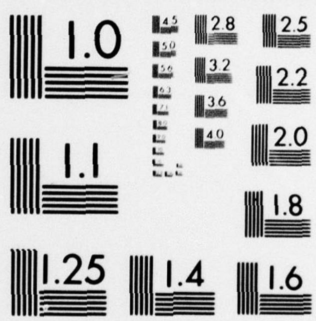
AD-A060 039 CONTROL DATA CORP MINNEAPOLIS MINN DIGITAL IMAGE SYS--ETC F/G 17/9
STATISTICAL EVALUATION OF FOP FEATURES.(U)
JUN 78 L KIRVIDA, D HANSON, L BAUMANN F30602-76-C-0349
UNCLASSIFIED RADC-TR-78-108 NL

1 OF 2
AD
A060039


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

② LEVEL II

5C

AD A060039

RADC-TR-78-108
Final Technical Report
June 1978

STATISTICAL EVALUATION OF FOP FEATURES

Mr. L. Kirvida
Mr. D. Hanson
Dr. L. Baumann

Control Data Corporation

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-78-108 has been reviewed and is approved for publication.

APPROVED: Ronald B. Haynes

RONALD B. HAYNES
Project Engineer

APPROVED: H Davis

HOWARD DAVIS
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRE) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-78-108	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): STATISTICAL EVALUATION OF FOP FEATURES		5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, July 1976 - September 1977
		6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Mr. L. Kirvida, Mr. D. Hanson, Dr. L. Baumann		8. CONTRACT OR GRANT NUMBER(s): F30602-76-C-0349
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Control Data Corporation/Digital Image Systems Div, 2800 East Old Shakopee Road, Minneapolis MN 55420		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63746F, 20370302
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRRE), Griffiss AFB NY 13441		12. REPORT DATE: June 1978
		13. NUMBER OF PAGES: 93
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same		15. SECURITY CLASS. (of this report): UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

12 100 p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Ronald B. Haynes (IRRE)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Radar exploitation
Radar change detection
Radar digital cueing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes research conducted to determine the potential of automatic classification of military targets from large area surveillance radar systems. The Air Force has developed the Modular Change Detector (MCD) for performing real-time change detection using a reference and mission pair of radar images. The automatic classification experiments performed under this contract established the feasibility of automatically screening certain classes of military targets. Three automatic classification experiments were demonstrated with 100%

over

DD FORM 1 JAN 73 1473

78 10 10 087

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 732

correct classification of military equipment in operational deployment. With the incorporating of this screening technique on the output of the MCD it is possible to cue an interpreter to particular targets of interest. Such cueing helps discriminate against false alarms changes, which previous studies have shown increases the interpreters coverage rate and his probability of detection.

PREFACE

This is the final report for Contract F30602-76-C-0349. The period covered is from July 1976 through September 1977.

The contract technical monitor is Mr. Ronald Haynes of RADC. His assistance and advice are gratefully acknowledged.

The project manager was Mr. L. Kirvida. Principal contributions were from Mr. D. Hanson and Dr. L. Baumann.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (CONTD.)

LIST OF TABLES

EVALUATION

The work completed under Contract F30602-76-C-0349 is significant because it has established the potential of correctly classifying military equipment in performing real-time radar change detection exploitation. With large area reconnaissance radar surveillance systems, methods for automatically classifying or cueing interpreters to targets is necessary if intelligence data is to be extracted from these systems in a timely manner. The Air Force has developed the Modular Change Detector (MCD) for performing real-time change detection. This contract established a means of automatically classifying targets which are displayed on the output of the MCD to cue interpreters to areas of interest. The work accomplished under this effort falls directly under TPO-R2C, Ground Target Detection and Identification whose objective is to improve the Air Force's real-time exploitation capabilities.

The technique developed in this effort is currently being applied to the advanced real-time UPD-4 Side Looking Radar exploitation system.

Ronald B. Haynes

RONALD B. HAYNES
Project Engineer

1.0 OBJECTIVE

Military tactical targets are often composed of a number of individual objects. For example, a mobile missile site contains radar units, support equipment and launchers. These objects are clustered with some general characterization such as the spacing between objects, the number of objects and, possibly, the pattern of deployment. Similarly, gun emplacements, armor, etc. are composed of clusters.

The objectives of this study were to 1) develop a cluster detection procedure, 2) evaluate the procedure with typical outputs from radar change detection, and 3) evaluate the potential for improvements in system effectiveness; either more area processed per unit time or better target discrimination.

1.1 SYSTEM CONFIGURATION

The relationship of the cluster detector to the radar exploitation system is shown in Figure 1-1. Changes detected by a radar change detection* system are the inputs. The cluster detector determines which changes are associated with clusters. The output format is under operator control. It could be all the changes with the clusters indicated by graphics, only the clusters or only clusters in designated target classes of interest.

The subsequent processing is the normal exploitation management and target reporting.

If the change detection is considered as a target screener, then the cluster detector adds another stage of filtering or discrimination. Rather than show all changes to the analyst, it either "flags" or passes only those with higher probability of being targets.

---

* Radar change detection is a procedure for detecting objects in radar imagery which have moved between two coverages of an area. The radar return from the object remains in the difference image.

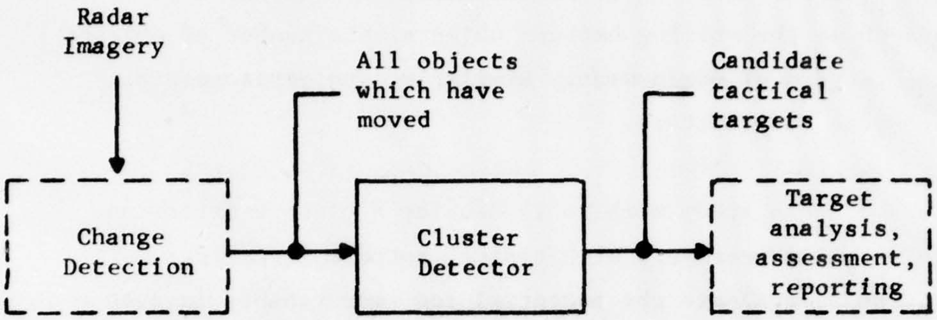


Figure 1-1 Relation of Cluster Detector to Radar Exploitation System.

1.2 PROCEDURE DESCRIPTION

There are two primary steps in the procedure; clustering and classification. These are illustrated in Figure 1-2. The clusterer searches for all clusters which satisfy some general parameters. Typical parameters are spacings between objects, size of cluster and number of objects. These parameters should include all targets of interest. Since the target categories are time and situation-dependent, the parameter values are under manual control.

The cluster classifier discriminates specific target classes of interest. This feature is useful in searching for specific targets of interest. Two further benefits of the classifier are 1) assignment of a confidence measure on the probability the cluster belongs to a target class and 2) computation of target features. These features could be used by the analyst in target identification or might be useful in cataloging the targets for subsequent review or selection from a data base. These benefits were not evaluated since they were beyond the scope of the effort.

1.3 RESULTS

Three experiments were performed to evaluate the procedure. The input data were imagery samples of known targets provided by RADC. Each sample was an image pair from two successive coverages of the same area. Only those cases where the targets moved between coverages were used. This ensured that the change detection procedures would give realistic outputs. That is, the change detection parameters were set to detect all the changes due to targets. The false alarms which occurred correspond to these same parameters. Thus the false alarm distribution is consistent with the target changes.

The imagery data were processed through a software change detection system. This software simulates the Modular Change Detector, MCD, developed

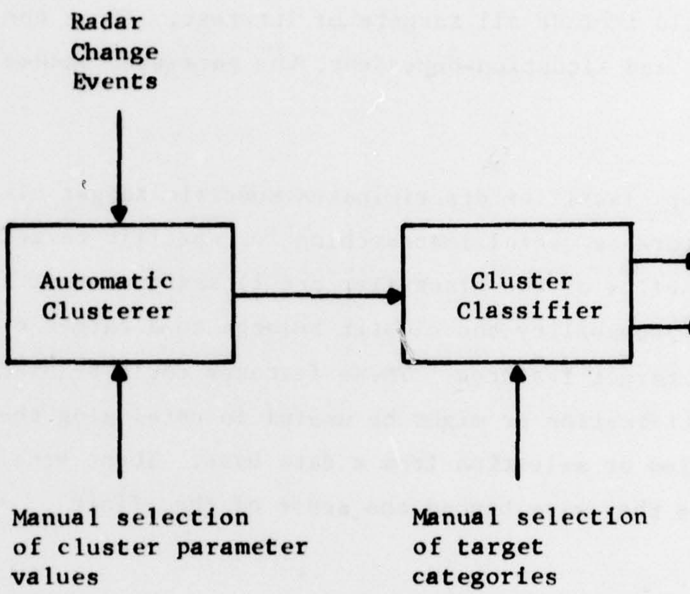


Figure 1.2 Basic Elements of Cluster Detector.

by Control Data for the Air Force.* Consequently the input data is representative of current change detection technology.

A series of three experiments was performed, each representing a further stage of development of the procedure. The first was for only one target class, gun emplacement arrays. The initial clusterer correctly grouped all eight of the gun arrays into separate clusters. The background or, false alarm, changes were grouped into 52 clusters. The classifier then correctly discriminated the eight gun arrays as target while calling one of the background clusters a target. These results are listed in Table 1-1.

1.4 CONCLUSIONS AND RECOMMENDATIONS

The principal result is that an effective procedure was developed to detect clusters of radar changes corresponding to military targets. It appears that this procedure would be a useful addition to radar change detection. Potential advantages are:

1. A reduction in the number of changes that have to be analyzed in subsequent processing. This can increase the search area and target completeness of an analyst.

2. Automatic computation of cluster features and probabilities of being a target. These could aid the analyst in target discrimination. Another use would be to use these features to prioritize or select the target candidates for subsequent analysis.

It is recommended that the following steps be taken.

1. The clustering procedure would be implemented in an interactive laboratory simulation. Then the procedures for manual selection of cluster and target parameters could be defined and evaluated.

---

* Air Force Contract F30603-73-C-0141

2. The clustering procedure (with manual parameter selection) would be configured and evaluated on an on-line implementation. Outputs from the MCD (or alternate) change detection would be used to provide a more extensive data set. These data would be further extended with synthetically generated cluster distributions.

3. An operational implementation would be defined, based on the above two steps.

The first two steps could be conducted on CDC's Cyber-Ikon facility. The third step would be based on the hardware configurations available. The Cyber-Ikon uses the same hardware (Flexible Processors) as the MCD. Thus it would be relatively easy to transfer the procedure from the laboratory to the MCD.

TABLE 1-1. AUTOMATIC CLASSIFICATION OF GUN EMPLACEMENT ARRAYS

GROUND TRUTH \ CLASSIFICATION RESUTS	TARGET	BACKGROUND
TARGET	8	0
BACKGROUND	1	51

In the second and third experiments, the targets were grouped into one class and backgrounds into the other class. No further class distinction was made since there were not enough examples to be statistically significant. Use of the initial clustering technique produced the results shown in Table 1-2. Subsequently a more powerful technique was developed resulting in the performance shown in Table 1-3. This more powerful technique showed excellent discrimination.

TABLE 1-2. AUTOMATIC CLASSIFICATION OF GUN EMPLACEMENT, HELICOPTER, AND BEACON ARRAYS.

GROUND TRUTH \ CLASSIFICATION RESULTS	TARGET	BACKGROUND
TARGET	17	0
BACKGROUND	5	93

TABLE 1-3. AUTOMATIC CLASSIFICATION OF HELICOPTER, HAWK, ARMOR, AND TANK ARRAYS

GROUND TRUTH \ CLASSIFICATION RESULTS	TARGET	BACKGROUND
TARGET	15	0
BACKGROUND	2	102

2.0 THEORETICAL DESCRIPTION OF TECHNIQUES

This section describes the automatic target pattern detection process. It involves four steps. First, a difference image is created using MCD change detection software. Second, change events are extracted from the difference image and clustered into arrays. Third, event ground truth and characteristic properties of the arrays are used to train the automatic classifier. Finally, the arrays are automatically classified and compared with ground truth to establish the probability of detection and the probability of false alarms. These procedures are described in the following paragraphs.

2.1 STEP 1: CONSTRUCTING THE DIFFERENCE IMAGE

The first step of the automatic classification process is the construction of a difference image. This is obtained from the SAR data base as shown in Figure 2-1. A dual coverage region is selected from the SAR data base, yielding a mission image and a reference image pair. The mission image is then registered and photonormalized to the reference image. Finally, the two images are subtracted to obtain the difference image.

SAR Data Base

The SAR data base for this study consists of imagery from Gallant Hand and Reforger '76. Regions of dual coverage (reference and mission) for change detection analysis containing gun emplacements (Gallant Hand) and a variety of groups of military vehicles (Reforger '76) are selected for processing.

Registration and Photonormalization

The mission and reference images are registerd, photonormalized, and subtracted to produce a difference image. The first step in this procedure is the manual selection of corresponding control points in the two images. These control points are used to evaluate the parameters in the warp equation, which mathematically relates the coordinates of an event in the two images.

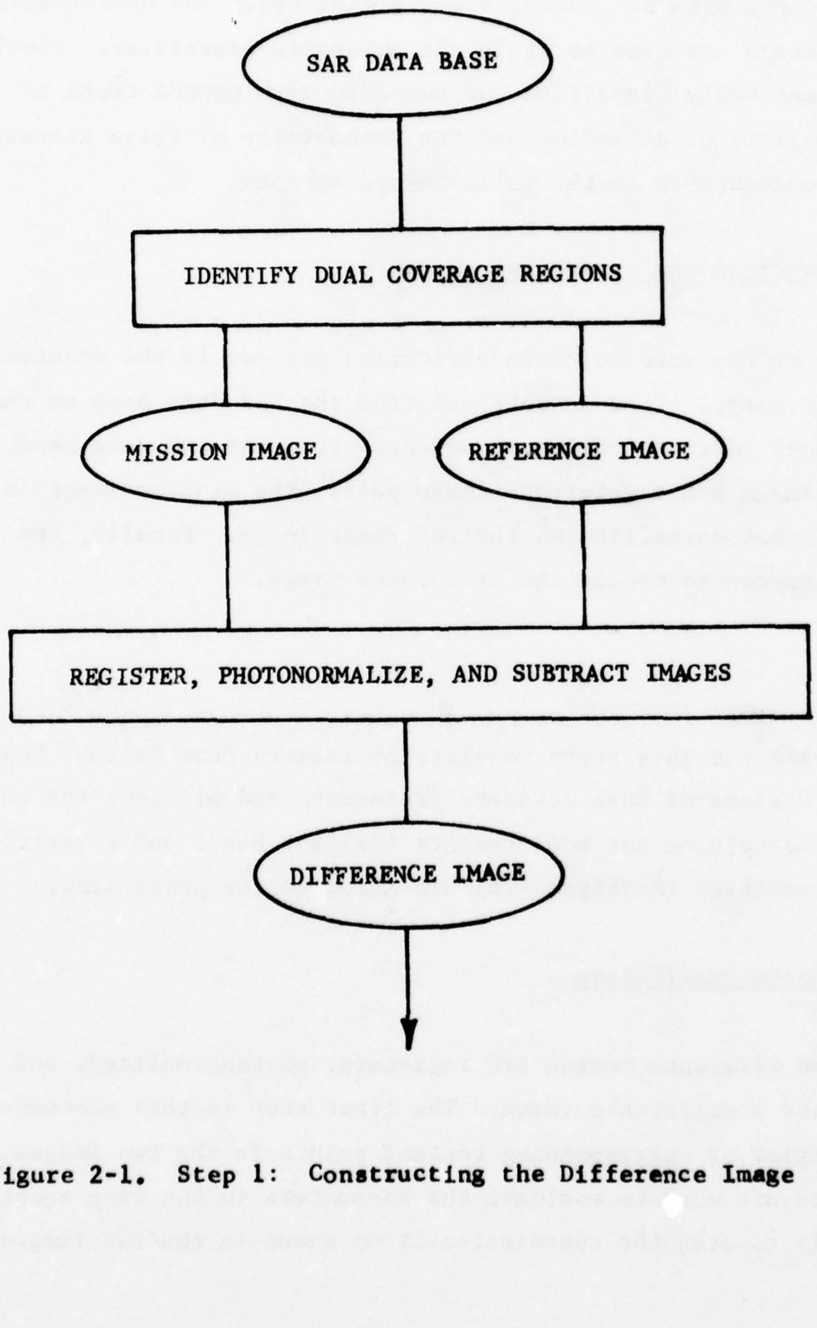


Figure 2-1. Step 1: Constructing the Difference Image

The output is a warped mission image which achieves point-by-point registration with the reference image.

After registration, the gray levels (intensity) of the warped mission image are photonormalized so that corresponding points on each image have approximately equal intensity. Then each pixel of the warped mission image is subtracted from the corresponding pixel in the reference image to produce a difference image. The difference image suppresses the background and provides change events.

2.2 STEP 2: CLUSTERING THE EVENTS INTO ARRAYS

The second stage of the automatic classification process is the clustering of change events into arrays, as illustrated in Figure 2-2. The change image pixels are thresholded in intensity so that only the strong return pixels which are in the mission image but not in the reference image remain. Next, contiguous strong return pixels in the thresholded image are connected together to form events. These events are thresholded according to size so that only the large events remain. These large events, representing large strong returns which are in the mission image but not in the reference image, are automatically clustered into arrays. Clustering is further described later in this section.

Adaptive Intensity and Size Threshold Selection

An intensity histogram of the change image is computed as illustrated in Figure 2-3. This histogram shows the number of pixels which have a particular intensity (gray scale). It has been found that the histogram is qualitatively Gaussian in shape except for low intensities. At these low intensities, a departure from a Gaussian shape occurs. The intensity at the point of departure from Gaussian shape was chosen as the intensity threshold. It was found that this selection procedure using the intensity threshold for extracting mission strong return change events, together with an event size threshold for two pixels, successfully extracted target events and suppressed noise events for the series of SAR regions processed.

