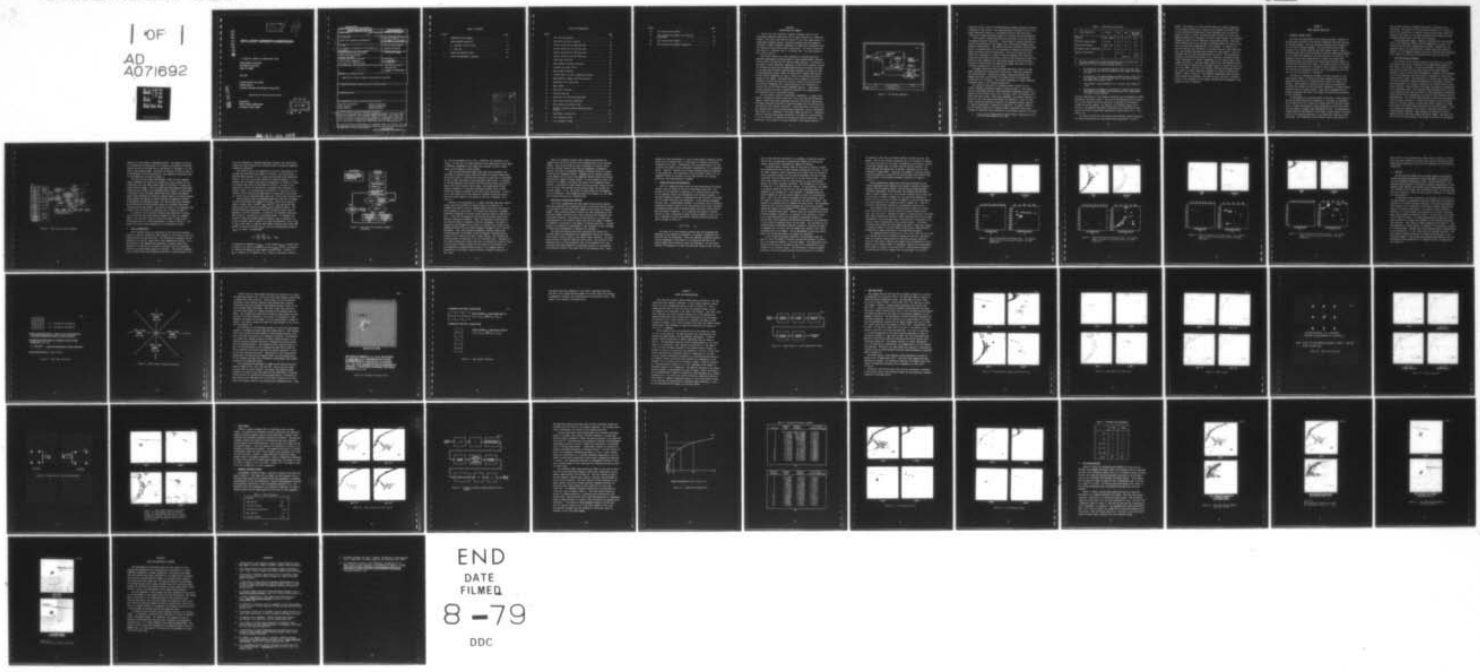
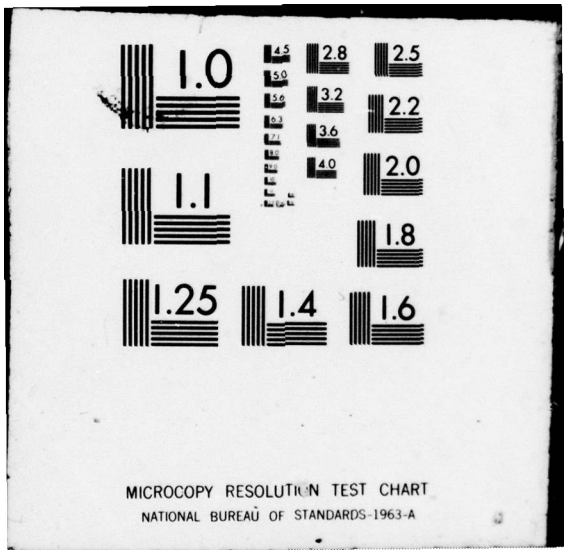
AD-A071 692 HUGHES RESEARCH LABS MALIBU CALIF F/G 17/2
INTELLIGENT BANDWIDTH COMPRESSION.(U)
JUN 79 D Y TSENG, B L BULLOCK, J D OLSEN DAAK70-78-C-0148
UNCLASSIFIED NL



END
DATE FILMED
8-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

INTELLIGENT BANDWIDTH COMPRESSION

D. Y. Tseng, B. L. Bullock, J. D. Olsen, and K. E. Olin

Hughes Research Laboratories
3011 Malibu Canyon Road
Malibu, CA 90265

June 1979

Contract DAAK70–78–C–0148
Quarterly Report 1
For period 1 November 1978 through 31 January 1979

Approved for public release; distribution unlimited.

Sponsored by
NIGHT VISION LABORATORY
Fort Belvoir, Virginia 22060

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

INTELLIGENT BANDWIDTH COMPRESSION

5. TYPE OF REPORT & PERIOD COVERED

Quarterly Report, Interim no. 1,
1 Nov 1978 - 31 Jan 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

D.Y. Tseng, B.L. Bullock, J.D. Olsen, and K.E. Olin

8. CONTRACT OR GRANT NUMBER(s)

DAAK70-78-C-0148

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Hughes Research Laboratories
3011 Malibu Canyon Road
Malibu, CA 90265

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

NIGHT VISION LABORATORY
Fort Belvoir, Virginia 22060

12. REPORT DATE

June 1979

13. NUMBER OF PAGES

39

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

58p.

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

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

18. SUPPLEMENTARY NOTES

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

Bandwidth compression	Feature extraction
Smart sensors	Pattern recognition
Image analysis	Target detection

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

This is the first quarterly technical progress report for the Intelligent Bandwidth Compression (IBC) program (DAAK70-78-C-0148). The report describes the results of the first three months of a seven-month technical effort to develop techniques and algorithms for achieving a 1,000:1 bandwidth compression in image data transmission rate for RPV applications.

The period covered by this report is 1 November 1978 to 31 January 1979.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

172 600 LB

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SECTION 1

INTRODUCTION AND SUMMARY

This is the first quarterly technical progress report for the Intelligent Bandwidth Compression (IBC) Program (Contract No. DAAK70-78-C-0148). The report describes the results of the first three months of a seven-month technical effort to develop techniques and algorithms for achieving a 1,000:1 bandwidth compression in image data transmission rate for remotely piloted vehicle (RPV) applications. The period covered by this report is 1 November 1978 to 31 January 1979. Mr. David Singer is the contract monitor.

The overall objective of this program is to develop high-ratio bandwidth-compression techniques based on image understanding, advanced scene analysis, and spatial and temporal image sampling. Our approach to achieving a 1,000:1 bandwidth compression of RPV derived imagery utilizes the information-extraction and image-understanding techniques developed in the Army ATAC and DARPA Terminal Homing programs in conjunction with a knowledge-based control scheme developed under a Hughes IR&D program. Scene-analysis techniques are used to extract key features and detect targets in the images. Knowledge-based control schemes are applied to adaptively evaluate and match the available targets with mission goals for priority assignments and to make decisions on the allocation of resources in the available channel bandwidth capacity. Conventional coding techniques are used to encode the resources for transmission to, and reconstruction by, the ground station.

The IBC system consists of three major components: an image interpreter, an adaptive priority controller, and a data link encoder/decoder. Figure 1 shows these major components and their interactive connections. The image interpreter operates on digitized images from the IBC data base and performs three primary operations: target detection, statistical target identification, and structural/feature analysis. Control information about where (i.e., addresses in the input image) and how (i.e., specifications of features and analysis procedures to be used) the interpretation is to be performed is supplied by the adaptive priority controller. This particular control path is the most important in the system because it

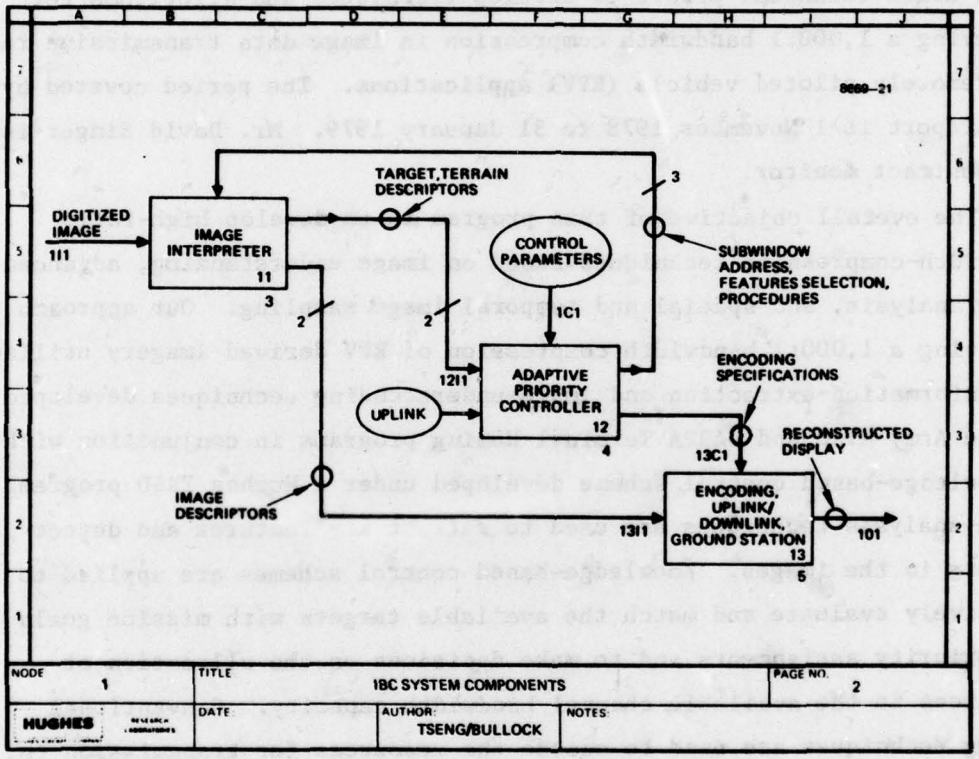


Figure 1. IBC system components.

closes the control loop on the interpretation system, allowing interactive refinement of the results. The image interpreter provides a list of the target descriptors that go to the adaptive priority controller and a collection of image descriptors that go to the data link encoder/decoder.

Three types of target descriptors are contained in the list sent to the adaptive priority controller: a cued target at a given location, an initial classification of the target, and a structurally refined scene-analysis classification. The adaptive priority controller utilizes these inputs to determine what type of processing is necessary and what the image interpreter should be doing next. The control policy for this decision is contained in a table of control parameters that lists the target and scene object priorities as well as the allocations allowed for different scene components for the encoded signals to be transmitted. Through the uplink input, the operator can change this table of control parameters dynamically to suit his needs.

The image descriptors from the image interpreter consist of edge image subwindows and target/object descriptors. These are encoded by the encoder/decoder, transmitted through a simulated downlink (with jamming and noise), and decoded for reconstruction in the simulated ground station for display. This display represents a 1,000:1 bandwidth compressed image of the original sensed scene.

The resource allocation for available channel capacity in the 1,000:1 bandwidth compression scheme is shown in Table 1. The channel capacity requirement used for image-compression comparison is based on an image data rate derived from a 512 x 512 pixel, 8-bit, 30 frames-per-sec (fps) video data stream. Under this assumption, the channel capacity required for full rate image transmission is 63 Mbit/sec. Thus, for the 1,000:1 bandwidth compression required, the available channel capacity for image transmission is reduced to 63 kbit/sec. As shown in Table 1, the IBC program produces the required 1,000:1 compression while maintaining all essential scene information by allocating resources as follows:

- A 32 x 32 pixel high-priority target window transmitted at full resolution and updated at 7.5 fps.

Table 1. IBC Resource Allocation

Data Composition	Pixels	FPS	BPP	Data Rate kbit/sec
Edge map	256 x 256[a]	1.0	1.0	16.4
Priority target window	32 x 32	7.5	1.5	11.5
Orientation window	128 x 128	1.0	1.0	16.4
Secondary target windows(6)	32 x 32	1.0	1.5	9.2
Symbolic Descriptors	-	1.0	-	9.5
			Total	63.0

a. Edge map transmitted at reduced resolution (128 x 128 pixel) and replicated to 256 x 256 size at ground station.

- An orientation (or reference) window of 128 x 128 pixel size centered about the detected (high-priority) targets and updated at 1 fps.
- An edge map of the image reduced in resolution by a factor of 2 and replicated to 256 x 256 pixel size at the ground station. The update rate is also 1 fps.
- Six secondary target windows of 32 x 32 pixel size updated at 1 fps.
- Approximately 200 symbolic descriptors to denote scene content information of interest, both supplementing and complementing the above stated resources.

