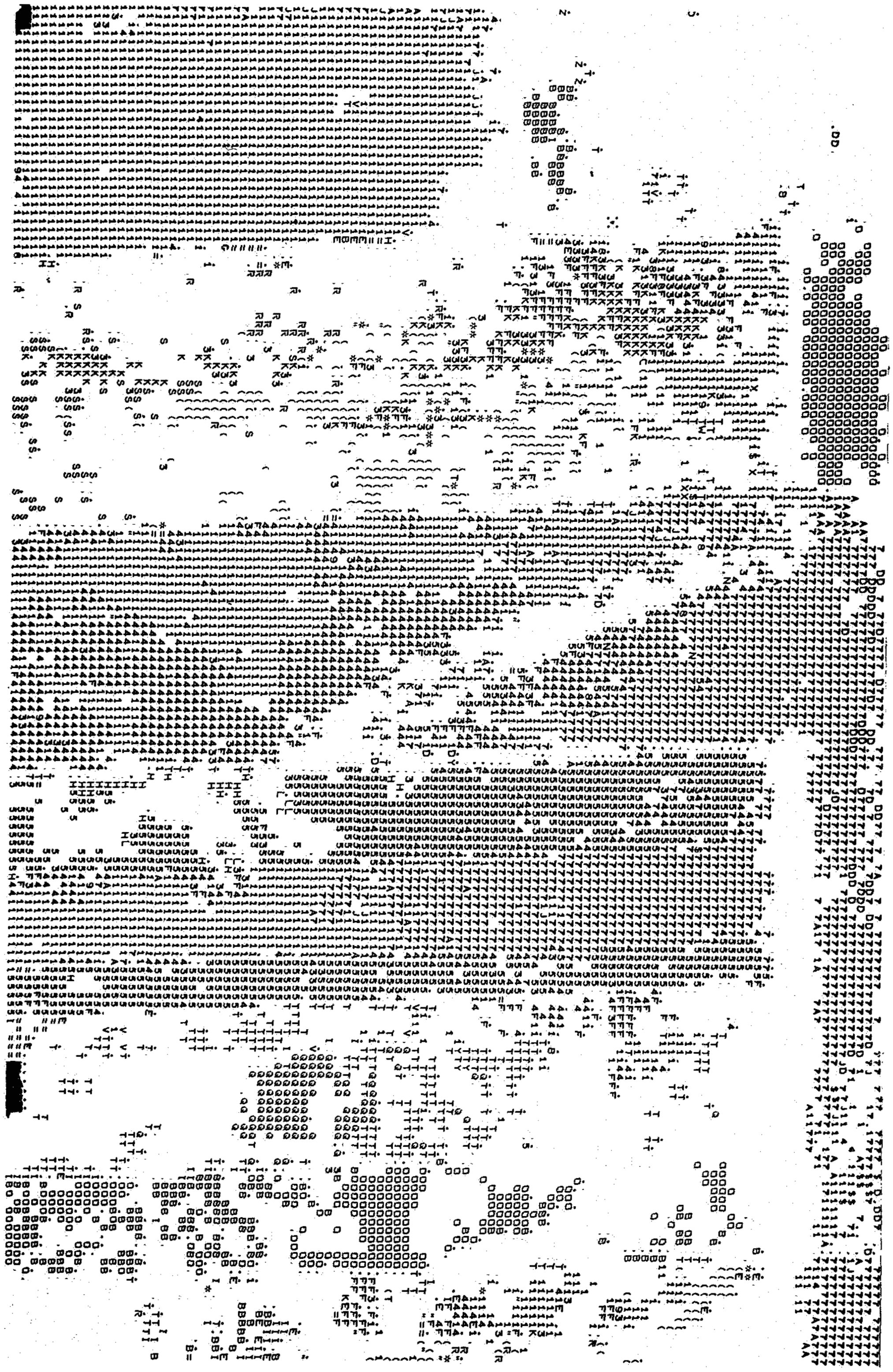

Figure 10. Final Cl classification map – Section 7

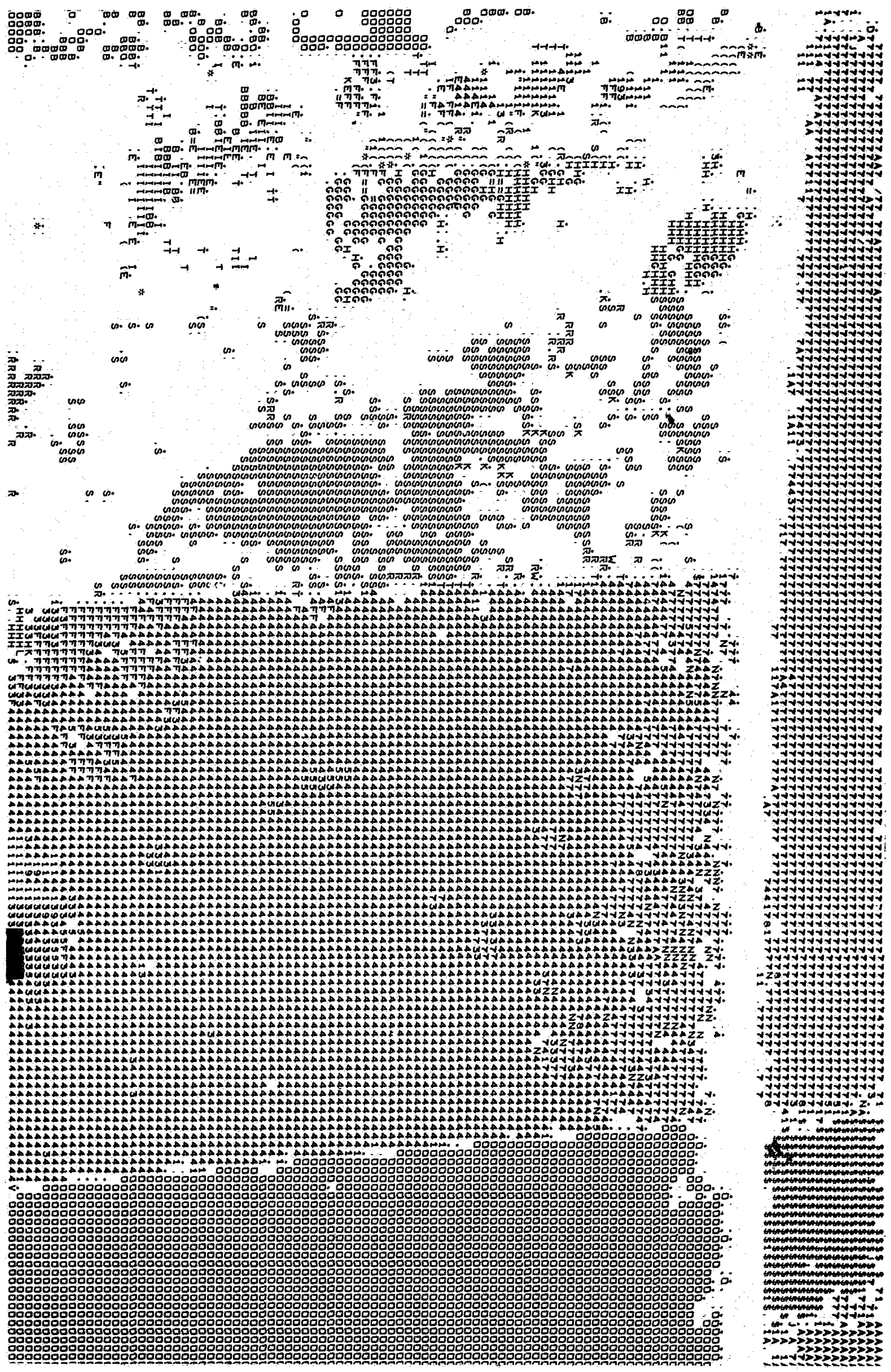

Figure 11. Final Cl classification map – Section 8.

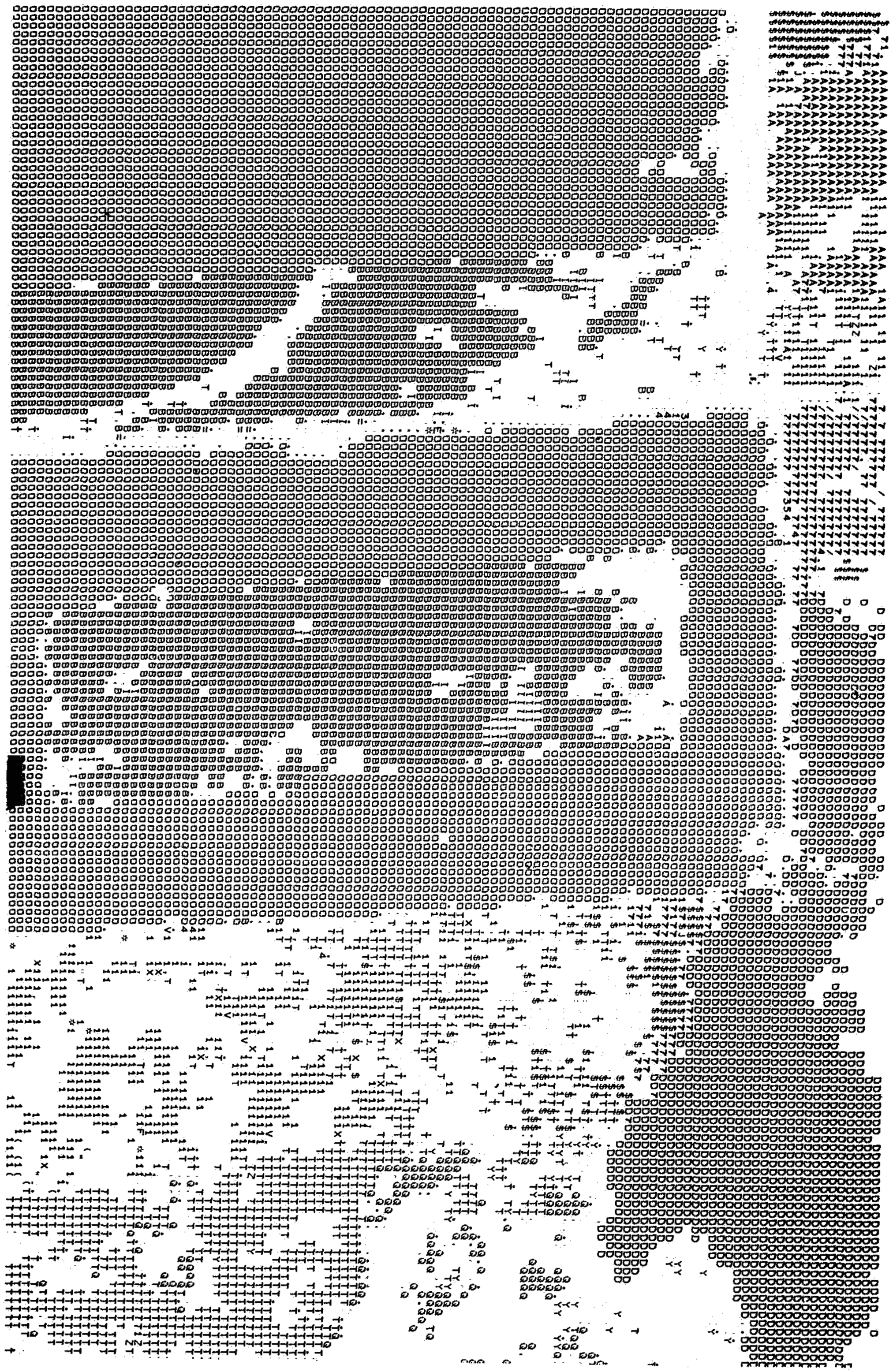

Figure 12. Final Cl classification map – Section 9.

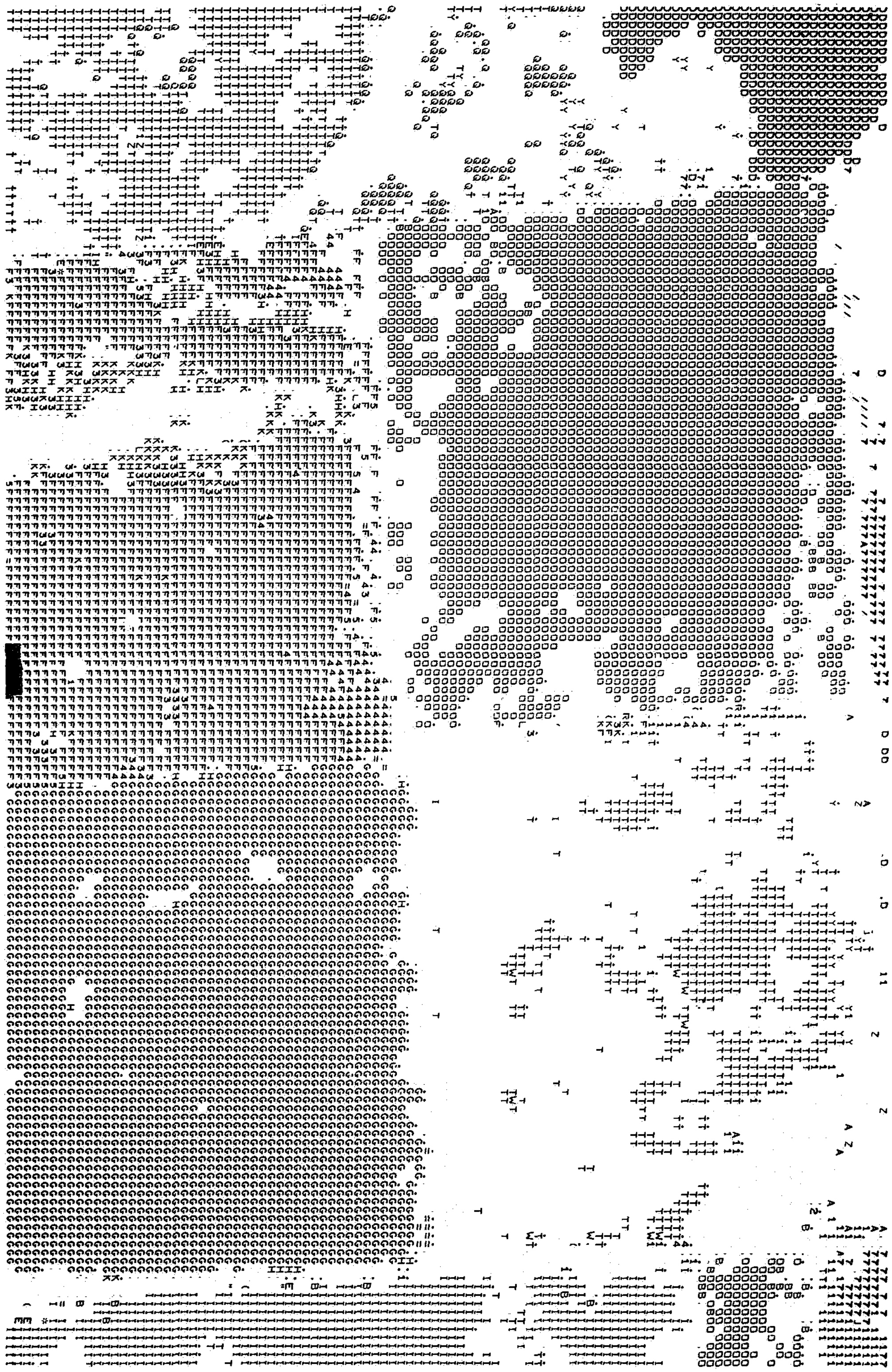

Figure 13. Final Cl classification map – Section 10.