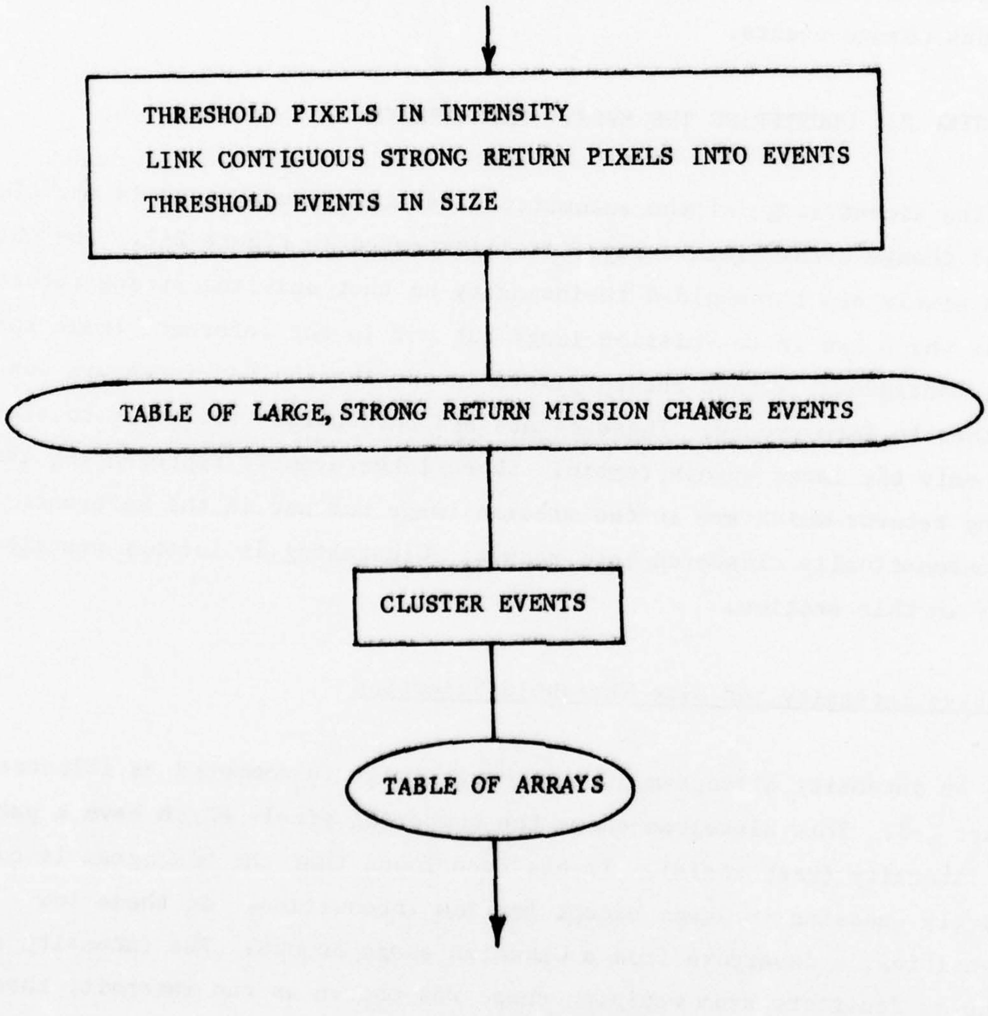


Figure 2-2. Step 2: Clustering the Events into Arrays

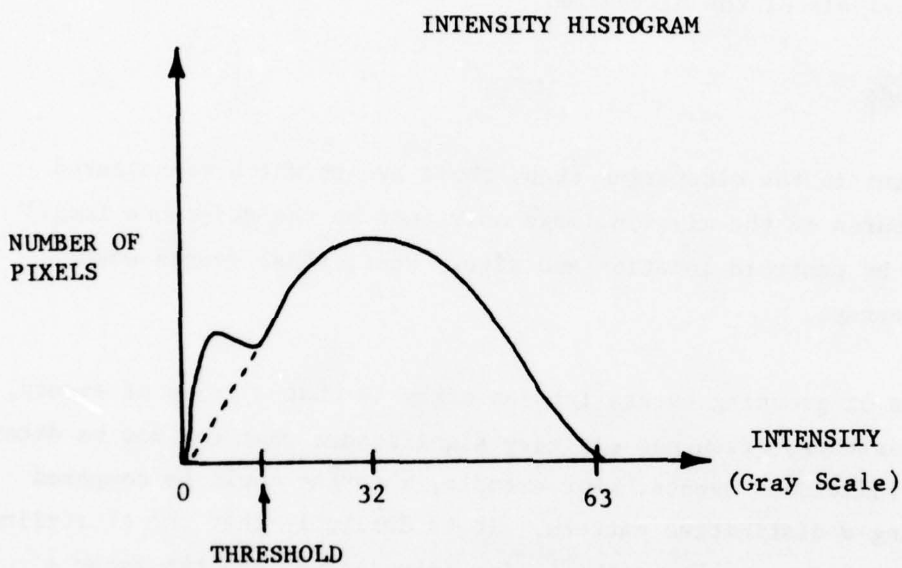


Figure 2-3. Mission Strong Return Event Intensity Threshold Selection on Change Image

The intensity histograms, and therefore the intensity thresholds, differed from one image to another image. Thus, the threshold selection can be considered to be adaptive. This method was successfully used in the SAR images processed during the later period of the contract.

The intensity at departure from Gaussian shape was a subjective judgement. However, this departure intensity could be automatically determined by a computer analysis of the histogram.

Clustering Methods

At this point in the clustering step, those events which represented large strong returns on the mission image only (not on the reference image) have been filed by centroid location and size. Next, these events were clustered into arrays.

The purpose of grouping events into an array is that a group of events, located close together, often has military significance that can not be determined from the individual events. For example, a target could be composed of events forming a distinctive pattern. It is desirable that the clustering algorithm cluster target events without also introducing into the array a large number of background events.

Two different clustering programs (Iterative Clustering and Minimal Spanning Tree Clustering) were investigated and are described below.

Iterative Clustering Method

This special-purpose two-dimensional clustering procedure was developed for this contract. It is discussed here in two parts: the analytically well-defined portion, which makes the assignment of events to clusters; and the portion which reviews the result, applies subjective completion criteria, and modifies the input parameters to drive the process towards a completion state. Quantitative details are summarized in Appendix A.

Cluster Assignment

The first part of the Iterative Clustering Method concerns cluster assignment. A linked list is provided to keep track of the cluster to which each event is assigned and to locate all events in a given cluster. The assignment process begins by assigning event number one to cluster number one. The next event is tested to determine whether it should be assigned to the same cluster or begin a new one. In every case the cluster assignment is determined by evaluating the conditional probability density function, (i.e. the likelihoods of each existing cluster at the location of the event to be assigned). The event is assigned to the maximum likelihood cluster unless it is determined that it lies out on the tail of that cluster's distribution. In these cases a new cluster is created.

Events on the tail of a cluster's distribution are identified by integrating the distribution from its mean value out to the event location. Since the distributions are normalized, the closer the result of the integration approaches one, the further out on the tail the event lies. The decision to begin a new cluster is made whenever the integral exceeds a threshold value. The threshold value is adjusted as required after each assignment, iteration (assignment of each event to some cluster) is completed.

This threshold value is one of two input parameters which controls the state of the clusters. The other is the size of the parameter which determines the initial clustering assignment and limits the maximum cluster size.

A probability density function is associated with each cluster. When a cluster is created it contains one event. Until two more events are added, the probability density function is assumed to be a circularly symmetric exponential. The initial rate of exponential decay with distance from the cluster center for clusters with less than three events is the second input parameter. With every addition of an event to a cluster, the exponential becomes broader because the kurtosis of the distribution function selected increases with the number of events assigned to the cluster.

A departure from circular symmetry is introduced by computing the two-dimensional covariance of the cluster when three or more events are assigned to the same cluster. The inverse of the covariance is used in place of the input parameter causing circular symmetry. It creates a preferred direction, permitting events lying in that direction to be more easily drawn into the cluster than equally distant events which are in a perpendicular to the preferred direction.

At the transition from use of the input parameter to use of the covariance, the breadth of the function is prevented from changing rapidly by requiring an equality in the value of comparable probabilities for the 2 and 3 event cluster distributions.

Cluster Assessment

The second part of the iterative clustering method concerns cluster assessment. The clustering process begins by creating a few very large clusters. The final solution obtained by any clustering process cannot be unique unless external requirements are imposed on the process. The process described here gives results which, if not unique, are consistently reasonable and which are judged, subject to experimental confirmation, to be nearly independent of event order.

The external requirement imposed to drive the cluster assignment process through another iteration is maximum permissible cluster size. A judgement is required about the scale size of a cluster that would be caused by the largest significant military activity of interest.

The clusters obtained after every pass are examined to identify a permissible solution. If the solution is favorably evaluated, the number of clusters obtained is fixed and the solution is submitted to a stability test. This is conducted by allowing events to move to different clusters but not allowing the number of clusters to change. The intent is to find a state of the solution which is relaxed; that is, each event lies in its maximum likelihood cluster. If the solution is not satisfactory, process parameters are adjusted to encourage the creation of new clusters until the largest cluster satisfies the size criterion imposed.

Minimal Spanning Tree (MST) Clustering Method

This method for automatically clustering events was written as a computer program by R. L. Page* and is based on an algorithm by C. T. Zahn**. The method involves the construction of the minimal spanning tree graph of the set of events.

Advantages of the method are that it requires little input other than the event locations, it is relatively insensitive to permutations in the order that event locations are inputted, and the clusters (arrays of events) it produces are similar to clusters detected visually by humans when the events are displayed as an image. The following discussion will define, exemplify, and characterize the minimal spanning tree clustering method.

An <u>edge-weighted linear graph</u> is composed of a set of points called <u>nodes</u> (or events) and a set of node pairs called <u>edges</u> with a <u>weight</u> (the distance between the pair of events) assigned to each edge. Figure 2-4 a depicts a weighted graph with six nodes and nine edges, each of which has a weight equal to its length. A <u>path</u> in a graph is a sequence of edges joining two nodes, as (ABCFE) or (BADF). A <u>circuit</u> is a closed path as (ABCA) or (ACFEDA). A connected graph has paths between any pair of nodes. A <u>tree</u> is a connected graph with no circuits, and a <u>spanning tree</u> of connected graph G is a tree in G which contains all nodes of G. Figures 2-4 b

* R. L. Page, "A Minimal Spanning Tree Clustering Method", <u>Comm. ACM, 17</u> (1974), 321-323.

**C. T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans. on Computers, C-20 (1971), 68-86.

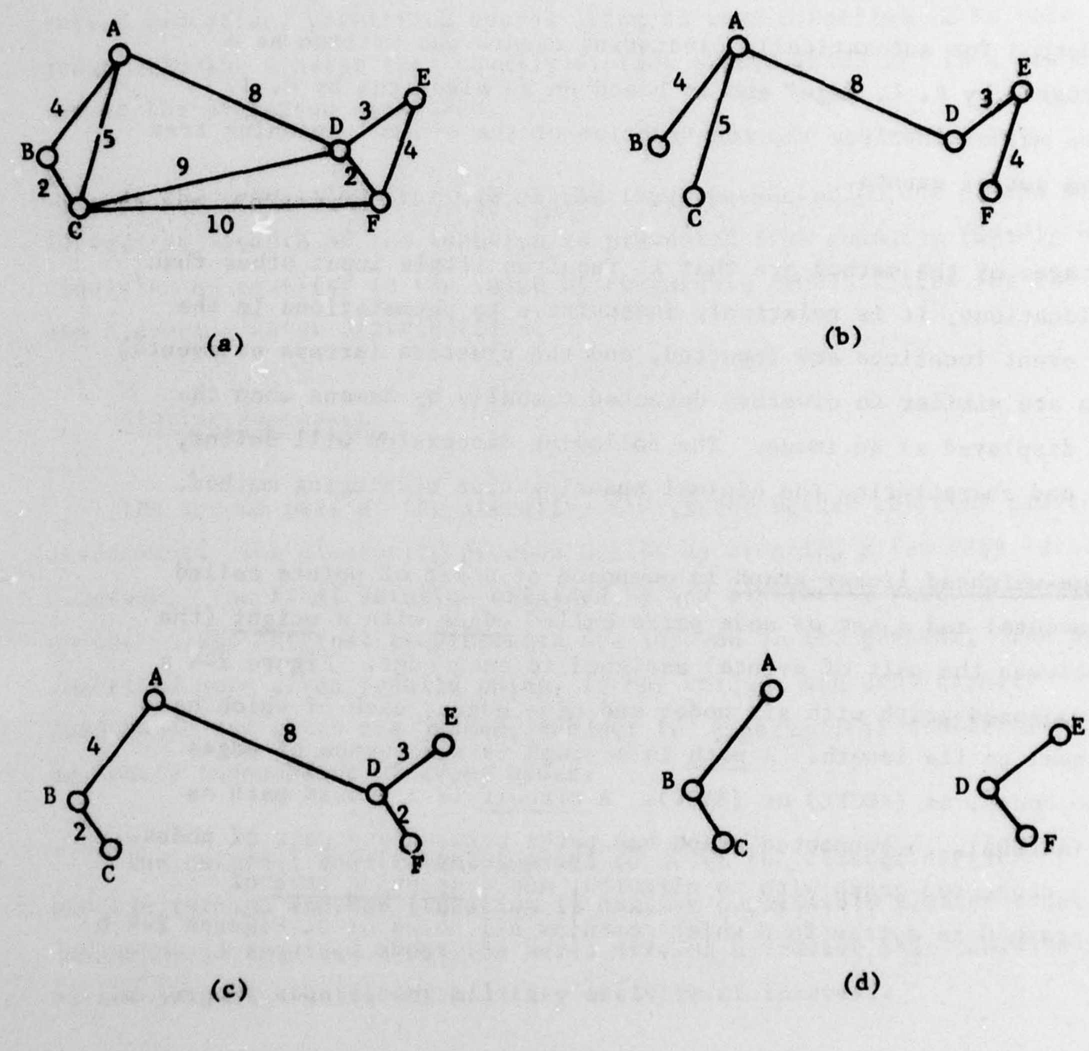


Figure 2-4. Cluster Identification with Minimal Spanning Tree
(a) Weighted Linear Graph. (b) Spanning Tree.
(c) Minimal Spanning Tree. (d) Two Subtrees
(Clusters) Formed After Deleting "Significantly"
Longer Edge AD from Minimal Spanning Tree.

and c represent spanning trees of the graph in Figure 2-4 a. If the weight of a tree is defined to be the sum of the weights of its constituent edges, then a minimal spanning tree (MST) of graph G is a spanning tree whose weight is minimum among all spanning trees of G. Figure 2-4 c is the MST for Figure 2-4 a.

The basic idea of the MST clustering method is to detect inherent separations in the event locations by deleting edges from the minimal spanning tree which are significantly longer than nearby edges. Such an edge is called "inconsistent." Zahn suggests the following criterion: an edge is inconsistent if (1) its length is more than f times the average of the length of nearby edges, and (2) its length is more than s standard deviations larger than the average of the lengths of nearby edges (the numbers f and s may be adjusted by the user). The question of determining which edges are "nearby" is also answered by the user. The event P is said to be nearby event Q if event P is connected to event Q by a path in the minimal spanning tree containing d or fewer edges (d is an integer specified by the user). Deleting the inconsistent edges breaks up the tree into several connected subtrees. The events in each connected subtree are the members of an array. Figure 2-4 d shows the two subtrees (clusters or arrays) formed after the significantly longer edge AD was deleted from the minimal spanning tree shown in Figure 2-4 c.

Thus the minimal spanning tree clustering program inputs the large, strong return change event centroid locations as nodes, forms the minimal spanning tree of the nodes, and deletes the significantly longer edges of the minimal spanning tree, thus forming groups of subtrees of nodes. The change events from either the reference or mission image are grouped into clusters in the same way. In general, any one cluster would be formed of changes detected on one of the images only.

These subtrees are the event arrays: all events in the same subtree are assigned to the same array.

All clustering done with the minimal spanning tree method used the following values of the parameters: d=2, f=1.3, and s=0.

2.3 STEP 3: TRAINING THE AUTOMATIC CLASSIFIER

The third stage of the automatic classification process deals with the training of the automatic classifier; as illustrated in Figure 2-5. At this point in the process, the events have been clustered into arrays. Each array is now labeled as a target or background on the basis of event ground truth information. Next, twenty features are evaluated for each of the arrays. Then, from the target arrays, twenty target feature means and the twenty-by-twenty target feature covariance matrix are calculated. Also, from the background arrays, the twenty background feature means and the twenty-by-twenty background feature covariance matrix are calculated. Then a cannonical transformation in the twenty-feature space is performed which determines the three best generalized features (linear combinations of the original twenty features) for distinguishing target arrays from background arrays. Finally, class-conditional probability density functions for target arrays and for background arrays are parameterized in this three-dimensional generalized feature space.

A more detailed description of the automatic classifier training process follows.

Application of Event Ground Truth

At this point in the processing the events have been clustered into arrays. Next, each array is labeled as a target array or a background array, using the ground truth information for each event. A particular array is labeled (a) a "target array" if the percentage of target events in the array is greater than a specified threshold; (b) a "background array" if the percentage of background events is greater than another threshold; and (c) "neither", if neither of the above conditions are met. In all of the processing, the thresholds in (a) and (b) were both set at 52%. Thus, if an array was composed of more than 52% target events, the array was labeled a target array. If an array was composed of more than 52% background events, the array was labeled a background array. However, if the array was composed of,

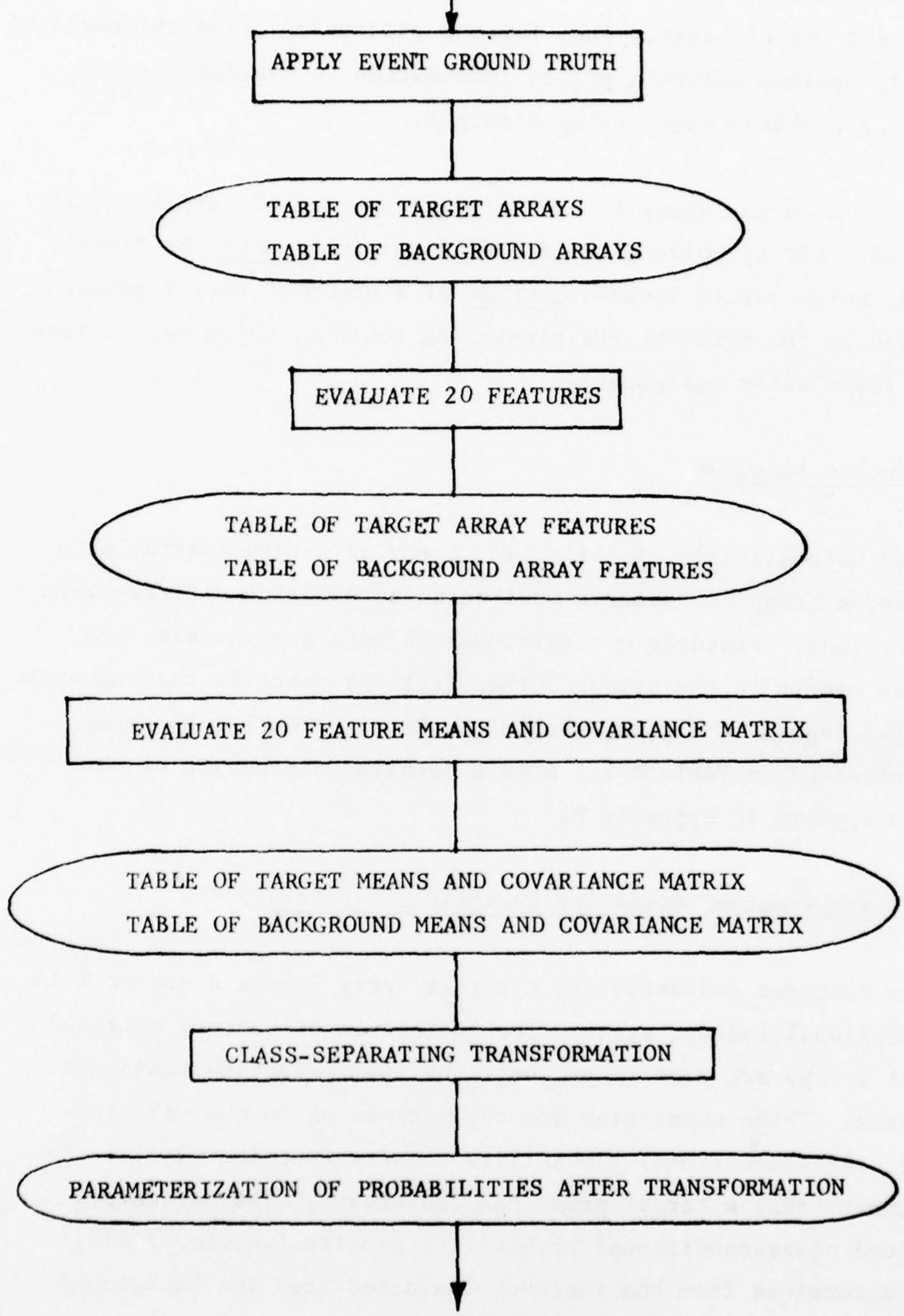


Figure 2-5. Training the Automatic Classifier

for example, 50% target events and 50% background events, the array was a "neither" array and was eliminated from further processing. The threshold of 50-50 is probably optimum unless a priori information is available as to the probability of a change event being a target.

The fact that a target array is not totally composed of target events is to be expected. First, noise (background) does occur among the target events. Second, noise events located outside of a group of target events might be included in the array by the clustering routine, which uses inter-event spacings for cluster assignments.

Evaluation of Twenty Features

In order to automatically classify each array as a target array or a background array, a group of features must be selected which differentiate between the two types. Features of individual objects are the size and intensity of the events in the array. Other features describe the geometric properties of the events as a group within the array. The twenty array features are presented in Table 2-1. A more detailed discussion of the twenty features appears in Appendix B.

Evaluation of Twenty-Feature Means and Covariance Matrices

The twenty features evaluated for a target array locate a vector $\vec{X}$ in the twenty-dimensional feature space. The collection of vectors obtained from the target arrays are used to evaluate the twenty-feature means and covariance matrix. These quantities are the parameters in the Gaussian-modeled target class-conditional probability density function $f(\vec{X}|T)$, the probability density that a target array has features $\vec{X}$. The parameters of the background class-conditional probability density function $f(X|B)$ are similarly determined from the features evaluated from the background arrays.