The exact composition of the resource allocation can be dynamically changed depending on the scene content. This can be accomplished either automatically by the adaptive priority controller or manually by the operator through the uplink command. For example, the secondary targets and a portion of the symbolic descriptor allocations can (if desired) be redirected to provide an additional high-priority target window at a 7.5 fps update rate.

Our efforts during the first quarter were directed toward developing the image-interpretation and encoding/decoding components of the IBC

system. The purpose is to first develop these two integral components of the system and then also to gain an early insight into which format of the reconstructed image would be most desirable from the operator/observer's point of view. The IBC data base—32 frames of visible band images divided into 10 sequences of distinct scene content—was processed through the ATAC auto-cuer simulation (algorithm) software. With some minor modifications in the low-level processing stage, the auto-cuer was able to produce good segmentations of the targets. In addition, the IBC data base was also processed by the edge-extraction (algorithm) software, and dominant edge maps were produced for these images. Section 2 discusses the auto-cuer and edge-extraction results. The encoding and decoding of the allocated resources (e.g. edge map, orientation window, target subwindows) for 1,000:1 compression of the original image is discussed in Section 3. Edge map coding, DPCM coding, and Hadamard coding of the allocated resources were performed subject to various levels of bit error rate (BER). Section 3 also includes reconstructed images showing the results of 1,000:1 bandwidth compression on typical IBC data base imagery. Several versions of the bandwidth compressed images using different formats are discussed. The final section, Section 4, discusses plans for the remainder of the program, including the development of the adaptive priority controller.

SECTION 2

SCENE CONTENT EXTRACTION

A. AUTOMATIC TARGET CUEING

The auto-cuer incorporated in the image interpreter component of the IBC system consists of the software simulation algorithm developed in the ATAC program. This auto-cuer had originally been designed for use with FLIR (infrared) imagery. The minor modifications needed (because of the visible band IBC data base imagery used) to adapt it involved mainly the low-level processor, where new weighting parameters for the linear discriminant of the auto-cuer had to be selected. The weighting factors were redistributed to reflect the better edge-definition targets contained in the visible band data base imagery as compared with the ATAC FLIR imagery. The salient features of the ATAC auto-cuer are presented here. Automatic target cueing of FLIR imagery has received much attention recently, and a representative literature list is given in References 1 through 12.

The auto-cuer simulation utilizes a two-level approach to automatic target cueing. A low-level processor examines the full-frame image and selects points of interest where targets are likely to be located. These points of interest are then sent to the high-level processor for further, and more detailed, examination in window regions centered about the interest points. Here, object segmentation and feature extraction take place, and targets are detected and identified by the classifier.

In determining the low-level points of interest, a small number of easily calculated statistical parameters of the image are considered. The motivation here is to form, by a proper choice of low-level parameters, a linear discriminant of these parameters so that target areas are more likely to possess high discriminant values. By appropriately thresholding the discriminant at this stage, a small number of interest points are produced and passed on to the high-level stage for further processing. The remaining, below threshold, points sufficiently far

from interest points are excluded from further consideration and processing. The objective of the low-level processor is to locate as many of the target/areas as possible using interest points, while discarding as many areas of the image as possible from complex and time-consuming calculations by the high level processor.

At the high-level stage, a window is centered about each interest point in the image and objects from the background, or clutter, are extracted with an object segmentation algorithm. Subsequently, all segmented objects are characterized by several high-level features. Finally, all the segmented objects are sent to the high-level classifier, where targets are detected and identified and clutter is rejected.

1. Low-Level Interest Operator

The function of the low-level stage of the auto-cuer is to select and isolate those points of interest in the image where targets are most likely to be located. Low-level processing, as defined here, is the process by which each 5x5 pixel region of the shrunk image is assigned a value, via the linear discriminant, intended to be representative of the probability that this region is partially or totally contained within a target. The objective is to utilize only a small number of easily calculated, statistical parameters of the image to extract a high percentage of the targets, by means of the derived interest points, while excluding a large portion of the image from further consideration by the high-level stage. The selection of the features used in the low-level processing were chosen not only for their usefulness in differentiating targets from the background, but also for their potential ease of implementation. The low-level features used in this auto-cuer simulation study are shown in Figure 2. As the figure shows, a local window consists of a 5x5 pixel region, a large local window consists of a 60x60 pixel region, and global values are taken over the entire image. These ten features are combined by the scalar to form eight contrast- and offset-invariant features. These features are then used to form a weighted sum for each small window (5x5 pixel) region. The small windows of the low-level processor can be made to slide in an overlapping

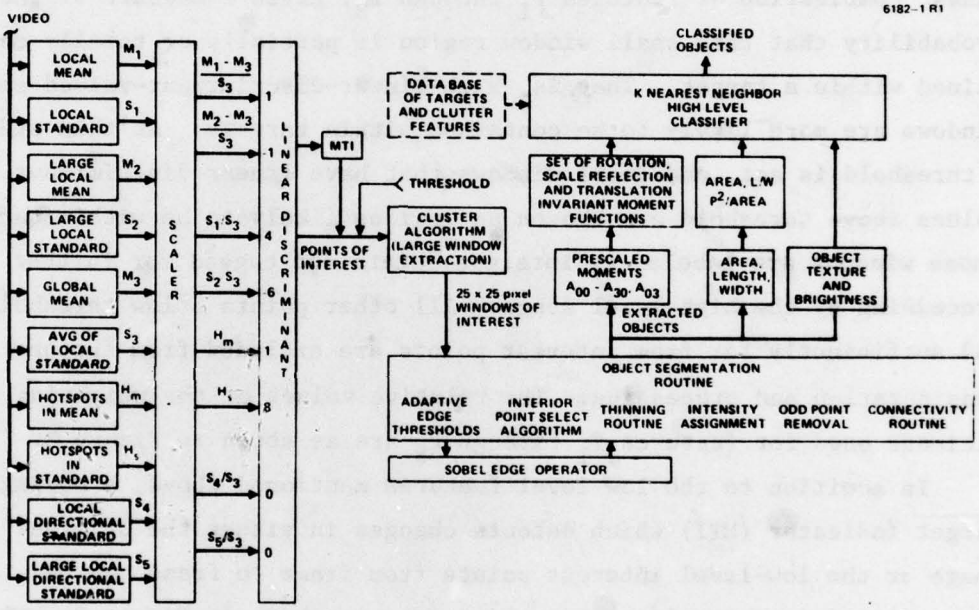


Figure 2. ATAC auto-cuer block diagram.

fashion or can be taken in contiguous sectors. The degree of overlap is variable and can range from 1 to 5 pixels, resulting in a low-level feature picture of the same size or one shrunk to a maximum of 1/5 the size of the original image. The degree of overlap can also be taken to be independent in the two directions. The weighting coefficients used in our simulations are shown in Figure 2.

The value of the linear discriminant, formed from the weighted linear combination of features F_1 through F_8, gives a measure of the probability that this small window region is partially or totally contained within a target. That is, high-linear-discriminant-valued small windows are more likely to be contained within targets. At this stage, a threshold is set, and small windows that have linear discriminant values above threshold are chosen as regions likely to be within targets. Those windows are labeled as interest points and tagged for further processing by the high-level stage. All other points below threshold and sufficiently far from interest points are excluded from further consideration and processing. The relative values of the weighting coefficients used for features F_1 through F_8 are as shown in Figure 2.

In addition to the low-level features mentioned above, a moving target indicator (MTI) which detects changes in either the original image or the low-level interest points from frame to frame can be incorporated into the linear discriminant, as shown in Figure 2. This feature has not yet been implemented in the simulation studies.

2. Object Segmentation

After interest points are identified by the low-level processor, regions of interest (defined by large windows 25 x 25 pixels in area) are centered about the interest points. The extraction of targets from the background, or clutter, takes place in the segmentation routine. It is the function of the segmentation algorithm to separate those areas believed to be interior to an object of interest from the background enclosed by a large window (region of interest). The accurate segmentation of objects from the background constitutes a critical stage of the

auto-cuer simulation. Reliable high-level features that characterize target and clutter objects can be derived only if accurately segmented targets are extracted.

The initial step in the segmentation routine is the generation of an edge value map within the region of interest. For this purpose, a modified Sobel operator of 3 x 3 pixel size was used in the simulation studies. The form of the Sobel operator is shown in Figure 8. It is assumed that the (Sobel) edge values corresponding to targets are among the largest edge values in the region of interest. As used in this algorithm, an object of interest is a contiguous area consisting of interior points. An interior point is defined as a pixel that is surrounded in most directions by edge points, an edge point being defined as a point with an associated edge value that is above some threshold value.

Edge threshold value is determined adaptively using the histogram of edge values within the region of interest. Based on the expected range of target sizes, the 80^{th} (E_{80}) and 95^{th} (E_{95}) percentile edge values within the windows are used as control end point values. The minimum percentile considered is the 80^{th} percentile value, and the maximum is the 95^{th} percentile. In addition, two absolute (arbitrary) edge values $T_1 = 100$ and $T_2 = 50$ are also chosen. To determine the threshold value for a given region of interest, E_{80} is compared to T_1. If E_{80} is greater than or equal to T_1, then the threshold is set to E_{80}. If E_{80} is less than T_1, then the next higher intensity value within the region of interest is picked (E_{next}), and its corresponding percentile (P_{next}) is noted. An expression

$$T_\Delta = \frac{(T_1 - T_2)}{P_{95} - P_{80}} (P_{next} - P_{80})$$

is computed and compared to E_{next}. If the current E_{next} is greater than or equal to T_Δ, then the threshold is set to E_{next}; if not, the process is repeated by picking the next higher intensity level. Finally, if E_{95} is reached, it is compared to T_2. If E_{95} is less than or equal to

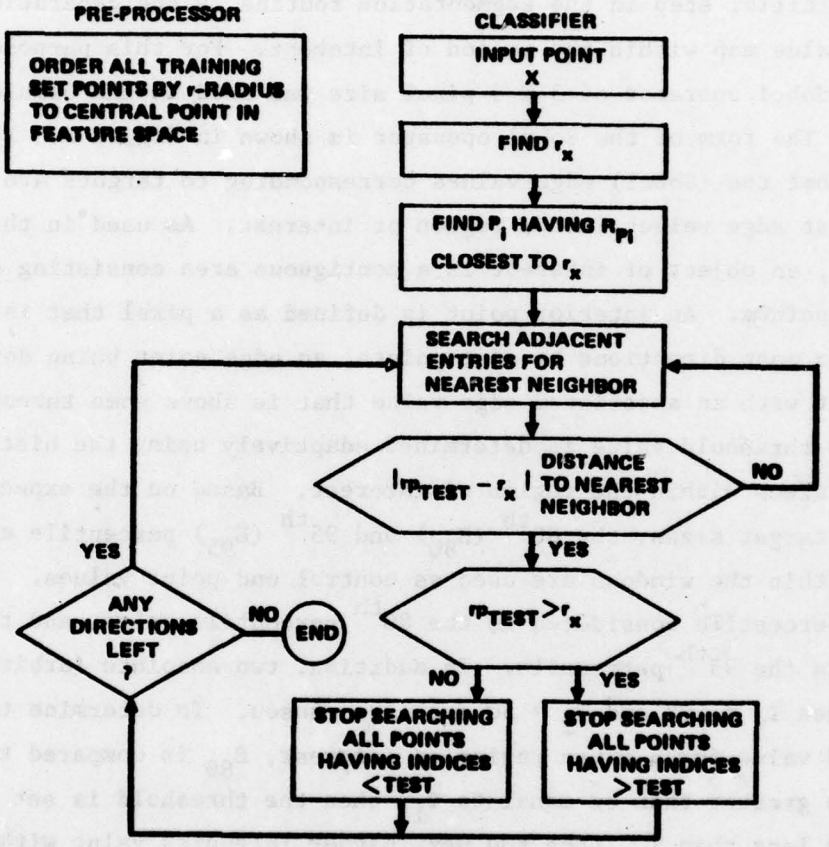


Figure 3. Flow chart for K-nearest-neighbor classifier.

T_2, then the threshold is set to T_2. Otherwise, the threshold is set to E_{95}. In this way, the edge threshold is set adaptively in each region of interest, depending on the target size, presence of other strong edges, and the amount of noise present.

Once the above-threshold edge points have been determined, the detection of interior points within the region of interest takes place. An interior point is (arbitrarily) defined as a pixel that is surrounded in six of eight directions by above-threshold edge points. A thinning or filling operation then removes isolated interior points and fills in gaps in the binary interior point image. The thinning and filling operator consists of a 3x3 pixel overlapping window, which is slid within the region of interest in both the x and y directions. If five out of nine pixels in the 3x3 window are occupied by interior points, then the center element of the window is set to 1; otherwise, it is set to 0.