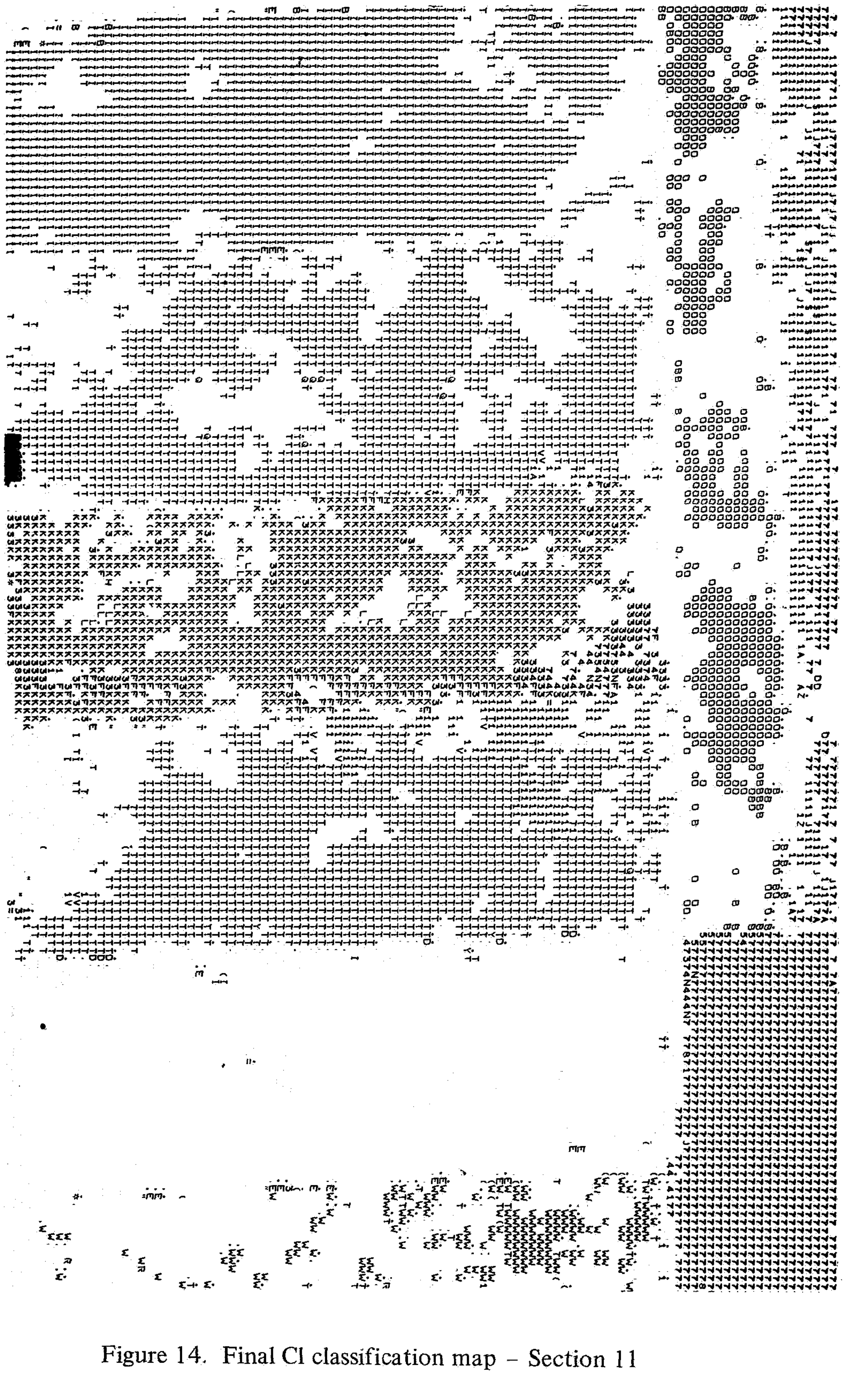

Figure 14. Final CI classification map – Section 11

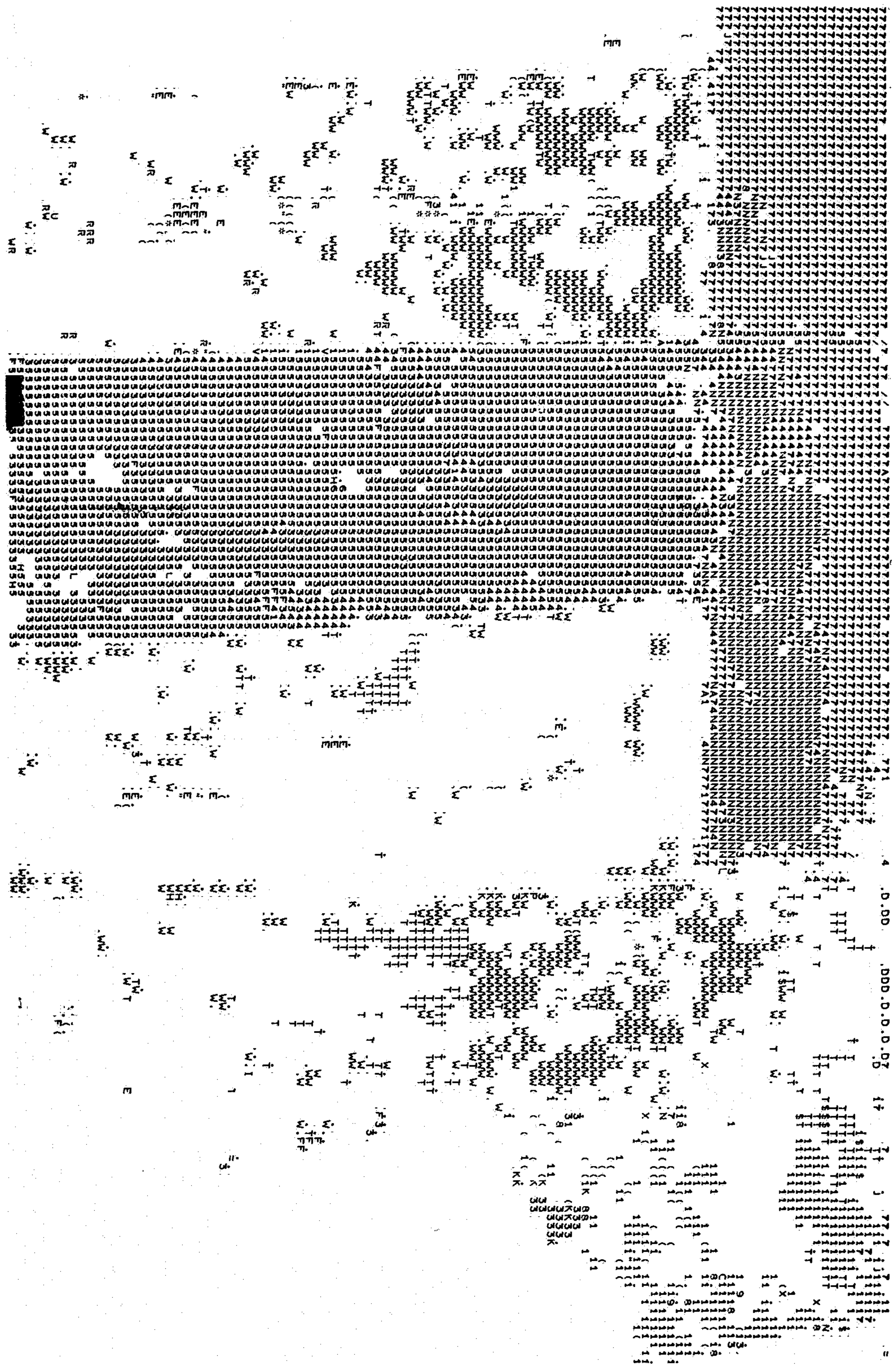

Figure 15. Final Cl classification map – Section 12.

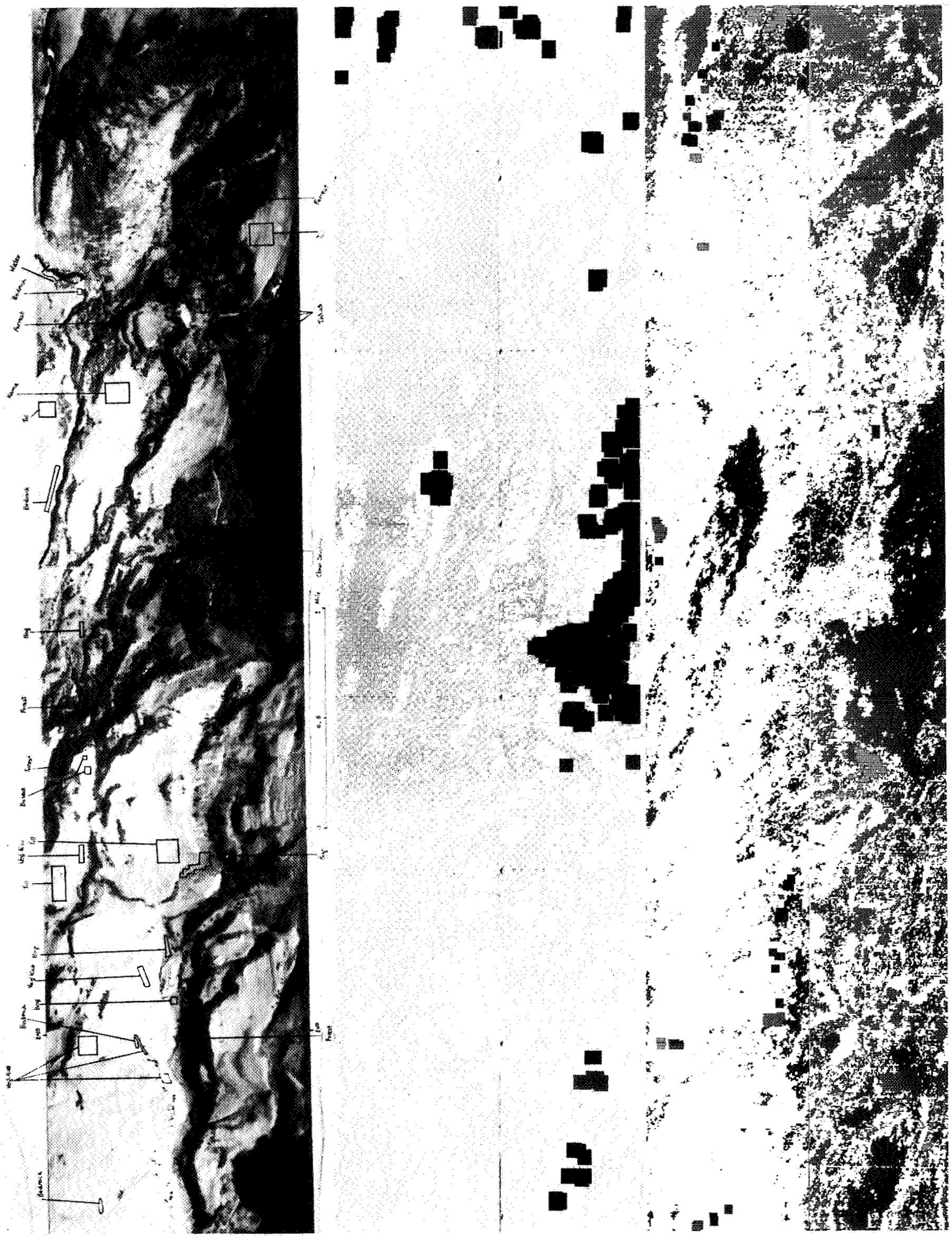

Figure 16. Yellowstone Park – Video reprint, cluster selection, initial classification, and secondary cluster selection.

The Yellowstone Park data were analyzed exactly as the Cl flight line data. The video reprint contains some ground truth and locations of training areas used by other investigators listed in References 11 and 12.

Table 9 gives the merging procedure for both classification passes since no multiple merges were encountered. No identification is given for the clusters because the cluster locations do not necessarily coincide with the training areas, and it is easier to identify the features after the final classification. The analysis of this data set indicates the type of problem involved when an inaccessible region is remotely sensed. It is difficult to estimate what types of features can be extracted from the data, and it may be economically advantageous for the investigator to collect the ground-truth information based upon the feature extraction map, rather than trying to anticipate the needed detailed ground truth

Table 10 was prepared for interpreting the final results based on information derived from the video reprint. Several meadow-like areas were discernible, but were only given meadow numbers corresponding to their class numbers because of lack of other information. The geologic terms till, talus, kame, and vegetated rock rubble are described in References 11 and 12, but are repeated here for convenience.

The class till consists of meadow areas underlain by glacial till They are grassland and sagebrush areas which were largely dormant at the time of data collection. Mineral soil is exposed in about one-fifth of the area and consists of silty to bouldery debris. Deer and elk manure are locally abundant in these areas.

The class talus includes blockfields, talus, and talus flows of basalt lava flows, volcanic tuff, and gneiss, formed by frost-riving and solifluction from outcrops. They are blocky and well-drained deposits, and trees are widely spaced or absent. The blocks are covered with dark gray lichens and range from a few cm to about 1 m in diameter Most are larger than 10 cm. The slopes in these areas range widely from 35 to 45 deg at the head to 5 deg or less at the toe.

The class kame is very similar to till except about one-fourth of the area is exposed mineral soil

The class vegetated rock rubble consists of locally derived angular rubble, frost-riven from basalt lavas, volcanic tuff and breccia, and gneiss. Grass, lichens, evergreen seedlings, and moss cover more than three-fourths of the surface underlain by this debris. The rocks range in diameter from less than 1 cm to about 1 m and occur on slopes from 0 to about 25 deg.

Three additionally known features in the ground scene did not appear in the classification map because they were not contained in a homogeneous area large enough to be selected by a cluster These features were water, bedrock, and bog. However, the areas where they appear on the classification map can be located with the aid of the

TABLE 9 MERGING PROCEDURE FOR YELLOWSTONE DATA

Classification Pass 1-10x10 Array		Classification Pass 2-6x6 Array	
Initial Cluster No.	Final Class No.	Initial Cluster No.	Final Class No.
1	1	11	11
2	2	12	11
3	2	13	11
4	3	14	11
5	3	15	11
6	2	16	11
7	4	17	11
8	4	18	11
9	3	19	11
10	4	20	12
11	4	21	13
12	5	22	14
13	4	23	14
14	6	24	15
15	6	25	15
16	4	26	16
17	4	27	16
18	4	28	16
19	4	29	16
20	7	30	17
21	8	31	18
22	9	32	19
23	10	33	19
24	10	34	20
25	10		

TABLE 10. INTERPRETATION OF YELLOWSTONE RESULTS

Class No.	Symbol	Brief Description
1	1	Meadow 1
2	2	Meadow 2
3	3	Meadow 3
4	4	Dense forest and shadow
5	5	Till
6	6	Kame
7	7	Meadow 7
8	8	Meadow 8
9	9	Meadow 9
10	0	Meadow 10
11	A	Trees
12	B	Trees
13	C	Talus
14	D	Till
15	E	Till
16	F	Till
17	G	Till
18	H	Till
19	I	Vegetated rock rubble
20	J	Meadow 20

video reprint, and they appear as unclassified areas. These classes could now be identified elsewhere in the data by using a supervised technique and selecting these unclassified resolution elements as training areas.

Figures 17 through 28 contain the final classification results for the Yellowstone Park test site. Figures 17 through 22 correspond to the left side of the video reprint, and Figures 23 through 28 correspond to the right side of the video reprint.

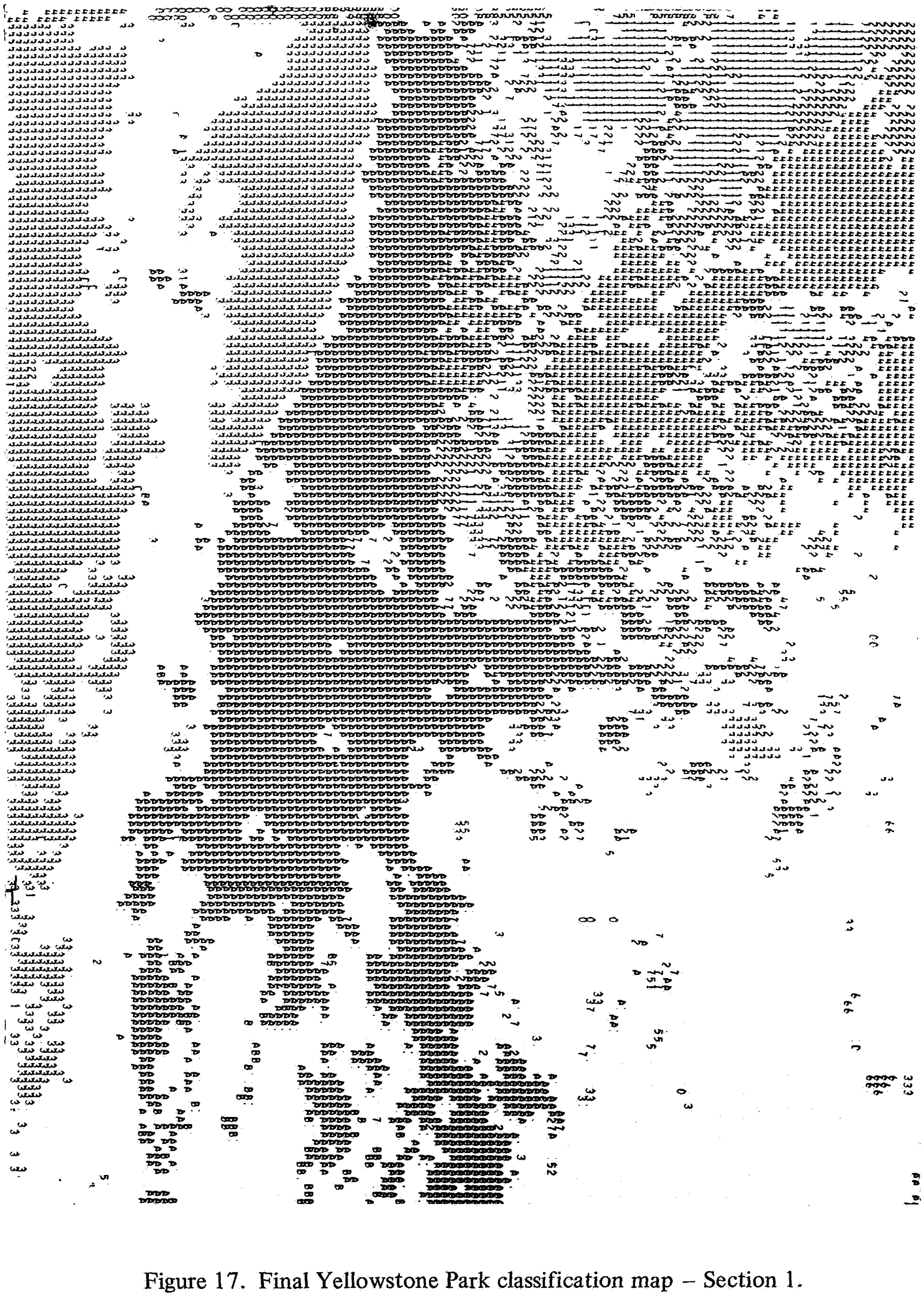

Figure 17. Final Yellowstone Park classification map – Section 1.