TABLE 2-1. FEATURE DEFINITION

FEATURE	DESCRIPTION
1	Number of events in the array.
2	Spacing of events.
3	Regularity of spacing.
4	Event size.
5	Uniformity of event size.
6	Event shape.
7	Uniformity of event shape.
8	Uniformity of event orientation.
9	Array size.
10	Array shape.
11	Orientation of events to the array.
12	Event area.
13	Array mass size.
15	Array mass shape.
16	Orientation of array mass to array.
17	Distance from mass to geometric centroids.
18	Array pixel density.
19	Event pixel density.
20	Uniformity of event pixel density.

The means and covariance matrices of the targets and background are used to determine the best class-separating transformation and to model the probability density functions after the transformation to generalized features.

Class-Separating Transformation

The number of feature variables is reduced from twenty ($\vec{X}$) to three ($\vec{X}'$) by determining the three independent linear combinations of the original twenty-feature variables which are best at separating (in the feature vector space) the target points from the background points. The three new variables are called "generalized features."

The linear transformation W relating the twenty features to the three generalized features can be written

$$\vec{X}' = W\vec{X} \tag{2.1}$$

where $\vec{X}'$ is (3x1), W is (3x20), and $\vec{X}$ is (20x1).

Equation 2.1 is used to evaluate the target-generalized feature means $\underline{\vec{X}'}$ in terms of the twenty-feature means $\underline{\vec{X}}$. The target twenty-feature covariance matrix C is related to the three generalized feature covariance matrix C' by

$$C' = WCW^T \tag{2.2}$$

where the T superscript denotes "transpose", C' is (3x3), W is (3x 20), C is (20x20), and W^T is (20x3).

With the same W, Equations 2.1 and 2.2 are used to relate the background twenty-feature means to the background three-generalized-feature means, and the background twenty-feature covariance matrix to the three-generalized-feature covariance matrix, respectively.

A more detailed discussion of the class-separating transformation is given in Appendix C.

Parameterization of Probabilities After Transformation

The class-conditional probability density functions for both target arrays and background arrays are parameterized in the three best class-separating generalized feature space. The parameters for targets are the three-component generalized feature vector means $\vec{\underline{X}}'$ and the three-by-three generalized feature covariance matrix C' for target arrays. The same is true for background arrays. Thus, the probability densities for target arrays, $\hat{f}(\vec{X}'|T)$, and for background arrays, $\hat{f}(\vec{X}'|B)$, for an array with generalized feature vector $\vec{X}'$ can be evaluated. These densities, together with a priori information about the relative abundance of target and background arrays, can be used to compute the probability that an array of unknown classification with generalized features $\vec{X}'$ is a target array, $P(T|\vec{X}')$, or a background array, $P(B|\vec{X}')$.

A more detailed discussion of the parameterization of the probability density functions in the generalized feature space is given in Appendix D.

This completes the discussion of the training of the automatic classifier illustrated in Figure 2-5.

2.4 STEP 4: AUTOMATIC CLASSIFICATION

The fourth and final stage of the automatic classification process deals with the actual automatic classification of arrays, as illustrated in Figure 2-6. At this point in the process, the automatic classifier has been trained. For each array, the probabilities of its being a target array and a background array are evaluated as a function of the array's generalized feature vector $\vec{X}'$. The array is automatically classified as that class which has the greater probability. This automatic classification of the array is then compared with the previously determined classification based on event ground truth. From the collection of arrays, the percentages of correct and incorrect classifications are presented as a confusion matrix.

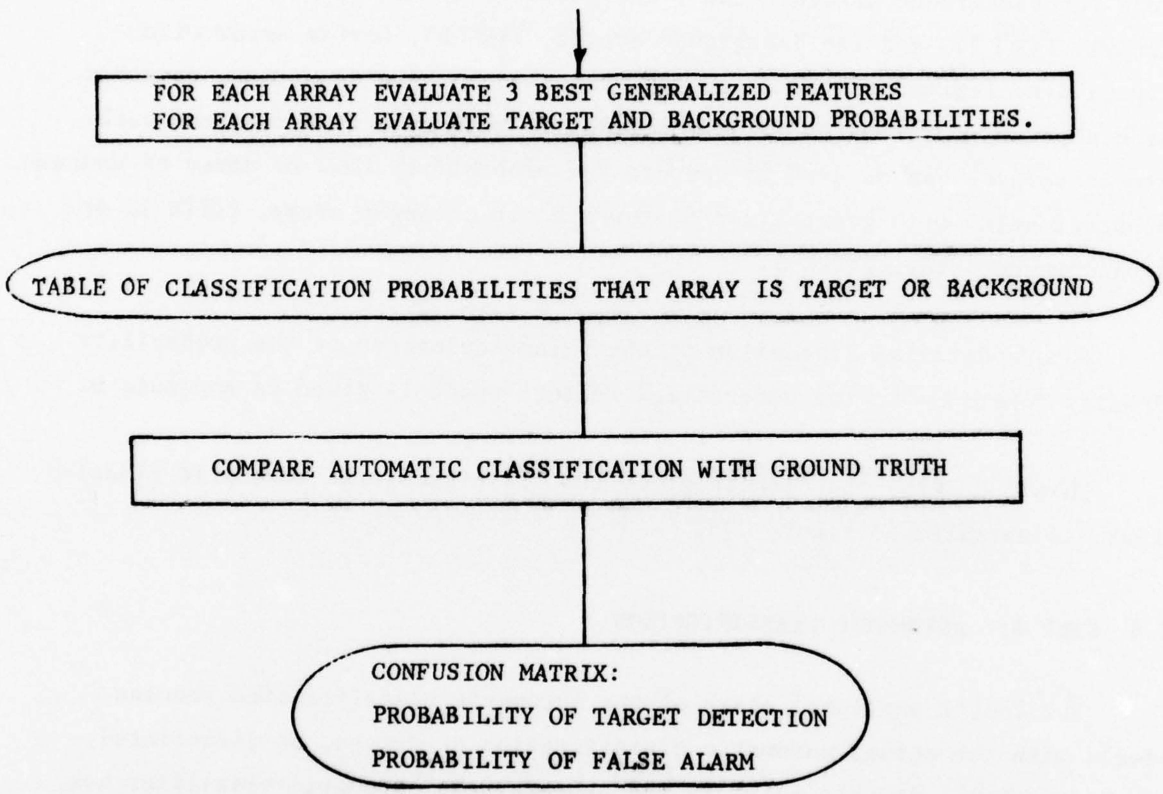


Figure 2-6. Step 4: Automatic Classification

This yields the probability of detection and the probability of false alarm. The following discussion describes the automatic classification in greater detail.

Evaluate Array Target and Background Probabilities

The arrays to be automatically classified are input one-at-a-time to the trained classifier. For each array, the twenty features are evaluated, yielding a twenty-vector $\vec{X}$ for that array. Using Equation 2.1, a generalized feature three-component vector $\vec{X}'$ is calculated. Finally, the class-conditional probability densities for targets, $\hat{f}(\vec{X}'|T)$, and for background, $\hat{f}(\vec{X}'|B)$ are evaluated for the array.

These densities are used to calculate the Bayes a posteriori (i.e., <u>after</u> the measurement of $\vec{X}'$) probabilities $P(T|\vec{X}')$, the probability that an array with generalized feature vector $\vec{X}'$ is a target array, and $P(B|\vec{X}')$, the probability that the array is a background array. Specifically,

$$P(T|\vec{X}') = \frac{q(T)\ \hat{f}(\vec{X}'|T)}{q(T)\ \hat{f}(\vec{X}'|T) + q(B)\ \hat{f}(\vec{X}'|B)} \tag{2.3}$$

$$P(B|\vec{X}') = \frac{q(B)\ \hat{f}(\vec{X}'|B)}{q(T)\ \hat{f}(\vec{X}'|T) + q(B)\ \hat{f}(\vec{X}'|B)} \tag{2.4}$$

where $q(T)$ and $q(B)$ are the a priori (i.e., before the measurement $\vec{X}'$) probabilities that the array is a target array and a background, respectively. If $P(T|\vec{X}') > .50$, the array is automatically classified as a target array. If $P(B|X') > .50$, it is automatically classified as a background array. Equations (2.3) and (2.4) have the property that

$$P(T|\vec{X}') + P(B|\vec{X}') = 1 \tag{2.5}$$

Automatic Classification vs Ground Truth

A comparison is made for each array between its classification from ground truth and its classification from the automatic classifier. A confusion matrix is constructed which lists the number of ground truth target arrays which are automatically classified as target arrays and background arrays. This is also done for the ground truth background arrays. From this matrix the probability of target detection and the probability of a false alarm can be calculated.

3.0 RESULTS OF AUTOMATIC CLASSIFICATION

SAR imagery taken from Gallant Hand and Reforger data was first processed to obtain a set of change images, each containing military targets. The processing was done with CDC computer programs which simulate the Modular Change Detector (MCD) processing required to create a difference image. Figures 3-1 through 3-9 show the results of this processing. In each case, except Figure 3-1, the reference, mission, warped mission, and difference images are presented. The target changes are shown as white in Figure 3-1 and black on the other images.

The next stage of the processing is performed on each difference image separately. First, a pixel intensity threshold is imposed on the difference image to create mission change events. Then an event size threshold is imposed so that only "large" events remain. The intensity and size thresholds are determined either by the user (supervised thresholding) or automatically (adaptive thresholding). At this point the change data correspond to the MCD output.

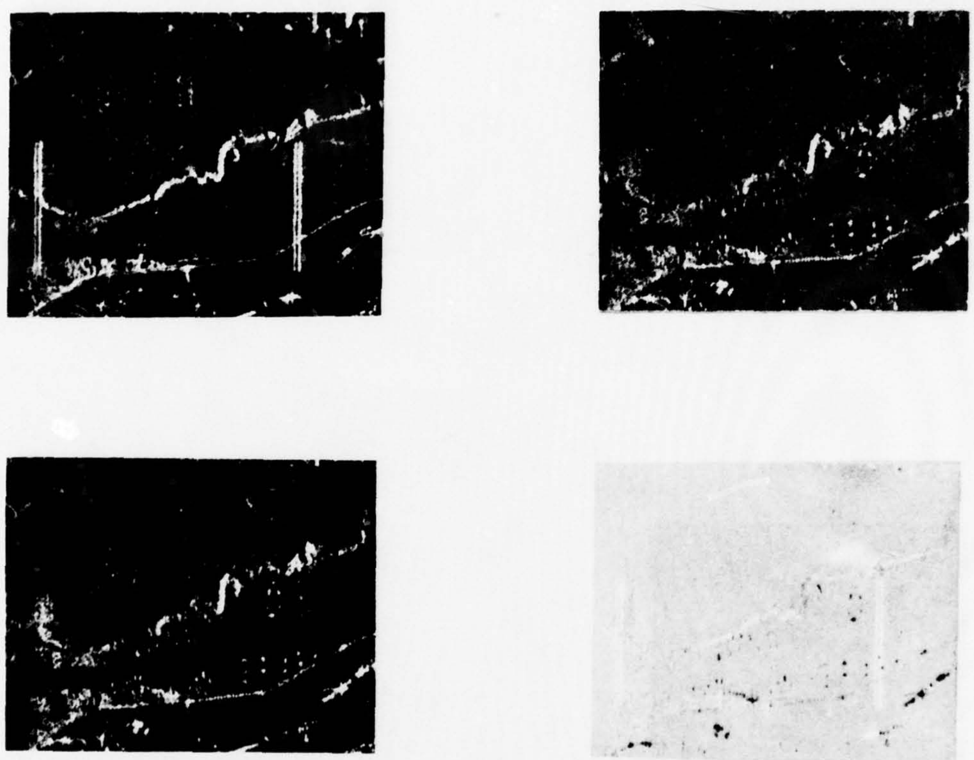
The resulting change event images are presented in Figures 3-9 through 3-17 (events are white). The caption of each image includes the thresholding procedure and values used. The target clusters are indicated by white hand-drawn boundaries. The other events are those changes which have passed the change criteria at this point in the processing. These can be either valid but non-interesting changes, or false changes induced by signal noise, scintillation, etc. The ground truth is not adequate to discriminate these background categories.

Next, the events in each image are clustered into arrays using either Iterative Clustering or Minimal Spanning Tree Clustering, which are described in Section 2.0. The captions of Figures 3-9 through 3-17 indicate the clustering method used.

The next stage of the processing (Figure 2-5) involves the use of event ground truth to establish array ground truth. Those arrays composed of more than 52% target events (labeled from ground truth) are labeled target arrays

Figure 3-1. Gallant Hand (Gun Emplacements) Difference Image.

REFERENCE MISSION



WARPED MISSION DIFFERENCE

Figure 3-2. Reforger Region A (Helicopters).

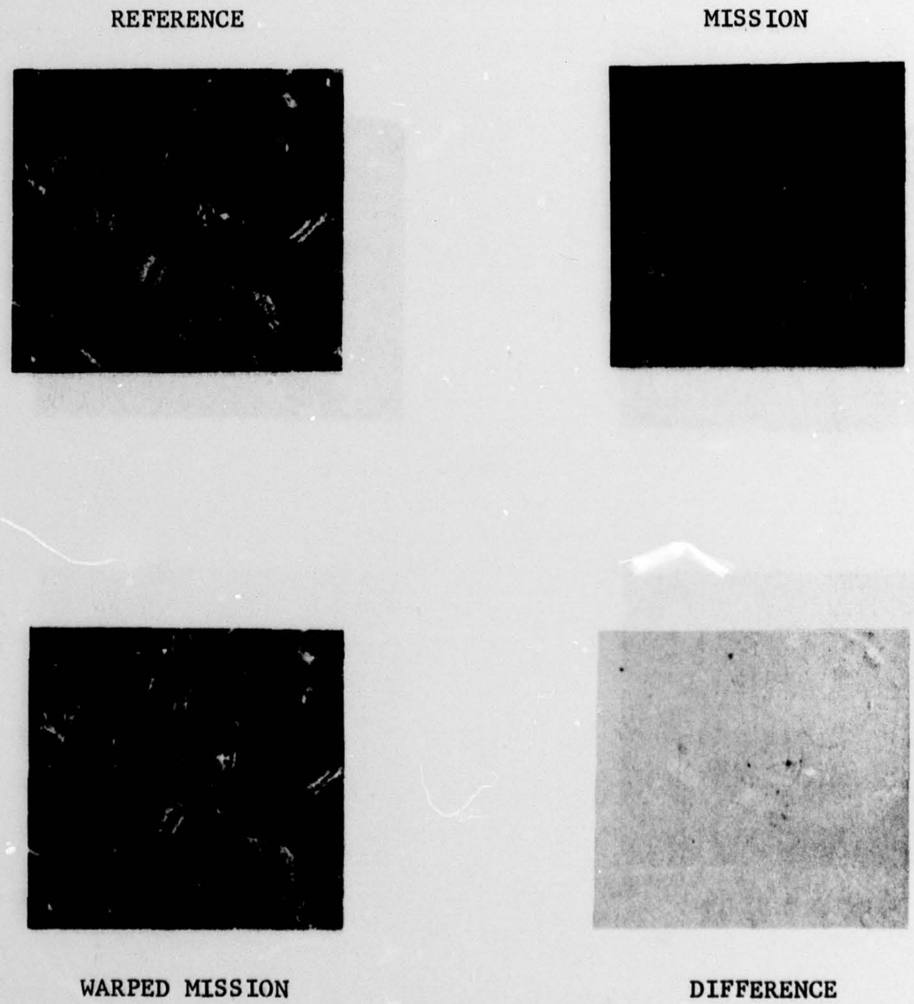


Figure 3-3. Reforger Region B (Hawk Site).

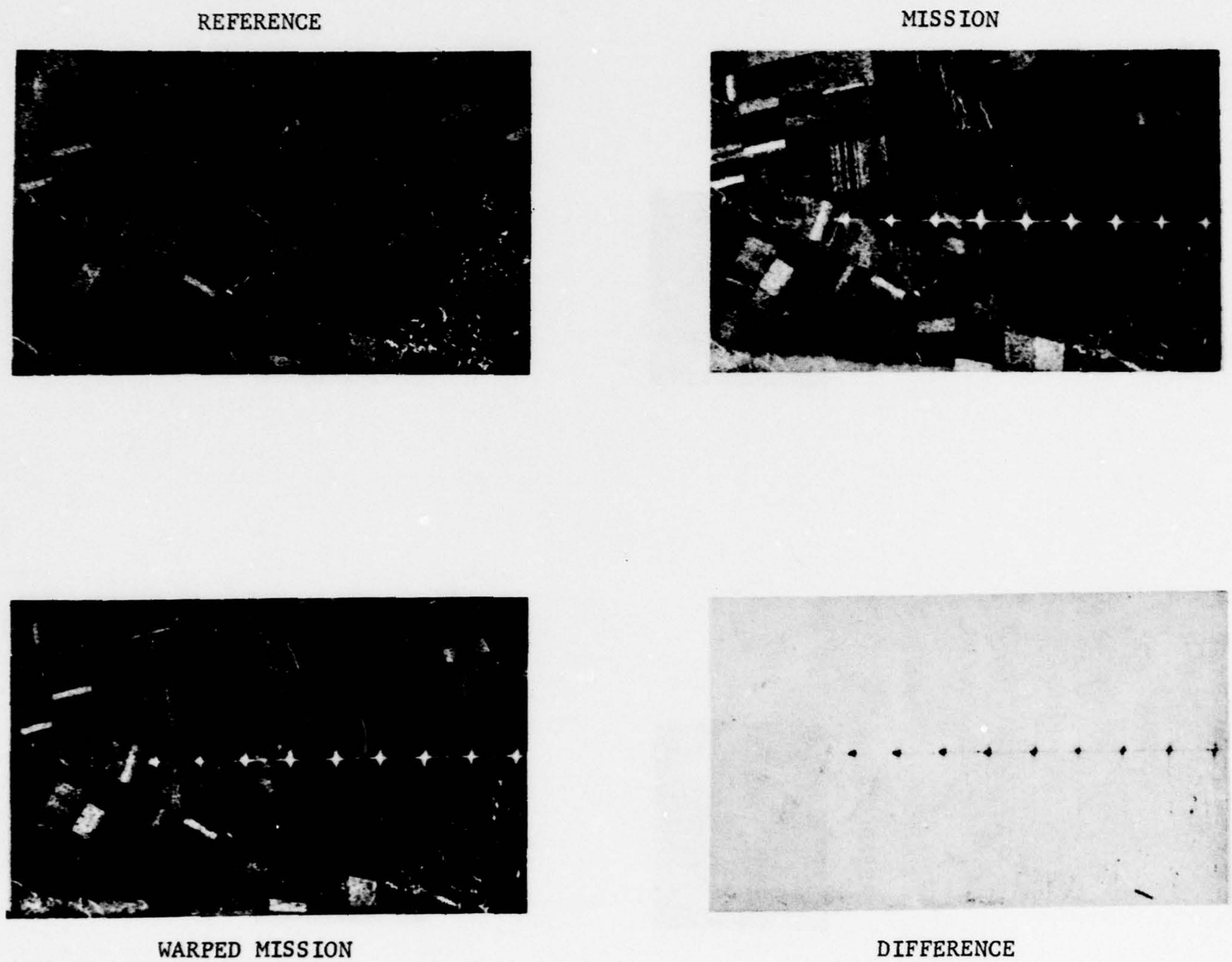


Figure 3-4. Reforger Region C (Beacon).