Because of the coarseness of the eight-direction edge point search, concave and convex portions of objects tend to be filled in, or shadowed; the same is true for regions between two objects that are in close proximity to each other. To overcome this drawback, a maximum-likelihood assignment of interior versus exterior points is carried out next. To do this, histograms of the previously derived interior and exterior regions are tabulated. Then, the intensity of each pixel in the interest window is compared to the two histograms. The pixel in question is now assigned as an interior or exterior point based on the maximum likelihood that this pixel intensity belonged to one of the two histograms. After this intensity assignment, another thinning and filling operation is performed, again to eliminate isolated interior points or fill in gaps in the newly formed objects. The thinning and filling operator is identical to the one described above. Subsequent to the second thinning and filling operation, a connectivity routine is used to merge those interior regions that are likely to be connected, but were segmented as separate areas because of, for example, gaps or degradations in the object caused by noise or thermal differences in the target.

Next, the connected interior point regions thus derived are compared with the first interior point classification in the interest window. Those areas having the most coincidence of interior points with the original interior point map are designated as segmented objects, and their corresponding original intensity values are substituted into the segmented objects. Depending on the scenario expected in the images, one or more of the segmented objects within the interest window can be tagged as likely targets. In this simulation study, the area with the largest coincidence was chosen as the segmented object. Finally, a calculation is made (based on area and object center) to see if it is likely that portions of the desired target lie outside the interest window. If so, the window is repositioned and the segmentation is recomputed. These segmented objects are passed on to the high-level feature extractor for classification.

3. Extraction of High-Level Features

Once object segmentation has been completed and interior points have been separated from background clutter, a high-level feature vector is calculated to characterize each segmented object within an interest window. These high-level-feature vectors are used subsequently in the K-nearest-neighbor classifier for target detection and recognition and for clutter rejection. One purpose of the features is to reduce the dimensionality of the decision regions from the total number of degrees of freedom of the pattern to the number of features extracted. A second purpose is to improve the performance of the classifier by only allowing relevant pattern parameters to influence its training.

The high-level-feature-extraction algorithm first detects all interior points as determined by the segmentation routine and replaces each detected interior point by its original pixel intensity. All other points within the interest window are set to zero. The centroid of the segmented object is then determined, and a set of moments (up to and including third-order moments) is calculated relative to the centroid, with distances scaled to the square root of the extracted area. These

moments are then transformed to a set of seven moment transforms, M1-M7, which has the property that it is invariant to rotation, reflection, translation, and scale. Additionally, the calculated moments are used to derive the length, width, and aspect (length/width) of the segmented object. Finally, a measure of the object intensity against background is also generated. The collection of all the feature vectors for the segmented objects of the data base are then used for the training, detection, and classification of targets.

4. Target Detection and Recognition

After the segmented objects have all been characterized by feature vectors, they are ready for use in the high-level classifier. The function of the high-level classifier is to perform clutter rejection on the segmented objects and recognition on the detected targets. A K-nearest-neighbor technique is implemented in the high-level classifier for both target detection and recognition. In the K-nearest-neighbor decision rule, a predetermined constant K is selected, and the K-nearest training set feature vectors to the test feature vector $\underline{X}$ are collected. A Euclidean distance measure between feature vectors is used to determine the nearest neighbors. If n_i represents the number of feature vectors of class i contained in these K nearest neighbors, and C_i is a set of constants (one for each class), then the feature vector $\underline{X}$ is said to belong to class j if

$$c_j n_j \geq C_i n_i \qquad \forall_i \quad .$$

The choice of a test procedure requires that the advantages and drawbacks of each method be considered. In general, recognition systems fall into two categories: those that perform recognition based on a priori knowledge of the objects to be recognized and those that require sample data on which to train the classifier. For the latter category of recognizers, of which the auto-cuer system is one, the best

way to train and test the system is to assemble a large data base and then to use a statistically representative sample for training and another statistically representative sample for testing.

A problem arises, however, when the data base is limited, because the available data must be split between training and test sets. On the one hand, choosing a large training set and a small test set results in better classifier design, but the tested performance will have a large uncertainty interval associated with it, or may even be wrong. On the other hand, making the test set large results in a good performance measure of a poorly designed classifier. This dilemma may be resolved by using the U, or leave-one-out, method. In this approach, one sample is initially chosen for the test set, and the remaining samples are assigned to the training set. The classifier is designed using the large training set, and the removed sample is used for testing performance. Next, the test sample is assigned to the training set, and a new sample is removed from the training set and used for testing. This procedure is repeated until all samples have been tested. This procedure results in both a better classifier design and a more statistically significant measure of performance.[13,14] In general, it requires redesigning the classifier many times. However, when the classifier used is the K-nearest-neighbor classifier, no redesign of the classifier is necessary because classification is performed by measuring distances in feature space and determining classification according to the population of the classes of neighbors about the test point. The robustness of the classifier may be determined using the U method by counting the number of neighbors out of K about the test sample that belong to the correct classification of that sample. In our measurement of auto-cuer performance, the U method was used as our basic test procedure.

In our K-nearest-neighbor classifier, a preprocessor stage was used to reduce the number of distance calculations necessary to determine the K nearest neighbors of the test sample. The preprocessor first calculates the radii of all sample points in the classifier data base relative to an arbitrary origin. The calculation of nearest-neighbor distances to the test sample proceeds until the radius of a training

set sample is less than the distance between the sample and the test sample. Then by the triangle inequality rule, all training set samples with radii less than the one under consideration will produce a distance to the test sample that is larger than the current distance. So these training set samples can be ignored. Similarly for the other direction of larger radii. Typically, the number of distance calculations was reduced by a factor of five for our runs, from 1,000 to 200. A flow diagram of the K-nearest-neighbor classifier is shown in Figure 3.

Some representative images which have been processed for the 1,000:1 compression ratio simulation are shown in Figures 4 through 7 together with the results derived from the auto-cuer and edge-map-extraction algorithms. In each figure, the original IBC data base image is shown in segment (a), the low-level interest point output from the auto-cuer is shown in segment (b), the high-level segmentation output from the auto-cuer is shown in segment (c), and the edge map of key scene contents in the images is shown in segment (d). (The discussion of edge-element extraction is presented in Section 2.B.)

These figures show that the interest points isolate the targets quite accurately in the low-level auto-cuer outputs (Figures 4(b), 5(b), 6(b), and 7(b)). Although the number of interest points could have been reduced somewhat by imposing a higher threshold value on the linear discriminant, this would have been at the expense of losing some of the interest points corresponding to low contrast targets. The linear threshold value chosen for these images represents a compromise between a reasonable number of interest points selected and a high extraction probability of target locations for the high-level processor.

The segmented objects (Figures 4(c), 5(c), 6(c), and 7(c)) are potential targets that must be identified through the auto-cuer classifier. These are normally characterized by unique and prominent features, which are then used to separate the targets from the background clutter. Because of the limited number of independent images and correspondingly small number of targets in this data base, a statistically meaningful classification of all the targets has not been made at this point.

8751-7

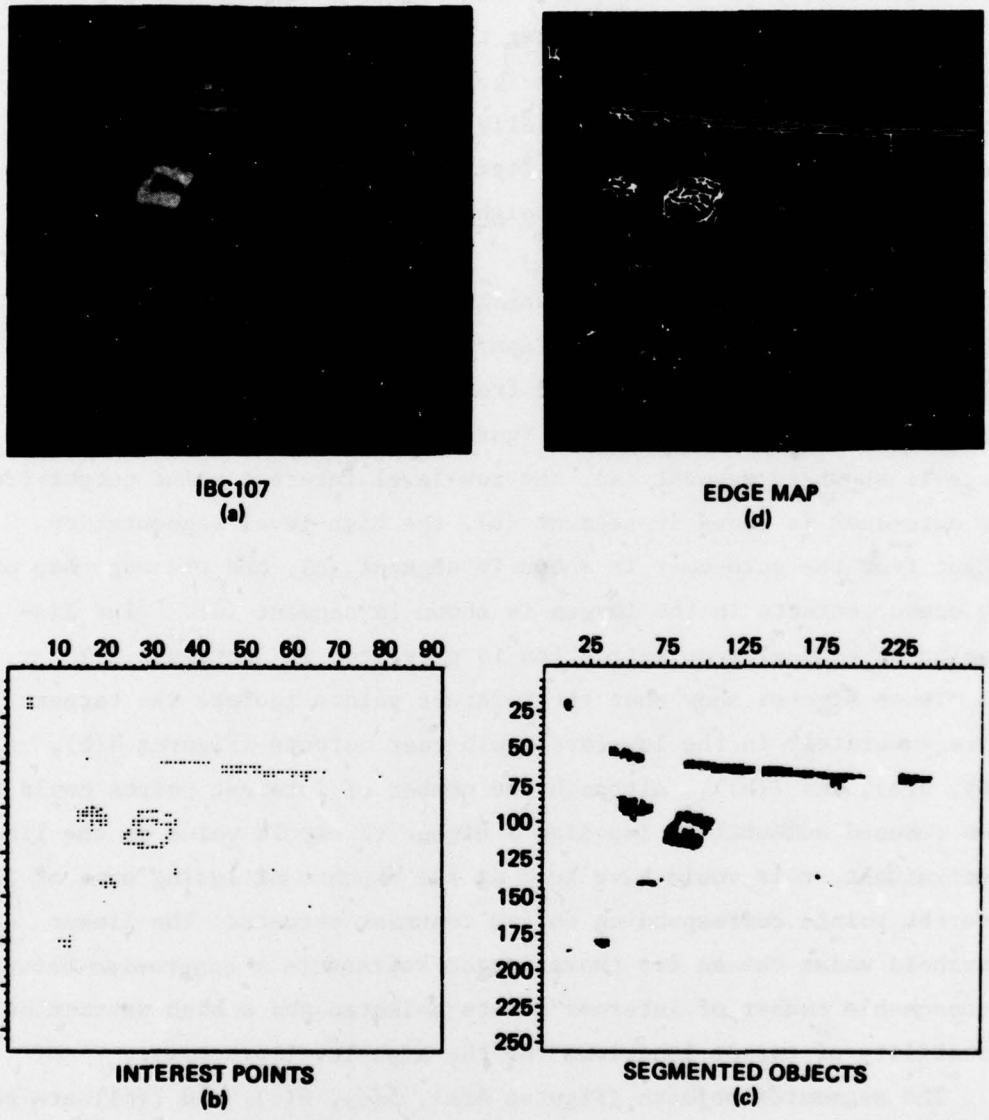


Figure 4. Feature extraction on IBC data base: (a) original image, (b) and (c) auto-cuer results, (d) edge extraction.