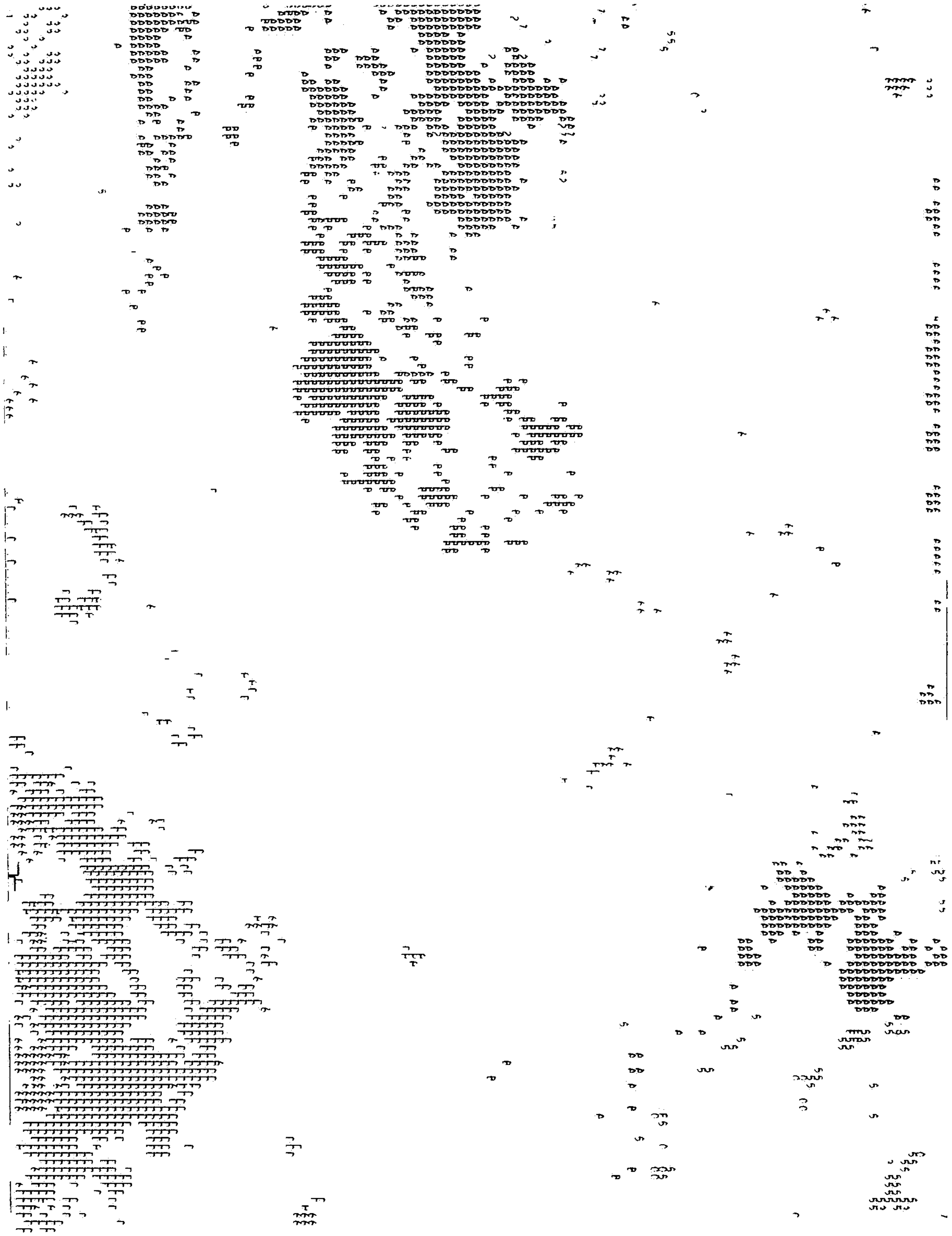

Figure 18. Final Yellowstone Park classification map – Section 2.

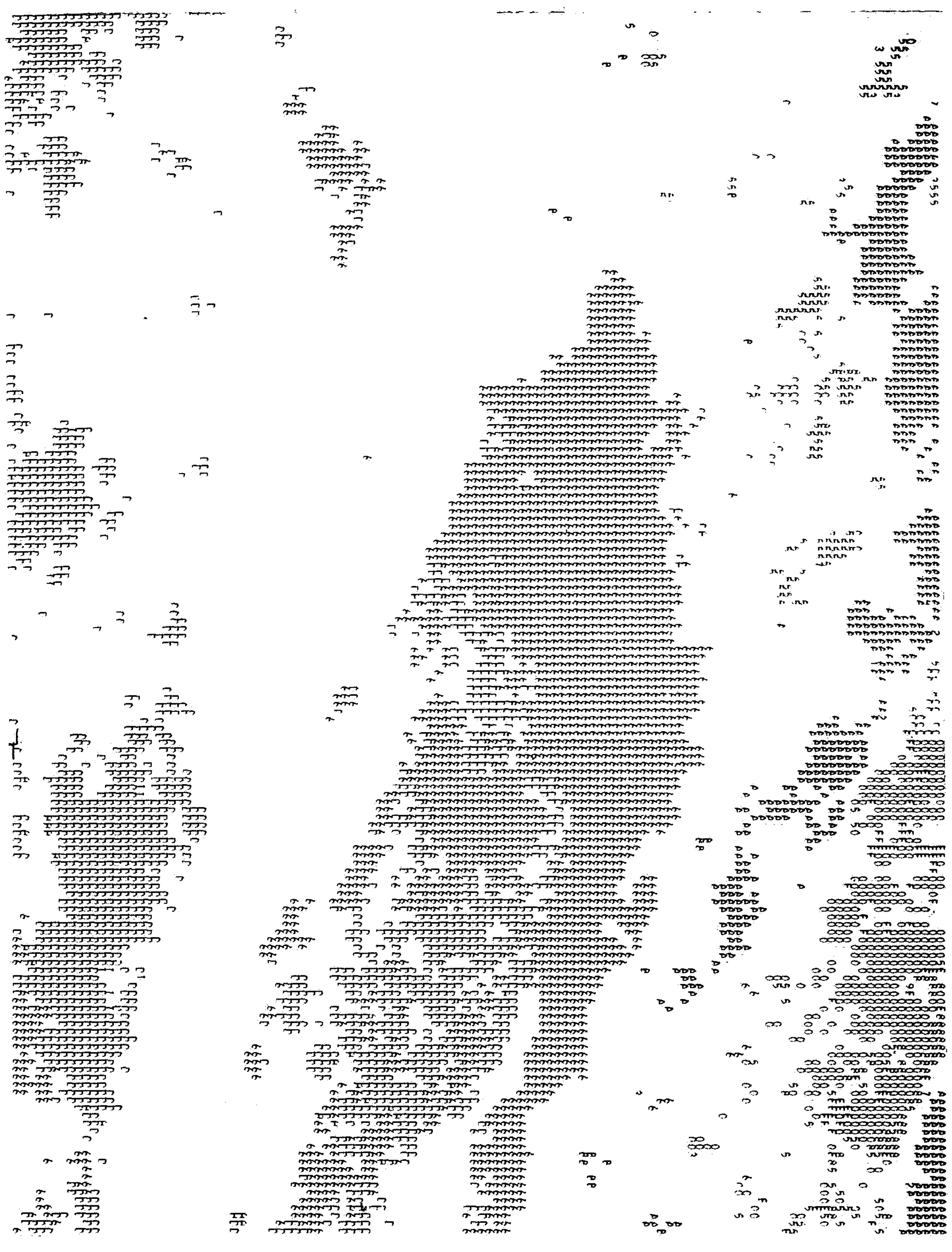

Figure 19. Final Yellowstone Park classification map – Section 3.

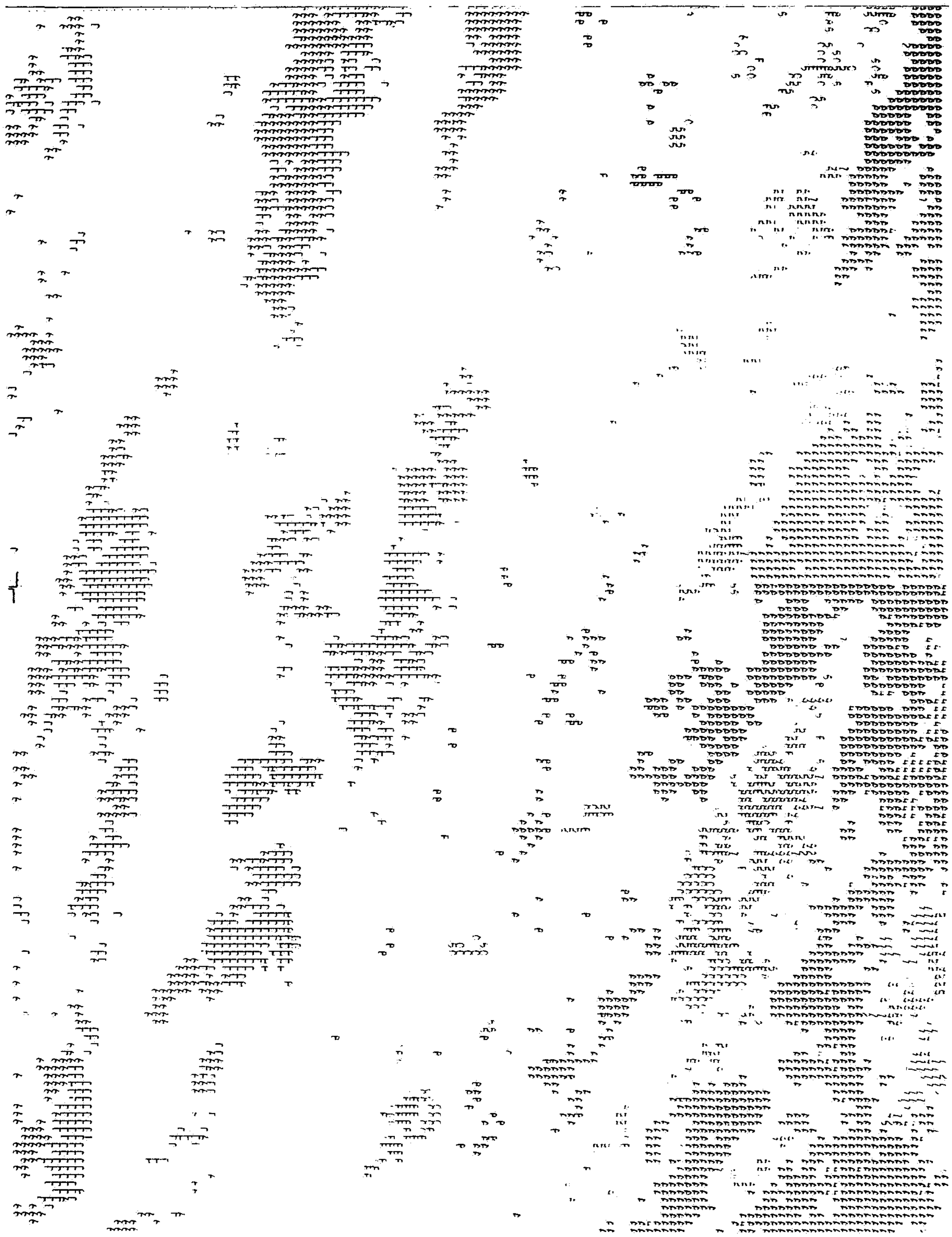

Figure 20. Final Yellowstone Park classification map – Section 4.