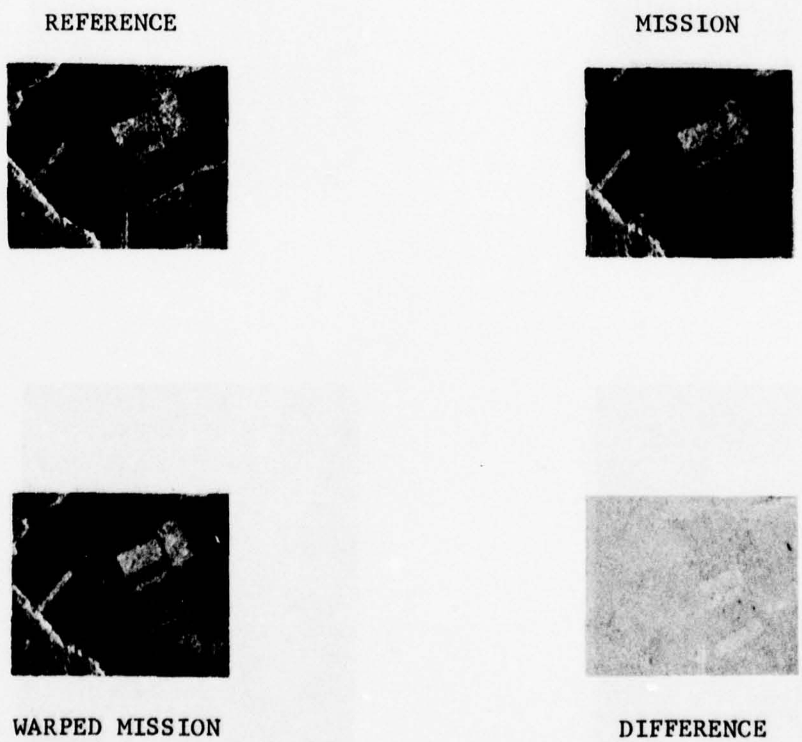


Figure 3-5. Reforger Region D (Hawk Site).

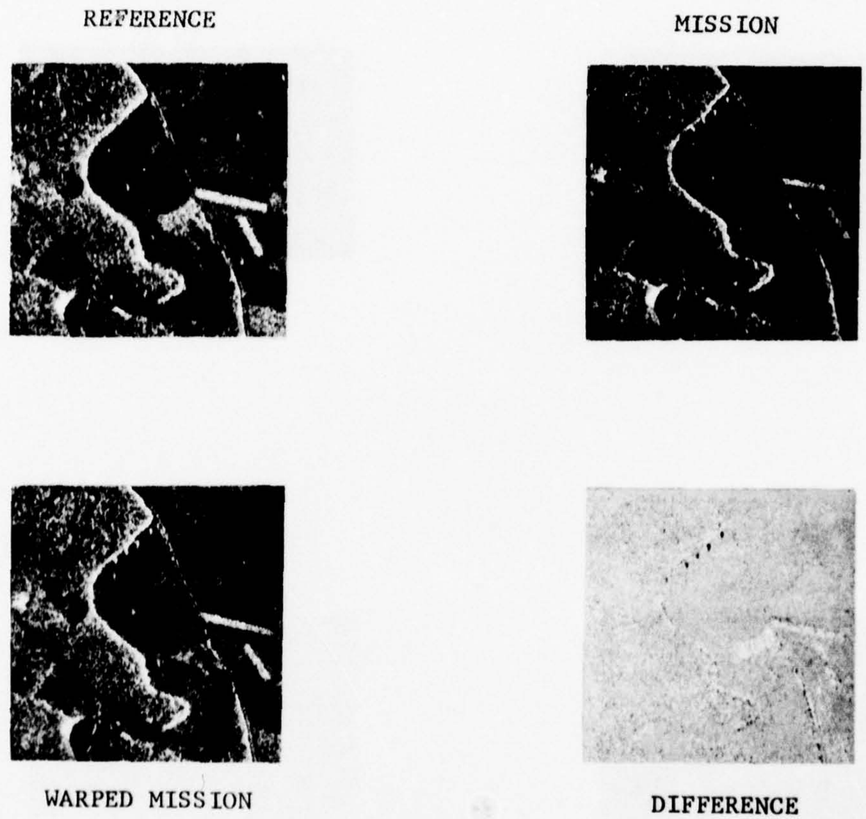


Figure 3-6. Reforger Region F (Helicopters).

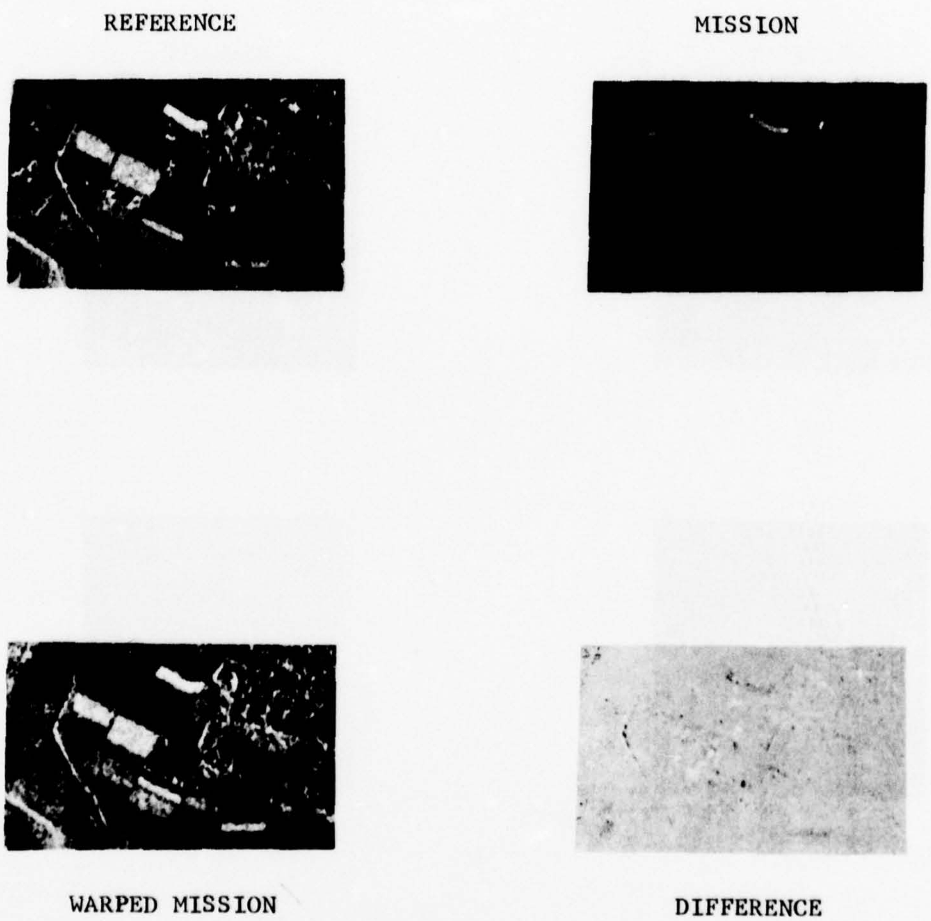
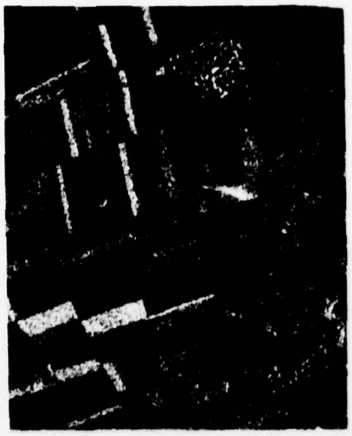


Figure 3-7. Reforger Region G (Armor).

REFERENCE



MISSION

WARPED MISSION

DIFFERENCE

Figure 3-8. Reforger Region N (Tanks).

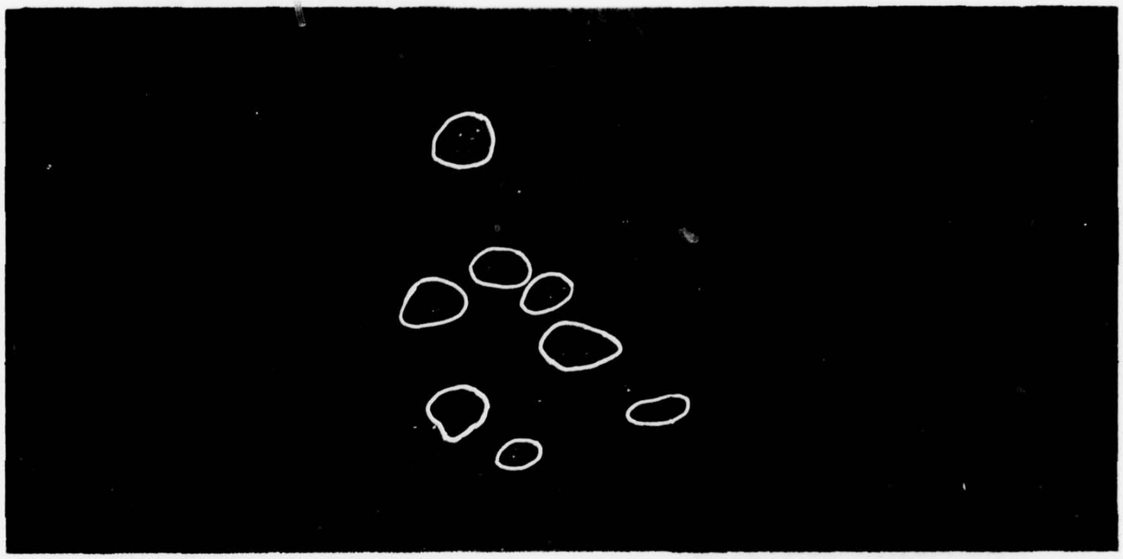


Figure 3-9. Gun Emplacement Event Image with Target Arrays Circled.
Thresholds: Supervised Size (13) and Intensity (24).
Clustering: Iterative.

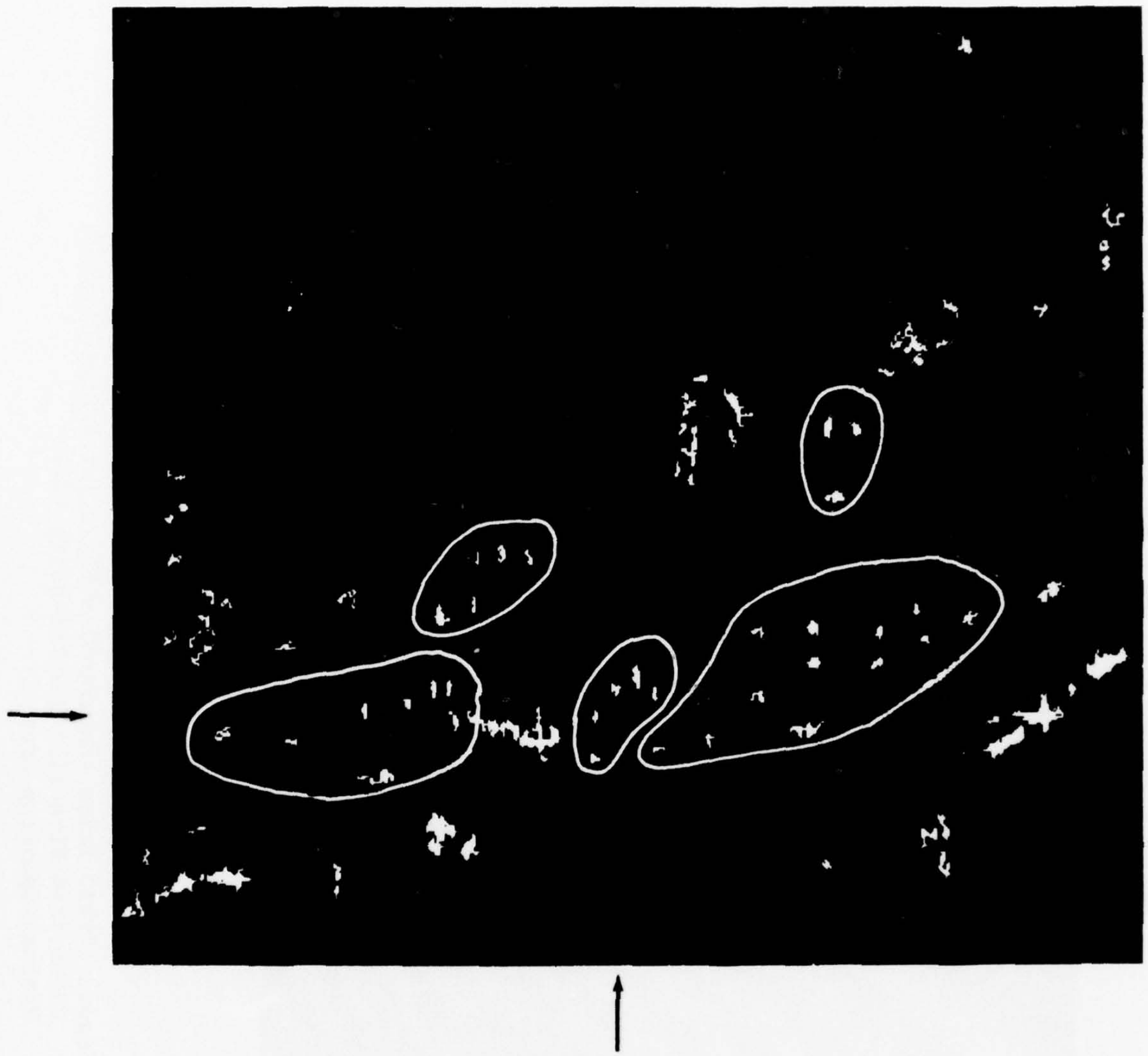


Figure 3-10. Region A (Helicopters) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Supervised Size (13) and Intensity (24)
Clustering: Minimal Spanning Tree

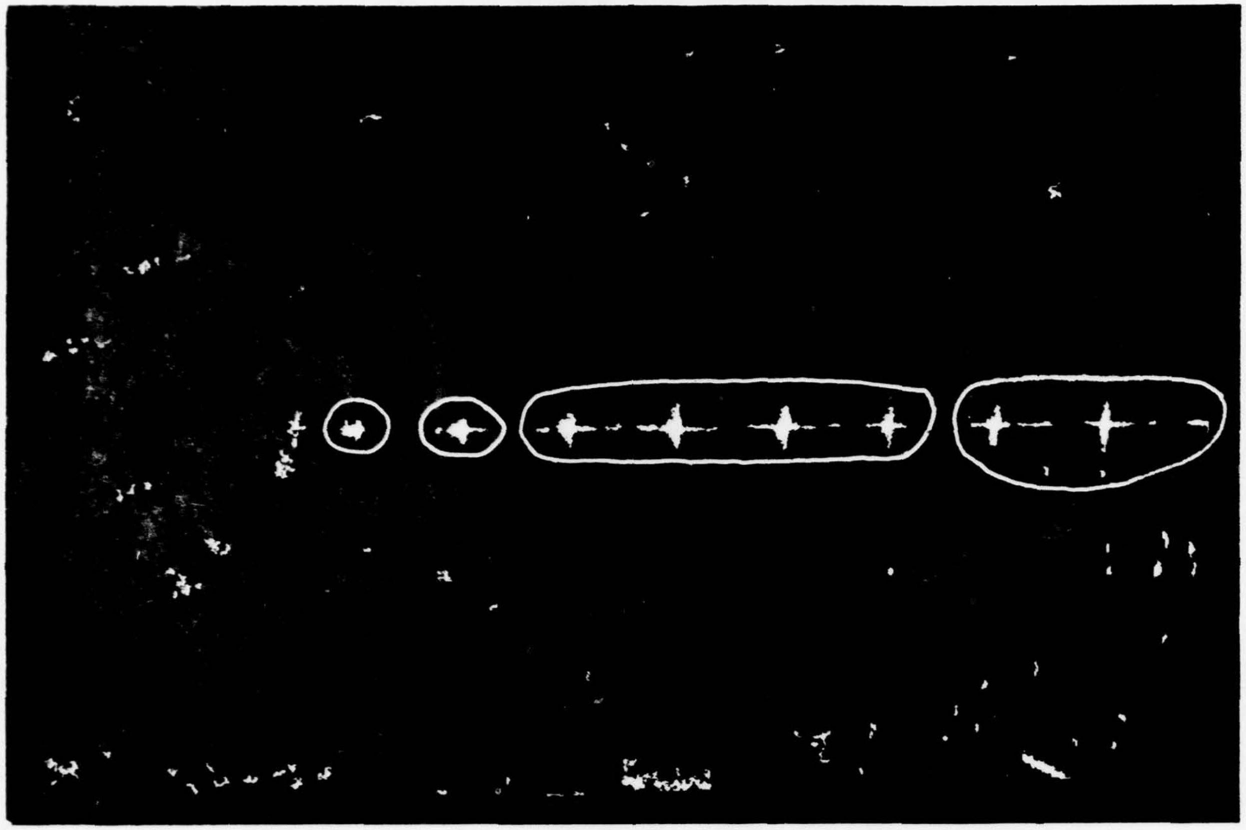


Figure 3-11. Region C (Beacon) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Supervised Size (13) and Intensity (24)
Clustering: Minimal Spanning Tree

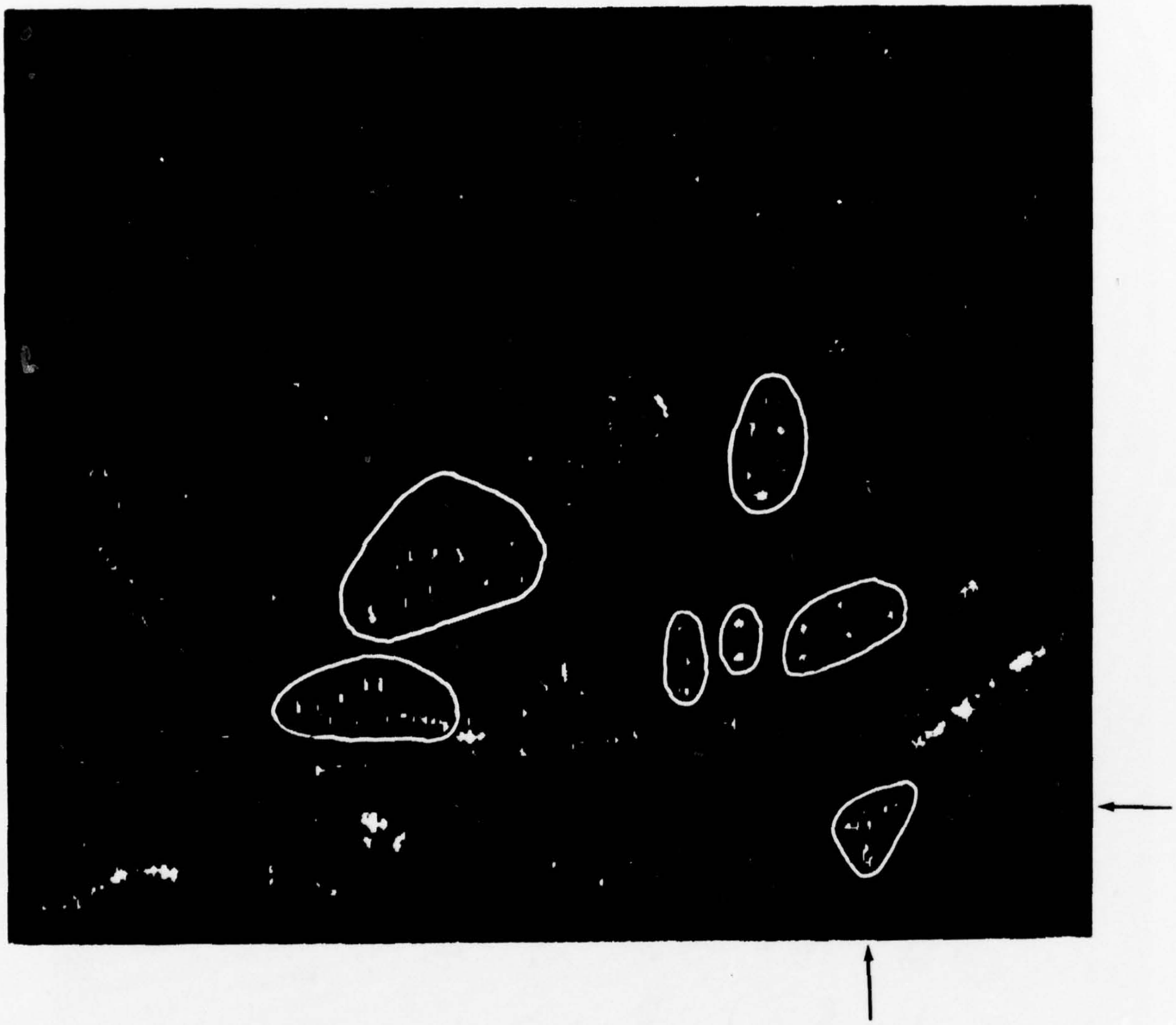


Figure 3-12. Region A (Helicopters) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Adaptive Size (2) and Intensity (17)
Clustering: Minimal Spanning Tree