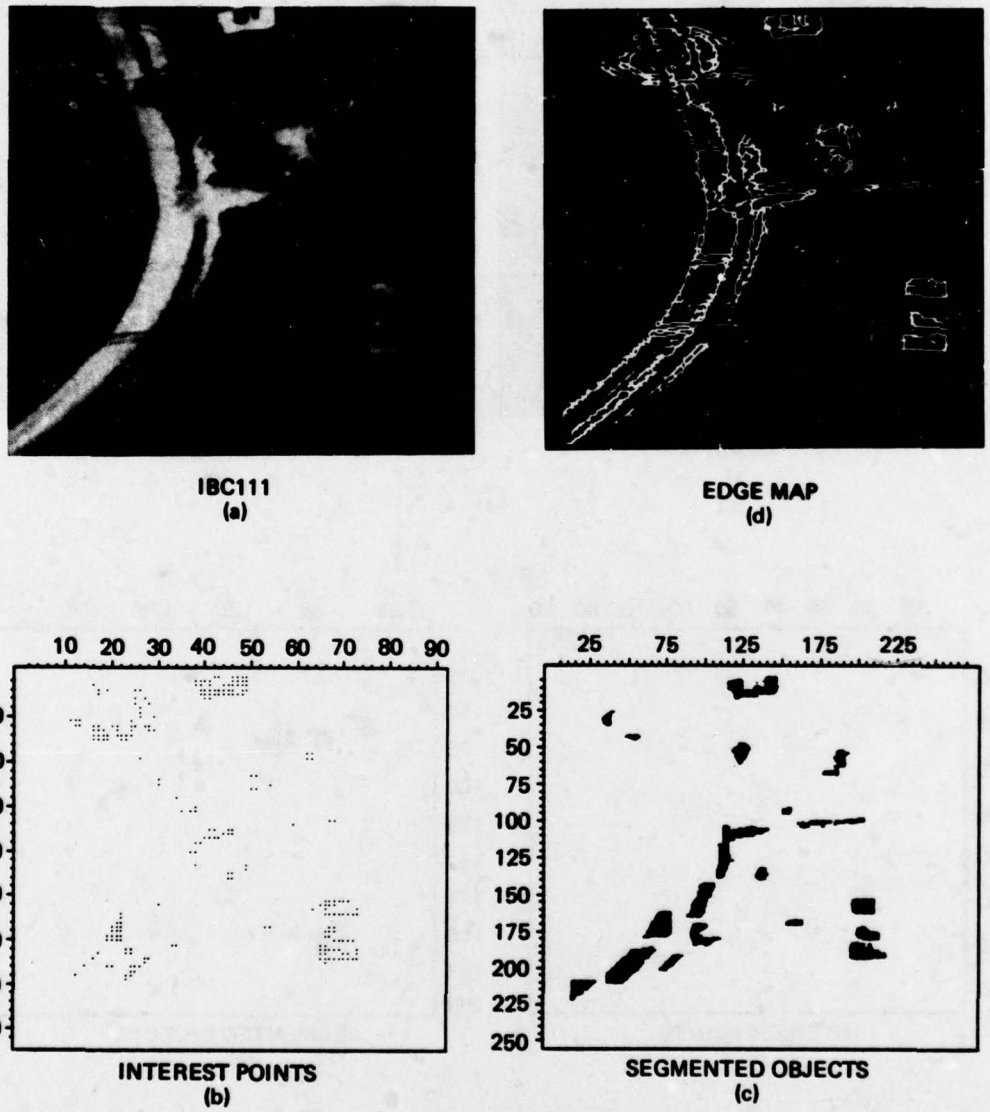
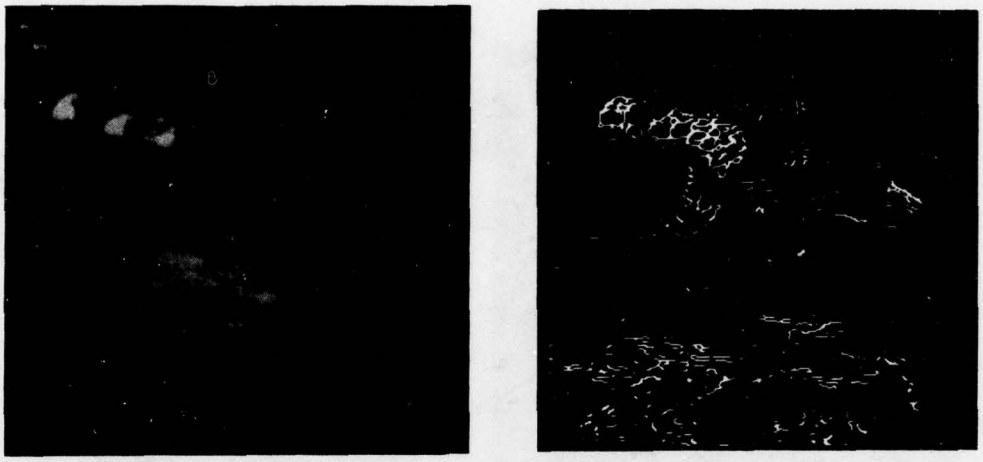
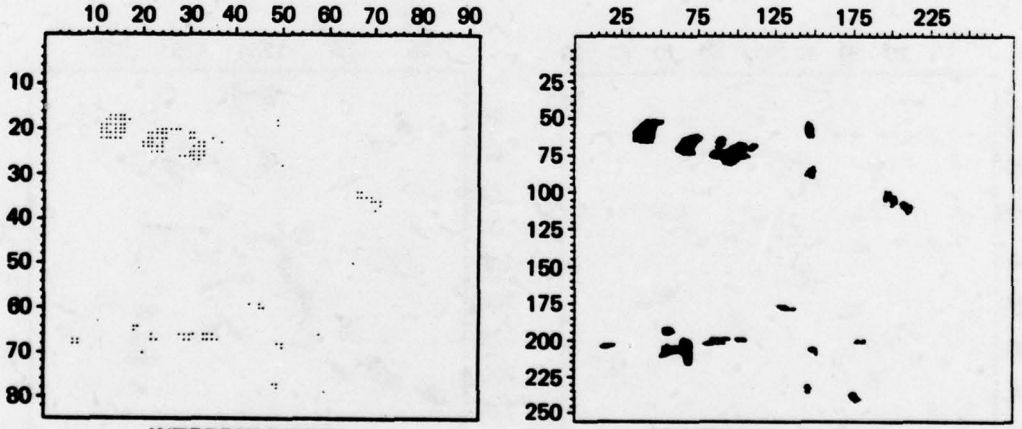


Figure 5. Feature extraction on IBC data base: (a) original image, (b) and (c) auto-cuer results, (d) edge extraction.



IBC201
(a)

EDGE MAP
(d)



INTEREST POINTS
(b)

SEGMENTED OBJECTS
(c)

Figure 6. Feature extraction on IBC data base: (a) original image, (b) and (c) auto-cuer results, (d) edge extraction.

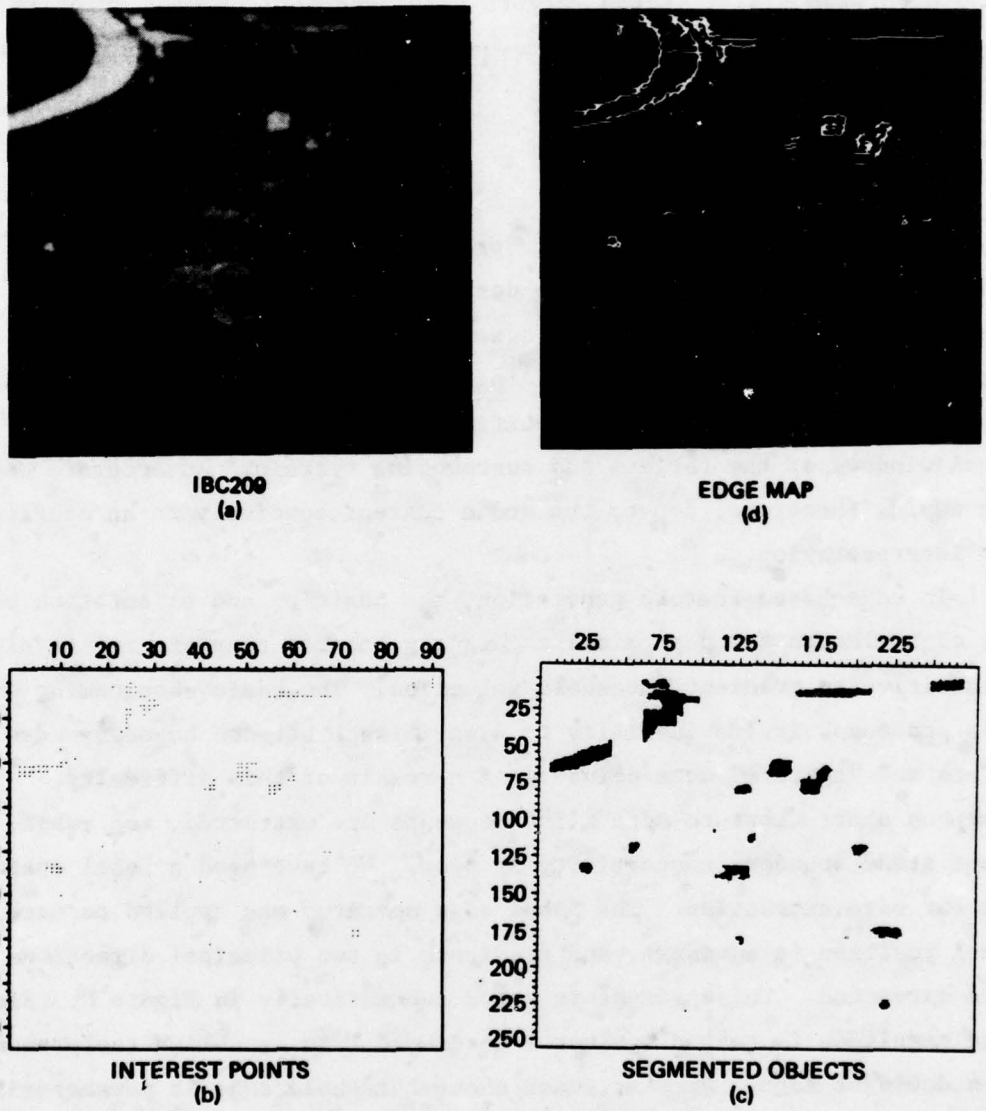


Figure 7. Feature extraction on IBC data base: (a) original image, (b) and (c) auto-cuer results, (d) edge extraction.

Those objects in the representative images shown in Figures 4 through 7 which have been classified as targets were used for the high priority target window allocations in the resource allocation and reconstruction simulations.

B. EDGE MAP

The IBC data base images were processed through the edge-based boundary-extraction algorithms to derive edge maps of the images for use in the bandwidth-compression scheme. As seen from Table 1, the edge map forms the basis of providing an overall representation of the dominant scene features, with specific local areas superceded by gray-level windows of the targets and surrounding terrain. An accurate edge map would, therefore, convey the scene content concisely to an operator for interpretation.

In edge-based feature generation, the position and orientation of the edge element are determined accurately because they are relatively insensitive to gradient threshold selection. The basic shortcoming of this processor is its inability to discriminate between boundary edge points and "texture" edge points. As a result of this difficulty, numerous short "texture-edge" line segments are extracted, and subsequent scene boundary connectivity is poor. We have used a Sobel operator for edge extraction. The Sobel edge operator was applied to each pixel position in an image, and gradients in two principal directions were extracted. This process is shown schematically in Figure 8. Each edge magnitude is tested against a threshold T to establish the presence of a dominant edge. Further, each above-threshold edge is parameterized by its orientation, $\theta = \tan^{-1} \Delta y/\Delta x$, and its principal direction. The edge element principal direction is defined as the direction of its principal gradient. This divides the extracted edge elements into four directional groups separated by 45° diagonals in the x-y image plane (see Figure 9).

7610-1

A	B	C
D	E	F
G	H	J

$$\nabla x = (C + 2F + J) - (A + 2D + G)$$

$$\nabla y = (G + 2H + J) - (A + 2B + C)$$

- SOBEL EDGE GRADIENTS, ∇x AND ∇y, ARE EXTRACTED WITH 9-ELEMENT WINDOW CENTERED AT PIXEL POSITION E.
- EDGES ARE ESTABLISHED BY THRESHOLD TEST OF EDGE MAGNITUDE, $|\nabla x| + |\nabla y|$

  IF $\frac{|\nabla x| + |\nabla y|}{T} > 1$, EDGE IS ESTABLISHED AT PIXEL POSITION E.

- EDGE ORIENTATION, $\theta = \tan^{-1} [\nabla y / \nabla x]$.

Figure 8. Sobel edge extraction.