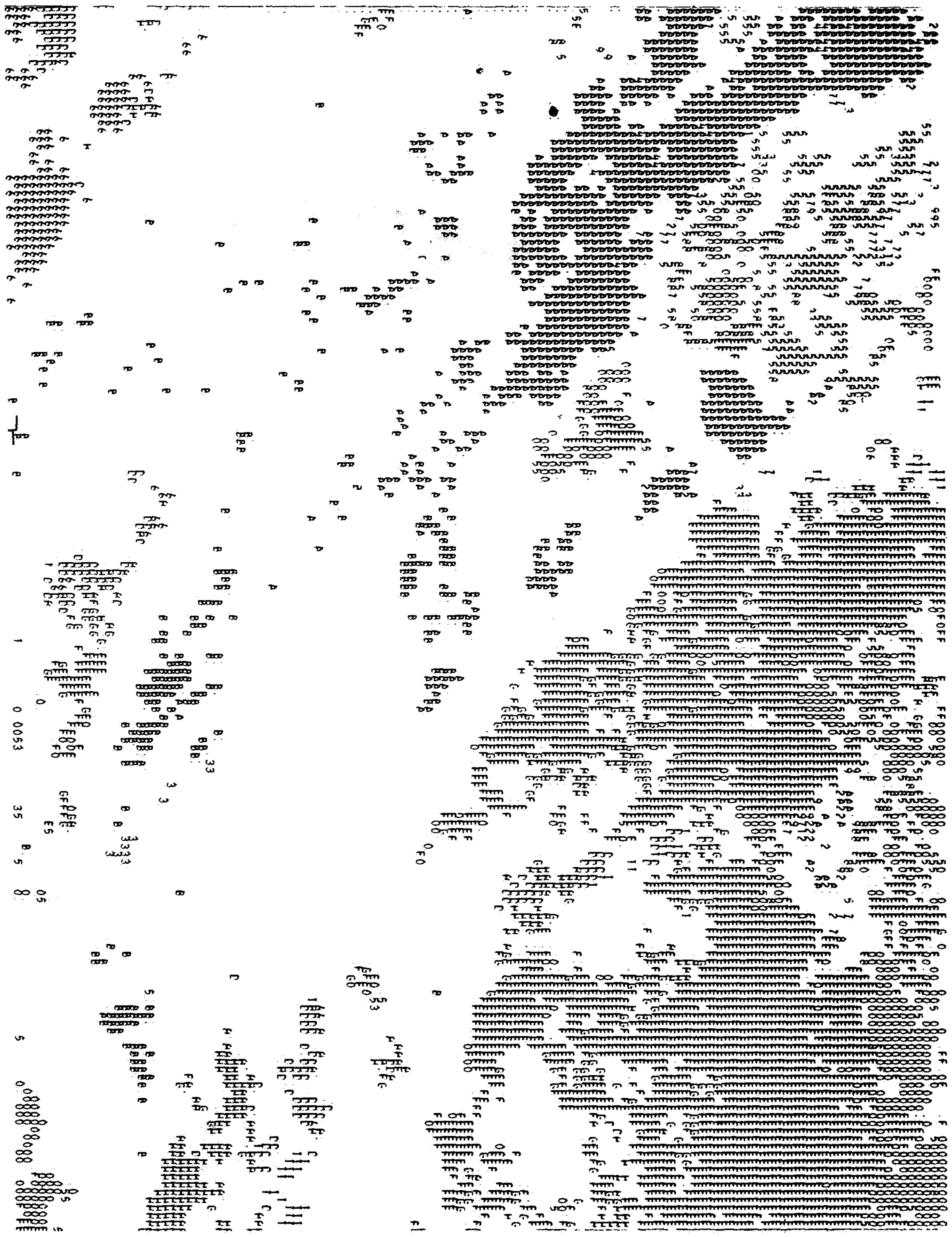

Figure 21 Final Yellowstone Park classification map – Section 5

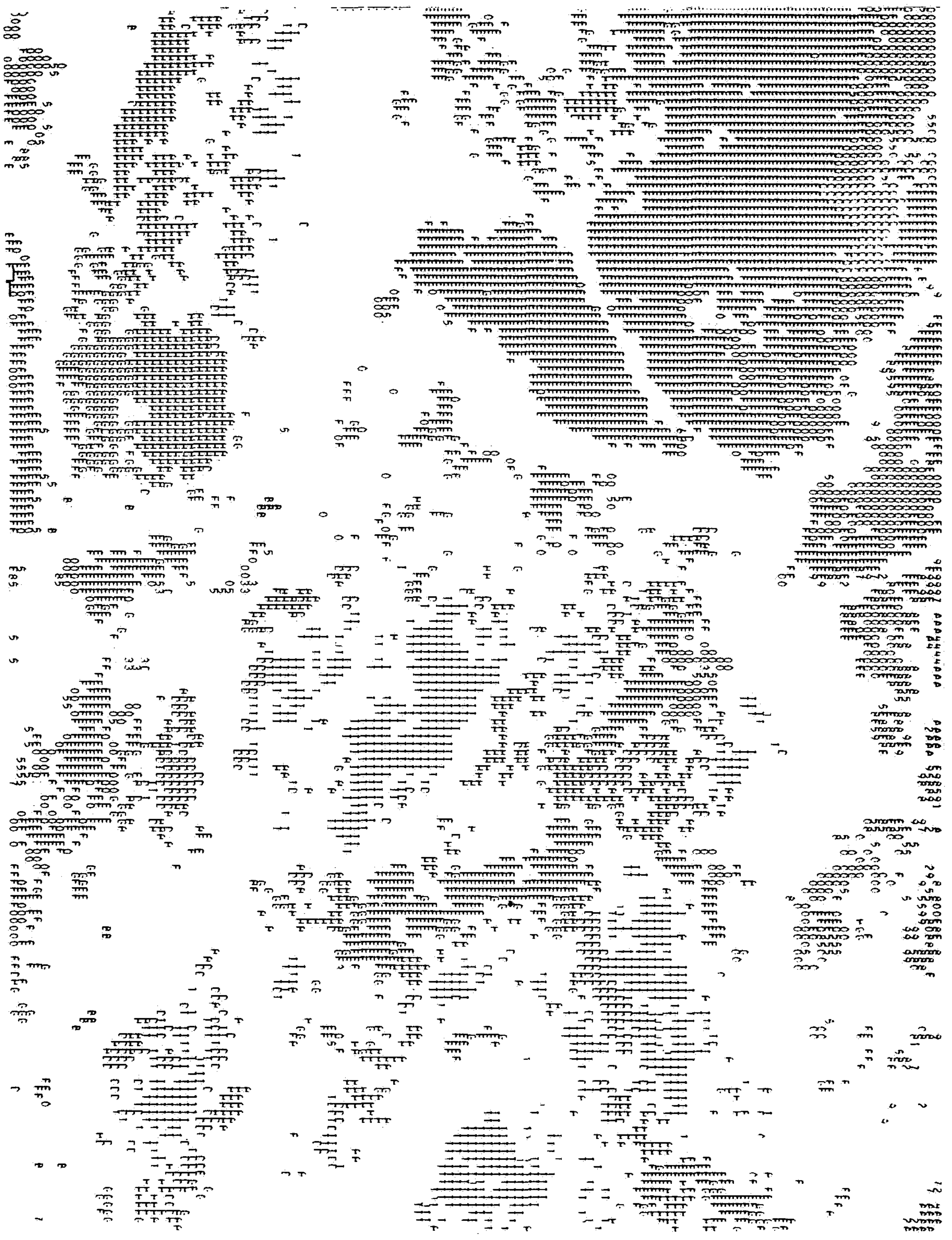

Figure 22. Final Yellowstone Park classification map – Section 6.

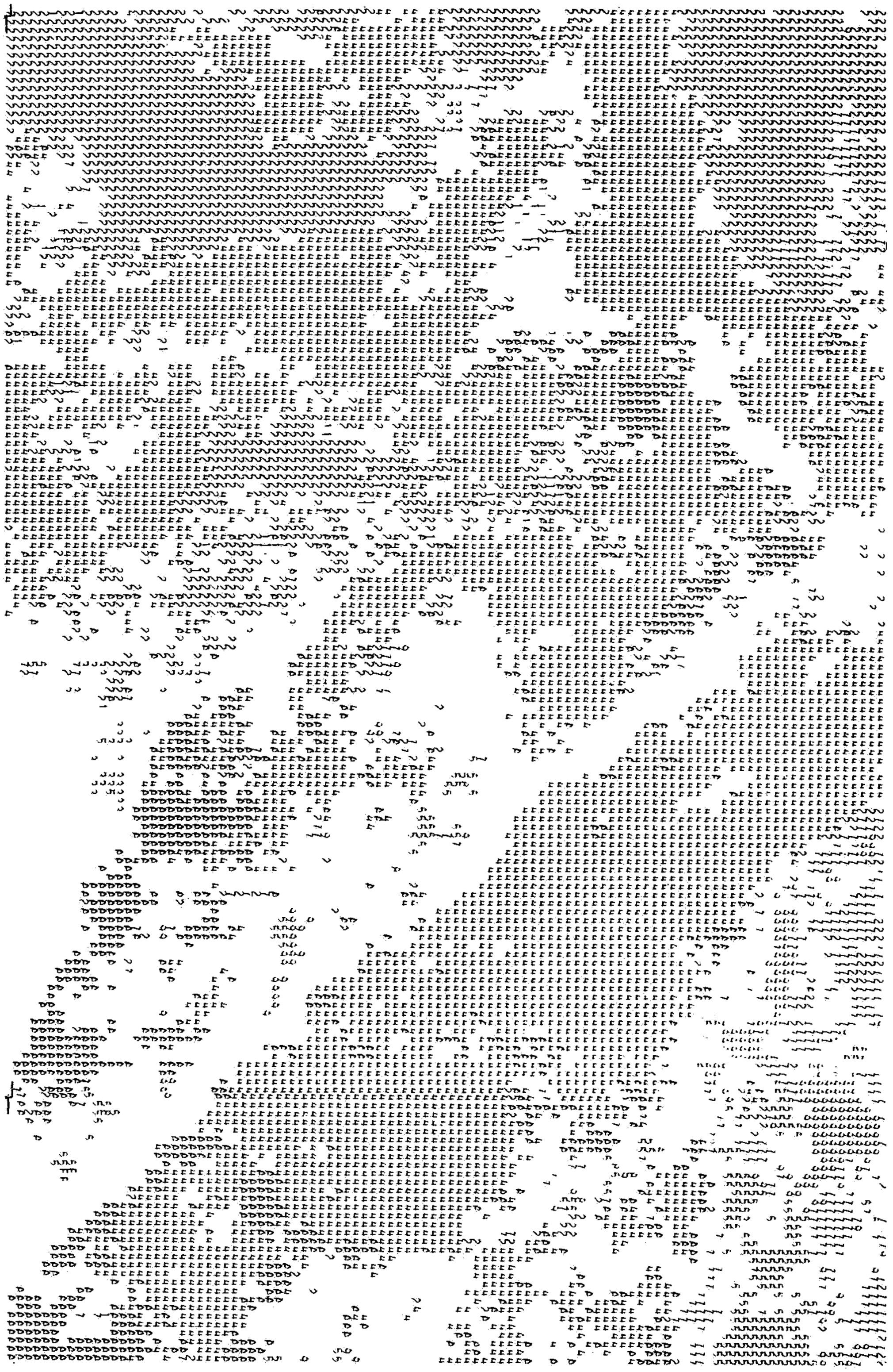

Figure 23. Final Yellowstone Park classification map – Section 7

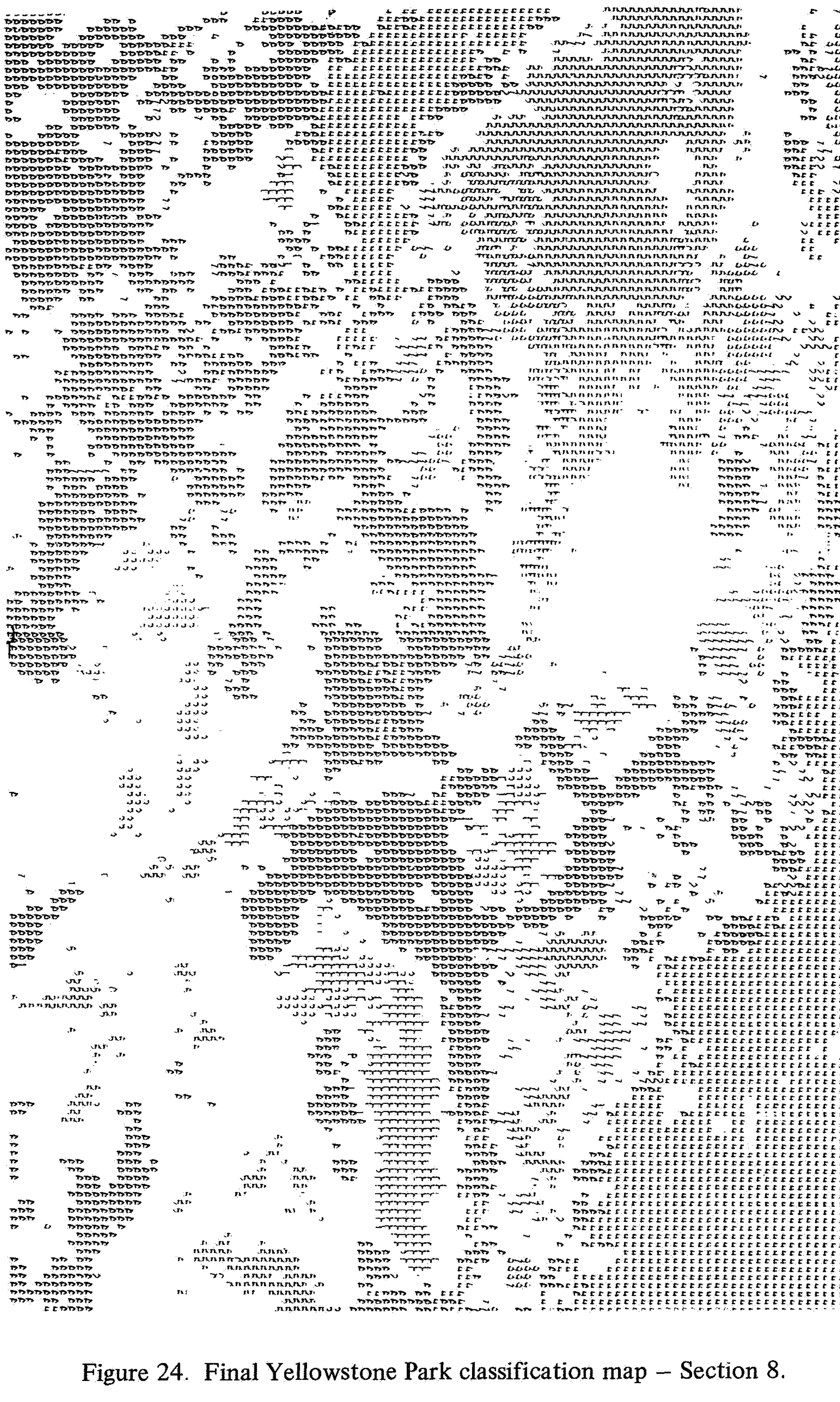

Figure 24. Final Yellowstone Park classification map – Section 8.

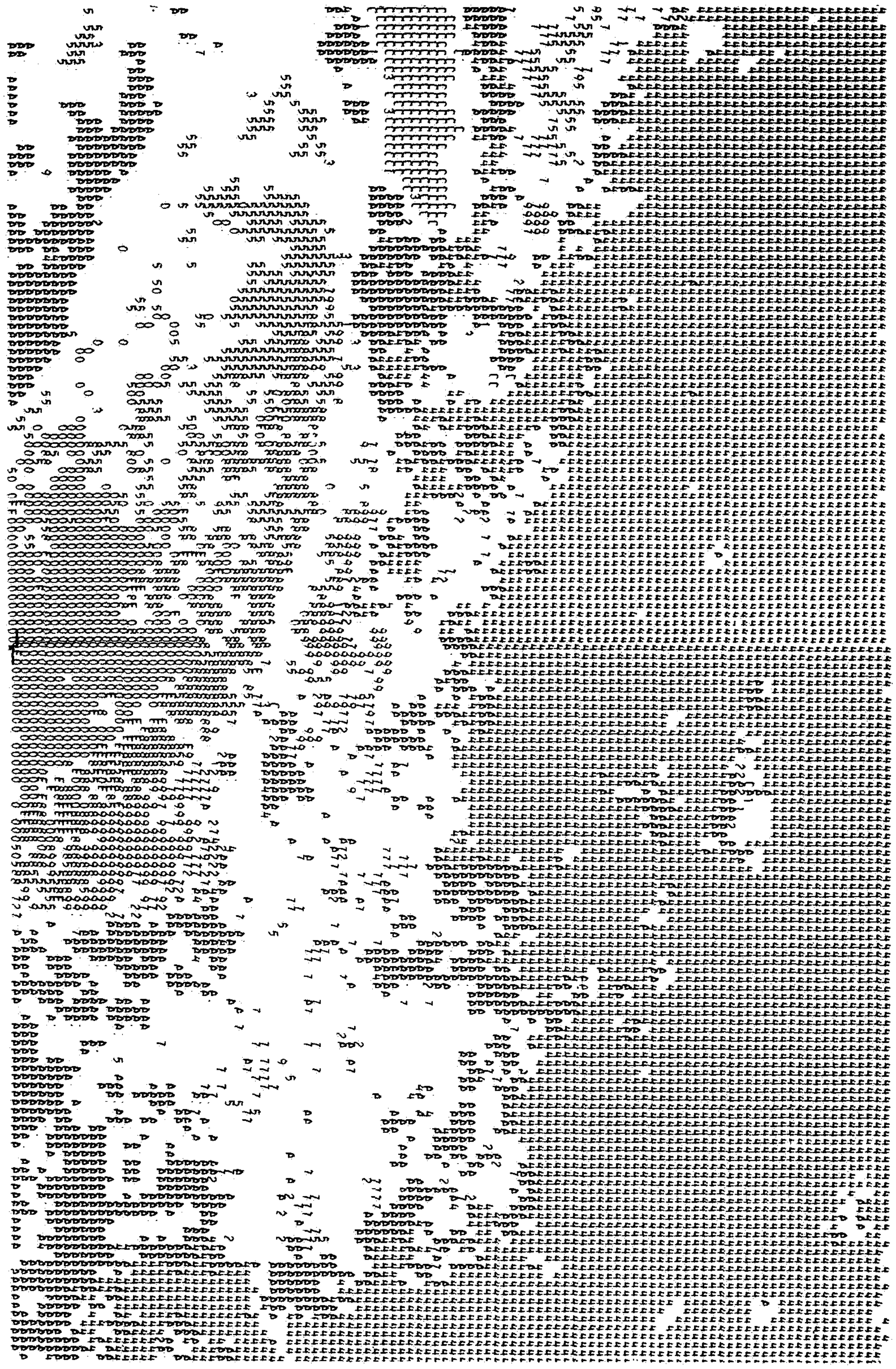

Figure 25 Final Yellowstone Park classification map – Section 9

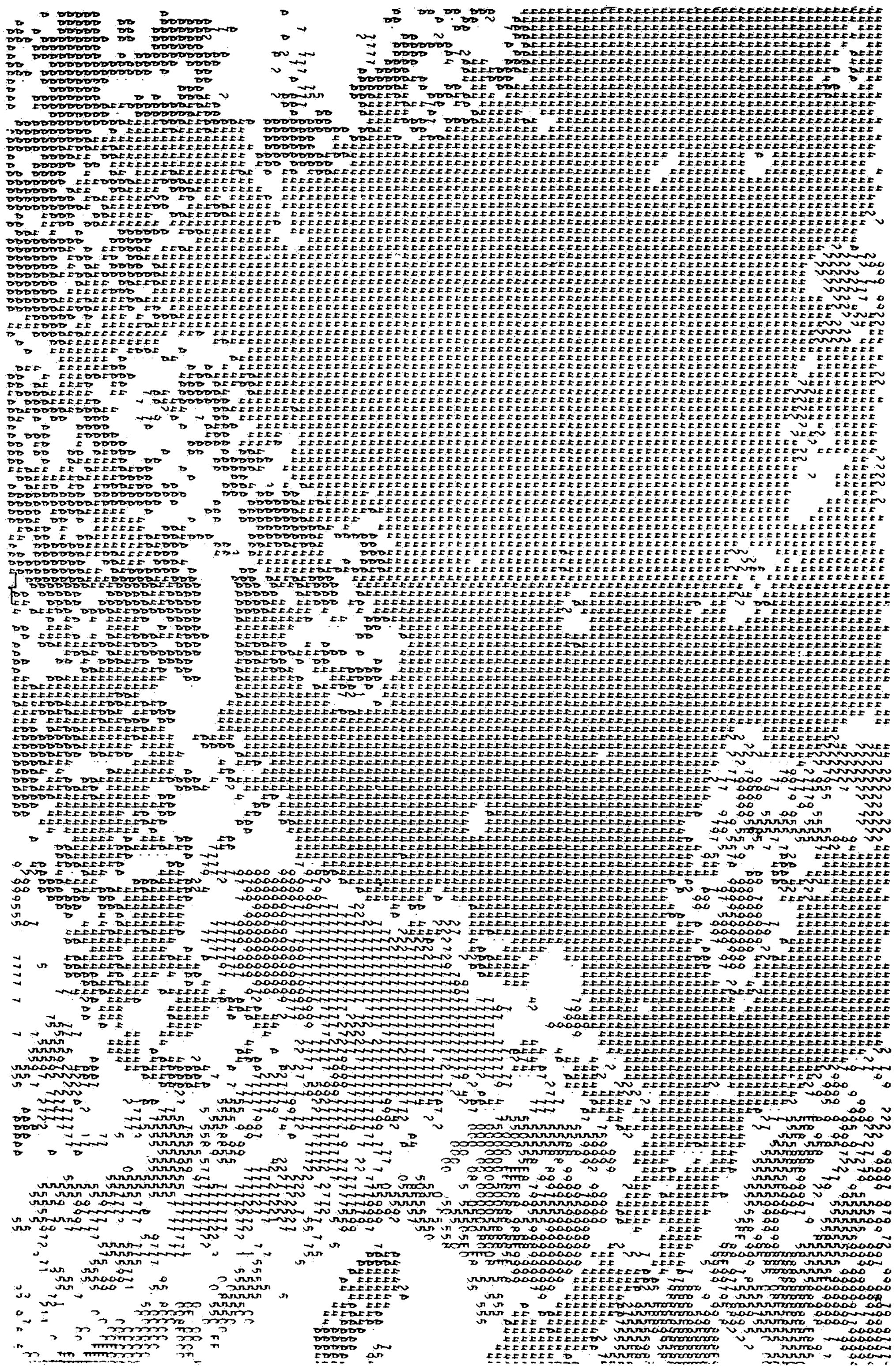

Figure 26. Final Yellowstone Park classification map – Section 10.