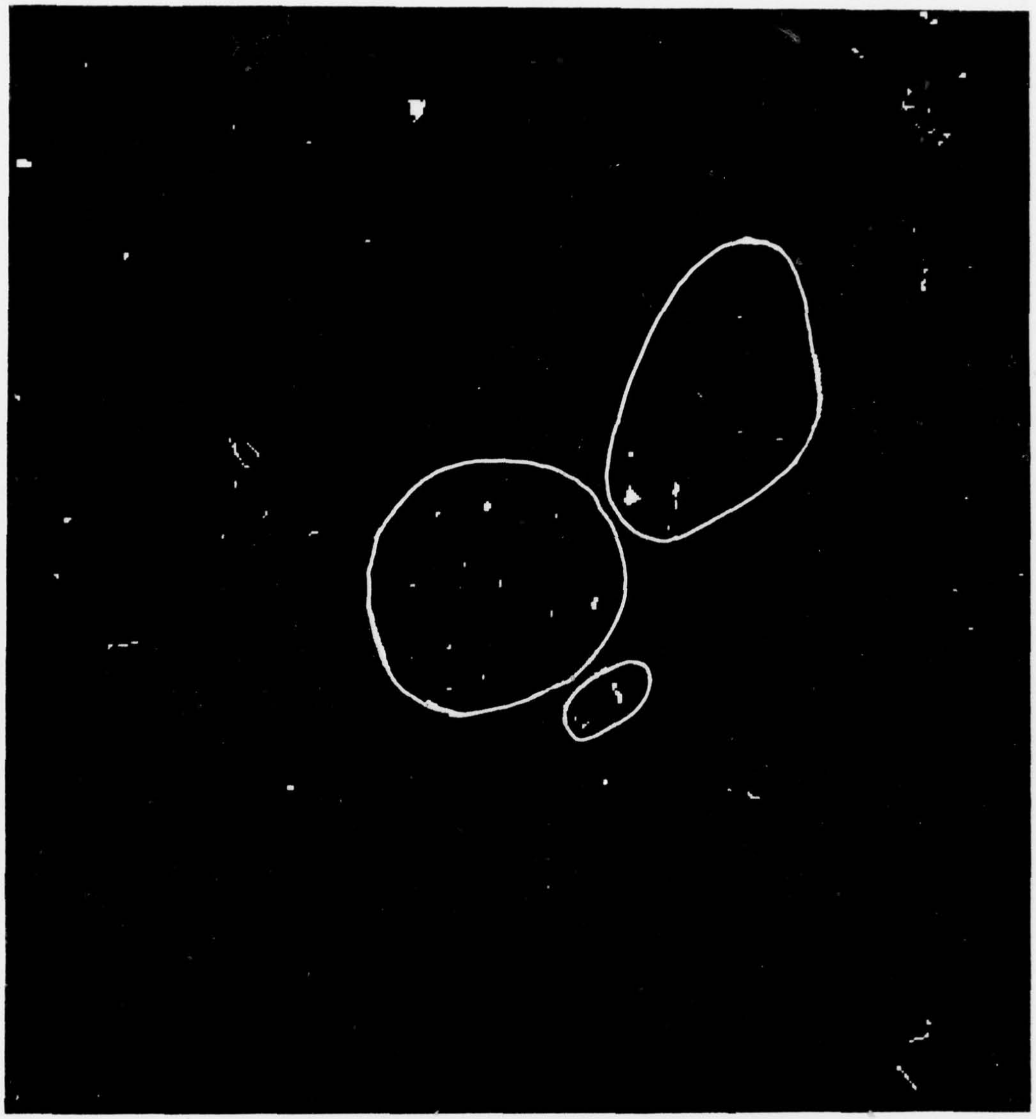


Figure 3-13. Region B (Hawk Site) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Adaptive Size (2) and Intensity (19).
Clustering: Minimal Spanning Tree.

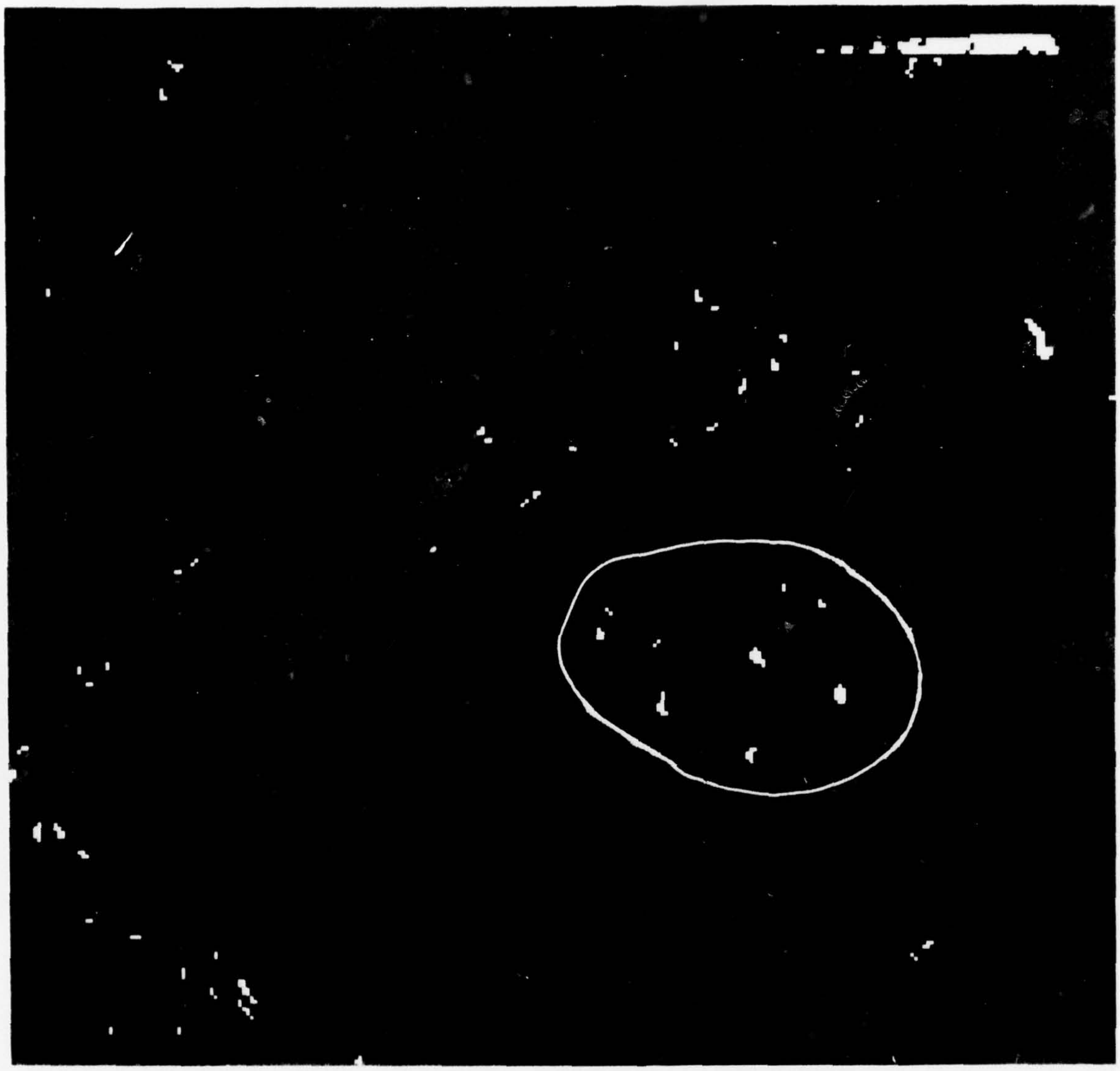


Figure 3-14. Region D (Hawk Site) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Adaptive Size (2) and Intensity (22).
Clustering: Minimal Spanning Tree.

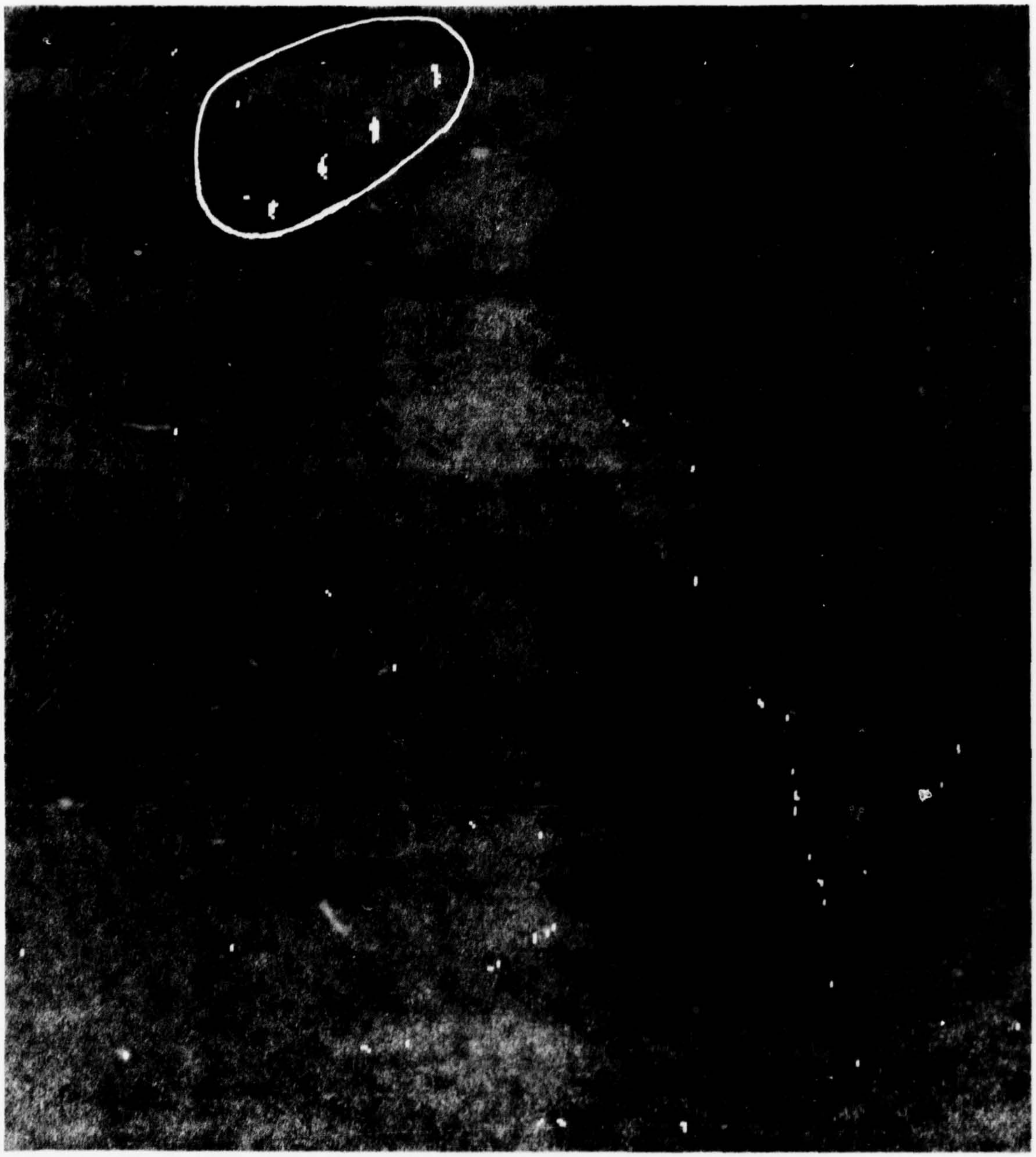


Figure 3-15. Region F (Helicopters) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Adaptive Size (2) and Intensity (12).
Clustering: Minimal Spanning Tree.

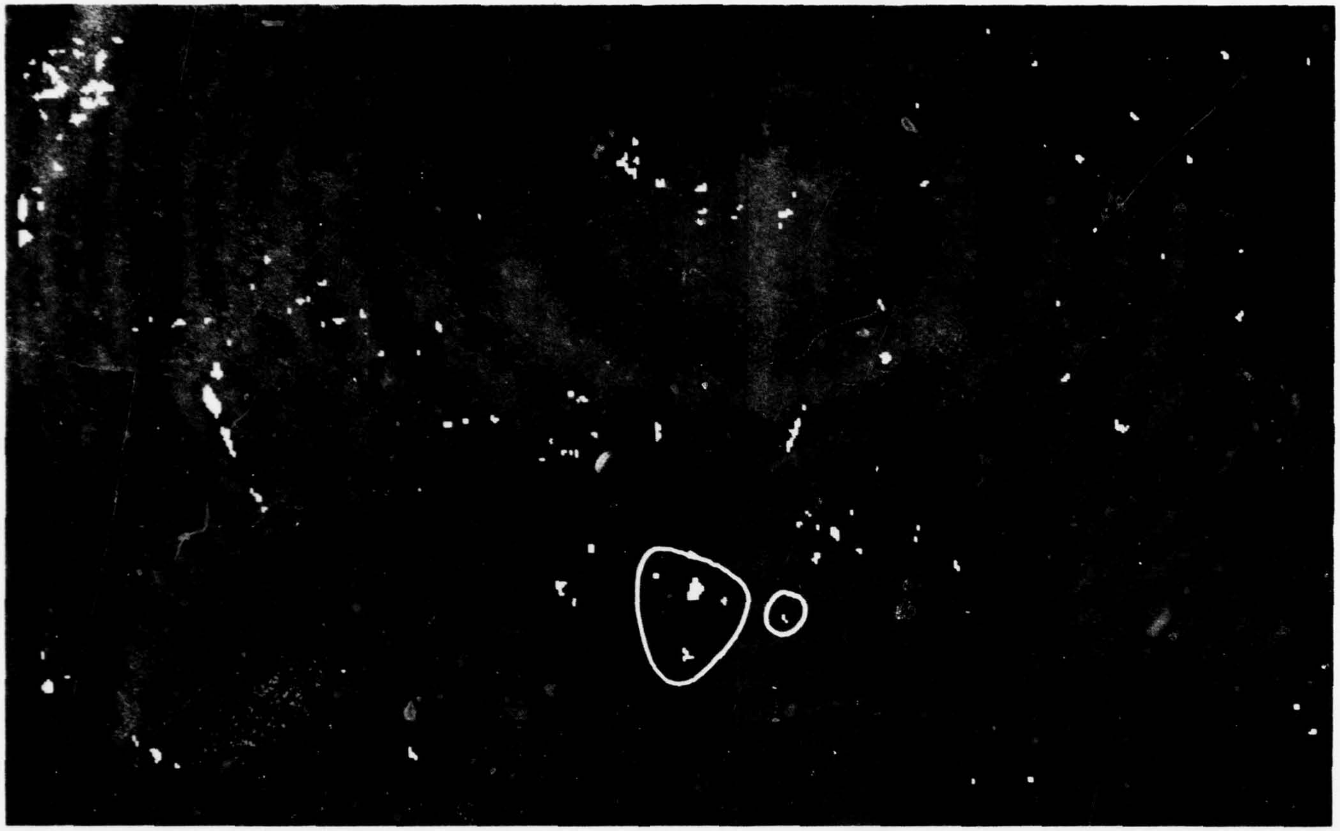


Figure 3-16. Region G (Armor) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Adaptive Size (2) and Intensity (19)
Clustering: Minimal Spanning Tree

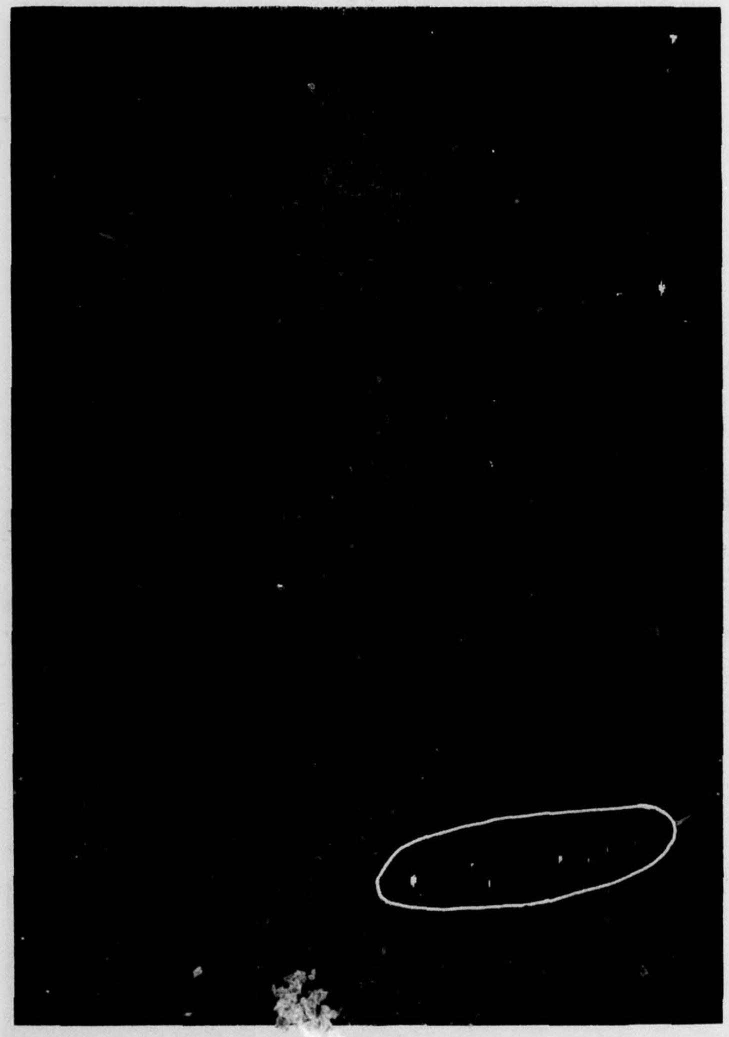


Figure 3-17. Region N (Tanks) Event Image (Enlarged) with Target Arrays Circled.
Thresholds: Adaptive Size (2) and Intensity (17).
Clustering: Minimal Spanning Tree.

and are circled in Figures 3-9 through 3-17.

Comparison of Figures 3-10 and 3-12 shows the differences which result when different thresholds are used on the same difference image. A single large target array in Figure 3-10 is replaced by three target arrays in Figure 3-12. The target array shown by arrows in Figure 3-10, is absent from Figure 3-12 because in the latter the array consisted of only 50% target events. (An array must consist of at least 52% target events to be labeled a target array and at least 52% background events to be labeled a background array.) Therefore, this array is not labeled and is removed from further consideration in classifier training and in automatic classification.

The target array shown by arrows in Figure 3-12 is missing in Figure 3-10 because of a modification in the event ground truth. Ground truth for Reforger Region A indicates the presence of helicopters, although which particular events are the helicopters is not specified. Therefore, all events which, on the basis of size and spacing from other events, could reasonably be designated helicopters were "identified" as such. The result of this event labeling (after clustering) is shown in Figure 3-10 where the target arrays are circled. At a later date, the decision was made to consider only the two classes, "military equipment" arrays and "background" arrays, because the small number of tank arrays, helicopter arrays, etc. precluded their being used as separate classes in the training of the classifier. The events comprising the array circled in Figure 3-12 but not in Figure 3-10 were believed to be of a military nature, even though the events did not appear to be helicopters. Thus, these events were labeled as target events so that the array in Figure 3-12 is labeled a target array.

The following material describes the three different classifier training and automatic classification experiments which were performed under this contract. Each experiment consists of a selection of images from among those of Figures 3-9 through 3-17. As illustrated in Figure 2-5, each array in a selected image is labeled as a target array or a background array on the basis of event ground truth. All of the target arrays in the selected images are used to evaluate the parameters in the target class-conditional probability density function, i.e., to train the classifier. The background arrays are used similarly. The trained classifier is then used to automatically classify each array as a target array or a background array. Comparison is then made between the classifications as given by the automatic classifier and by the ground truth assignment. These comparisons are summarized for the experiment as a confusion matrix which presents the number of target arrays classified as target arrays (a quantity related to the probability of detection), the number of target arrays classified as background arrays (a quantity related to the false alarm rate), and the corresponding quantities calculated for the background arrays.