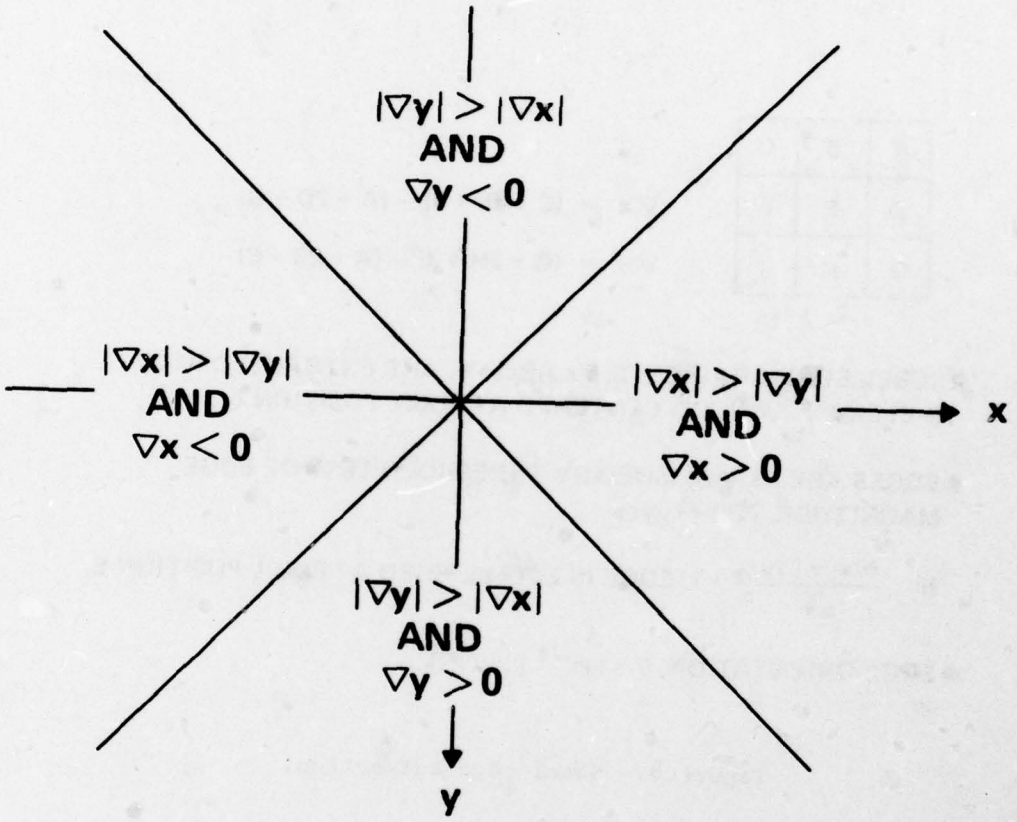


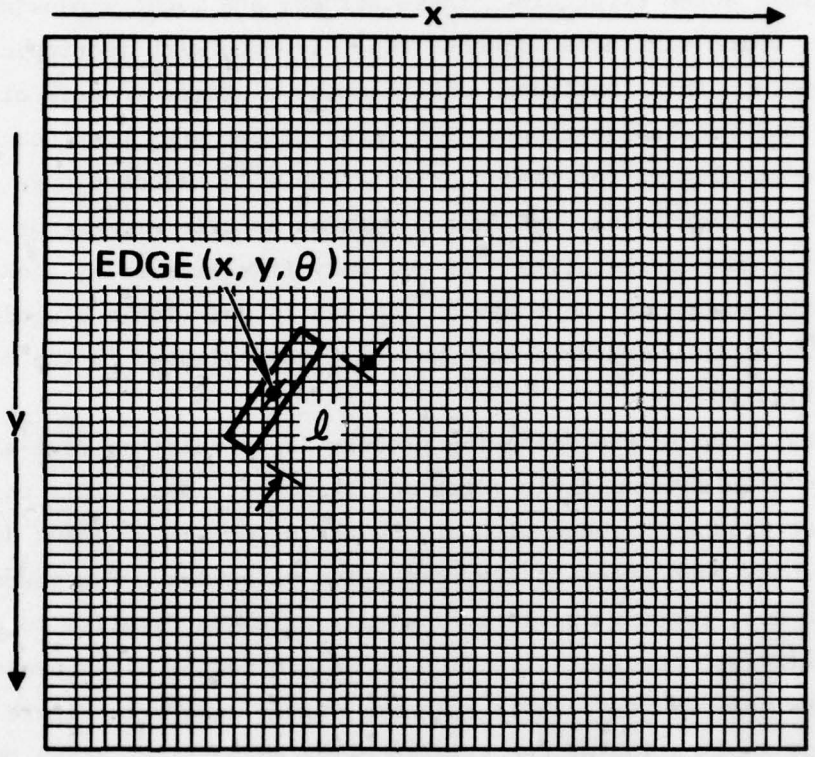
Figure 9. Edge element principal directions.

Several steps of edge element filtering were carried out to reduce the image edge density and to keep only those edge elements contributing to significant scene structure. These filters are local windowing operations, which eliminate adjacent edge elements with isotropic orientations and thin contiguous edge element clusters in each of the four principal Sobel directions. The first filter step tests each extracted edge element for common orientation with adjacent edge elements. The straight-line edge filter retains an edge element at pixel position Pi,j with orientation θ if and only if a minimum of n edge elements with angles in the range $\theta \pm \Delta\theta$ are formed within a window of length l and width w centered at pixel Pi,j and oriented in the θ direction (see Figure 10).

The next step in the filtering process is to thin the edge-element clusters to eliminate multiple edges and shadow effects. This thinning filter first groups the edge elements by the principal gradient directions and thins each group in its respective direction. The operation is a simple one, testing triplets of pixels oriented parallel to the selected principal gradient direction to determine the pixel position with the maximum gradient. This thinning filter, shown in Figure 11, and a second application of the straight-line edge filter using the filter parameters previously described are applied separately in each of the four principal directions. Application of the thinning filter skeletonizes the multiple and shadowed edges; the second application of the straight-line edge filter eliminates noncontiguous edge elements revealed by the thinning process.

Some representative edge-maps of images in the IBC data base are shown in Figures 4(d), 5(d), 6(d) and 7(d). Most of the key scene contents (such as roads, vehicles, structures, and dominant ground textures) are shown to have been extracted and represented in detailed outline form by the thinned and filtered edge points. The density of edge points in the edge maps can be controlled by adjusting the tolerances and threshold levels in the filtering and thinning process. These

5866–7



EDGE PICTURE

AN EDGE ELEMENT (x, y, θ) IS RETAINED IF AND ONLY IF A MINIMUM OF N EDGE ELEMENTS WITH ANGLES IN THE RANGE ($\theta \pm \Delta\theta$) ARE FOUND WITHIN A WINDOW OF LENGTH ℓ CENTERED AROUND THE EDGE ELEMENT (x, y, θ)

Figure 10. Straight-line edge filter.

THINNING FILTER FOR ∇x DIRECTIONS

7610-4

$P_{i-1,j}$	$P_{i,j}$	$P_{i+1,j}$

EDGE ELEMENT $e_{i,j}$ IS RETAINED ONLY IF
$\nabla x_{i,j} > \nabla x_{i-1,j}$ AND $\nabla x_{i,j} > \nabla x_{i+1,j}$

THINNING FILTER FOR ∇y DIRECTIONS

$P_{i,j-1}$
$P_{i,j}$
$P_{i,j+1}$

EDGE ELEMENT $e_{i,j}$ IS RETAINED ONLY IF
$\nabla y_{i,j} > \nabla y_{i,j-1}$ AND $\nabla y_{i,j} > \nabla y_{i,j+1}$

Figure 11. Edge element thinning.

edge maps form a key component of the 1,000:1 compressed image and are sent to the encoder/decoder segment of the IBC system for encoding, transmission, decoding, and reconstruction at the ground station. This aspect of the system is discussed below.

SECTION 3

CODING AND RECONSTRUCTION

The source and channel coding schemes being considered for the IBC system have been computer simulated. A block diagram showing a general model of a video transmission system is shown in Figure 12. From a communications point of view, the source and channel encoder are the two highest-level components of the remote IBC terminal. Note that, from this viewpoint, the source encoder contains all of the sophisticated image-interpretation and priority-control functions of the IBC system. The function of the source encoder is to represent the image as efficiently as possible (i.e., with the fewest bits) while the channel encoder intentionally adds redundancy to make the information less subject to errors (i.e., jamming).

Three classes of data are extracted by the IBC system for transmission on the downlink: the edge map video, the conventionally compressed sub-image windows, and the symbolic descriptors. A typical channel resource allocation a 1,000:1 compression ratio is shown in Table 1. The preliminary edge map source encoding method is 1-bit scanning at 1/4 resolution. This requires 16.4 kbit/sec at 1 fps update. A 32 x 32 pixel sub-image, centered on the highest priority target, is encoded at 1.5 bits per pixel (bpp) and updated at 7-1/2 fps. A 128 x 128 pixel orientation (or reference) window is transmitted at 1 bpp and 1 fps to provide orientation information to the ground station operator. The channel resource allocation also provides for transmission of sub-windows of 32 x 32 pixel size for secondary targets. These are updated at 1 fps and would be coded at 1 to 2 bpp, depending on the number of secondary targets to be transmitted. The symbolic descriptors are merely digital messages of approximately 48 bits in length. Hence, performance of the communications system in transmitting these messages can be readily evaluated (e.g., bit error rate curve) and was not simulated during the first quarter. Computer simulation results of source and channel coding of the edge map and conventionally compressed image subwindows, as well as the reconstructed IBC images, are discussed below.

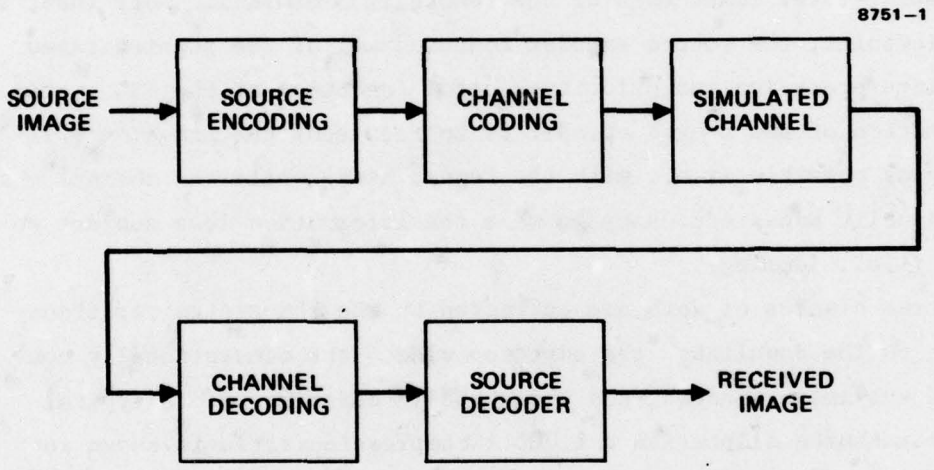


Figure 12. General model of video transmission system.

1. Edge Map Coding

Four images from the IBC data base are shown in Figure 13, with the corresponding full resolution (256 x 250) edge maps shown in Figure 14. Using the raster transmission format, the edge maps are seen in Figure 15 to be rather robust. Even at BER = 10^{-1}, the edge map is still very usable. However, a typical modem operating point is more like BER = 10^{-3}. A ground-restoration technique called edge-map filtering was used to remove isolated point errors. The filtering algorithm removes any indicated edge point that does not have nearest neighbors, as diagrammed in Figure 16. The performance of these filtering techniques can be readily estimated. At an error point in a black region of the edge map, the probability that more of the eight neighbors are also in error is $(1-P_B)^8$, where P_B is the channel bit error rate. Substituting a channel BER = 10^{-2} give the result that 92 percent of the channel errors should be removed by the filtering algorithm. The result of applying the filtering algorithm on the noise-corrupted edge maps of Figure 15 are shown in Figure 17. Note that at BER = 10^{-2}, 644 corrections were made out of 676 errors, which is consistent with the 92 percent error correction prediction. The channel capacity at 1,000:1 compression (i.e., 63 kbit/sec) is insufficient for transmission of the full resolution edge map. Therefore, the edge images were reduced in resolution by a factor of two in each dimension using the reduction/expansion algorithm shown in Figure 18.

The final results of the edge map coding simulation are shown in Figure 18 for a channel BER = 10^{-2}. Note that these images have been processed as follows: (1) resolution reduction, (2) channel error simulation, (3) edge map filtering, and (4) expansion to a 256 x 250 resolution.

Encoding of the IBC data base using the two conventional techniques of differential pulse code modulation (DPCM) and one-dimensional Hadamard transform is discussed below.