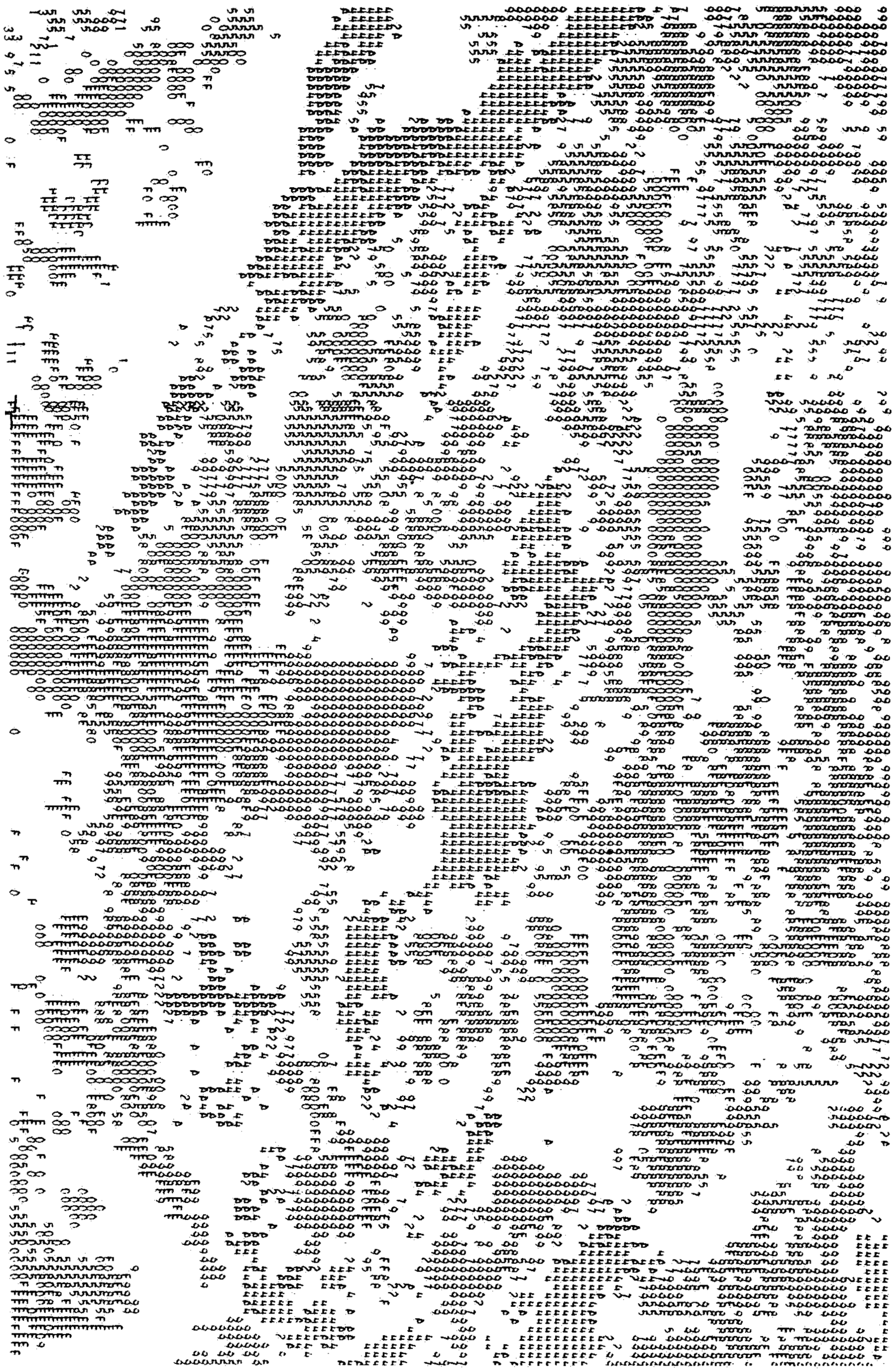

Figure 27 Final Yellowstone Park classification map – Section 11

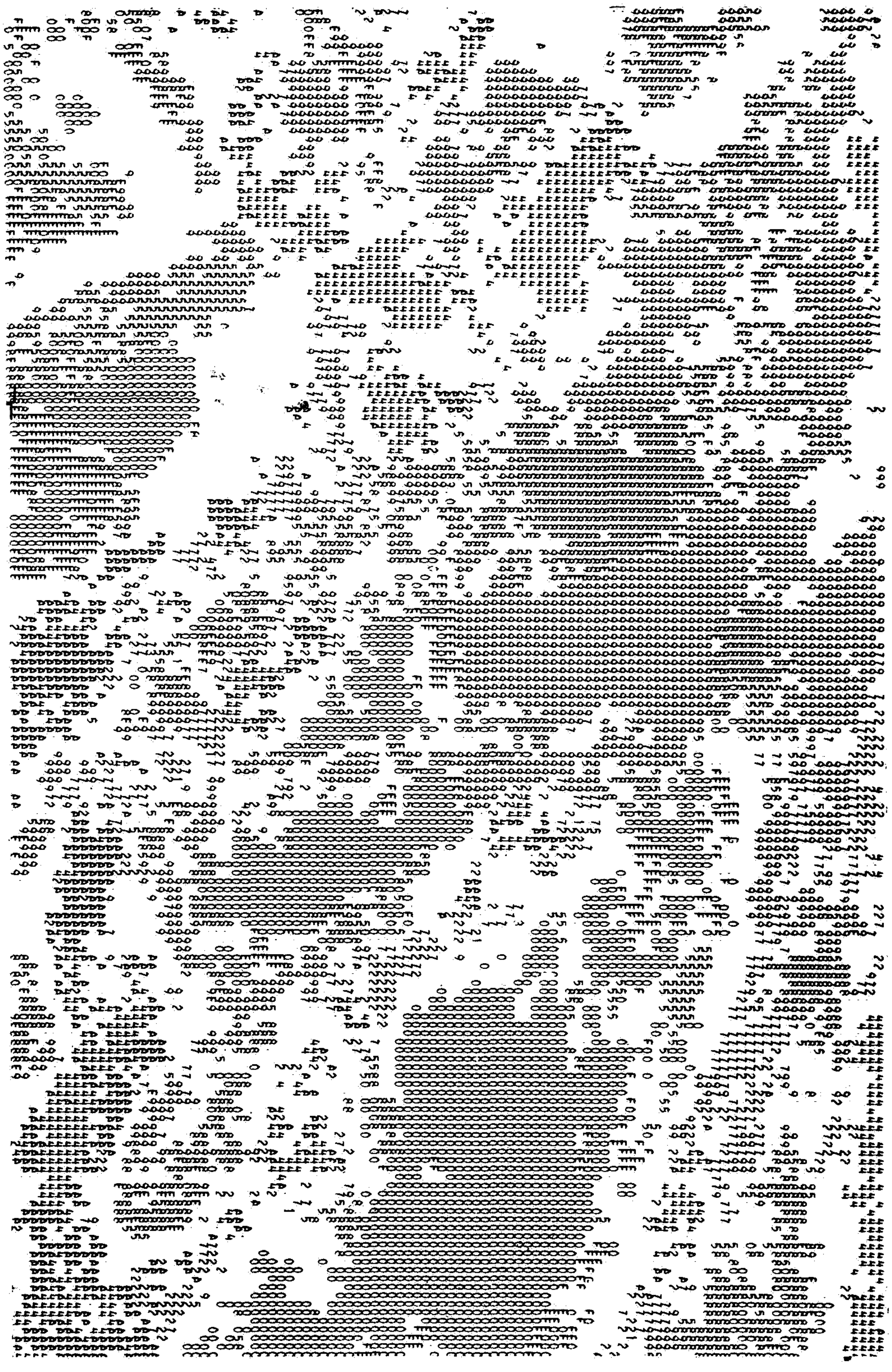

Figure 28. Final Yellowstone Park classification map – Section 12.

V. CONCLUSIONS

The feature extraction can be interpreted as being an extension to photography. Photography is capable of presenting patterns in colors or various shadings for interpretation by an observer, but, often, some features appear which are similar, but they are not similar, and vice versa. Computer feature extraction adds the capability to distinguish subtle differences in multiple digital data images and to make a decision concerning the similarity of those features in question. This extension can be exploited to its fullest limits, in some cases, when a data bank can be used for actual identification of the features.

The purpose of this work was to demonstrate the capability of extracting features from digital data images without involving an observer in the data-processing loop and to compress unmanageable data into manageable and useful information. An observer would still be needed to interpret the results.

The success of the data compression is significant because of the 12 channels of data that were compressed into 1 and because of 200 022 resolution elements that were reduced to a maximum of 45 distinct categories.

The computer programs presented in this work were purposely developed to be as general as possible, and the ultimate success of these computer programs for information extraction can only be determined when the programs are tailored and applied to solving a specific user or user agency's needs. The present utilization of these programs for feature surveying, such as the vegetation and geological categorizations presented in the text, can be interpreted as being successful since the comparison of results with those previously published in the references appears very favorable.

The two keys to success in using the unsupervised feature extraction program are the production of a boundary map which cleanly separates homogeneous areas belonging to different features and the choice of the decision rule for spectrally merging similar features.

The over abundance of boundaries in the Yellowstone data indicates the need for improvement in the mathematical definition of a boundary or at least a means for improving the threshold using the present boundary definition. If repetitive coverage were available for a particular test site, the optimum threshold for a boundary decision could probably be determined. Otherwise, the experience gained from working with other types of data sets or from perusing the literature would have to provide the impetus for determining a better mathematical definition of a boundary The two most important properties that need to be considered for this definition are most probably the resolution and some measure of roughness in the pictorial scene of the digital data image.

The use of a p x p array for obtaining initial cluster statistics in a homogeneous area may have an advantage over the manual selection of training areas since the cluster areas can have a fairly arbitrary shape rather than being rectangular and the data selected from these areas do not contain unusual data points which could bias the statistical calculations. The use of the p x p array does have the definite advantage because it is faster than manually selecting the coordinates of the training area and entering these coordinates into the computer

The use of an elliptical decision rule for merging and classification appears to be very acceptable as evidenced by the fact that practically all of the resolution elements used for calculating the statistics of a feature are classified as belonging to that feature. This fact is even evident when several initial clusters have been merged to form a final class. Additional supporting evidence to this conclusion is that the initial clusters selected on the second pass through the data did not merge with any of the final classes obtained from the first pass through the data, and an almost unnoticeable amount of resolution elements had their original classification changed. Most of the error in classification occurs near boundaries and near the edges of the scan lines The first type of error is probably caused by the data changing from one feature to another in the vicinity of a boundary and in the process passing through a decision region of a third different feature. The misclassification near the edge of the data is because of an optical effect called scan angle error The angle at which the ground scene is viewed at the edge of a scan is usually 30 to 40 deg off from the local vertical and, as a consequence, the signal that is recorded by the sensor reflects an angular dependence. It is reasonable to assume, however, that the use of an elliptical decision rule could consider the angular dependence. The angular effect should be approximately linear dependent for the different channels of data and this would amount to a length stretching of the principal axes of the classification ellipses. It is apparent that the classification maps probably contain more detailed information than is actually desired. The detailed information present can be further compressed by visual interpretation and manually merging the desired features by changing the symbol output on the classification map.

When the features are manually merged, caution must be exercised in interpreting a given feature extracted from the data, e.g., there are several features which represent soy, corn, and a mixture of corn and soy Although there is a temptation to attach a simple description that is commonly used, the description may be incomplete with respect to the information presented by the data, and a logical manual merging may not be possible. Detailed ground truth would be needed to provide the qualifications and adjectives for a complete description. For this reason, it may be important to perform the data analysis before prejudging the information content of the data rather than using the ground truth to assist in the analysis of the data.

Finally, it must be emphasized that the development of the unsupervised feature extraction computer programs was directed toward obtaining a computer logic that could extract information from remotely sensed data with a moderate degree of success, which meant that computer running time necessarily took a "back seat." Since the development work has been completed, efforts can now be directed toward optimizing the computer time and efficiency of the programs.

APPENDIX A. PROGRAM DESCRIPTION

The computer programs which utilize the equations in the text are written in the form of subroutines and have been included as an integral part of a much larger computer program called an Earth Resources Processor. This processor contains several preprocessing and data display routines in addition to the classification programs and is documented and flow charted in detail in the final report of contract NAS8-26797 by IIT Research Institute, Chicago, Illinois [25]. Hence, only a brief description of these subroutines will be given. A small executive program can be written for controlling the sequence of calling these subroutines by following the logic flow previously described in Figure 2 and discussed in Section III.

Subroutine BWNDR3

Subroutine BWNDR3 is the first stage of processing or the boundary program and reads the raw data tape as input. This subroutine contains equations (8) through (16) and the logic for using these equations. The boundary program also contains two subroutines which will be discussed. Subroutine GET6 is used for getting the data off digital tape and putting it into the computer. The content of this subroutine has to be specialized to the type of digital tape format desired. Subroutine JNTPB is a joint probability distribution program.

Subroutine JNTPB

Subroutine JNTPB is used for calculating the joint probability distribution of s_x versus s_y in equation (8) and the decision rule $as_x^2 + bs_y^2 + cs_xs_y \leq 1$ in equation (16). This program has a fixed storage allocation, but the shape of the joint distribution can be completely arbitrary because a search mode of operation is used rather than a table look-up procedure. This program will accept data of any dynamic range and output a scatter diagram bounded by the minimum and maximum values of the data. This program contains three subroutines: (1) DJCOBI is an IBM library system subroutine for calculating the eigenvalues and eigenvectors used in the ellipse equation; (2) LABEL6 is used for labeling the axes of the scatter diagram, while subrouting PLTBF6 is specialized for use with the Stromberg-Carlson 4020 peripheral equipment to obtain microfilm copies of the boundary map; and (3) JNTPB also outputs the boundary tape for use in the second stage of processing.

Subroutine CLASFY

Subroutine CLASFY contains the second, third, and fourth stages of processing via subroutines TRUCK, SEQMRG, and CLASS, respectively This subroutine also controls the number of classification passes desired and the size of the p x p cluster selection array via subroutine TRUCK.

Subroutine TRUCK

The second stage of processing is subroutine TRUCK, which reads the boundary map tape as input data and locates the homogeneous areas that are large enough to contain a p x p array. This subroutine also performs the spatial merging when two different clusters collide. The output of this subroutine is a boundary map tape containing the location of the initial clusters, which is input to the third stage of processing. Subroutines LABELA and PLTBFA are used for labeling scan and column numbers and obtaining microfilm, respectively, for the visual display of the map obtained from subroutine TRUCK.