3.1 FIRST EXPERIMENT

In this experiment, only the gun emplacement data from Gallant Hand, shown in Figure 3-9, was used. Of the 61 arrays obtained by clustering the events, eight were labeled as target arrays, 52 as background arrays, and one as neither (since it was composed of 50% target events and 50% background events). The eight target arrays are circled in Figure 3-9.

To train the classifier as indicated in Figure 2-5, 20 features were evaluated for each of the target arrays (or gun emplacement clusters). These eight points in the 20-dimensional feature space were used to evaluate the parameters (the 20-feature means and covariance matrix for the target arrays) in the Gaussian-modeled gun emplacement class-conditional probability density $f(\vec{X}/T)$. Next, the same 20 features were evaluated for each of the 52 background (on the basis of event ground truth) arrays. These 52 points in the 20-dimensional feature space were used to evaluate the parameters (the 20 feature means and covariance matrix for the background arrays) in the Gaussian-modeled background class-conditional probability density $f(\vec{X}/B)$.

Next, as shown in Figure 2-6, the number of feature variables was reduced from 20 ($\vec{X}$) to three ($\vec{X}'$) by determining the three independent linear combinations of the original 20 feature variables which are best at separating in feature space the target points from the background points. The three new variables are called "generalized features".

The linear transformation W relating the 20 features to the three generalized features can be written as in Equation 2.1. The 20 feature target and background means and covariance matrices are related to the three generalized feature target and background means and covariance matrices by Equations 2.1 and 2.2. Thus, the probability density functions for target arrays, $\hat{f}(\vec{X}'/T)$, and background arrays, $\hat{f}(\vec{X}'/B)$, for this experiment were evaluated.

The final step in the experiment was the inputting of each of the 60 arrays to the trained classifier for automatic classification as shown in Figure 2-4. For each array, the 20 features were evaluated, yielding a 20-vector $\vec{X}$ for that array. Using Equation 2.1, a generalized feature three-vector $\vec{X}'$ was calculated. Finally, the class-conditional probability densities for target, $\hat{f}(\vec{X}'/T)$, and for background, $\hat{f}(\vec{X}'/B)$, were evaluated for the array. These densities were used to calculate the Bayes a posteriori (i.e., <u>after</u> the measurement of $\vec{X}'$) probabilities $P(T/\vec{X}')$, which is the probability that the array with generalized features evaluated at $\vec{X}'$ is a target array, and $P(B/\vec{X}')$ which is the probability that the array is a background array. $P(T/X')$ and $P(B/\vec{X}')$ are given by Equation 2.3 and 2.4, where the a priori probabilities q(T) and q(B) are given by

$$q(T) = \frac{8}{60} \qquad (3.1)$$

$$q(B) = \frac{52}{60} \qquad (3.2)$$

If $P(T/\vec{X}') > .50$, the array is automatically classified as a target array; otherwise, it is automatically classified as a background array. Comparison is then made with the classification of the array based on event ground truth. The resulting confusion matrix is shown in Table 3-1.

Since all target arrays were correctly identified, the probability of detection is 100%. Since one background array was incorrectly identified, the probability of a false alarm was 1/52, which is approximately 2%.

TABLE 3-1. RESULTS OF AUTOMATIC CLASSIFICATION FOR FIGURE 3-9

GROUND TRUTH \ CLASSIFICATION RESULTS	TARGET	BACKGROUND
TARGET	8	0
BACKGROUND	1	51

3.2 SECOND EXPERIMENT

In this experiment, the Gallant Hand Gun Emplacement image (Figure 3-9), the Reforger Region A helicopter image (Figure 3-10), and the Reforger Region C beacon image (Figure 3-11) were used. The 115 arrays were obtained by clustering each of the three images separately. Seventeen arrays were labeled target arrays and 98 were labeled background arrays on the basis of the event ground truth. The 17 target arrays are circled in Figures 3-9, 3-10, and 3-11.

The procedures used in this experiment are the same as those described in the first experiment. To train the classifier as indicated in Figure 2-5, 20 features were evaluated for each of the 17 target arrays. The resulting 12 points in the 20-feature space were used to evaluate the parameters (the 20-feature means and covariance matrix for the target arrays) in the Gaussian-modeled target array class-conditional probability density $f(\vec{X}/T)$. The same 20 features were evaluated for each of the 98 background arrays. The resulting 98 points in the 20-feature space were used to evaluate the parameters (the 20-feature means and covariance matrix for the background arrays) in the Gaussian-modeled background class-conditional probability density $f(\vec{X}/B)$.

Next, as shown in Figure 2-6, the three generalized feature variables which are linear combinations of the original 20 feature variables and which are best at separating in feature space the target points from the background points were determined by the class-separating transformation W determined in <u>this</u> experiment. The 20-feature target and background means and covariance matrices are related to the three generalized feature target and background means and covariance matrices by Equations 2.1 and 2.2. Thus, the probability density functions for targets, $\hat{f}(\vec{X}'/T)$, and background, $\hat{f}(\vec{X}'/B)$, for this experiment were evaluated, so the classifier was trained.

Finally, each of the 115 arrays was inputted to the trained classifier for automatic classification, as shown in Figure 2-4. For each array, the 20 features were evaluated, yielding a 20-vector $\vec{X}'$ for that array. Using Equation 2.1, a generalized feature vector $\vec{X}'$ was determined for the array. Then, the class-conditional probability densities for target arrays, $\hat{f}(\vec{X}'/T)$, and for background arrays, $\hat{f}(\vec{X}'/B)$, were evaluated for the array. These densities were used to calculate the Bayes a posteriori probabilities $P(T/\vec{X}')$ and $P(B/\vec{X}')$ that the array with generalized feature $\vec{X}'$ is a target array and a background array, respectively. $P(T/\vec{X}')$ and $P(B/\vec{X}')$ are given by Equations 2.3 and 2.4, where the a priori probabilities q(T) and q(B) are given by

$$q(T) = \frac{17}{115} \tag{3.3}$$

$$q(B) = \frac{98}{115} \tag{3.4}$$

The confusion matrix shown in Table 3-2 is the result of the comparison of classification of the arrays based on automatic classification and event ground truth. Since all target arrays were correctly identified, the probability of detection was 100%. Since five out of 98 background arrays were automatically classified incorrectly as targets, the probability of false alarm was 5/98, which is approximately 5%. Thus, the addition of Region A with helicopter arrays (Figure 3-10) and Region C with beacon arrays (Figure 3-11) to the gun emplacement image (Figure 3-9) resulted in an increase in the probability of false alarm from 2% to 5%. However, the probability of target array detection remained 100%.

TABLE 3-2. RESULTS OF AUTOMATIC CLASSIFICATION FOR FIGURES 3-9, 3-10, AND 3-11

GROUND TRUTH \ CLASSIFICATION RESULTS	TARGET	BACKGROUND
TARGET	17	0
BACKGROUND	5	93

3.3 THIRD EXPERIMENT

In this experiment, Reforger Regions A, B, D, F, G, and N, shown in Figures 3-12 through 3-17, respectively, were used. Region C was not used because the beacons had patterns qualitatively different from those of the other regions. The 119 arrays were obtained by clustering each of these images separately. Fifteen arrays were labeled target arrays (Region A has helicopter arrays; Region B, a hawk site; Region D, a hawk site; Region F, helicopters; Region G, an armor array; and Region N, a tank array). Thus, the target arrays were arrays of different types of military equipment. In addition, there were 104 background arrays. The arrays were labeled as target or background on the basis of the event ground truth. The 15 target arrays are circled in Figures 3-12 through 3-17.

The procedures used in this experiment were the same as those described in the first and second experiments. To train the classifier as indicated in Figure 2-5, 20 features were evaluated for each of the 15 target arrays. These points were used to evaluate the 20-feature means and the covariance matrix for the target arrays so that the Gaussian-modeled target array class-conditional probability density $f(\vec{X}/T)$ was parameterized. The 20 features evaluated for the 104 background arrays were used to evaluate the corresponding parameters for the background array probability density $f(\vec{X}/B)$.

Next, as shown in Figure 2-6, the three generalized feature variables, which are linear combinations of the original 20 feature variables and which are best at separating in feature space the target points from the background points were determined by the class-separating transformation W determined in this experiment. The 20-feature target and background means and covariance matrix are related to the three generalized feature target and background means and covariance matrix by Equations 2.1 and 2.2 Thus, the probability density functions for targets, $\hat{f}(\vec{X}'/T)$, and background, $\hat{f}(\vec{X}'/B)$, for this experiment were evaluated so that the classifier was trained.

Finally, each of the 119 arrays was inputted to the trained classifier for automatic classification, as shown in Figure 2-6. For each array, the 20 features were evaluated, yielding a 20-vector $\vec{X}$ for that array. Using Equation 2.1, a generalized feature vector $\vec{X}'$ is determined for the array. Then, the class-conditional probability densities for target arrays, $\hat{f}(\vec{X}'/T)$, and for background arrays, $\hat{f}(\vec{X}'/B)$, were evaluated for the array. These densities were used to calculate the Bayes a posteriori probabilities $P(T/\vec{X}')$ and $P(B/\vec{X}')$ that the array with generalized feature $\vec{X}'$ is a target array and a background array, respectively. $P(T/\vec{X}')$ and $P(B/\vec{X}')$ are given by Equation 2.3 and 2.4, where the a priori probabilities q(T) and q(B) are given by

$$q(T) = \frac{15}{119} \qquad (3.5)$$

$$q(B) = \frac{104}{119} \qquad (3.6)$$

The confusion matrix shown in Table 3-3 is the result of the comparison of classification of the arrays based on automatic classification and event ground truth. Since all target arrays were correctly identified, the probability of detection was 100%. Since two out of 104 background arrays were automatically classified incorrectly as targets, the probability of false alarm was 2/104, which is approximately 2%.

TABLE 3-3. RESULTS OF AUTOMATIC CLASSIFICATION FOR FIGURES 3-12 THROUGH 3-17

GROUND TRUTH \ CLASSIFICATION RESULTS	TARGET	BACKGROUND
TARGET	15	0
BACKGROUND	2	102

4.0 CONCLUSIONS AND RECOMMENDATIONS

All of the three automatic classification experiments had 100% correct classification of the military equipment in operational deployment. The percentage of correct classification of background arrays ranged from 95% to 98%. In the first classification experiment, the only military equipment in the imagery consisted of gun emplacements. In the second classification experiment, the military equipment consisted of gun emplacements, helicopters and beacons. In the third experiment, the military equipment consisted of helicopters, hawk sites, armor and tanks.

The use of "military equipment" as a class was necessary because of the limited number of arrays of any particular type available for training the classifier. When more training data becomes available, helicopter arrays, etc., can be considered as separate target classes. Then the clasifier can be extended from the two-class system (background and target) to a multi-class system (background, helicopter arrays, tank arrays, etc.). The classifier output would then specify military equipment as to target type, i.e., helicopter arrays would be identified as helicopters and tank arrays would be identified as tanks.

The Air Force has developed the Modular Change Detector (MCD) for performing real-time change detection using a reference and mission pair of radar images. The MCD increases operator effectiveness by cueing him to changes. Going further, the three automatic classification experiments performed under this contract establish the feasibility of automatically screening for certain classes of targets consisting of groups of changes, e.g., missile sites and clusters of tanks or helicopters. Incorporating this screening on the output of the MCD could cue a user to particular targets of interest. Such cueing discriminates against both false alarm changes and legitimate changes which are not of interest. Previous studies have shown this could increase the operator's coverage rate and his probability of detection.

APPENDIX A
ITERATIVE CLUSTERING METHOD

Problem

Assign n_e events to n_c clusters in such a way that $n_c \leq n_e$ and all events assigned to the same cluster appear to the eye to be associated, and that all events that appear to the eye to be associated by proximity are assigned to the same cluster.

Approach

The first step is to define a probability density for each cluster. This is a function of the distance from the cluster center and the number and covariance of events assigned to the cluster. The next step is to iteratively pass through the list of n_c events assigning each event to the maximum likelihood cluster, determined by evaluating the conditional probability density for each existing cluster at the location of the event being examined, or creating a new cluster of one event.

Upon completion of each iteration, the result is examined to decide whether to accept the solution or modify the parameters and perform another iteration.

Process Details

The n_e events to be clustered are each described by a set of six numbers:

$$\overline{X} = \frac{1}{n_p} \sum_{i=1}^{n_p} X_i$$

$$\overline{Y} = \frac{1}{n_p} \sum_{i=1}^{n_p} Y_i$$

n_p = number of pixels in the event

$$C_{11} = \frac{1}{n_p - 1} \sum_{i=1}^{n_p} (X_i - \overline{X})^2$$

$$C_{12} = \frac{1}{n_p - 1} \sum_{i=1}^{n_p} (X_i - \overline{X})(Y_i - \overline{Y})$$

$$C_{22} = \frac{1}{n_p - 1} \sum_{i=1}^{n_p} (Y_i - \overline{Y})^2$$

where X_i, Y_i are the image coordinates of pixel i in the event.

It will be convenient to refer to the properties of the $j^{\underline{th}}$ event in the following way:

$$R_j^T = (\overline{X}, \overline{Y})^T{}_j \equiv \text{Event centroid}$$

$$C_j = \begin{pmatrix} C_{11} & C_{12} \\ C_{12} & C_{22} \end{pmatrix}_j \equiv \text{Event covariance}$$

The probability density function which describes the distribution of events in a cluster depends on the distance and direction from the cluster center and the number of events, n_c, in the cluster. The following notation is used.

$$\overline{R}_i = \frac{1}{n_c} \sum_{j=1}^{n_c} R_j \equiv \text{Cluster Centroid}$$

$$S_i = \frac{1}{n_c - 1} \sum_{j=1}^{n_c} \left(\frac{n_p - 1}{n_p} \right)_j C_j \qquad C_j \equiv \text{Cluster Covariance}$$

where the index j runs over all events included in cluster i.

S_i is defined as written only for $n_c > 1$. In the current implementation, the preceding definition is used for $n_c \geq 3$. For $n_c \leq 2$,

$$S_i = \begin{pmatrix} D^2 & 0 \\ 0 & D^2 \end{pmatrix}$$

It is thereby assumed that a cluster with one or two events is circularly symmetric about its mean with a characteristic dimension of D.

The probability density is written as

$$f(r,n) = \frac{\beta^4}{\alpha^2 + \beta^2} \quad (1 + \alpha^2 d^2) e^{-\beta^2 d^2}$$

where

$$d^2 = h(n,s)\ (r-\overline{R})^T\ S^{-1} (r-\overline{R})$$

$$\alpha = \frac{n-1}{n}$$

$$\beta^2 = \frac{1}{1 + \alpha^2}$$

The function h(n,s) is a smoother to prevent the size of the function f(r,n) from changing rapidly from a circularly symmetric function of dimension D to an elliptically symmetric function with dimensions defined by the covariance matrix S. The transformation which diagonalizes S leads to eigenvalues corresponding to the major and minor axes of an ellipse as follows:

$$U^T S\ U = \begin{pmatrix} D_1^2 & 0 \\ 0 & D_2^2 \end{pmatrix}$$

h(n,s) is defined by requiring that, for n=3, the area of an ellipse with axes D_1 and D_2 equals the area of the circle with diameter D. For larger values of n, the distribution is allowed to gradually assume a size depending only on the covariance of events in the cluster. Thus

$$h(n,s) = \left(1.0 - \left(1.0 - \frac{D^2}{D_1 * D_2} \right) e^{-(n-3)} \right)$$

Assignment of an event located at $R_j^T = (\overline{X}, \overline{Y})_j^T$ to the maximum likelihood cluster is accomplished by evaluating $f_i(R_j, n_c)$ for each cluster i, and assigning event j to cluster i if

$$f_i(R_j, n_c) \geq f_k(R_j, n_c) \text{ for any value of k.}$$

Event j is assigned to the maximum likelihood cluster or begins a new cluster as the probability volume from the cluster centroid to the contour of equal likelihood passing through R_j is less than or greater than a threshold value P. Fortunately there is an analytical form for this integral. It can be shown that

$$2\pi \int_0^x f(r,n)\ rdr = P(r,n),$$

where

$$P(r,n) = 1 \left(1 + \frac{\alpha^2 \beta^2}{\alpha^2 + \beta^2} r^2 \right) e^{-\beta^2 r^2}$$

The values of $P(r,n)$ at $r = 0$, and $r = \infty$ show that $f(r,n)$ is in fact a probability density: $P(o,n) = 0$; $P(\infty,n) = 1$.

Upon completion of one iteration, every event has been assigned to some cluster. Subsequent interations are conducted, for each event, by removing the event from the cluster to which it is assigned (in order that the cluster statistics used to make the maximum likelihood cluster selection will not be biased in favor of the previous decision), and making a new decision.

If the parameters D and P are not changed, two results may be found. Either the iteration cycles are terminated by detecting that no further changes are introduced by iterating, or changes always occur. In the last case, it has been found that this usually means that two or more events are transferred back and forth between the same pair of clusters. **Situations** of this type are detected and treated as appropriate to prevent the process from continuing indefinitely.

In the first case, when an iteration produces no changes in cluster assignments, a decision is required. Either the solution is accepted, or the solution is disturbed to produce changes. The disturbance may be intended either to produce more clusters or fewer clusters.

Alternatives available for disturbing the solution include:

1. Adjustment of the parameter values D and P.
2. Selecting and removing from its cluster an event to create a new cluster.
3. Restarting the process and preventing actions which lead to an unsatisfactory solution.

Selection of an appropriate alternative requires an evaluation of the current solution. Each solution (the state of the system after completion of an iteration) is characterized by three numbers. These are, the number of clusters, N_c, the number of events moved from one cluster to a new cluster, N_m, and the number of clusters with a characteristic dimension exceeding a threshold value, N_1.

Before discussing the role of these numbers in selecting an alternative, it is appropriate to discuss some additional parameters used to start the clustering process.

There are two special values of D, the size of the circularly symmetric distribution used to start clusters; its initial value, and its maximum value. Both of these are input parameters. Whenever D is set to a value larger than the maximum value, the symmetric form of the probability density function is used for all computations without regard to the number of events in a cluster.

The initial value of D may be set to a value larger than the maximum value to accomplish a swift gross clustering by distance only. A clustering solution is never accepted as final when D exceeds the maximum permitted value. When the clustering is performed, the first iteration typically has a large value of D and produces a small number of very large clusters.

The second iteration is performed with D set down to the maximum permitted value, and probability density functions are computed as already described. By this operation the few large clusters formed initially fall apart into several smaller ones. Subsequent iterations are performed to remove residual order dependence and to find a stable solution that satisfies other imposed requirements.

Other parameters used to begin the clustering process include an initial estimate of the number of clusters in the solution. (If this estimate is omitted, a one will be used. If provided, the number of iterations needed to reach a satisfactory solution may be reduced.) Also included are P, the probability threshold for beginning a new cluster; L, a characteristic dimension associated with the largest significant activity of interest; N_L, the smallest number of events associated with the largest activity.

At the end of each iteration, the three numbers which describe the state of the solution are examined. Iterations continue until the largest cluster in the solution with more than N_L events, has a characteristic dimension smaller than L. The parameters D and P are adjusted at each iteration to encourage an increase or a decrease in the number of clusters obtained. Decreasing P encourages new clusters; increasing P discourages new clusters; similarly for D. D is never raised above the maximum value except when restarting is necessary. P is adjusted between the limits of 0.5 and 1.0.

When several iterations with readjusted parameters fail to change the number of clusters, direct action is taken. If more clusters are needed, a permanent new cluster is formed by removing from the largest cluster, the event which has the largest Mahalanobis distance from the cluster center. A large increase in the number of clusters which results from this action is not permitted. If parameter adjustment fails to recombine some of them, the offending action is recorded and the clustering process is reinitiated with a very large value of D. When the process returns to the offending action, it is prevented and another alternative is forced.

When a solution is reached in which all clusters are of satisfactory size, and the number of clusters in the solution is reached through an orderly process, a stability test is performed. In this situation iterations are performed which permit events to move freely from one cluster to another; no new clusters are created and no old ones are destroyed. This is accomplished by assigning each event to the maximum likelihood cluster and by refusing to destroy a cluster by reassigning the cluster nucleus (The event nearest the cluster centroid; these are determined for each cluster before the stability test begins). Iterations continue until no reassignments are needed or until it is apparent that the reassignments are due to oscillation of two or more events between two or more clusters. In either event the system is then judged to be in a relaxed state.