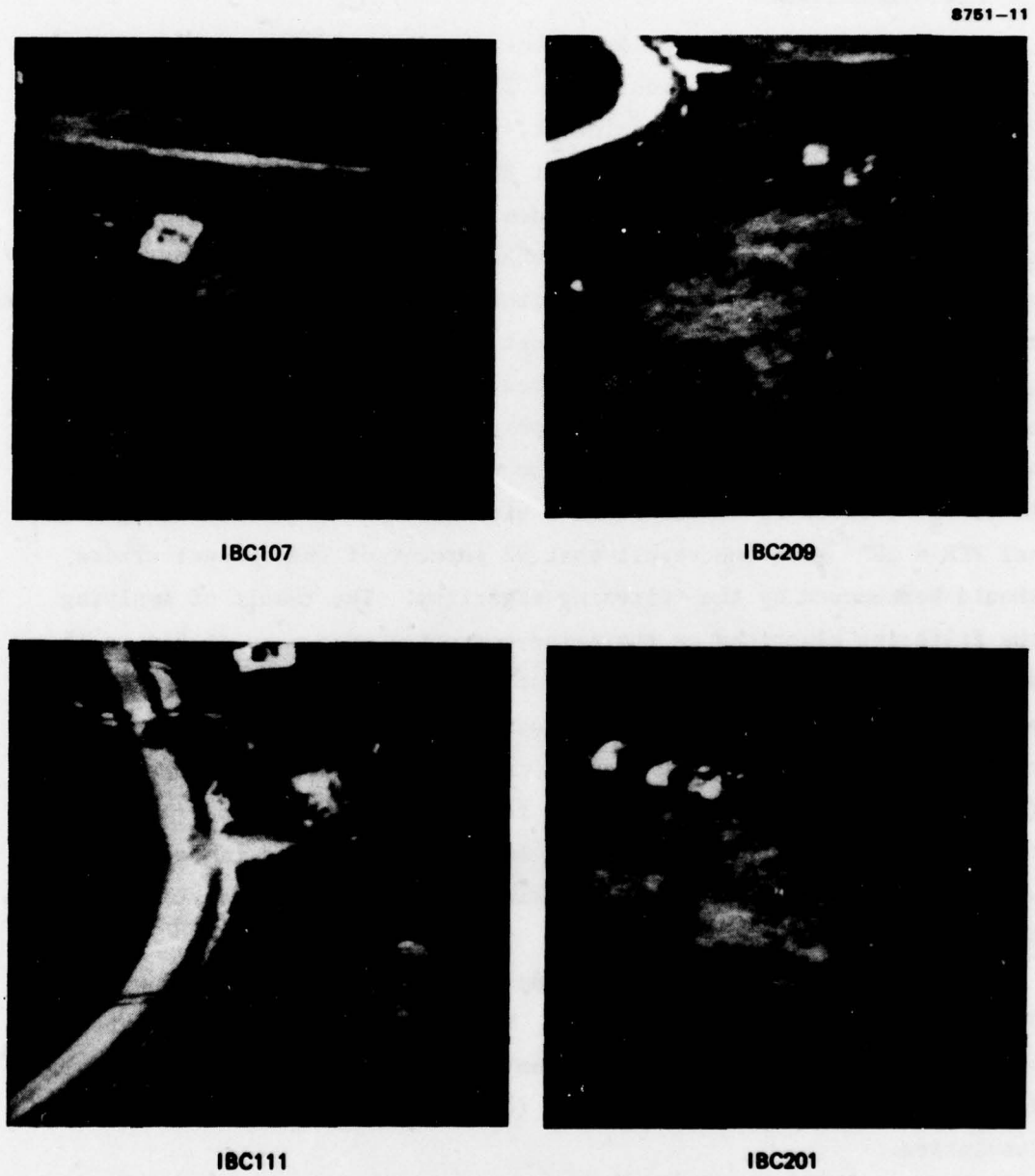
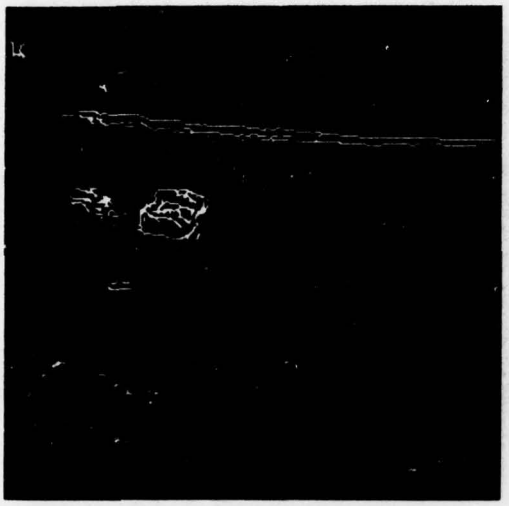


Figure 13. Representative images from IBC data base.

8751-12



IBC107

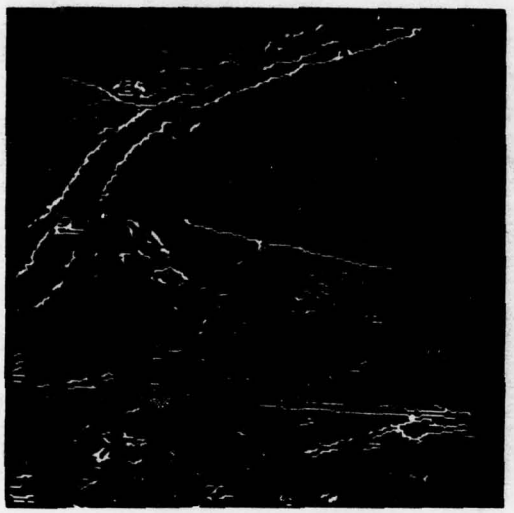
IBC209

IBC111

IBC201

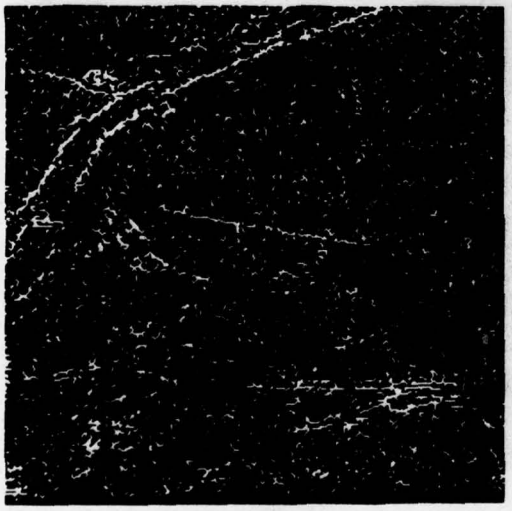
Figure 14. Edge maps (full resolution).

8751–13



IBC101

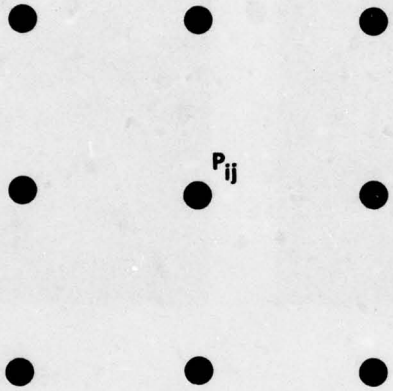
BER = 10^{-3}



BER = 10^{-1}

BER = 10^{-2}

Figure 15. Edge coding.

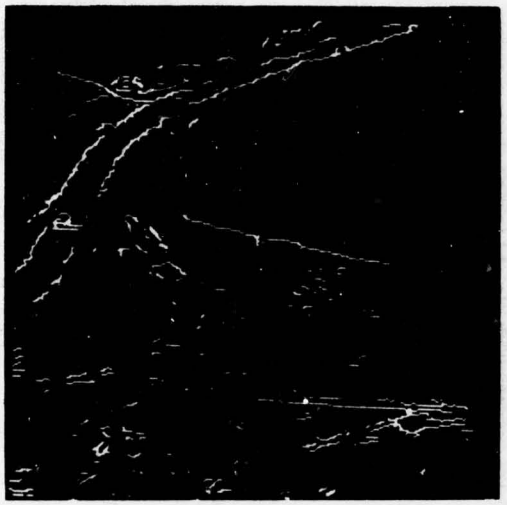


FILTERING ALGORITHM: IF P_{ij} = 1 (EDGE POINT)
AND NONE OF ITS 8 NEIGHBORS ARE = 1, THEN SET P_{ij} = 0.

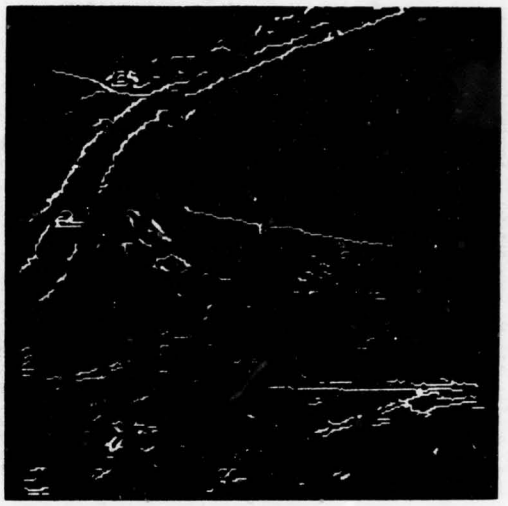
NOTES: AT BER = 10^{-1} P (NO NEIGHBORS IN ERROR/P_{ij} = ERROR) = $(1 - P_B)^8$ = 43%
AT BER = 10^{-2} PROB = 92%

Figure 16. Edge point filtering.

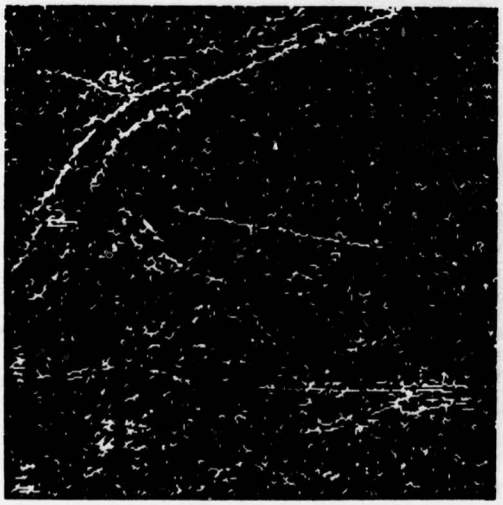
8751-14



IBC101



BER = 10^{-3}
(158/78 REMOVED)



BER = 10^{-1}
(2605/6371 REMOVED)

BER = 10^{-2}
(644/676 REMOVED)

Figure 17. Filtered edge map.

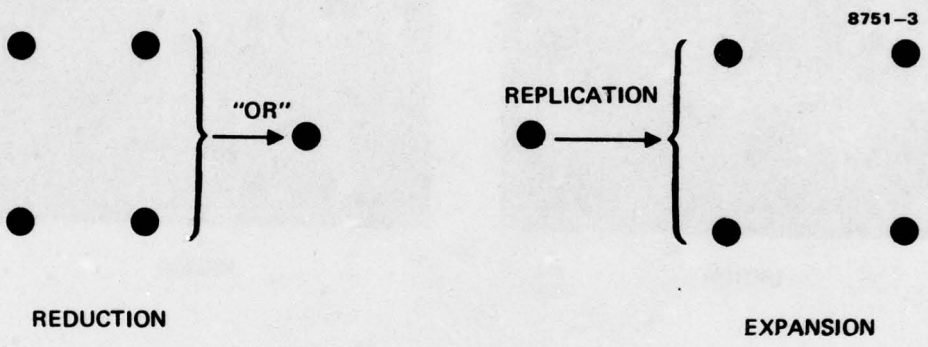
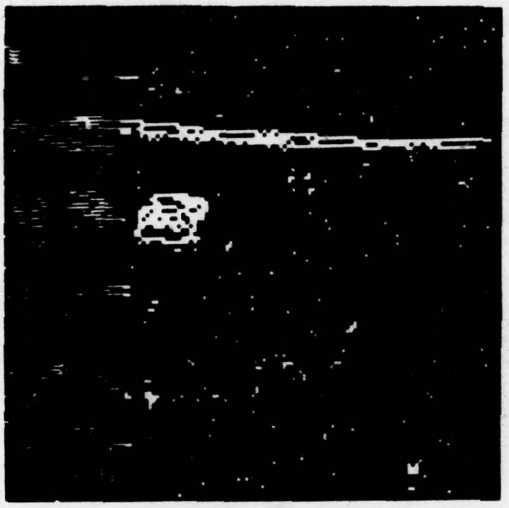


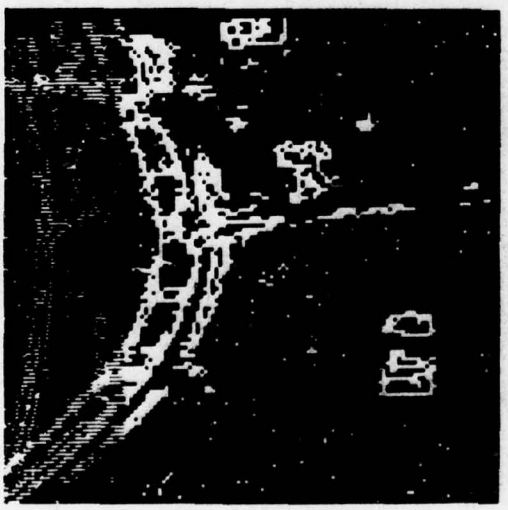
Figure 18. Reduction and replication algorithm.

8751-15



IBC107

IBC209



IBC111

IBC201

Figure 19. Edge coding (reduced resolution). (Note: the replication error at the left columns of these figures was due to a minor problem in software. This has been corrected, and the corrected photographs will be used in final report.)

2. DPCM Coding

DPCM is a simple technique that is frequently used for image coding. By encoding the difference between a predicted pixel amplitude (based on previous samples) and the actual amplitude, a dynamic range reduction and subsequent bandwidth reduction are achieved. The mean and variance on both a per pixel and a differential base were computed on subject images from the NVL data base. The statistics for image IBC 101 are shown in Table 2. Note the reduction in variance by a factor of >20 achieved by differential encoding. The computed inter-pixel correlation coefficients, which are used by the predictor, were typically very large for this data base. However, a large prediction coefficient causes channel errors to propagate extensively. Hence, a prediction coefficient of only 0.9 was used in all DPCM coding simulations, the results of which are shown in Figure 20 for 4-bit DPCM. At BER = 10^{-1}, the image is seen to be very streaked due to error propagation.