Subroutine SEQMRG

The third stage of processing is subroutine SEQMRG and uses the boundary and cluster location tape and the raw data tape as input. The boundary and cluster tape is used to locate and retrieve raw data belonging to each cluster. Statistics utilizing equations (17) through (30) are then calculated for each cluster and used for deciding whether to spectrally merge clusters. The clusters are read in sequentially, and each new cluster has the opportunity of merging with any or several of the previous clusters. Subroutine SKRBIN is an IBM system subroutine and allows for skipping binary records on the magnetic tape. This routine is primarily used with the cluster selection since the clusters may be located anywhere on the tape. Subroutine FETCOR is used to retrieve and calculate the centroid of each cluster and to calculate the cross-channel correlations for each cluster. The means are subtracted from the correlations to form a covariance matrix in subroutine AMTRX, which becomes an input to the eigenvalue and eigenvector subroutine DJCOBI. Subroutine ROTA is a rotation matrix calculation used to rotate a feature vector into the coordinate system of a cluster Subroutine KCHECK is used to check whether the centroids of two clusters fall within each other's ellipses or merge. If two clusters will merge, the statistics of both clusters are combined and updated. The output of SEQMRG is a tape containing the statistics for the final set of classes.

Subroutine FETCOR

In addition to retrieving and correlating the raw data for each cluster, subroutine FETCOR keeps track of the starting and stopping scan lines and columns of the clusters that are in the computer This information is used with the subroutine BSRECD, which is an IBM system library subroutine for backspacing records. The use of this subroutine allows for backspacing the tape to locate the data for the next cluster, rather than rewinding the entire tape and searching for the next cluster's data.

Subroutine CLASS

The fourth stage of processing is subroutine CLASS and uses the tape containing the statistics for each class, the boundary tape, and the raw data tape as input. This subroutine mainly contains equation (31) and the provisions for outputting the classification map on standard computer printout or microfilm. The classification map is also output on tape for use in subroutine CLASFY to obtain additional clusters and classification passes.