APPENDIX B

FEATURES FOR DESCRIBING ARRAYS

After the clusters have been isolated, that is, individual events have been grouped in an array, characteristics about these arrays can be used to identify targets. The most obvious characteristics or features are size and intensity of individual events and combinations of size and intensity of events in an array. A proposed set of features is described which will be used in the multi-variate group-separating procedures. The features describe the geometric properties of a group of change events in an array.

Subroutines to perform the computations necessary to transform binary data describing an array of change events into the set of twenty features are on hand. The following measurements are accumulated for each array.

N_j = Number of pixels in event j

$\vec{r}_j$ = Centroid of event j

C_j = Covariance of event j about the event centroid

N = Number of events in the array

$\vec{R}$ = Geometric centroid of the array

C = Geometric covariance of the array

$\vec{R}_m$ = Mass centroid of the array

C_m = Mass covariance of the array

The quantitative definitions of these measurements are given below.

$$\vec{r}_{ij} = \text{image coordinate vector to pixel } i \text{ of event } j$$

$$= \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

b_j = set of all pixels in event j

$$\vec{r}_j = \frac{1}{N_j} \sum_{b_j} \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

R_j = matrix of pixel vectors in event j

$$= (\vec{r}_{1j} - \vec{r}_j, \vec{r}_{2j} - \vec{r}_j, \ldots, \vec{r}_{N_j j} - \vec{r}_j)$$

$$C_j = \frac{1}{N_j} R_j R_j^T$$

$$\vec{R} = \frac{1}{N} \sum_j \vec{r}_j$$

R = matrix of event centroids about the array centroid

$$= (\vec{r}_1 - \vec{R}, \vec{r}_2 - \vec{R}, \ldots, \vec{r}_N - \vec{R})$$

$$C = \frac{1}{N} R R^T$$

$$\vec{R}_M = \frac{1}{\sum_{j=1}^{N} N_j} \left(\sum_j N_j r_j \right)$$

R_M = matrix of event centroids about the mass centroid

$$= (\vec{r}_1 - \vec{R}_M, \vec{r}_2 - \vec{R}_M, \ldots, \vec{r}_N - \vec{R}_M)$$

$$C_M = \frac{1}{N} R_M R_M^T$$

Define three diagonalizing transformations as follows:

$$\begin{pmatrix} \lambda_{1j} & 0 \\ 0 & \lambda_{2j} \end{pmatrix} = H_j C_j H_j^T$$

$$H_j = (H_{1j}, H_{2j})$$

$$H_{ij} = \begin{pmatrix} h_{i1} \\ h_{i2} \end{pmatrix}_j$$

$$\begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix} = U\, C\, U^T$$

$$U = (U_1, U_2)$$

$$U_i = \begin{pmatrix} U_{i1} \\ U_{i2} \end{pmatrix}$$

$$\begin{pmatrix} \Gamma_1 & 0 \\ 0 & \Gamma_2 \end{pmatrix} = M\, C_M\, M^T$$

$$M = (M_1, M_2)$$

$$M_i = \begin{pmatrix} M_{i1} \\ M_{i2} \end{pmatrix}$$

The eigenvalues of the three covariance matrices are found on the diagonals of the left hand side and the eigen vectors which define the three principal directions are:

$$H_1 = \begin{pmatrix} h_{11} \\ h_{12} \end{pmatrix}, \quad U_1 = \begin{pmatrix} U_{11} \\ U_{12} \end{pmatrix}$$

and $M_1 = \begin{pmatrix} M_{11} \\ M_{12} \end{pmatrix}$

Define the distance from event i to its nearest neighbor as

$$\vec{d}_i = \min_j \left| (\vec{r}_i - \vec{r}_j) \right|$$

A set of twenty features which characterize the array is computed from these measurements. The description is based on the size, shape, arrangement, orientation and uniformity of events in the array. It is also based on the array size and shape and on the mass size, shape and displacement. The following features are used.

1. Array Size

Feature 1 = $\log_{10} N$

2. Spacing of Events

This is the only feature which requires a search of the data.

Define $\overline{d} \equiv \frac{1}{N} \sum_{i=1}^{N} |\vec{d}_i|$

Feature 2 = $\log_{10} (\overline{d})$

3. Regularity of Spacing

Feature 3 = $\sqrt{\frac{1}{N} \sum_{i=1}^{N} (d_i - \overline{d})^2}$

4. Event size

$$\text{Define } \overline{\lambda}_1 \equiv \frac{1}{N} \sum_{i=1}^{N} \lambda_{1i}$$

$$\text{Feature 4} = \log_{10} \left(\sqrt{\overline{\lambda}_1}\right)$$

5. Uniformity of Event Size

$$\text{Feature 5} = \left(\frac{1}{N} \sum_{i} (\lambda_{1i} - \overline{\lambda}_1)^2 \right)^{1/4}$$

6. Event Shape

$$\text{Feature 6} = \overline{\ell} = \frac{1}{N} \sum_{j} \ell_j \quad \text{where } \ell_j = \sqrt{\frac{\lambda_{2j}}{\lambda_{1j}}}$$

7. Uniformity of Event Shape

$$\text{Feature 7} = \sqrt{\frac{1}{N} \sum_{j} (\ell_j - \overline{\ell})^2}$$

8. Uniformity of Event Orientation

$$\text{Feature 8} = \sqrt{\frac{1}{N} \sum_{j} \{1 - (\hat{H}_{1j} \cdot \hat{p})^2\}}$$

where

$$\hat{p} = \frac{\sum_{j} (h_{11}, h_{12})_j}{\sum_{j} (h^2_{11} + h^2_{12})_j}$$

and

$$\hat{H}_{1j} = (h_{11}, h_{12})_j$$

is a unit vector in the principal direction of event j.

9. Array Size

$$\text{Feature } 9 = \log_{10}\left(\sqrt{\frac{\Lambda_1}{\bar{\lambda}_1}}\right)$$

where Λ_1 is the first eigenvalue of the array using the array centroid.

10. Array Shape

$$\text{Feature } 10 = \sqrt{\Lambda_2/\Lambda_1}$$

11. Orientation of Events to Array

$$\text{Feature } 11 = \sqrt{\frac{1}{N} \sum_j \{1 - (U_1 \cdot H_{1j})^2\}}$$

12. Event Area

$$\text{Define } \overline{N} = \frac{1}{N} \sum_{j=1}^{N} N_j$$

$$\text{Feature } 12 = \log_{10} (\overline{N})$$

13. Uniformity of Event Area

$$\text{Feature } 13 = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \{\log_{10}(N_j) - \log_{10}(\overline{N})\}^2}$$

14. Array Mass Size

$$\text{Feature 14} = \log_{10}\left\{\sqrt{\Gamma_1}/\log_{10}\left(\sqrt{\frac{\Lambda_1}{\overline{\lambda}_1}}\right)\right\}$$

where Γ_1 is the first eigenvalue.

15. Array Mass Shape

$$\text{Feature 15} = \sqrt{\Gamma_2/\Gamma_1}$$

16. Orientation of Array Mass to Array

$$\text{Feature 16} = \sqrt{1 - (\hat{M}_1 \cdot \hat{U}_1)^2}$$

17. Distance from Mass to Geometric Centroids

$$\text{Feature 17} = \log_{10}(|\vec{R} - \vec{R}_M|$$

18. Array Pixel Density

$$\text{Feature 18} = -\log_{10}\left\{\frac{\sum_j N_j}{(1+\sqrt{\Gamma_1})(1+\sqrt{\Gamma_2})}\right\}$$

19. Event Pixel Density

$$\text{Define } P_j = \frac{N_j}{(1+\sqrt{\lambda_{1j}})(1+\sqrt{\lambda_{2j}})}$$

$$\text{Feature 19} = \overline{P} = \frac{1}{N}\sum_{j=1}^{N} P_j$$

20. Uniformity of Event Pixel Density

$$\text{Feature } 20 = \sqrt{\frac{1}{N} \sum_{j=1}^{N} (\rho_j - \overline{\rho})^2}$$

APPENDIX C

CLASS-SEPARATING TRANSFORMATION

The following discussion develops the "optimum" linear transformation W of the array vectors in feature space which both clusters arrays of the same class and separates arrays that belong to different classes. The measure of clustering or separation between two arrays is the Euclidean distance between the arrays after their linear transformation to the new feature space.

The effect of such a transformation is illustrated in Figure C-1, where like arrays of each class have been clustered and the two clusters have been separated from each other.

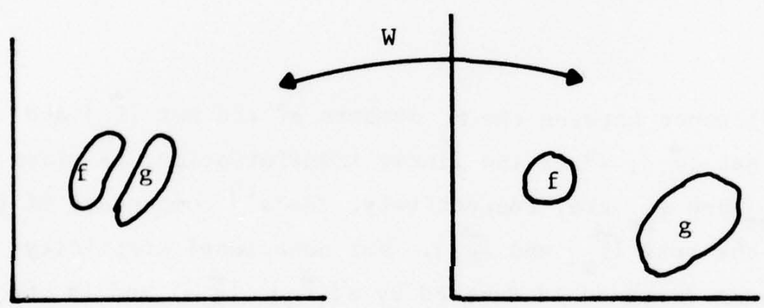


Figure C-1. Separation of Classes after the Linear **Transformation** W to New Feature Space.

One transformation that accomplishes the stated objective can be specified as follows: Find the linear transformation W that maximizes, after transformation, the mean-square distance between points that belong to different classes (the mean-square interset distance) subject to the constraint that the mean-square distance between the points of one class (the mean-square intraset distance of the class) is held constant after transformation.

The particular linear transformation W that maximizes atter transformation the mean-square **interset** distance while holding the mean-square intraset distance of one class constant after transformation is developed below. The purpose of the transformation is to separate arrays of different classes while clustering those that belong to the same class.

The linear transformation W maps the N-dimensional feature vector $\vec{f}$ into $\vec{f'}$:

$$\vec{f'} = W \vec{f} \qquad \text{(C.1)}$$

$$\begin{pmatrix} f_1' \\ f_2' \\ \vdots \\ f_N' \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix} \qquad \text{(C.2)}$$

so that $$f_i' = \sum_j w_{ij} f_j \qquad \text{(C.3)}$$

The mean-square distance between the M_1 members of the set $\{\vec{f}_m\}$ and the M_2 members of the set $\{\vec{g}_p\}$, after the linear transformation, is given by Equation C.4, where f_{ms} and g_{ps} are, respectively, the s^{th} components of the m^{th} and p^{th} arrays of the sets $\{\vec{f}_m\}$ and $\{\vec{g}_p\}$. For notational simplicity, this mean-square interset distance is denoted by $S(\{\vec{f}_m\}, \{\vec{g}_p\})$ and is the quantity to be maximized by a suitable choice of the linear transformation W. The choice of the notation above is intended to signify that the transformation to be found is a function of the two sets.

$$S(\{f_m\}, \{g_p\}) = \frac{1}{M_1 M_2} \sum_{m=1}^{M_1} \sum_{p=1}^{M_2} (\vec{f}_m{}' - \vec{g}_p{}') \cdot (\vec{g}_m{}' - \vec{g}_p{}') \qquad \text{(C.4a)}$$

$$= \frac{1}{M_1 M_2} \sum_{m=1}^{M_1} \sum_{p=1}^{M_2} \sum_{n=1}^{N} (f'_{mn} - g'_{pn})^2 \qquad \text{(C.4b)}$$

$$= \frac{1}{M_1 M_2} \sum_{m=1}^{M_1} \sum_{p=1}^{M_2} \sum_{n=1}^{N} \left[\sum_{s=1}^{N} w_{ns} (f_{ms} - g_{ps}) \right]^2 \qquad \text{(C.4c)}$$

The constraint that the mean-square distance θ between the points of one set, say $\{\vec{f}_m\}$, is a constant is expressed by Equation C.5.

$$\theta = \frac{1}{(M_1-1)M_1} \sum_{m=1}^{M_1} \sum_{p=1}^{M_1} (\vec{f}_m{}'-\vec{f}_p{}') \cdot (\vec{f}_m{}'-\vec{f}_p{}') \tag{C.5a}$$

$$= \frac{1}{(M_1-1)M_1} \sum_{m=1}^{M_1} \sum_{p=1}^{M_1} \sum_{n=1}^{N} (f'_{mn}-f'_{pn})^2 \tag{C.5b}$$

$$= \frac{1}{(M_1-1)M_1} \sum_{m=1}^{M_1} \sum_{p=1}^{M_1} \sum_{n=1}^{N} \left[\sum_{s=1}^{N} w_{ns}(f_{ms}-f_{ps}) \right]^2 \tag{C.5c}$$

$$= K, \text{ a constant}$$

Both equations C.5 and C.4 can be simplified by expanding the squares as double sums and interchanging the order of summations. Thus

$$S(\{\vec{f}_m\}, \{\vec{g}_p\}) = \sum_{n=1}^{N} \sum_{s=1}^{N} \sum_{r=1}^{N} w_{ns}\, w_{nr}\, x_{sr} \tag{C.6a}$$

where

$$x_{sr} = x_{rs} = \frac{1}{M_1M_2} \sum_{m=1}^{M_1} \sum_{p=1}^{M_2} (f_{ms}-g_{ps})(f_{mr}-g_{pr}) \tag{C.6b}$$

and

$$\theta = \sum_{n=1}^{N} \sum_{s=1}^{N} \sum_{r=1}^{N} w_{ns}\, w_{nr}\, t_{sr} = K \tag{C.7a}$$

where

$$t_{sr} = t_{rs} = \frac{1}{(M_1-1)M_1} \sum_{m=1}^{M_1} \sum_{p=1}^{M_1} (f_{ms}-f_{ps})(f_{mr}-f_{pr}) \qquad (C.7b)$$

The coefficient x_{sr} is the general element of the matrix X which is of the form of a moment matrix and arises from considerations of cross-set distances. The matrix T with general coefficient t_{sr} arises from considerations involving distances between the points of only one set, the set $\{\vec{f}_m\}$.

Equation C.6 can be maximized, subject to the constraint of Equation C.7 by the method of Lagrange multipliers. Since dw_{ns} is arbitrary in Equation C.8, Equation C.9 must be satisfied.

$$dS - \lambda d\theta = \sum_{n=1}^{N} \sum_{s=1}^{N} dw_{ns} \sum_{r=1}^{N} w_{nr}(x_{sr} - \lambda t_{sr}) = 0 \qquad (C.8)$$

$$\therefore \sum_{r=1}^{N} w_{nr}(x_{sr} - \lambda t_{sr}) = 0 \qquad n=1, 2, \ldots, N;\ s=1, 2, \ldots, N \qquad (C.9)$$

Equation C.9 can be written in matrix notation to exhibit the solution in an illuminating way. Let w_n be a row vector with N components $(w_{n1}, \ldots, \vec{w}_{nN})$, then Equation C.9 can be written as

$$\begin{aligned} &\vec{w}_1\ (X-\lambda T) = 0 \\ &\ldots\ldots\ldots \\ &\vec{w}_n\ (X-\lambda T) = 0 \\ &\ldots\ldots\ldots \\ &\vec{w}_N\ (X-\lambda T) = 0 \end{aligned} \qquad (C.10a)$$

Multiplying both sides of the equation from the right by T^{-1}, Equation C.10b, which is of the form of an eigen value problem, is obtained.

$$\vec{w}_1 \; (XT^{-1}-\lambda I) = 0$$
$$\cdots\cdots\cdots\cdots$$
$$\vec{w}_n \; (XT^{-1}-\lambda I) = 0 \qquad \text{(C.10b)}$$
$$\cdots\cdots\cdots\cdots$$
$$\vec{w}_N \; (XT^{-1}-\lambda I) = 0$$

T^{-1} exists since T is positive definite. Equations C.10a and C.10b can be satisfied in either of two ways. Either $\vec{w}_n$ (the row vector which is the n^{th} row of the linear transformation W given by Equation C.2) is identically zero, or it is an eigenvector of the matrix $XT.^{-1}$ In order to find the solution that maximizes S, a substitution in the mean-square interset distance given by Equation C.6a must be made. Note first that through matrix notation, Equations C.6a and C.7a can be written as Equations C.11 and C.12.

$$S(\{\vec{f}_m\},\{\vec{g}_p\}) = \sum_{n=1}^{N} \vec{w}_n \, X \, \vec{w}_n^{\,T} \qquad \text{(C.11)}$$

$$\theta = \sum_{n=1}^{N} \vec{w}_n \, T \, \vec{w}_n^{\,T} = K \qquad \text{(C.12)}$$

But from Equation C.10a, $\vec{w}_n X$ can always be replaced by $\lambda \vec{w}_n T$. Thus Equation C.11 can be written as Equation C.13, where the constraint of Equation C.12 is used.

$$S(\{\vec{f}_m\},\{\vec{g}_p\}) = \lambda \sum_{n=1}^{N} \vec{w}_n \, T \, \vec{w}_n^{\,T} = \lambda K \qquad \text{(C.13)}$$

Thus the <u>largest</u> eigenvalue of $(X-\lambda T) = 0$ determines the transformation that maximizes the mean-square interset distance, subject to the constraint that the mean-square intraset distance in the set $\{\vec{f}_m\}$ is a constant. The transformation is given by Equation C.14, where $\vec{w}_1 \equiv (w_{11}, w_{12}, \ldots, w_{1N})$ is the eigenvector corresponding to the largest eigenvalue, λ_1, and the other $\vec{w}_n$ of Equations C.10a and C.10b are identically zero.

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ 0 & 0 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix} \qquad (C.14)$$

The transformation of this equation is singular, which expresses the fact that the projection of a point in feature space onto the line of maximum mean-square interset distance and constant mean-square intraset distance for set $\{\vec{f}_m\}$ is the single most important feature determining class membership. This is illustrated in Figure C-2, where the line aa' is in the direction of the eigenvector with the largest eigenvalue of the matrix XT^{-1}. The point $\vec{p}$ represents an array of unknown classification with known values of Feature 1 and Feature 2. The point's projection onto line aa' is the single best class-separating feature. The point $\vec{p}$ is classified as class $\{\vec{g}_p\}$ because the mean-square difference between its projection on line aa' and the projection of points belonging to set $\{\vec{g}_p\}$, $S(\vec{p},\{\vec{g}_p\})$, is less than $S(\vec{p},\{\vec{f}_m\})$, the corresponding difference with members of set $\{\vec{f}_m\}$.

For the twenty feature problems considered in the contract, the three best class-separating features (or projections) were used. This transformation is given by Equation C.15, where $\vec{w}_1 \equiv (w_{11}, w_{12}, \ldots, w_{1N})$, $\vec{w}_2 \equiv (w_{21}, w_{22}, \ldots, w_{2N})$, and $\vec{w}_3 \equiv (w_{31}, w_{32}, \ldots, w_{3N})$ are the eigenvectors corresponding to the three largest eigenvalues (λ_1, λ_2, and λ_3), and where the $\vec{w}_n$ of Equations C.10a and C.10b are identically zero.

$$W = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ w_{31} & w_{32} & \cdots & w_{3N} \\ 0 & 0 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix} \qquad (C.15)$$

Thus, for each array, the three best generalized features $\vec{f}'$ are calculated from the original twenty features $\vec{f}$ using Equations C.15 and C.1.