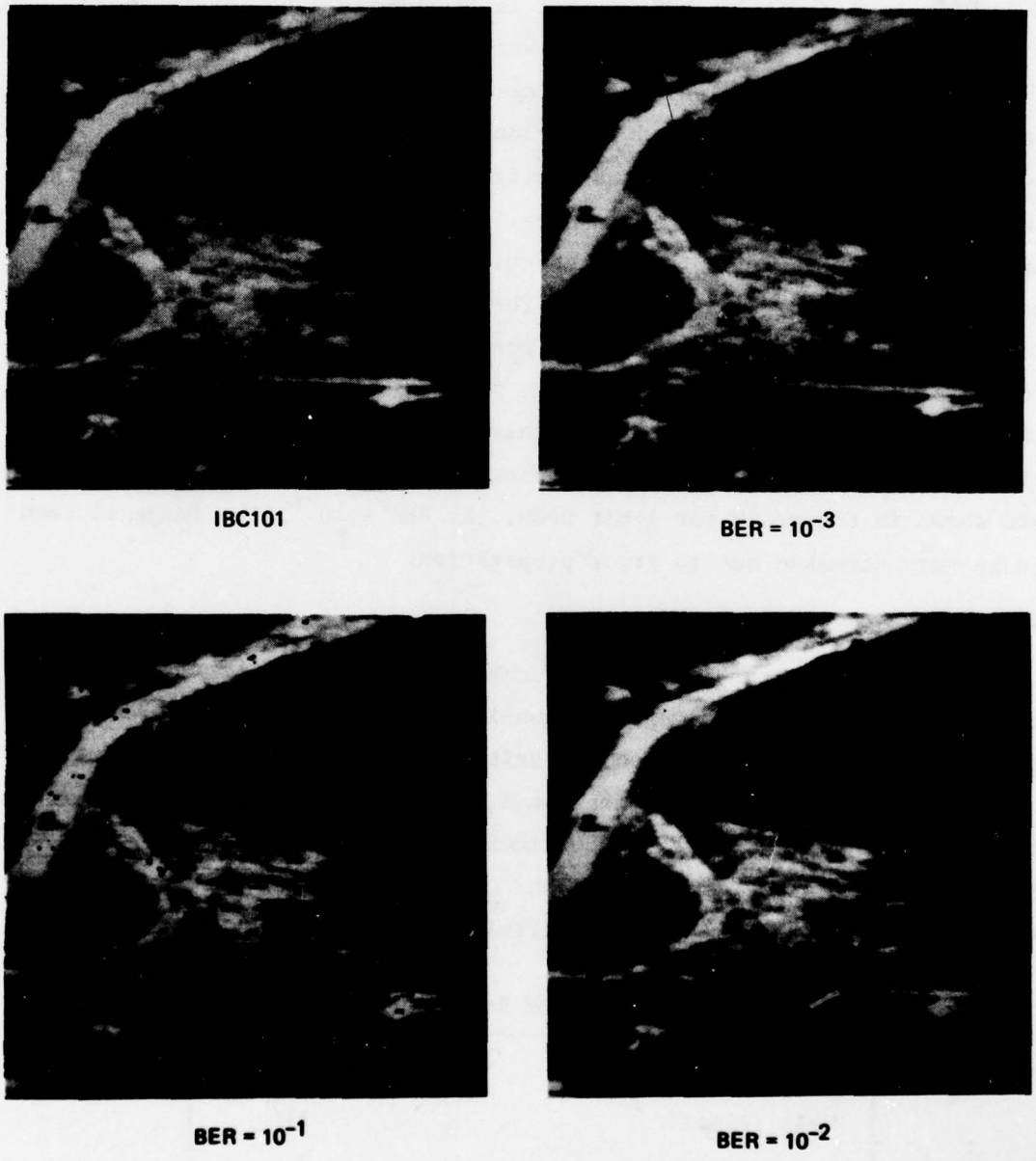
3. Hadamard Transform Coding

The Hadamard transform coding simulation is shown by the block diagram of Figure 21. The image (or sub-image) is first operated on by a one-third power memory-less nonlinearity and then undergoes a coordinate transformation via the one-dimensional, 16-point Walsh-Hadamard transform. Each transform coefficient is then individually quantized (i.e., the bandwidth reduction) depending on the component's importance in terms of energy contents and the human sensitivity to errors in that component.

Table 2. DPCM Statistics

IBC 101	
Mean (Pixel)	116
Variance (Pixel)	1841
Correlation Coefficient	0.98
Mean (Delta)	-0.1
Variance (Delta)	76.8

8751-16



IBC101

BER = 10^{-3}

BER = 10^{-1}

BER = 10^{-2}

Figure 20. DPCM coding with channel noise.

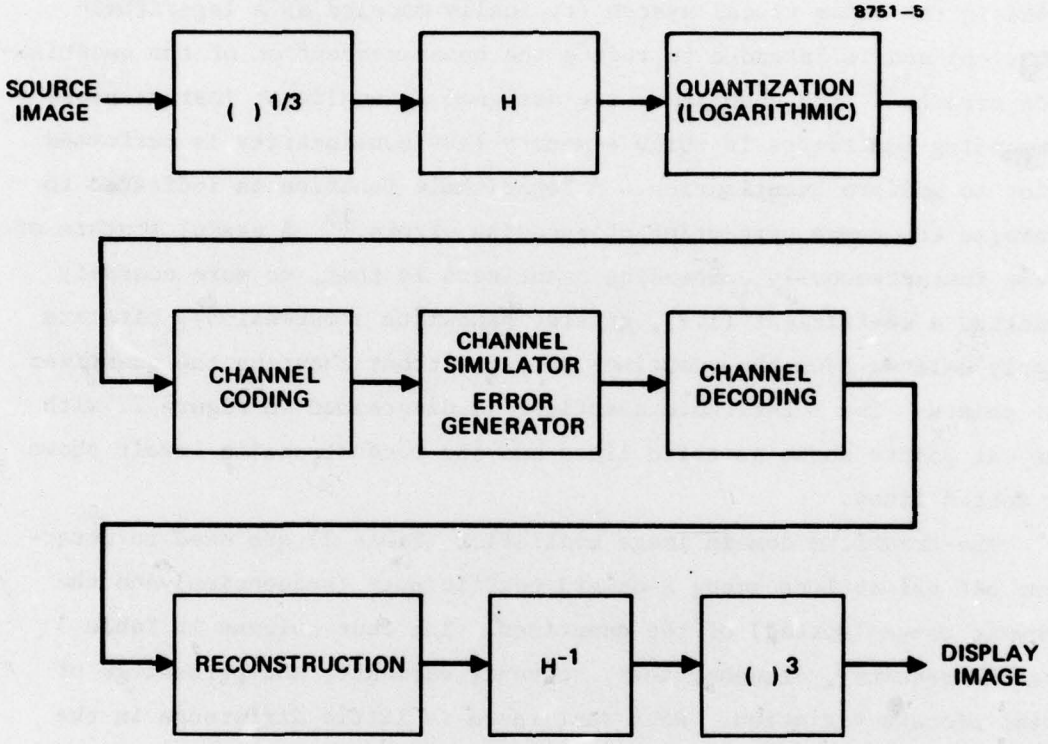


Figure 21. Hadamard transform coding simulation block diagram.

The quantized coefficients would then be error correction encoded and corrupted with bit errors by the channel simulator. The decoding operation is essentially the reverse of the encoding operation.

The one-third power preprocessing mimics the nonlinear processing found in the human visual system (typically modeled as a logarithmic function) and is intended to reduce the human perception of the quantization errors.[15] The quantizers are designed as nonlinear instantaneously companding quantizers in which a memory-less nonlinearity is performed prior to uniform quantization. A logarithmic function is indicated to minimize the human perception of encoding errors.[15] A useful feature of these instantaneously companding quantizers is that, to more coarsely quantize a coefficient (i.e., greater bandwidth compression), bits are simply deleted from the quantizer output without changing the quantizer cut points. The logarithmic quantizer is diagrammed in Figure 22 with the cut points shown as solid lines and the reconstruction levels shown as dotted lines.

The transform domain image statistics (Table 3) are used to determine bit allocations among Hadamard coefficients (sequencies) and the dynamic range (spread) of the quantizer. The four columns in Table 3 are the sequency, sequency mean, sequency variance, and percentage of total picture variation. Note that there is little difference in the image variance distribution among transform components whether the one-third power preprocessing is used (Table 3a) or is not used (Table 3b). The Hadamard transform coefficient bit allocations for 2 bpp and 1 bpp are shown in Table 4. Note that these allocations account for human sensitivity to encoding errors and were not just derived from a rate-distortion theory and mean-squared-error computation based on image statistics of Table 3 (the allocations are substantially different). The results of Walsh-Hadamard coding of the NVL data base are shown in Figure 23 for 2 bpp and in Figure 24 for 1 bpp. The quantizer designs were not perfected at this point, which is apparent in the 1 bpp coded images.

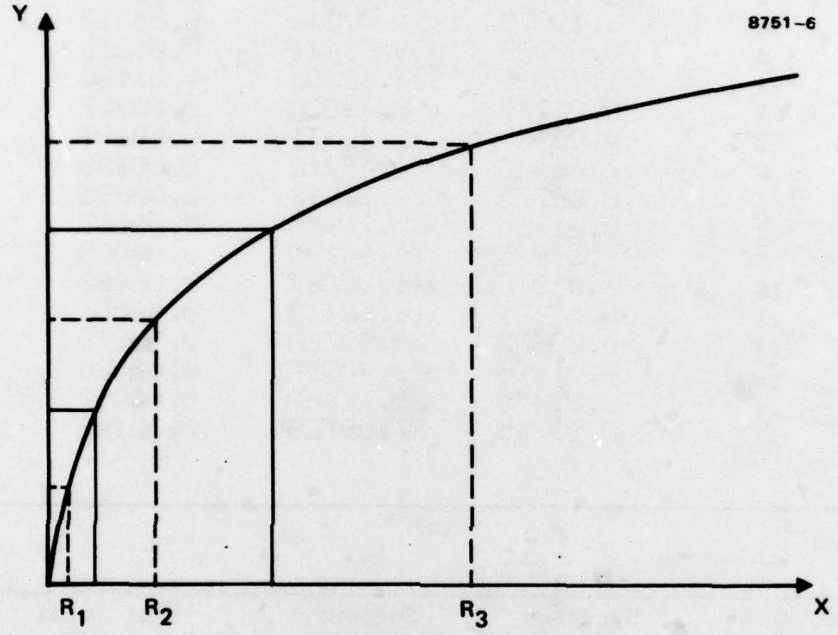


Figure 22. Logarithmic quantization.

Table 3. Hadamard Statistics (IBC101)

Sequency	Sequency Mean	Sequency Variance	% of Total Picture Variation
0	482.02475	24083.15600	0.90545
1	2.10875	1391.49840	0.05232
2	0.08025	353.83481	0.01330
3	1.33525	308.42336	0.01160
4	-0.21775	60.48983	0.00227
5	0.03925	77.48471	0.00291
6	-0.06425	79.43212	0.00299
7	0.62775	78.37918	0.00295
8	0.01925	17.71188	0.00067
9	-0.10675	16.64785	0.00063
10	0.05075	16.85267	0.00063
11	0.07175	16.62110	0.00062
12	-0.09825	20.33760	0.00076
13	0.01675	23.40997	0.00088
14	-0.01475	25.78803	0.00097
15	0.38725	27.99729	0.00105

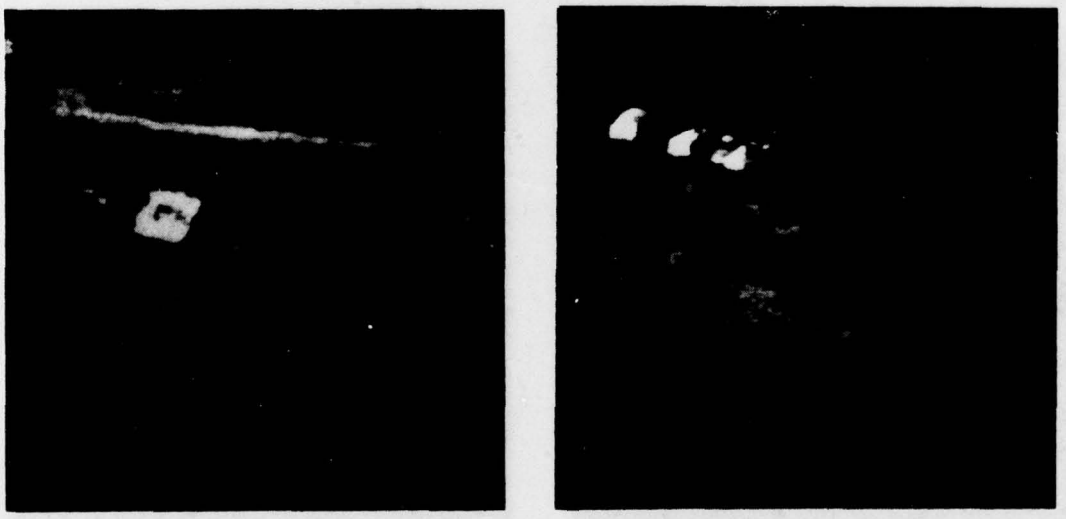
(a)

Sequency Sequency	Sequency Mean	Sequency Variance	% of Total Picture Variation
0	19.63224	5.73112	0.84962
1	0.03497	0.55466	0.08223
2	0.00086	0.14132	0.02095
3	0.02163	0.12356	0.01831
4	-0.00267	0.02505	0.00371
5	0.00007	0.03159	0.00468
6	-0.00135	0.03257	0.00483
7	0.01032	0.03243	0.00481
8	0.00002	0.00629	0.00123
9	-0.00169	0.00759	0.00113
10	0.00090	0.00762	0.00113
11	0.00106	0.00734	0.00109
12	-0.00116	0.00894	0.00133
13	0.00021	0.01014	0.00150
14	-0.00026	0.01121	0.00166
15	0.00585	0.01209	0.00179

(b)

8751-17

IBC101 IBC209



IBC107 IBC201

Figure 23. 2-bit Hadamard Coding.