APPENDIX B. COMPUTER PROGRAM LISTINGS

```
      SUBROUTINE BWNDR3
      DIMENSION X(12,256),Y(12,256),MCHAN(12),NN(   256),KSYM(49),JSYM(
     1256)
      DIMENSION NWHICH(12)
      NAMELIST/INPUT6/NSCANS,NSTART,NSPS,NCH,NVAR,NSYM,ISUM,NBTLG,
     1MODE,ITYPE,MSFC,,NSKIP,NBLK,INCX,INCY,NSTX,NSTY,NCRE
      NAMELIST/NCHUSE/NWHICH
      EQUIVALENCE (NSCAN,NSCANS)
      EQUIVALENCE (NSTRT,NSTART)
      EQUIVALENCE (NCOL,NSPS)
      EQUIVALENCE (NCHAN,NCH)
      ICARD=5
      IPRINT=6
      INTAPE=10
      IOTAPE=11
      READ(ICARD,INPUT6)
      WRITE (IPRINT,INPUT6)
      READ(ICARD,NCHUSE)
      WRITE(IPRINT,NCHUSE)
    1 FORMAT(1X,7I4)
    3 FORMAT(1X,12I1)
      READ(ICARD,5)(KSYM(I),I=1,NSYM)
    5 FORMAT(1X,60A1)
      NFLAG=0
      AVE=ISUM
      APOP=0.0
      DXAVE=0.0
      DYAVE=0.0
      DZAVE=0.0
      NSAV=NSCAN
      IF (NSKIP .EQ. 0) GO TO 98
      DO 97 I=1,NSKIP
      CALL SKRBIN(INTAPE,1,NOP)
 97   CONTINUE
 98   CONTINUE
    2 FORMAT(1H1)
    4 FORMAT(5X,111I1)
      II=1
      KK=NSTRT-1
  160 IF(II.EQ.NSCAN) GO TO 510
      II=II+1
      NFLAG2=1
      KK=KK+1
      IF(II.NE.2) GO TO 290
      DO 170 JJ=1,NCOL
      CALL GET6(X(1,JJ),NCOL,0,NCHAN,NSCANO,INTAPE,IERR,NFLAG2,NSTRT,
     1NBTLG,MODE,NCRE,ITYPE,MSFC)
  170 CONTINUE
  290 CONTINUE
      NFLAG2=1
      DO 300 JJ=1,NCOL
      CALL GET6(Y(1,JJ),NCOL,0,NCHAN,NSCANO,INTAPE,IERR,NFLAG2,NSTRT,
     1NBTLG,MODE,NCRE,ITYPE,MSFC)
  300 CONTINUE
      DO 380 JJ=2,NCOL
      IJ=JJ-1
      XSUM=0.0
      YSUM=0.0
      ZSUM=0.0
      DO 360 ICHAN=1,ISUM
      IICHN=NWHICH(ICHAN)
      XDIFF=Y(IICHN,JJ)-Y(IICHN,IJ)
      YDIFF=Y(IICHN,JJ)-X(IICHN,JJ)
      XSUM=XSUM+XDIFF*XDIFF
      YSUM=YSUM+YDIFF*YDIFF
      ZSUM=ZSUM+XDIFF*YDIFF
  360 CONTINUE
      XSUM=XSUM/AVE
      YSUM=YSUM/AVE
      ZSUM=ZSUM/AVE
      APOP=APOP+1.0
      AA=1.0/APOP
      BB=1.0-AA
      DXAVE=BB*DXAVE+AA*YSUM
      DYAVE=BB*DYAVE+AA*XSUM
      DZAVE=BB*DZAVE+AA*ZSUM
      XSUM=SQRT(XSUM)
      IF (XSUM .LT. 53.4) GO TO 365
 364  XSUM=53.0
      YSUM=53.0
      GO TO 366
 365  CONTINUE
      YSUM=SQRT(YSUM)
      IF (YSUM .LT. 53.4) GO TO 366
      GO TO 364
 366  CONTINUE
      CALL JNTPB(YSUM,XSUM,NFLAG,0,0,KSYM(3),NPOP,
     2DXAVE,DYAVE,
     3DZAVE,
     1JJ,NSCAN,II,NCOL,JSYM,X,NSTRT,NN,INCX)
      NSCAN=NSAV
  380 CONTINUE
      DO 500 JJ=1,NCOL
      DO 490 ICHAN=1,ISUM
      IICHN=NWHICH(ICHAN)
      X(IICHN,JJ)=Y(IICHN,JJ)
  490 CONTINUE
  500 CONTINUE
      GO TO 160
 510  CONTINUE
      NFLAG=1
      CALL JNTPB(YSUM,XSUM,NFLAG,0,0,KSYM(3),NPOP,
     2DXAVE,DYAVE,
     3DZAVE,
     1JJ,NSCAN,II,NCOL,JSYM,X,NSTRT,NN,INCX)
      REWIND IOTAPE
      REWIND INTAPE
C     CALL CLEAN
      RETURN
      END
      SUBROUTINE LABEL6(NSTART,NSTOP,INCRE
      DIMENSION IOUT(120)
      NDIF=(NSTOP-NSTART+1)/INCRE
      II=0
      DO 1 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/1000
1     CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 2 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/100-I/1000*10
2     CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 3 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/10-I/100*10
3     CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 4 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I-I/10*10
      IF (IOUT(II) .LE. 0 ) IOUT(II)=0
4     CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
10    FORMAT (11X,120I1)
      RETURN
      END
      SUBROUTINE JNTPB(DATAH,DATAV,NFLAG,MIX,MIY,ALPNUM,NPOP,
     2DXAVE,DYAVE,
     3DZAVE,
     1JJ,NSCAN,ISCAN,NCOL,ISYM,NX,NSTRT,NN,INCXY)
      DIMENSION INCXY(1)
      DIMENSION NP(54,54)
      DIMENSION DATA(12)
      DIMENSION IBIN(255 )
      DIMENSION ISYM(1),IQUIT(100)
      DIMENSION NX(1),NN(1)
      DIMENSION ALPNUM( 1),ALPHA(120),CORDX(3)
      DOUBLE PRECISION A(2,2),EIGEN(2,2)
      INTEGER ALPNUM,ALPHA,BLANK
      DATA ASTRIK/1H*/
      DATA XMARK/1HX/
      DATA BLANK/6H      /
      DATA NFLAG4/0/
1060  FORMAT(48X,25HDATA SWITCH HAS OCCURRED      )
1061  FORMAT(49X,30HJOINT PROBABILITY DISTRIBUTION   )
1062  FORMAT(1H1)
1063  FORMAT(44X,11H X-AXIS IS ,I6,6X,11H Y-AXIS IS ,2I6)
1066  FORMAT (30X,6HDXAVE=,F15.7,6HDYAVE=,F15.7,6HDZAVE=,F15.7     )
1040  FORMAT(1H ,67H MAXIMUM PROBABILITY OF UNCOMMONALITY EXCEEDED- CONT
     1INUE EXCUTION       ,2I6)
1064  FORMAT(1X,26HSYMBOL         N/SYMBOL      )
1065  FORMAT(11X,121(1H*),/,11X,1H*,55X,5HPART ,I1,4H OF ,I1,53X,1H*,/
     1,11X,121(1H*))
      IF (NFLAG4 .GT. 0) GO TO 80
      NFLG3=0
1000  FORMAT(1X,47A1)
      NFLG=0
      NI=2
      IRW=1
      NFLGEN=0
      IRT=11
      DO 1 I=1,54
      DO 1 J=1,54
      NP(I,J) =0
1     CONTINUE
      REWIND IRT
      REWIND IRW
      NFLAG4=1
80    CONTINUE
      IF (NFLAG.GT. 0) GO TO 13
      NC=DATAV+1.5
      NR=DATAH+1.5
      IF (NC .LT. 1) NC=1
      IF (NR .LT. 1) NR=1
      NP(NR,NC)=NP(NR,NC)+1
      I=54*(NC-1)+NR
      IBIN(JJ)=I
      IF (JJ .LT. NCOL) GO TO 15
      IBIN(1)=IBIN(2)
      WRITE(IRW)(IBIN(II),II=1,NCOL)
15    CONTINUE
```

```
      RETURN
 13   CONTINUE
 17   CONTINUE
      REWIND IRW
      IOPT=1
      IN=2
      IM=2
      RHO=1.0/(10.0**5)
      A(1,1)=DXAVE
      A(2,2)=DYAVE
      A(1,2)=DZAVE
      A(2,1)=DZAVE
      CALL DJCOBI(A,IM,IN,IOPT,RHO,ERR,EIGEN)
C     WRITE(6,1067) A(1,1),A(1,2),EIGEN(1,1),EIGEN(1,2)
C     WRITE(6,1067) A(2,1),A(2,2),EIGEN(2,1),EIGEN(2,2)
 1067 FORMAT(1X,2F15.7,10X,2F15.7)
      DXAVE=1.0/A(1,1)
      DYAVE=1.0/A(2,2)
      A(1,1)=EIGEN(1,1)*DXAVE
      A(1,2)=EIGEN(2,1)*DXAVE
      A(2,1)=EIGEN(1,2)*DYAVE
      A(2,2)=EIGEN(2,2)*DYAVE
      DXAVE=EIGEN(1,1)*A(1,1)+EIGEN(1,2)*A(2,1)
      DYAVE=EIGEN(2,1)*A(1,2)+EIGEN(2,2)*A(2,2)
      DZAVE=EIGEN(1,1)*A(1,2)+EIGEN(1,2)*A(2,2)+EIGEN(2,1)*A(1,1)
     1+EIGEN(2,2)*A(2,1)
      WRITE(6,1066) DXAVE,DYAVE,DZAVE
      I=0
      DO 130 NC=1,54
      DO 130 NR=1,54
      I=I+1
      IF (NP(NR,NC) .EQ. 0) GO TO 130
      XXX=NR*NR
      YYY=NC*NC
      ZZZ=NR*NC
      SUM=DXAVE*XXX+DYAVE*YYY
     1+DZAVE*ZZZ
      IF(SUM.GE.1.0)GO TO 115
      NX(I)=0
      GO TO 130
 115  NX(I)=-1
  130 CONTINUE
      WRITE (6,1062)
      WRITE(6,1061)
      WRITE(6,1066) DXAVE,DYAVE,DZAVE
      WRITE(6`,1064)
C     CALCULATE TABLE
      MAXKNT=0
      DO 22 NC=1,53
      DO 22 NR=1,53
      IF (NP(NR,NC) .GT. MAXKNT) MAXKNT=NP(NR,NC)
 22   CONTINUE
      IF (MAXKNT .LT. 46 ) MAXKNT=46
      NFACT=MAXKNT/45
      XXX=FLOAT(MAXKNT)/46.0
      NFAC=0
      IF (NFACT .LT. 1) NFACT=1
      WRITE(6,1050) BLANK,NFAC
      NFAC=NFAC+NFACT
      DO 23 I=1,46
      WRITE(6,1050) ALPNUM(I),NFAC
 1050 FORMAT (3X,A6,6X,I6)
      NFAC=NFAC+NFACT
 23   CONTINUE
      WRITE(6,1062)
C     PRINT DISTRIBUTION ON PAGE
      CORDX(1)=0.0
      CORDX(2)=CORDX(1)+60.0
      CORDX(3)=CORDX(1)+110.0
 1021 FORMAT (1H1)
      DO 65 IEND=1,54
      NC=55-IEND
      DO 66 I=1,54
      ALPHA(I)=BLANK
 66   CONTINUE
      IF (NFLG .EQ. 1) GO TO 69
      DO 68 I=1,54
      IBIN(I)=0
 68   CONTINUE
 69   CONTINUE
      DO 64 NR=1,54
      XX=FLOAT(NP(NR,NC))
      IF (NFLG .EQ. 1) GO TO 91
      IBIN(NR)=NP(NR,NC)
 91   CONTINUE
      ICAR=XX+(1.001-1.0/XXX)
      IF (ICAR .GT. 46 ) ICAR=46
      ALPHA(NR)=ALPNUM(ICAR)
 64   CONTINUE
      CORDY=FLOAT(NC)
      YMARG=XMARK
      IF (NC .NE. 54-IEND/10*10) YMARG=ASTRIK
      WRITE(6,1008) CORDY,YMARG,(ALPHA(I),I=1,54)
 1008 FORMAT(1X,F8.1,2X,A1,120A1)
 65   CONTINUE
      WRITE(6,1010)
      WRITE(6,1011) CORDX(1),CORDX(2),CORDX(3)
 1011 FORMAT (6X,F10.4,50X,F10.4,40X,F10.4)
      NFLG=1
      WRITE(6,1062)
      NSUB=1
      LWER=1
      LOW=NSTRT
 705  CONTINUE
      NHI=LOW+120-1
      NUPPER=LWER+120-1
      IF (NUPPER .GT. NCOL) NUPPER=NCOL
      WRITE(6,1062)
      CALL LABEL6(LOW,NHI,1)
      DO 131 II=NI,NSCAN
      READ(IRW) (IBIN(JJ),JJ=1,NCOL)
      DO 135 JJ=1,NCOL
      ICHECK=IBIN(JJ)
      JCHECK=NX(ICHECK)
      IF (JCHECK .NE. 0) GO TO 117
      ISYM(JJ)=ALPNUM(NSUB-1)
      NN(JJ)=0
      GO TO 135
117   ISYM(JJ)=ALPNUM(NSUB)
      NN(JJ)=-1
135   CONTINUE
      IF (NFLGEN .NE. 0) GO TO 136
      WRITE(IRT) (NN(JJ),JJ=1,NCOL)
      CALL PLTPF6(ISYM,NCOL,NBLK,INCXY(1),INCXY(2),INCXY(3),
     1INCXY(4),NCRE)
136   CONTINUE
      WRITE(6,1036) II,(ISYM(JJ),JJ=LWER,NUPPER)
1036  FORMAT(5X,I6,120A1)
1035  FORMAT(1X,I6)
131   CONTINUE
      NFLGEN=1
      REWIND IRW
      LWER=NUPPER+1
      LOW=NHI+1
      IF  (NUPPER .LT. NCOL) GO TO 705
995   CONTINUE
1010  FORMAT (11X,12(10HX*********))
      NFLGEN=0
      RETURN
      END
      SUBROUTINE CLASFY
      COMMON /LAB1/XBAR(43,12),SIGMA(43,12),ROT(43,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IXXX,IYYY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSEC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      NAMELIST/PASSES/NPASS,NCLUST
      NAMELIST/INPUTA/NSPS,NSCANS,NCHAN,LT1,LT9,LT10,LT11,LT12,LT13,
     1NSTART,NSTOP,NBTLG,MODE,ITYPE,MSEC,I4,NCRE,NSKIP,INCX,INCY,NSTX,
     2NSTY,IXXX,IYYY
      READ(5,PASSES)
      READ(5,INPUTA)
      WRITE(6,INPUTA)
      READ(5,1006) (ALPHA(I),I=1,48)
 1006 FORMAT(1X,60A1)
      KOUNT=NCLUST
      NSCANS=NSCANS-1
      INITCL=NCLUST+1
      DO 1  I=1,NPASS
      CALL  TRUCK(NCLUST,NPASS    )
      CALL  SEQMRG (NCLUST,KOUNT,INITCL        )
      CALL  CLASS  (KOUNT , I,  NPASS      )
      NCLUST=KOUNT
      INITCL=KOUNT+1
 1    CONTINUE
      RETURN
      END
      SUBROUTINE TRUCK(NCCNT,NPASS     )
      DIMENSION NNACC(12,256),MTAB(11),IPRT(256),IPLOT(256)
      DIMENSION NTBL(400)
      COMMON /LAB1/XBAR(43,12),SIGMA(43,12),ROT(43,12,12)
      COMMON /LAB2/X(12),NSYM(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IXXX,IYYY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSEC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      NFLGXX=0
      NFLAGX=0
      REWIND LT11
      REWIND LT1
      NFLAG1=0
      MFIN=0
      IXIY=IXXX*IYYY
      DO 10 I=1,IYYY
      MTAB(I)=I
10    CONTINUE
      DO 50 I=1,400
      NTBL(I)=I
50    CONTINUE
      DO 11 I=1,IYYY
      READ(LT11) (NNACC(I,JJ),JJ=1,NSPS)
      MFIN=MFIN+1
11    CONTINUE
      NUP=NSPS-IXXX+1
      NCNT=NCCNT+1
      NFLAG=0
200   CONTINUE
      III=MTAB(1)
      DO 110 JJ=1,NSPS
      IF (JJ .GT. NUP) GO TO 102
      IJ=JJ
      JI=JJ+IXXX-1
      NZERO=0
      IKNT=0
      ISUM=0
      JIJ=MTAB(1)
      NTEMP= NCCNT+1
      DO 101 I=IJ,JI
      DO 100 JIJ=1,IYYY
      IIJ=MTAB(JIJ)
      IF (NNACC(IIJ,I) .LE. NCCNT .AND. NNACC(IIJ,I) .NE. 0) GO TO 102
      IF (NNACC(IIJ,I)) 102,107,106
106   IF (NNACC(IIJ,I) .GT. NTEMP) NTEMP=NNACC(IIJ,I)
      GO TO 100
107   NZERO=NZERO+1
100    CONTINUE
101    CONTINUE
      IF (NZERO .NE. IXIY) GO TO 105
      DO 103 I=IJ,JI
      DO 104 JIJ=1,IYYY
      NNACC(JIJ,I)=NCNT
104   CONTINUE
```

```
103   CONTINUE
      NCNT=NCNT+1
      IF (NCNT .GT. 400) GO TO 999
      GO TO 110
105   CONTINUE
      DO 108 I=IJ,JI
      DO 108 JIJ=1,IYYY
      IF (NNACC(JIJ,I) .EQ. 0) NNACC(JIJ,I)=NTEMP
108   CONTINUE
      GO TO 110
102   CONTINUE
110    CONTINUE
      DO 111 JJ=1,NSPS
      IF (JJ .EQ. NSPS ) GO TO 111
      IF (NNACC(III,JJ) .LE. NCCNT) GO TO 111
      IF (NNACC(III,JJ+1) .LE. NCCNT) GO TO 111
      IF (NNACC(III,JJ) .LE. 0) GO TO 111
      IF (NNACC(III,JJ+1) .LE. 0) GO TO 111
      IF (NNACC(III,JJ) .EQ. NNACC(III,JJ+1)) GO TO 111
      IJ=NNACC(III,JJ)
      JI=NNACC(III,JJ+1)
      IF (JI .GT. 400 .OR. IJ .GT. 400) GO TO 111
      IF (NTBL(JI) .GT. NTBL(IJ)) GO TO 125
      NTBL(IJ)=NTBL(JI)
      GO TO 111
125   CONTINUE
      NTBL(JI)=NTBL(IJ)
111   CONTINUE
1007  FORMAT (1X,I6)
      WRITE(LT1) (NNACC(III,JJ),JJ=1,NSPS)
      IF (MFIN .GE. NSCANS) GO TO 999
      IYI=MTAB(1)
      READ(LT11) (NNACC(IYI,JJ),JJ=1,NSPS)
      MFIN=MFIN+1
      NTEMP=MTAB(1)
      IYY=IYYY-1
      DO 121 I=1,IYY
      MTAB(I)=MTAB(I+1)
121   CONTINUE
      MTAB(IYYY)=NTEMP
      GO TO 200
999   CONTINUE
      DO 122 I=2,IYYY
      III=MTAB(I)
      DO 112 JJ=1,NSPS
      IF (JJ .EQ. NSPS ) GO TO 112
      IF (NNACC(III,JJ) .LE. NCCNT) GO TO 112
      IF (NNACC(III,JJ+1) .LE. NCCNT) GO TO 112
      IF (NNACC(III,JJ) .LE. 0) GO TO 112
      IF (NNACC(III,JJ+1) .LE. 0) GO TO 112
      IF (NNACC(III,JJ) .EQ. NNACC(III,JJ+1)) GO TO 112
      IJ=NNACC(III,JJ)
      JI=NNACC(III,JJ+1)
      IF (JI .GT. 400 .OR. IJ .GT. 400) GO TO 112
      IF (NTBL(JI) .GT. NTBL(IJ)) GO TO 126
      NTBL(IJ)=NTBL(JI)
      GO TO 112
126   CONTINUE
      NTBL(JI)=NTBL(IJ)
112   CONTINUE
      WRITE(LT1) (NNACC(III,JJ),JJ=1,NSPS)
122   CONTINUE
      END FILE LT1
      REWIND LT1
      REWIND LT11
      REWIND LT12
      WRITE(6,1007) (NTBL(I),I=1,400)
      DO 113 I=1,400
      IF (NTBL(I) .EQ. I) GO TO 113
      JI=I+1
      IF (NTBL(JI) .NE. I) GO TO 114
      NTBL(JI)=NTBL(I)
114   CONTINUE
113   CONTINUE
      II=1
      NTEMP=II
      DO 116 I=2,400
      IF (NTBL(I)-NTBL(I-1)) 117,118,119
119   IF (NTBL(I) .NE. I) GO TO 117
      NTBL(I-1)=NTEMP
      II=II+1
      NTEMP=II
      GO TO 116
118   NTBL(I-1)=NTEMP
      GO TO 116
117   N=NTBL(I)
      NTBL(I-1)=NTEMP
      NTEMP=NTBL(N)
116   CONTINUE
      NTBL(400)=NTEMP
      WRITE(6,1007) (NTBL(I),I=1,400)
      LWER=1
      LOW=NSTART
705   CONTINUE
      NUPPER=LWER+120-1
      NHI=LOW+120-1
      IF (NUPPER .GT. NSPS ) NUPPER=NSPS
      IDIF=NUPPER-LWER+1
      WRITE(6,1005)
1005  FORMAT(1H1)
      CALL LABELA(LOW,NHI,1)
      DO 710 II=1,NSCANS
      READ(LT1) (NNACC(1,JJ),JJ=1,NSPS)
      DO 115 JJ=1,NSPS
      IB=NNACC(1,JJ)
      IF (IB .LE. 0) GO TO 115
      NNACC(1,JJ)=NTBL(IB)
115   CONTINUE
      IF (NFLGXX .GT. 0) GO TO 127
      WRITE(LT12) (NNACC(1,JJ),JJ=1,NSPS)
```

```
127   CONTINUE
      JI=0
      DO 711 JJ=LWER,NUPPER
      JI=JI+1
      N=NNACC(1,JJ)-(NNACC(1,JJ)-1)/45*45+2
      IPRT(JJ)=NSYM(N)
      IPLOT(JI)=NSYM(N)
711   CONTINUE
      WRITE(6,1003) II,(IPRT(JJ),JJ=LWER,NUPPER)
      CALL PLTBEA(IPLOT,IDIF,NBLK,INCX,INCY,NSTX,NSTY,
     1NCRE,NFLAGX,NFLAG1)
1003  FORMAT(4X,I6,1H*,120A1)
710   CONTINUE
      REWIND LT1
      NFLAGX=0
      NFLGXX=1
      NFLAG1=0
      LWER=NUPPER+1
      LOW=NHI+1
      IF (NUPPER .LT. NSPS ) GO TO 705
570   CONTINUE
      NCCNT=NCNT-1
      END FILE LT12
      REWIND LT12
      REWIND LT1
      IXXX=IXXX-4
      IYYY=IYYY-4
      LT11=13
      RETURN
      END
      SUBROUTINE SEQMRG(NCLUST,KOUNT,INITCL    )
      COMMON /LAB1/XBAR(43,12),SIGMA(43,12),ROT(43,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IXXX,IYYY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSEC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      DOUBLE PRECISION A(12,12),EIGEN(12,12)
      DIMENSION MERGE(200),MPOP(200),NEXEC(20),C(43,78),B(12,12)
      DIMENSION COM(24)
      EQUIVALENCE(COM(1),NSPS)
1000  FORMAT(1X,I6,12F10.3)
1001  FORMAT (1X,4HXBAR    )
1002  FORMAT(1X,16HDID NOT CONVERG      )
1003  FORMAT(1X,7HICLUST=   ,I6,14HMERGE(ICLUST)=   ,I6)
1004  FORMAT (1X,5HRHO=   ,F15.7,5HERR=   ,F15.7)
1005  FORMAT (1X,12F10.4)
1006  FORMAT(1X,12I6)
1007  FORMAT (1X,23HMERGING WILL TAKE PLACE    )
1008  FORMAT(1H )
1009  FORMAT (13H COV. MATRIX    )
1010  FORMAT (12H NORM EIGEN    )
1011  FORMAT (18H P.A. COV. MATRIX    )
1012  FORMAT(1H ,6HASUM= ,F15.7,7HCLUSTER,I4)
1013  FORMAT(1X,28HXBAR(I,J),J=1,12),I=1,KOUNT     )
1014  FORMAT (1X,29HSIGMA(I,J),J=1,12),I=1,KOUNT     )
1015  FORMAT(1X,55HROT(I,ICHAN,JCHAN),JCHAN=1,12),ICHAN=1,12),I=1,KOUNT
     1)      )
1016  FORMAT (1X,I6,(12F10.3))
      NFLG=0
      CZECH=FLOAT(NCHAN)-2.0
      REWIND LT10
      REWIND LT12
      RHO=1.0/(10.0**5)
      IF (NSKIP .EQ. 0) GO TO 6
      DO 7 I=1,NSKIP
      CALL SKRBIN(LT10,1,NOP)
7     CONTINUE
6     CONTINUE
      DO 5 ICLUST=1,NCLUST
      MERGE(ICLUST)=ICLUST
5     CONTINUE
      IM=NCHAN
      IN=IM
      IOPT=1
      DO 10 ICLUST=INITCL,NCLUST
      IF (NFLG .GT. 0) GO TO 11
      IF (KOUNT .GE. 43 ) GO TO 11
      KOUNT=KOUNT+1
      IFLAG=KOUNT
      CALL FETCOR(IFLAG,      C,       MPOP,NFLG,INITCL    )
      WRITE(6,1001)
      WRITE(6,1000) IFLAG,(XBAR(IFLAG,I),I=1,12)
      MI=1
      MJ=12
      MK=12
      DO 500 MM=1,12
      WRITE(6,1000) IFLAG,(C(IFLAG,MR),MR=MI,MJ)
      MI=MJ+1
      MJ=MJ+MK-MM
500   CONTINUE
      CALL AMTRX (IFLAG,XBAR,C,A,NCHAN)
      WRITE(6,1008)
      WRITE(6,1009)
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
      CALL DJCOBI (A,IM,IN,IOPT,RHO,ERR,EIGEN)
      WRITE(6,1004) RHO,ERR
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
      WRITE(6,1008)
      WRITE(6,1005) ((EIGEN(MI,MJ),MJ=1,12),MI=1,12)
      IF (ERR .EQ. 0.0) GO TO 15
      MERGE(ICLUST)=0
      KOUNT=KOUNT-1
      WRITE(6,1002)
      GO TO 10
15    CONTINUE
      CALL ROTA (IFLAG,ROT,EIGEN,NCHAN,A,SIGMA)
      MERGE(ICLUST)=KOUNT
      WRITE(6,1003) ICLUST,MERGE(ICLUST)
      MPOP(KOUNT)=MPOP(ICLUST)
      IF (KOUNT .EQ. 1) GO TO 10
      MCLUST=KOUNT-1
      DO 20 ICHECK=1,20
      NEXEC(ICHECK)=0
20    CONTINUE
      MCHECK=1
      DO 25 JCLUST=1,NCLUST
      IF(MERGE(JCLUST).LT.MCHECK)GO TO 25
      MCHECK=MCHECK+1
      IF (MERGE(JCLUST) .EQ. KOUNT) GO TO 26
      JFLAG=MERGE(JCLUST)
      DO 30 ICHAN=1,NCHAN
      X(ICHAN)=XBAR(JFLAG ,ICHAN)-XBAR(KOUNT,ICHAN)
30    CONTINUE
      IFLAG=KOUNT
      CALL KCHECK(IFLAG,      ROT,X,SIGMA,ASUM,NCHAN)
      WRITE(6,1008)
      WRITE(6,1012)ASUM,JFLAG
      IF (ASUM .GT. CZECH) GO TO 25
      IFLAG=JFLAG
      CALL KCHECK(IFLAG,      ROT,X,SIGMA,ASUM,NCHAN)
      WRITE(6,1008)
      WRITE(6,1012)ASUM,JFLAG
      IF (ASUM .GT. CZECH) GO TO 25
      NEXEC(1)=NEXEC(1)+1
      NSUB=NEXEC(1)+1
      NEXEC(NSUB)=JFLAG
25    CONTINUE
26    IF (NEXEC(1) .EQ. 0) GO TO 10
      DO 501 KK=1,NSUB
      WRITE(6,1006) KK,NEXEC(KK)
501   CONTINUE
      MSUB=NEXEC(1)+1
      TOTAL=MPOP(KOUNT)
      DO 31 IRUN=2,MSUB
      NSUB=NEXEC(IRUN)
      SUM=MPOP(NSUB)
      TOTAL=TOTAL+SUM
31    CONTINUE
      INUM=0
      DEN=MPOP(KOUNT)
      DO 35 ICHAN=1,NCHAN
      X(ICHAN)=XBAR(KOUNT,ICHAN)*DEN/TOTAL
      DO 40 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      B(ICHAN,JCHAN)=C(KOUNT,INUM)*DEN/TOTAL
40    CONTINUE
35    CONTINUE
      DO 45 IRUN=2,MSUB
      NSUB=NEXEC(IRUN)
      INUM=0
      DEN=MPOP(NSUB)
      DO 50 ICHAN=1,NCHAN
      X(ICHAN)=X(ICHAN)+XBAR(NSUB,ICHAN)*DEN/TOTAL
      DO 55 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      B(ICHAN,JCHAN)=B(ICHAN,JCHAN)+C(NSUB,INUM)*DEN/TOTAL
55    CONTINUE
50    CONTINUE
45    CONTINUE
      DO 60 ICHAN=1,NCHAN
      DO 65 JCHAN=ICHAN,NCHAN
      A(ICHAN,JCHAN)=B(ICHAN,JCHAN)-X(ICHAN)*X(JCHAN)
      A(JCHAN,ICHAN)=A(ICHAN,JCHAN)
65    CONTINUE
60    CONTINUE
      WRITE(6,1009)
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
      CALL DJCOBI(A,IM,IN,IOPT,RHO,ERR,EIGEN)
      WRITE(6,1004) RHO,ERR
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
      WRITE(6,1008)
      WRITE(6,1005) ((EIGEN(MI,MJ),MJ=1,12),MI=1,12)
      IF (ERR .NE. 0.0) GO TO 10
      WRITE(6,1007)
      IFLAG=NEXEC(2)
      MPOP(IFLAG)=TOTAL
      INUM=0
      DO 70 ICHAN=1,NCHAN
      XBAR(IFLAG,ICHAN)=X(ICHAN)
      DO 75 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(IFLAG,INUM)=B(ICHAN,JCHAN)
75    CONTINUE
70    CONTINUE
```

```
      CALL ROTA (IFLAG,ROT,EIGEN,NCHAN,A,SIGMA)
      DO 80 JCLUST=1,NCLUST
      DO 85 IRUN=2,MSUB
      NSUB=NEXEC(IRUN)
      IF (MERGE(JCLUST) .NE. NSUB) GO TO 85
      MERGE(JCLUST)=IFLAG
85    CONTINUE
80    CONTINUE
      MERGE(ICLUST)=IFLAG
      IF(NEXEC(1).EQ.1)GO TO 94
      ISW=0
      JCHECK=1
      DO 90 JCLUST=1,NCLUST
      IDUM=MERGE(JCLUST)
   91 IF(MERGE(JCLUST).LT.JCHECK)GO TO 90
      IF(MERGE(JCLUST).GT.JCHECK)GO TO 92
      IF(ISW.EQ.1)GO TO 93
      JCHECK=JCHECK+1
      IF(JCHECK.EQ.KOUNT)GO TO 94
      GO TO 90
   92 MERGE(JCLUST)=MERGE(JCLUST)-1
      ISW=1
      GO TO 91
93    IF (JCHECK .GT. KOUNT) GO TO 94
      ISW=0
      INUM=0
      MPOP(JCHECK)=MPOP(IDUM)
      DO 95 ICHAN=1,NCHAN
      XBAR(JCHECK,ICHAN)=XBAR(IDUM,ICHAN)
      SIGMA(JCHECK,ICHAN)=SIGMA(IDUM,ICHAN)
      DO 100 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(JCHECK,INUM)=C(IDUM,INUM)
      ROT(JCHECK,ICHAN,JCHAN)=ROT(IDUM,ICHAN,JCHAN)
      ROT(JCHECK,JCHAN,ICHAN)=ROT(IDUM,JCHAN,ICHAN)
100   CONTINUE
95    CONTINUE
      DO 96 LCLUST=1,NCLUST
      IF(MERGE(LCLUST).NE.IDUM)GO TO 96
      MERGE(LCLUST)=JCHECK
96    CONTINUE
      JCHECK=JCHECK+1
90    CONTINUE
   94 KOUNT=KOUNT-NEXEC(1)
10    CONTINUE
11    CONTINUE
      WRITE(LT9) (COM(I),I=1,24)
      WRITE(LT9) ((XBAR(I,J),I=1,KOUNT),J=1,12)
      WRITE(LT9) ((SIGMA(I,J),I=1,KOUNT),J=1,12)
      WRITE(LT9) (((ROT(I,ICHAN,JCHAN),I=1,KOUNT),ICHAN=1,NCHAN),
     1JCHAN=1,NCHAN)
      WRITE(6,1013)
      DO 510 I=1,KOUNT
      WRITE(6,1000) I,(XBAR(I,J),J=1,12)
510   CONTINUE
```

```
      WRITE(6,1014)
      DO 511 I=1,KOUNT
      WRITE(6,1000) I,(SIGMA(I,J),J=1,12)
511   CONTINUE
      WRITE(6,1015)
      DO 512 I=1,KOUNT
      WRITE(6,1016) I,((ROT(I,ICHAN,JCHAN),JCHAN=1,12),ICHAN=1,12)
512   CONTINUE
      DO 513 I=1,NCLUST
      IF(MERGE(I).GT.KOUNT)GO TO 514
      WRITE(6,515)I,MERGE(I)
  515 FORMAT(1X,7HCLUSTER,I4,1X,5HCLASS,I4)
  513 CONTINUE
  514 CONTINUE
      DO 660 I=1,KOUNT
      DO 620 ICHAN=1,NCHAN
      DO 610 JCHAN=1,NCHAN
      B(ICHAN,JCHAN)=ROT(I,JCHAN,ICHAN)/SIGMA(I,ICHAN)
610   CONTINUE
620   CONTINUE
      DO 650 ICHAN=1,NCHAN
      DO 640 KCHAN=1,NCHAN
      SUM=0.0
      DO 630 JCHAN=1,NCHAN
      SUM=SUM+ROT(I,ICHAN,JCHAN)*B(JCHAN,KCHAN)
630   CONTINUE
      A(ICHAN,KCHAN)=SUM
640   CONTINUE
650   CONTINUE
      WRITE(6,600) I
600   FORMAT(1X,13HCLASS ELLIPSE,I4)
      WRITE(6,1005) ((A(IA,JA),JA=1,NCHAN),IA=1,NCHAN)
660   CONTINUE
      REWIND LT9
      RETURN
      END
      SUBROUTINE FETCOR(IFLAG,C,NPOP,NFLG,N    )
      COMMON /LAB1/XBAR(43,12),SIGMA(43,12),ROT(43,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IX,IY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSEC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      DIMENSION NPOP(1),C(43,78),NDAT(255)
      DATA NCNT/0/
      INUM=0
      NFLG1=0
      NFLG3=0
      DO 5 ICHAN=1,NCHAN
      XBAR(IFLAG,ICHAN)=0.0
      DO 10 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(IFLAG,INUM)=0.0
 10   CONTINUE
 5    CONTINUE
      KNT=0
 40   CONTINUE
      IF (NCNT .GE. NSCANS) GO TO 70
      NFLG2=0
      READ(LT12)(NDAT(JJ),JJ=1,NSPS)
      NCNT=NCNT+1
      NFLG1=1
      NFLAG2=1
      DO 20 JJ=1,NSPS
      CALL GET(  X(1),NSPS,0,NCHAN,NSCANO,LT10,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSEC    )
      IF (NDAT(JJ) .NE. N) GO TO 30
      KNT=KNT+1
      NFLG2=1
      AI=FLOAT(KNT)
      INUM=0
      DO 25 ICHAN=1,NCHAN
      XBAR(IFLAG,ICHAN)=(1.0-1.0/AI)*XBAR(IFLAG,ICHAN)+X(ICHAN)/AI
      DO 26 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(IFLAG,INUM)=(1.0-1.0/AI)*C(IFLAG,INUM)+X(ICHAN)*X(JCHAN)/AI
 26   CONTINUE
 25   CONTINUE
      GO TO 20
 30   CONTINUE
      IF (NFLG3 .EQ. 1) GO TO 20
      IF (NDAT(JJ) .NE. N+1) GOTO 20
      NSAV=NCNT
      NFLG3=1
 20   CONTINUE
C     WRITE(6,1000) NCNT,KNT,NSAV,NBCKUP,N,NFLG2,NFLG3
 1000 FORMAT(1X,7I8)
      IF (NFLG2 .NE. 0) GO TO 40
      IF (NFLG3 .EQ. 0) GO TO 40
      NBCKUP=NCNT-NSAV+1
      CALL BSRECD(LT10,NBCKUP*I4,RE)
      SUBROUTINE AMTRX(IFLAG,XBAR,C,A,NCHAN)
      DIMENSION XBAR(43,12),C(43,78)
      DOUBLE PRECISION A(12,12)
      INUM=0
      DO 1 ICHAN=1,NCHAN
      DO 2 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      A(ICHAN,JCHAN)=C(IFLAG,INUM)-XBAR(IFLAG,ICHAN)*XBAR(IFLAG,JCHAN)
      A(JCHAN,ICHAN)=A(ICHAN,JCHAN)
2     CONTINUE
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROTA(IFLAG,ROT,EIGEN,NCHAN,A,SIGMA)
      DIMENSION ROT(43,12,12)
      DIMENSION SIGMA(43,12)
      DOUBLE PRECISION A(12,12)
      DOUBLE PRECISION EIGEN(12,12)
      DO 1 ICHAN=1,NCHAN
      SIGMA(IFLAG,ICHAN)=A(ICHAN,ICHAN)
      DO 2 JCHAN=1,NCHAN
      ROT(IFLAG,ICHAN,JCHAN)=EIGEN(JCHAN,ICHAN)
2     CONTINUE
1     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KCHECK (IFLAG,       ROT,X,SIGMA,ASUM,NCHAN)
      DIMENSION ROT(43,12,12),SIGMA(43,12),X(1)
      ASUM=0.0
      DO 3 ICHAN=1,NCHAN
      SUM=0.0
      DO 4 JCHAN=1,NCHAN
      SUM=SUM+ROT(IFLAG,ICHAN,JCHAN)*X(JCHAN)
4     CONTINUE
      ASUM=ASUM+SUM*SUM/SIGMA(IFLAG,ICHAN)
3     CONTINUE
      RETURN
      END
      SUBROUTINE CLASS(NCLASS,NTEST,NPASS    )
      COMMON /LAB1/XBAR(43,12),SIGMA(43,12),ROT(43,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IX,NDUMMY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSEC,I4,NCRF,
     1NSKIP,INCX,INCY,NSTX,NSTY
      DIMENSION W(12),MTAB(3)
      DIMENSION NDAT(255,3),PRNT(255)
      DIMENSION COM(24)
      EQUIVALENCE(COM(1),NSPS)
      REWIND LT9
      REWIND LT10
      REWIND LT1
      REWIND LT12
      REWIND LT13
      CZECH=NCHAN
      IF (NSKIP .EQ. 0) GO TO 601
      DO 602 I=1,NSKIP
      CALL SKRBIN(LT10,1,NOP)
602   CONTINUE
601   CONTINUE
      READ(LT9) (COM(I),I=1,24)
      READ(LT9) ((XBAR(I,J),I=1,NCLASS),J=1,12)
      READ(LT9) ((SIGMA(I,J),I=1,NCLASS),J=1,12)
      READ(LT9) (((ROT(I,ICHAN,JCHAN),I=1,NCLASS),ICHAN=1,NCHAN),
     1JCHAN=1,NCHAN)
      DO 1 I=1,3
1     MTAB(I)=I
      DO 10 IEND=1,NSCANS
      READ(LT12) (NDAT(I,1),I=1,NSPS)
      NFLAG2=1
      DO 20 ISUBN=1,NSPS
      CALL GET(X(1),NSPS,0,NCHAN,NSCANO,LT10,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRF,ITYPE,MSEC     )
      IF (NDAT(ISUBN,1) .GT. 0 ) NDAT(ISUBN,1)=0
      SMALL=1.75*CZECH
      DO 25 ICLASS=1,NCLASS
      DO 30 ICHAN=1,NCHAN
      W(ICHAN)=X(ICHAN)-XBAR(ICLASS ,ICHAN)
30    CONTINUE
      ASUM=0.0
      DO 35 ICHAN=1,NCHAN
      SUM=0.0
      DO 40 JCHAN=1,NCHAN
      SUM=SUM+W(JCHAN)*ROT(ICLASS,ICHAN,JCHAN)
40    CONTINUE
41    ASUM=ASUM+SUM*SUM/SIGMA(ICLASS,ICHAN)
35    CONTINUE
      IF (ASUM .GT. SMALL) GO TO 25
      SMALL=ASUM
      NDAT(ISUBN,1)=ICLASS
25    CONTINUE
1016  FORMAT(1X,3I6,F10.2)
20    CONTINUE
```

```
      WRITE(LT1)(NDAT(I,1),I=1,NSPS)
10    CONTINUE
      END FILE LT1
      REWIND LT1
      REWIND LT12
      REWIND LT10
      DO 610 IZ=1,NSCANS
      IY=MTAB(1)
      READ(LT1) (NDAT(IA,IY),IA=1,NSPS)
      NTEMP=MTAB(1)
      MTAB(1)=MTAB(2)
      MTAB(2)=MTAB(3)
      MTAB(3)=NTEMP
      IF (IZ .LT. 3) GO TO 610
      DO 620 IA=NSTART,NSTOP
      IF (IA .EQ. 1) GO TO 620
      IF (IA+1 .GT. NSTOP) GO TO 620
      IIY=MTAB(2)
      IF (NPASS .EQ. NTEST) GO TO 621
      IF (NDAT(IA,IIY) .LT. 0) GO TO 620
621   CONTINUE
      IM=MTAB(1)
      IN=MTAB(2)
      M=NDAT(IA,IM)
      N=NDAT(IA-1,IN)
      IF (M .NE. N) GO TO 650
      IL=MTAB(3)
      L=NDAT(IA,IL)
      IF (M .NE. L) GO TO 650
      NDAT(IA,IIY)=M
      GO TO 620
650   IM=MTAB(1)
      M=NDAT(IA-1,IM)
      IN=MTAB(3)
      N=NDAT(IA-1,IN)
      IF (M .NE. N) GO TO 620
      L=NDAT(IA+1,IM)
      IF (M .NE. L) GO TO 620
      NDAT(IA,IIY)=M
620   CONTINUE
      IF (IZ .LT. 3) GO TO 610
      L=MTAB(1)
      WRITE(LT13) (NDAT(I,L),I=1,NSPS)
610   CONTINUE
      DO 611 I=2,3
      L=MTAB(I)
      WRITE(LT13) (NDAT(IL,L),IL=1,NSPS)
611   CONTINUE
      REWIND LT1
      REWIND LT9
      REWIND LT13
      REWIND LT10
      LOW=NSTART
      LWER=1
800   CONTINUE
```

```
      NHI=LOW+120-1
      NUPPER=LWER+120-1
      IF (NUPPER .GT. NSPS ) NUPPER=NSPS
      IDIF=NUPPER-LWER+1
      WRITE(6,1007)
1007  FORMAT(1H1)
      CALL LABELA(LOW,NHI,1)
      DO 801 II=1,NSCANS
      READ(LT13) (NDAT(JJ,1),JJ=1,NSPS)
      DO 803 JJ=LWER,NUPPER
      IB=NDAT(JJ,1)
      IRND=IB-(IB-1)/45*45+2
      PRNT(JJ)=ALPHA(IRND)
803   CONTINUE
      CALL PLTBEA(PRNT(LWER),IDIF,NBLK,INCX,INCY,NSTX,NSTY,NCRE,
     1NFLAGX,NFLAG1)
      WRITE(6,1008) II, (PRNT(JJ),JJ=LWER,NUPPER)
1008  FORMAT(4X,I6,1H*,120A1)
801   CONTINUE
      REWIND LT13
      NFLAG1=0
      NFLAGX=0
      LWER=NUPPER+1
      LOW=NHI+1
      IF (NUPPER .LT. NSPS ) GO TO 800
802   CONTINUE
      NSCANS=NSCANS-7
      RETURN
      END
```