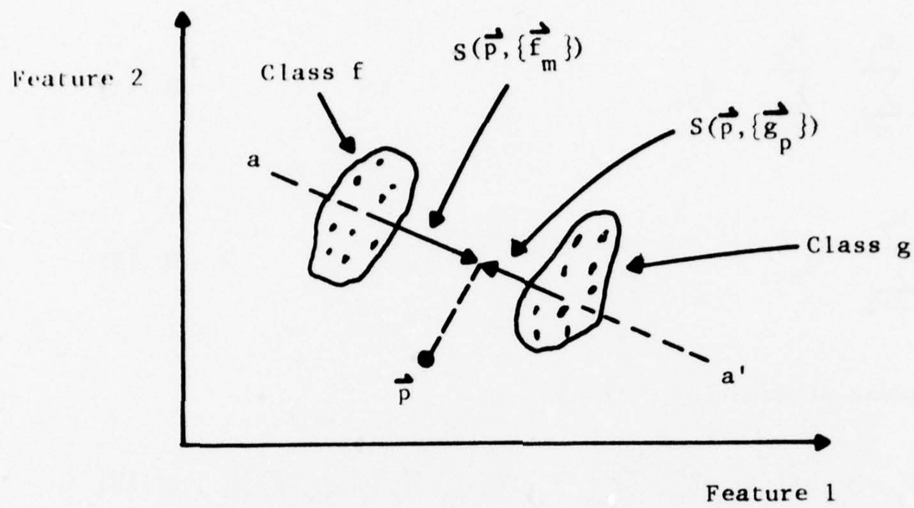


Figure C-2. Singular Class-Separating Transformation

In order to determine the transformation W in Equation C.15, the eigenvectors $\vec{w}$ and eigenvalues λ of XT^{-1}, as indicated in Equation C.10b, must be obtained. The elements of the matrices X and T are given by Equations C.6b and C.7b, respectively. The following equations indicate how these matrices can be expressed in terms of various moment matrices of the two classes $\{\vec{f}_m\}$ and $\{\vec{g}_p\}$.

From Equation C.6b,

$$X_{sr} = X_{rs} = \frac{1}{M_1 M_2} \sum_{m=1}^{M_1} \sum_{p=1}^{M_2} (f_{ms} - g_{ps})(f_{mr} - g_{pr}) \qquad (C.16)$$

Define

$$\overline{g}_s = \frac{1}{M_1 M_2} \sum_{m=1}^{M_1} \sum_{p=1}^{M_2} g_{ps} \qquad (C.17)$$

$$\overline{g}_s = \frac{1}{M_2} \sum_{p=1}^{M_2} g_{ps} \qquad (C.18)$$

Rewrite the argument of Equation C.16 as

$$(f_{ms}-g_{ps})(f_{mr}-g_{pr})=([f_{ms}-\overline{g}_s]-[g_{ps}-\overline{g}_s])([f_{mr}-\overline{g}_r]-[g_{pr}-\overline{g}_r]) \qquad (C.19)$$

Using Equations C.19 and C.17, Equation C.16 becomes

$$X_{rs} = \overline{([f_s-\overline{g}_s]-[g_s-\overline{g}_s])([f_r-\overline{g}_r]-[g_r-\overline{g}_r])} \qquad (C.20)$$

$$X_{rs} = \overline{[f_s-\overline{g}_s][f_r-\overline{g}_r]}-\overline{[f_s-\overline{g}_s][g_r-\overline{g}_r]} \qquad (C.21)$$

$$-\ \overline{[g_s-\overline{g}_s][f_r-\overline{g}_r]}+\overline{[g_s-\overline{g}_s][g_r-\overline{g}_r]}$$

The second term of Equation C.21 can be evaluated.

$$\overline{[f_s-\overline{g}_s][g_r-\overline{g}_r]}=\overline{[f_s-\overline{g}_s]}\,\overline{[g_r-\overline{g}_r]} = 0 \qquad (C.22)$$

since

$$\overline{[g_r-\overline{g}_r]}= \overline{g}_r-\overline{g}_r = 0 \qquad (C.23)$$

The same result is obtained for the third term of Equation C.21.

Thus Equation C.21 becomes

$$X_{sr} = \overline{[f_s-\bar{g}_s][f_r-\bar{g}_r]} + \overline{[g_s-\bar{g}_s][g_r-\bar{g}_r]} \tag{C.24}$$

Equation C.24 indicates that X_{sr} is the sum of the covariance of class f about the mean of class g and the covariance of class g about the mean of class g. Using matrix notation, Equation C.24 can be written.

$$X = F(\bar{g}) + G(\bar{g}) \tag{C.25}$$

where $F(\bar{g})$ is the twenty-feature covariance matrix of class f about the twenty-feature mean, $\bar{g}$, of class g, and $G(\bar{g})$ is the covariance of class g about the mean of class g. If, instead of Equation C.19, the following substitution is made

$$(f_{ms}-g_{ps})(f_{mr}-g_{pr}) = ([f_{ms}-\bar{f}_s]-[g_{ps}-\bar{f}_s])([f_{mr}-\bar{f}_r]-[g_{pr}-\bar{f}_r]) \tag{C.26}$$

Then Equation C.25 becomes

$$X = F(\bar{f}) + G(\bar{f}) \tag{C.27}$$

The matrix T can be evaluated from Equation C.7b.

$$t_{sr} = t_{rs} = \frac{1}{(M_1-1)M_1} \sum_{m=1}^{M_1} \sum_{p=1}^{M_1} (f_{ms}-f_{ps})(f_{mr}-f_{pr}) \tag{C.28}$$

Rewrite the argument of Equation C.28 as

$$(f_{ms}-f_{ps})(f_{mr}-f_{pr})=([f_{ms}-\bar{f}_s]-[f_{ps}-\bar{f}_s])([f_{mr}-\bar{f}_r]-[f_{pr}-\bar{f}_r]) \tag{C.29}$$

Using Equations C.29 and C.17, Equation C.28 becomes

$$t_{sr} = \frac{M_1}{(M_1-1)} \left\{ \overline{[f_s-\bar{f}_s][f_r-\bar{f}_r]} - \overline{[f_s-\bar{f}_s]}\,\overline{[f_r-\bar{f}_r]} - \overline{[f_s-\bar{f}_s]}\,\overline{[f_r-\bar{f}_r]} + \overline{[f_s-\bar{f}_s][f_r-\bar{f}_r]} \right\} \tag{C.30}$$

Using Equation C.23, Equation C.30 becomes

$$t_{sr} = \frac{M_1}{(M_1-1)} \; 2\,\overline{[f_s-\bar{f}_s][f_r-\bar{f}_r]} \tag{C.31}$$

$$T = \frac{M_1}{(M_1-1)} \; 2\,F\,(\bar{f}) \tag{C.32}$$

Thus, using Equations C.32 and C.27,

$$XT^{-1} \alpha \,[F(\bar{f}) + G(\bar{f})]\; F^{-1}(\bar{f}) \tag{C.33}$$

$$= I + G(\bar{f})\; F^{-1}(\bar{f}) \tag{C.34}$$

Thus the eigenvectors $\vec{w}$ and eigenvalues λ of XT^{-1} can be obtained from the eigenvectors and eigenvalues of $G(\bar{f})\; F^{-1}(\bar{f})$. $G(\bar{f})$ is the covariance matrix in the twenty-feature space of the arrays of class g calculated about the mean of the arrays of class f. $F^{-1}(\bar{f})$ is the inverse of the covariance matrix of the arrays of class f calculated about the mean of the arrays of class f.

Throughout this section the class-separating transformations were developed by reference to the existence of two sets, $\{\vec{f}_m\}$ and $\{\vec{g}_p\}$. The results obtained by these methods are more general, however, because they apply directly to the separation of an arbitrary number of sets. For example, in the maximization of the mean-square interset distance, there is no reason why the matrix X should involve interset distances between only two sets. An arbitrary number of sets may be involved, and the interset distances are simply all those distances measured between two points not in the same set. Similar arguments are valid for all the other matrices involved. The only precaution that must be taken concerns the possible use of additional constraints specifying preferential or nonpreferential treatment of classes. These additional constraints may require, for instance, that the mean square intraset distance of all sets be equal or be related to each other by constants. Aside from these minor considerations, the results apply to the separation of any number of classes.

The eigenvectors of $G(\overline{f})\ F^{-1}(\overline{f})$ are needed to obtain the transformation to the three best class-separating generalized features in the twenty-feature space. The eigenvectors are obtained as follows.

Theorem: If $G(\overline{f})$ and $F(\overline{f})$ are symmetric, positive definite matrices, then there exists a transformation W such that

(a) $$WF(\overline{f})\ W^T = I, \tag{C.35}$$

where I is the identity matrix;

(b) $$WG(\overline{f})\ W^T = D \tag{C.36}$$

where D is a diagonal matrix

and

(c) $$W(G(\overline{f})\ F^{-1}(\overline{f}))\ W^{-1} = D \tag{C.37}$$

Since Equation C.37 is a similarity transform of the matrix GF^{-1}, D (which is the same diagonal matrix as in (b)) has diagonal elements which are the eigenvalues of GF^{-1} and the rows of W are the eigenvectors of GF^{-1}. W is constructed as follows: Let H be the similarity transform for F.

$$HF(\overline{f})\, H^{-1} = D_1 \tag{C.38}$$

where D_1 is the diagonal matrix with diagonal elements equal to the eigenvalues of F. Since F is symmetric

$$H^{-1} = H^T \tag{C.39}$$

Thus

$$HF(\overline{f})\, H^T = D_1 \tag{C.40}$$

Since the eigenvalues of F are positive definite,

$$D_1^{-\frac{1}{2}}\, HF(\overline{f})\, H^T D_1^{-\frac{1}{2}} = I \tag{C.41}$$

Apply this same operator to G to define $\overline{G}$,

$$D_1^{-\frac{1}{2}}\, HG(\overline{f})\, H^T\, D_1^{-\frac{1}{2}} \equiv \overline{G}(\overline{f}) \tag{C.42}$$

Let M be the similarity transform for $\overline{G}$. (C.43)

Since $\overline{G}$ is symmetric

$$M^{-1} = M^T \tag{C.44}$$

Thus

$$M\overline{G}(\overline{f})\, M^T = D \tag{C.45}$$

where the elements of the diagonal matrix D are the eigenvalues of $\overline{G}$. The matrix W which has the properties given in Equations C.35, C.36, and C.37 is constructed as

$$W = MD_1^{-\frac{1}{2}} H \tag{C.46}$$

To show that Equation C.35 is true,

$$WF(\overline{f})\, W^T = MD_1^{-\frac{1}{2}}\, HF(\overline{f})\, H^T\, D_1^{-\frac{1}{2}}\, M^T \tag{C.47}$$

Using Equations C.41 and C.44

$$WF(\overline{f})\, W^T = I \tag{C.48}$$

Thus Equation C.35 is true when W is given by Equation C.46.

To show that Equation C.36 is true,

$$WG(\overline{f})\, W^T = MD_1^{-\frac{1}{2}}\, HG(\overline{f})\, H^T\, D_1^{-\frac{1}{2}}\, M^T \tag{C.49}$$

Using Equations C.42 and C.43

$$WG(\overline{f})\, W^T = D \tag{C.50}$$

Thus Equation C.36 is true when W is given by Equation C.46.

To show that Equation C.37 is true, multiply Equation C.35 and C.36 together.

$$WG(\overline{f})\, W^T\, WF(\overline{f})\, W^T = D \tag{C.51}$$

From Equation C.35

$$F(\overline{f})\, W^T = W^{-1} \tag{C.52}$$

$$F(\overline{f}) = W^{-1}(W^T)^{-1} = (W^TW)^{-1} \tag{C.53}$$

$$F^{-1}(\overline{f}) = W^TW \tag{C.54}$$

Using Equation C.52 in Equation C.51,

$$WG(\overline{f})\ W^T\ W\ W^{-1} = D \tag{C.55}$$

Using Equation C.52 in Equation C.53,

$$WG(\overline{f})\ F^{-1}\ (\overline{f})\ W^{-1} = D \tag{C.56}$$

Therefore Equation C.37 is true. Since Equation C.56 is a similarity transform of the matrix GF^{-1}, the diagonal elements of D are the eigenvalues of GF^{-1}, and the rows of W are the eigenvectors of GF^{-1}. These eigenvectors of GF^{-1} are not orthonormal, since

$$W^T \neq W^{-1} \tag{C.57}$$

so that

$$WW^T \neq I \tag{C.58}$$

Equation C.57 is obtained from Equation C.46, C.44, and C.39, where D_1 is the diagonal matrix with diagonal elements equal to the eigenvalues of F, as given by Equation C.38.

In summary, W is the transformation matrix which maps a twenty-feature vector of an array into the generalized feature space, the best three of which will be used to classify the array. W, constructed as in Equation C.46, has the properties given by Equations C.35, C.36, and C.37.

APPENDIX D

PARAMETERIZATION OF PROBABILITY DENSITIES

The class-conditional probability densities for the arrays in the two classes f and g are modeled as multi-variate normal distributions in the n-dimensional feature space, with n = 20. For the class f, the probability density of obtaining the twenty-feature measurement X is

$$P(X|f) = \frac{1}{(2\pi)^{n/2}} |F^{-1}(\bar{f})|^{\frac{1}{2}} \exp\{-\tfrac{1}{2}(X-\bar{f})^T F^{-1}(\bar{f})(X-\bar{f})\} \qquad (D.1)$$

$F^{-1}(\bar{f})$ is the inverse of the twenty-feature covariance matrix of the class f about the twenty-feature mean, $\bar{f}$, of class f. $|F(\bar{f})|$ is the determinant of $F(\bar{f})$. $(X-\bar{f})$ is the twenty-feature measurement relative to the mean of class f.

For the class g, the probability density of obtaining the twenty-feature measurement X is

$$P(X|g) = \frac{1}{(2\pi)^{n/2}} |G^{-1}(\bar{g})|^{\frac{1}{2}} \exp\{-\tfrac{1}{2}(X-\bar{g})^T G^{-1}(\bar{g})(X-\bar{g})\} \qquad (D.2)$$

The class-conditional probability densities in the best generalized feature space are found as follows. Equation D.2 involves the quadratic from

$$(X-\bar{f})^T F^{-1}(\bar{f})(X-\bar{f}) \equiv y^T F^{-1}(\bar{f})y \qquad (D.3)$$

where $y \equiv (X-\bar{f})$ by definition. From Equation C.1 the transform W takes a feature vector from the twenty-feature space to the generalized feature space.

$$y' = Wy \qquad (D.4)$$

Thus

$$y = W^{-1}y' \tag{D.5}$$

$$y^T = (y')^T(W^{-1})^T \tag{D.6}$$

The right side of Equation D.3 becomes

$$\begin{aligned} y^TF^{-1}\,(\overline{f})\;y &= (y')^T(W^{-1})^TF^{-1}(\overline{f})W^{-1}y' \\ &= (y')^T(WF(\overline{f})W^T)^{-1}\;y' \end{aligned} \tag{D.7}$$

Thus in the generalized feature space

$$F'^{-1}(\overline{f}) = (WF(\overline{f})\;W^T)^{-1} \tag{D.8}$$

in order that the quadratic forms in the generalized and twenty-feature spaces are equal. Using Equation C.35

$$F'^{-1}(\overline{f}) = I \tag{D.9}$$

so that

$$|F'^{-1}(\overline{f})| = 1 \tag{D.10}$$

Using Equations D.9 and D.8 in Equation D.7

$$y^TF^{-1}(\overline{f})\;y = (y')^T\;Iy' \tag{D.11}$$

Substituting for y

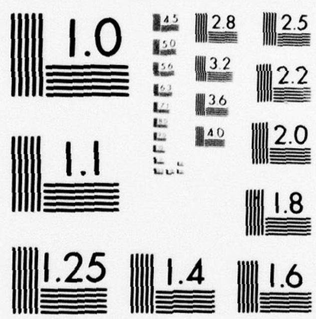
$$\begin{aligned} (X-\overline{f})^T\;F^{-1}(\overline{f})(X-\overline{f}) &= (X'-\overline{f}')^T\;I(X'-\overline{f}') \\ &= (X'-\overline{f}')\;(X'-\overline{f}') \end{aligned} \tag{D.12}$$

AD-A060 039 CONTROL DATA CORP MINNEAPOLIS MINN DIGITAL IMAGE SYS--ETC F/G 17/9
STATISTICAL EVALUATION OF FOP FEATURES.(U)
JUN 78 L KIRVIDA, D HANSON, L BAUMANN F30602-76-C-0349

UNCLASSIFIED RADC-TR-78-108 NL

2 OF 2
AD
A060039

END
DATE
FILMED
12-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Substituting Equation D·12 and D·10 into Equation D·1,

$$P(X' | f) = \frac{1}{(2\pi)^{n/2}} \exp \{ -\tfrac{1}{2}(X'-\overline{f}')^T (X'-\overline{f}') \} \tag{D·13}$$

The determination of Equation D.2 in the generalized feature space proceeds similarly. As in Equation D.8

$$G'^{-1}(\overline{g}) = (WG(\overline{g})\, W^T)^{-1} \tag{D·14}$$

However, Equation D.14 involves $G(\overline{g})$, rather than $G(\overline{f})$, which is used in Equation C.36. However

$$G_{Sr}(\overline{f}) \equiv \overline{(g_S-\overline{f}_S)(g_r-\overline{f}_r)}$$

$$= \overline{([g_S-\overline{g}_S] - [f_S-\overline{g}_S])([g_r-\overline{g}_r] - [f_r-\overline{g}_r])}$$

$$= \overline{[g_S-\overline{g}_S]\, [g_r-\overline{g}_r]} - \overline{[g_S-\overline{g}_S]}\, [\overline{f}_r-\overline{g}_r]$$

$$-[\overline{f}_S-\overline{g}_S]\, \overline{[g_r-\overline{g}_r]} + [\overline{g}_S-\overline{f}_S]\, [\overline{g}_r-\overline{f}_r]$$

$$G_{Sr}(\overline{f}) = G_{Sr}(\overline{g}) + \overline{G}_{Sr}(\overline{f}) \tag{D·15}$$

where Equation C·23 has been used, and $\overline{G}_{Sr}(\overline{f})$ is the covariance of the mean of class $g,\overline{g}$, about the mean of class $f,\overline{f}$. In matrix notation Equation D.15 becomes

$$G(\overline{f}) = G(\overline{g}) + \overline{G}(\overline{f}) \tag{D.16}$$

Thus

$$G(\overline{g}) = G(\overline{f}) - G(\overline{f}) \tag{D.17}$$

so that Equation D.14 becomes

$$G'^{-1}(\bar{g}) = (W[G(\bar{f}) - \bar{G}(\bar{f})]\, W^T)^{-1} \tag{D.18}$$

From Equation C.36

$$G'^{-1}(\bar{g}) = (D-S)^{-1} \tag{D.19}$$

where S is defined to be

$$S \equiv W\bar{G}(\bar{f})\, W^T \tag{D.20}$$

and where, from Equation C.37, D is a diagonal matrix with diagonal elements equal to the eigenvalues of $G(\bar{f})\, F^{-1}(\bar{f})$.

Thus using Equation D.19 in Equation D.2,

$$P(X'|\, g) = \frac{1}{(2\pi)^{n/2}} \left| (D-S)^{-1} \right|^{\frac{1}{2}} \exp \left\{ -\tfrac{1}{2}(X'-\bar{g}')^T (D-S)^{-1} (X'-\bar{g}') \right\} . \tag{D.21}$$

Thus Equations D.13 and D.21 are the class-conditional probability density functions for classes f and d, respectively.

Equation D.13 gives the probability density $P(X'|f)$ that the generalized feature vector X' will be obtained for an array in class f. Equation D.21 gives the probability density $P(X'|\, g)$ that X' will be obtained for an array in class g. However, the problem to be addressed in automatic classification is: given that an array of <u>unknown</u> classification has features locating a vector X' in generalized feature space, what is the probability $P(f|X')$ that the array belongs to class f and the probability $P(g|X')$ that it belongs to class g? Bayes's Theorem states that

$$P(f|X') = \frac{q(f)\, P(X'|\, f)}{q(f)\, P(X'|\, f) + q(g)\, P(X'|\, g)} \tag{D.22}$$

$$P(g \mid X') = \frac{q(g)\ P(X' \mid g)}{q(g)\ P(X' \mid g) + q(f)\ P(X' \mid f)} \tag{D.23}$$

q(f) and q(g) are the a priori probabilities that an array belongs to class f and class g, respectively. Thus, an array of unknown classification has probability q(f) of being a class f array. However, when the twenty features X are measured and (through the transformation W) the generalized feature X' determined for the array, this additional information about the array is used to calculate the probability $P(f \mid X')$ that an array of unknown classification with generalized features X' belongs to class f. The same is true for class g.

An array of unknown classification with generalized features X' is classified as an f-class array if $P(f \mid X') > P(g \mid X')$. Otherwise, it is classified as a g-class array.

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

**Printed by
United States Air Force
Hanscom AFB, Mass. 01731**