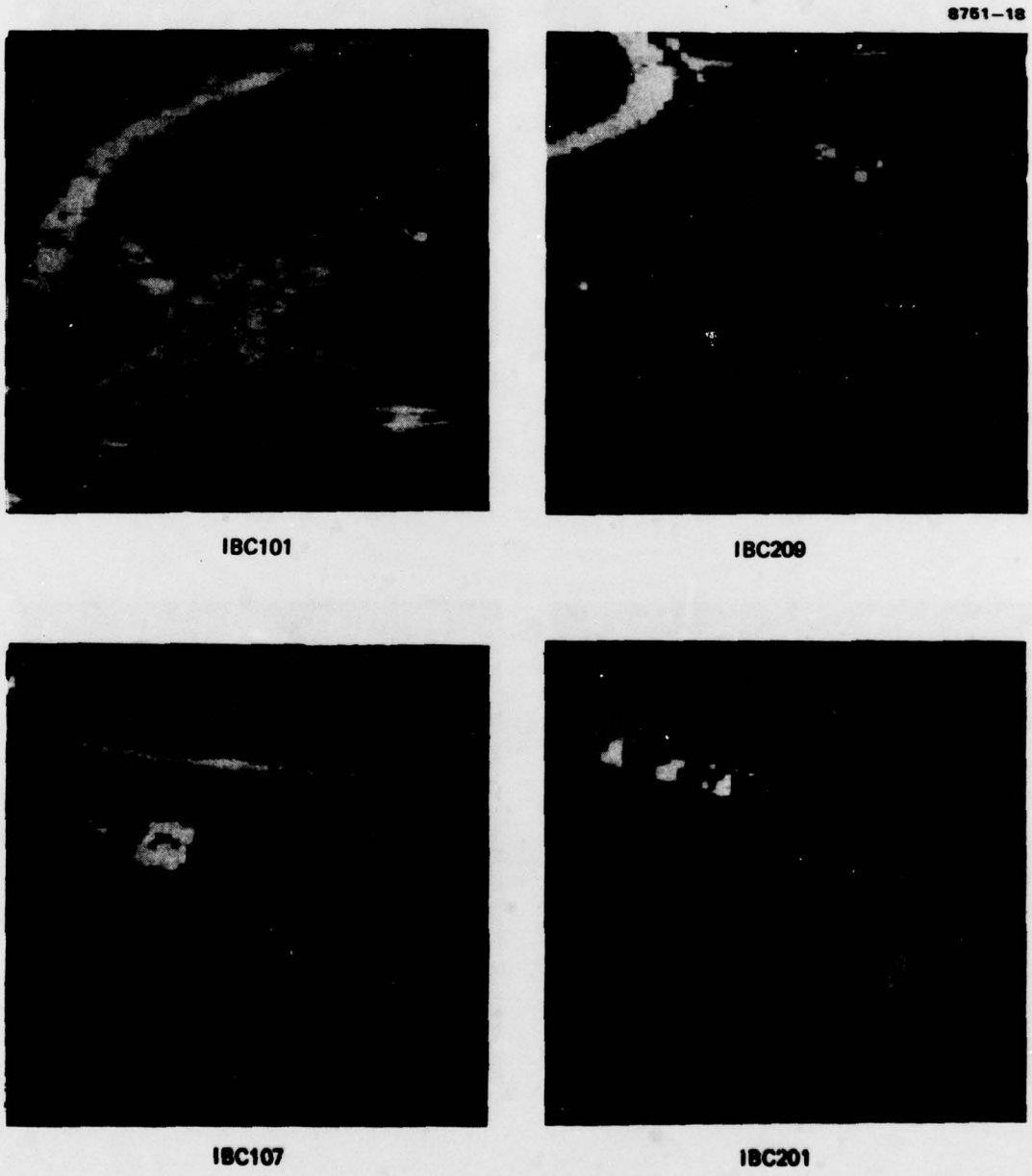


Figure 24. 1-bit Hadamard coding.

Table 4. Hadamard Bit Assignments

	2 bpp	1 bpp
S_0	5	4
S_1	5	4
S_2 S_3	3	2
S_4 S_7	2	1
S_8 S_{15}	1	0
Total	32	16

4. IBC Reconstruction

Figure 25 shows the simulated ground display at a data rate of 50 kbps, with Hadamard sub-image coding. This display shows the edge-map, the 128 x 128 orientation window, and a 32 x 32 priority target sub-image. We felt that using the average background gray scale as the non-white level in the edge maps would enhance the ground display. This enhancement technique is shown in Figure 26 with a background level of 127 out of a range of 0 to 255. If the gray scale had actually been chosen as the average background level, the display would have been even more pleasant.

Figure 27 shows a reconstructed image that has been completely processed by the computer-simulated IBC system. The auto-cuer results indicated 5 detections, 2 of which were false alarms. Each indicated target was coded with a 32 x 32 pixel subwindow at 2 bpp. The 128 x 128 orientation window is coded at 1 bpp and the edge map was transmitted at $(1/2)^2$ resolution. In Figure 27, the subimages were coded using DPCM @ BER = 10^{-2}, while in Figure 28 a comparison is made with Hadamard transform coding. Note that sharp edges (e.g., the road) in the orientation window are much better preserved with the transform coding.

8751-19

IBC101

1000:1 BANDWIDTH COMPRESSION
HADAMARD CODING
(NO CHANNEL ERRORS)

Figure 25. IBC reconstructed imagery, data rate <50 kbs.

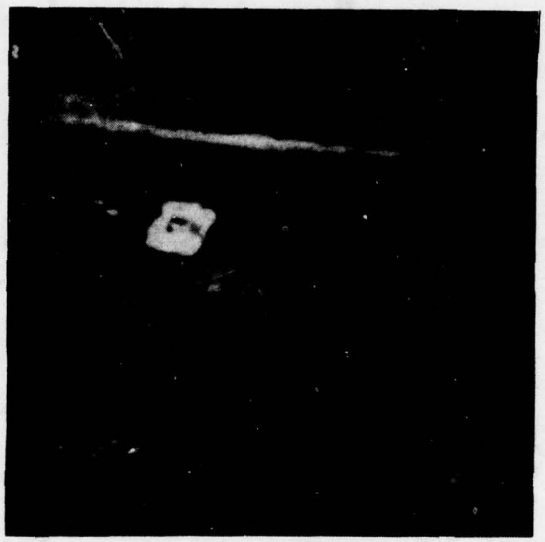
8751-20

IBC101

1000:1 BANDWIDTH COMPRESSION
(EDGE MAP GREY SCALE = 127)

Figure 26.
IBC reconstructed imagery with edge map enhancement, data rate < 50 kbs.

8751-21



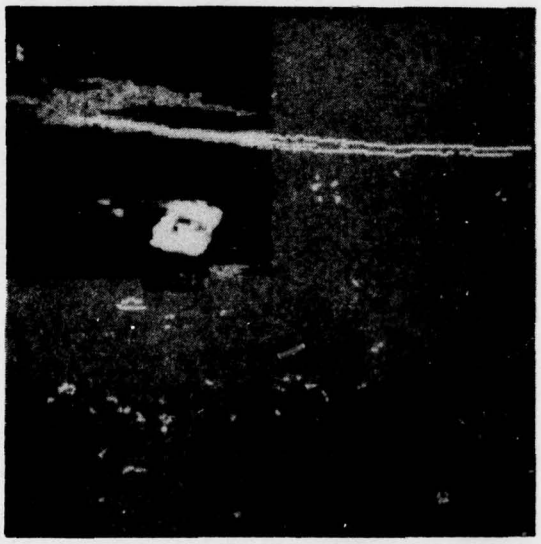
IBC107

1000:1 BANDWIDTH COMPRESSION
DPCM CODED, BER = 10^{-2}

Figure 27. IBC reconstructed imagery, data rate <50 kbs.

8751-22

DCPM CODED
(BER = 10^{-2})



HADAMARD CODED
(NO CHANNEL ERRORS)

Figure 28.
IBC reconstructed imagery comparison.

SECTION 4

PLANS FOR REMAINDER OF PROGRAM

The investigations performed during the first quarter of this program have demonstrated the feasibility and usefulness of a 1,000:1 bandwidth compression in image transmission. The results have shown that key features and target information of the image can be accurately extracted and reconstructed for display to a ground station operator. The results have also shown that the composite display of an edge map, an orientation gray scale window, and gray scale full resolution target windows can accurately and vividly portray the major image content information of value to an RPV mission to the ground station operator.

For the remainder of this program, the major emphasis will be placed on the development of the adaptive priority controller system. The assignment of priorities to the target descriptors, the sorting of these priority descriptors into a priority queue, the decision of how to further refine the detected targets for better classification, and the allocation of channel capacity to accommodate the available resources are all to be developed and tested using the IBC image data base.

In addition, more efficient source edge map coding will be investigated. In particular, attention will be directed at a form of (synchronizable) run-length coding. For comparison, the images will also be coded with one-dimensional discrete cosine transform, two-dimensional transform (e.g., 4 x 4 Walsh Hadamard, and a hybrid Hadamard/DPCM. The effects of error protection encoding will be simulated based on the (7,4) Hamming code. As time permits, the effort will be extended to include the (23,12) Golay code.

REFERENCES

1. "Demonstration of Westinghouse Automatic Cueing Techniques Using NVL Imagery," Contract DAAG53-75-C-0025 to Night Vision Laboratory.

2. "FLIR Image Analysis with the Autoscreener Computer Simulation," Vols. 1 and 2, Contract DAAG53-75-C-0246 to Night Vision Laboratory.

3. "A Discussion of Hardware Implementation for an Automatic Target Cueing System," Quarterly Report (October 31, 1976) on Contract DAAG53-76-C-0138.

4. "A Discussion of Design Goals and Hardware Implementation for an Automatic Target Cueing System," Quarterly Report (July 30, 1976) on Contract DAAG53-76-C-0138, Westinghouse Defense and Electronic Systems Center.

5. D. Milgram, "Region Extraction Using Convergent Evidence," Proc.: Image Understanding Workshop, April 1977, Science Applications, Inc.

6. D. Panda, "Segmentation of FLIR Images Pixel Classification," Proc.: Image Understanding Workshop, April 1977, Science Applications, Inc.

7. M. Geokezas, R. Jennewine, and G.P. Swanlund, "A Real Time Imagery Screener," Report on work conducted on USAF Contract AFAL-TR-74-184, October 1974.

8. "Processing of FLIR Data on DICIFER," Contract DAAG53-75-C-0277 for Night Vision Laboratory, by Pattern Analysis and Recognition Corp.

9. M. Geokezas and R. Jennewine, "Target Screener/FLIR System," Proceedings of the SPIE, Vol. 101, pp. 84-90, April 1977.

10. G.E. Tisdale, "A Digital Image Processor for Automatic Target Cueing, Navigation, and Change Detection," Proceedings of the SPIE, Vol. 101, pp. 112-119, April 1977.

11. "A Discussion of Hardware Implementation and Fabrication for an Automatic Target Cueing System," Quarterly Report (Jan 31 1977) on Contract DAAG53-76-C-0138.

12. G.R. Nudd, P.A. Nygarrd, and J.L. Erickson, "Image Processing Techniques Using Charge-Transfer Devices," Proc.: Image Understanding Workshop, October 1977, Science Applications, Inc.

13. P.A. Lachenbruch and M.R. Mickey, "Estimation of Error Rates in Discriminant Analysis," Technometrics, Vol. 10, No. 1, pp. 1-11, February 1968.

14. Keinosuke Fukunaga and David L. Kessel, "Estimation of Classification Error," IEEE Trans. on Comput. C-20, No. 12, 1521-1527 (Dec. 1971).

15. D.J. Sakrison, H. Halter, and M. Mostafavi, "Properties of the Human Visual System as Related to the Encoding of Images." In New Directions in Signal Processing in Communications and Control, NATO Advanced Studies Institute Series, Nordhoff International Publishing Company, 1974.