REFERENCES

1. Lueder, D. R.. Aerial Photographic Interpretation. New York, McGraw Hill Book Company, Inc., 1959.

2. Thompson, M. M. (Editor): Manual of Photogrammetry. Falls Church, Virginia, American Society of Photogrammetry, 1966.

3. Jensen, N.. Optical and Photographic Reconnaissance Systems. New York, John Wiley and Sons, Inc., 1968.

4. Detchmendy, D. M., and Pace, W. H.. A Model for Spectral Signature Variability. TRW IOC 4913.7-71-193.

5. Eppler, W. G., Helmke, C. A., and Evans, R. H.. Table Look-Up Approach to Pattern Recognition. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971.

6. Remote Multispectral Sensing in Agriculture. Purdue University Research Bulletin No. 844, 1968, and No. 873, 1970.

7. Bauer, M. E., Swain, P H., et al.. Detection of Southern Corn Leaf Blight by Remote Sensing Techniques. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971

8. Hoffer, R. M., and Goodrick, F. E.. Variations in Automatic Classification over Extended Remote Sensing Test Sites. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971.

9. Anuta, P. E., Kristof, S. J., et al.: Crop, Soil, and Geological Mapping from Digitized Multispectral Satellite Photography. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971.

10. Stoner, E. R., and Horvath, E. H. The Effect of Cultural Practices on Multispectral Response from Surface Soil. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971.

11. Smedes, H. W., Lennerud, H. J., et al. Digital Computer Mapping by Clustering Techniques Using Color Film as a Three Band Sensor. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971.

12. Smedes, H. W., Spencer, M. M., and Thomson, F. J.: Preprocessing of Multispectral Data and Simulation of ERTS Data Channels to Map Computer Terrain Maps of a Yellowstone National Park Test Site. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971.

REFERENCES (Concluded)

13. Bond, A. D., Dasarathy, B. V., and Atkinson, R. J.. Feature Selection and Supervised Non-Parametric Classification Applied to Earth Resources Multispectral Scanner Data. NASA Contractor Report, Contract NAS8-21805, 1971.

14. Wacker, A. C., and Landgrebe, D. A. Boundaries in Multispectral Imagery by Clustering. IEEE Symposium on Adaptive Processes, 1970.

15. Roth, C. B., and Baumgardner, M. F. Correlation Studies with Ground Truth and Multispectral Data. Effect of Size of Training Field. Proc. Seventh Int. Sym. on Remote Sensing of Environment, 1971.

16. Bendat, J. S., and Piersol, A. G.. Measurement and Analysis of Random Data. New York, John Wiley and Sons, Inc., 1966.

17. Remote Sensing of Earth Resources, A Literature Survey with Indexes. NASA SP-7036, 1970.

18. Su, M. Y.: The Composite Sequential Clustering Technique for Analysis of Multispectral Scanner Data. Northrop Services Inc., TR-250-1141, NASA Contract NAS8-27364, 1972.

19 Nagy, G., Shelton, G., and Tolaba, J. Procedural Questions in Signature Analysis. Proc. Seventh Int. Sym. on Remote Sensing of the Environment, 1971

20. Nagy, G., and Tolaba, J. Nonsupervised Crop Classification Through Airborne Multispectral Observations. IBM J. Res. Dev., 1971.

21. Ayres, F.. Matrices. New York, Schaum Publishing Co., 1962.

22. Kendall, M. G., and Stuart, A. The Advanced Theory of Statistics. New York, Hafner Publishing Co., Vol. 3, 1966.

23. Sebestyen, G. S. Decision-Making Processes in Pattern Recognition. New York, The Macmillan Co., 1962.

24. Keeping, E. S. Introduction to Statistical Inference. Princeton, New Jersey, D. Van Nostrand Co., Inc., 1962.

25. Phillips, M. R. Correlation Signatures of Wet Soils and Snows. IIT Research Institute Final Report J6243-6, NASA Contract NAS8-26797, 1972.

NASA-Langley, 1973 —— 13

> *"The aeronautical and space activities of the United States shall be conducted so as to contribute . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*
>
> —National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546