CR-152007

CR 152007

STUDY OF ADAPTIVE METHODS FOR DATA COMPRESSION OF MULTISPECTRAL SCANNER DATA

(NASA-CR-152007) STUDY OF ADAPTIVE METHODS FOR DATA COMPRESSION OF SCANNER DATA (TRW Defense and Space Systems Group) 153 p HC A07/MF A01 CSCL 09B

N81-19792

Unclas

G3/61 18657

FINAL REPORT

Contract No. NAS2-8394

TRW No. 26566

March 1977

Prepared for

NASA Ames Research Center
Moffett Field, California 94035

TRW
DEFENSE AND SPACE SYSTEMS GROUP

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

1. INTRODUCTION

The value of satellite-based earth resources observation has been well established by NASA's LANDSAT Program. As users have become familiar with the capabilities and opportunities provided by the current system, the potential operational uses of such data have become better defined. This, in turn, has led to increased user requirements on spatial and spectral resolution, spatial and spectral coverage, and timely delivery of data. Accomplishment of these objectives will require sensors operating at hundreds of megabits per second. At the same time, increased ground processing capability will be required to effectively take advantage of the large amounts of data. This will necessitate efficient data archiving, speedy transmission of data to the user, and effective image processing to extract the desired information.

Data compression (or source encoding) can play a significant role in each step of the data dissemination chain. By exploiting statistical properties of the image data, the data rate required for transmission to the ground can be reduced, simplifying data transmission and reducing on-board storage. Similarly, the amount of data to be stored on the ground can be reduced. Since a reduced rate implies faster transmission over a channel with limited capacity, compression has the potential of speeding data to the end user, and even making possible rapid interaction between a user and a distant archive.

Going beyond the considerable amount of research in image data compression over a number of years, several systems have been put into operation recently. These include a NASA system for transmitting weather satellite imagery from Wallops Island to Suitland, Maryland and a NASA/ARC video compression experiment using the CTS.

On-board implementation of data compressors has also become practical now because of increases in reliability and reductions in size and cost of digital logic.

1.1 BACKGROUND

This study extends the work completed in the Study of On-Board Compression of Earth Resources Data[1] (NAS2-8394). The preceding study emphasized nonadaptive methods for data compression; that is, methods that do not adapt to the local statistics of an image. The current study had as its goal evaluation of potential gains achievable by allowing adaptability to local statistics. In general, this means calculating data statistics for each block of some fixed size and coding the data in the block in different ways, depending on the statistics.

The current study also considered extending the application of compression algorithms to data dissemination tasks other than the transmission from a satellite sensor to a ground station. Thus, the applicability of compression algorithms to archiving and ground-to-ground transmission was also considered.

1.2 STUDY TASKS

As previously stated, the main objectives of the current study were to evaluate the performance of adaptive image compression techniques and to determine the applicability of a variety of techniques to the various steps in the data dissemination process. To effectively accomplish these goals, the study was broken down into three phases. The study flow is shown in Figure 1-1. The specific study tasks as defined at the outset of the study are described in the following paragraphs.

Phase 1. The first phase of the effort includes a thorough review of the adaptive compression strategies proposed in the literature in addition to those methods or combinations initially considered for study and evaluation by the contractor. The first monthly report contains a summary of the literature review and the list of adaptive methods proposed for investigation. The characteristics, advantages, and limitations are included in this review and reported in the monthly reports.

The adaptive methods include the following general approaches:

a) Adaptive transform coding methods using:

1) An activity index

2) Recursive quantization

3) Phase and amplitude quantization

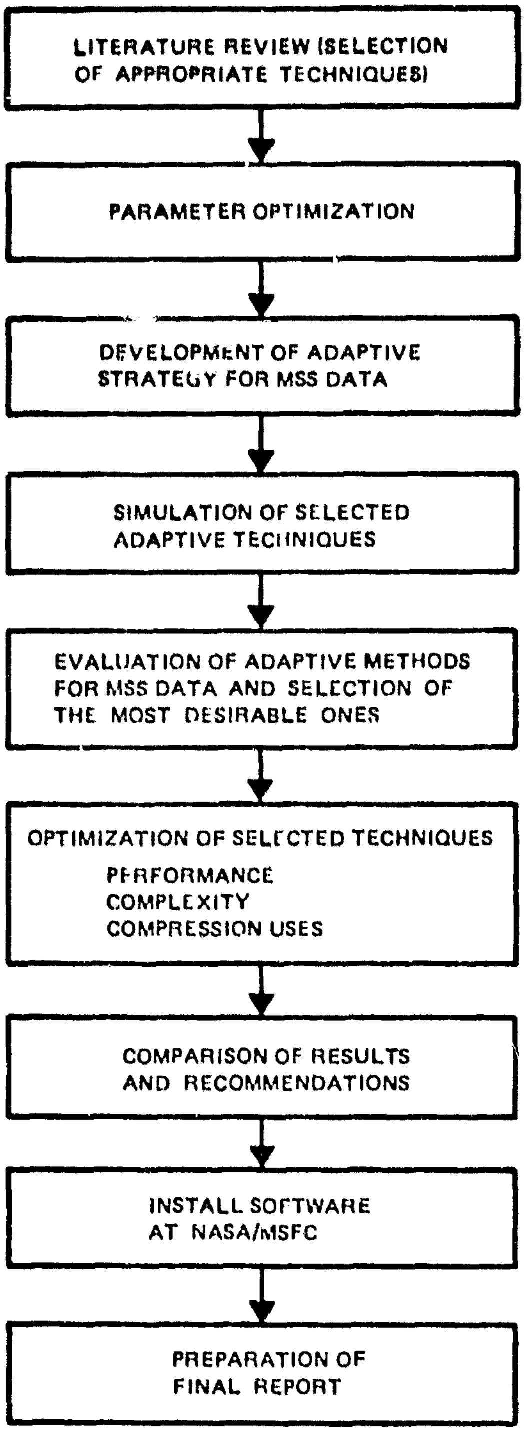

LITERATURE REVIEW (SELECTION OF APPROPRIATE TECHNIQUES)
PARAMETER OPTIMIZATION
DEVELOPMENT OF ADAPTIVE STRATEGY FOR MSS DATA
SIMULATION OF SELECTED ADAPTIVE TECHNIQUES
EVALUATION OF ADAPTIVE METHODS FOR MSS DATA AND SELECTION OF THE MOST DESIRABLE ONES
OPTIMIZATION OF SELECTED TECHNIQUES
PERFORMANCE
COMPLEXITY
COMPRESSION USES
COMPARISON OF RESULTS AND RECOMMENDATIONS
INSTALL SOFTWARE AT NASA/MSFC
PREPARATION OF FINAL REPORT

Figure 1-1. Study Flow Chart

b) Adaptive DPCM methods using:

   1) Switched quantizers

   2) Variable threshold - reconstruction levels

   3) Adaptive blocked DPCM encoders

c) Adaptive hybrid methods such as:

   1) Transform coding in the spectral and horizontal dimension and adaptive blocked DPCM in the vertical dimension

   2) Transform coding in the spectral dimension and two-dimensional blocked DPCM in the vertical direction

d) Adaptive cluster coding methods using:

   1) A variable number of clusters in each block

   2) Entropy coding to encode centroids and the classified picture.

Factors to be considered in parameter optimizations and tradeoffs include:

a) Bit assignments for transformed vectors

b) Quantization cut-points and representative values

c) Types of transforms and DPCM algorithms

d) Types of activity index used

e) Numbers and kinds of modes used.

To the extent possible, theoretical or experimental justification is given for choices of factors, parameters, and techniques considered under this task.

Phase 2. The second phase of the study is primarily devoted to:

a) Developing the proposed adaptive strategies for application to MSS data

b) Developing computer programs to simulate the adaptive compression algorithms

c) Evaluating the techniques with the specified standard scenes of MSS data

d) Optimizing compression algorithm parameters from three points of view:

   1) Compression performance

   2) Hardware implementation complexity

   3) Compression uses in the data system functions:

      On-board processing
      Transmission
      Central storage
      Ground distribution
      User storage
      User analysis/interpretation process.

Phase 3. The final study phase consists of:

a) Comparing results from developed adaptive methods

b) Recommending methods best suited for each compression application in the data system function

c) Preparing final report

d) Documenting and delivering to NASA computer programs of promising compression algorithms, and integrating these programs on the NASA MSFC computer.

The criteria for performance evaluation of each of the adaptive methods studied and tested on MSS scenes includes:

a) Mean-squared error between the original and compressed/reconstructed scenes

b) Classification accuracy

c) Computational complexity

d) Estimate of hardware implementation complexity

e) Subjective quality.

Evaluation shall be done primarily through computer simulations of algorithms on MSS data.

Selection of MSS scenes for algorithm evaluation shall have approval from the NASA technical monitor. NASA will supply to the contractor the MSS scenes to be used as standards for evaluation of the compression algorithms.

1.3 SUMMARY OF RESULTS

The most significant study results are the following:

a) The adaptive techniques investigated are capable of providing higher SNR with less complexity than the methods recommended in the previous study. Without requiring the complex KL spectral transform, the adaptive DPCM and adaptive hybrid techniques achieve results approximately 2 to 3 dB better than those obtained by nonadaptive techniques with the KL transform.

b) For low bit rates (below 1 bit per pixel), adaptive cluster coding produces the highest SNR reconstructions.

c) At 2 bits per pixel per band, all the selected adaptive techniques produce sharp reconstructed imagery with very few artifacts.

d) Below 2 bits per pixel per band, each algorithm introduces different kinds of distortion. Adaptive DPCM tends to blur the image slightly, the degree of blur depending on bit rate. Adaptive hybrid Hadamard introduces a blocky structure that might be partially reducible using post filtering. The adaptive clustering technique produces crisp images but introduces contouring. The adaptive 2D Hadamard technique produces some blockiness (less than the Hybrid technique) and also yields edge precursors. The 2D Cosine transform introduces slight blurring.

e) Effects of the various techniques on classification consistency are extremely scene-dependent. Radiometric distinctness of the various classes in the scene and the type of distortion introduced by the compressor both affect the results obtained.

f) Use of a fixed spectral transform preceding the spatial compression algorithms yields mixed results. For one scene, less than 1 dB improvement was obtained using the Haar spectral transform. For the other scene, a 2 dB degradation in performance resulted from using the spectral transform. This is due to the fact that the scene models that the spatial compressors take advantage of to reduce the bit rate are not as appropriate for the transformed bands as for the original bands. Alternate methods for taking advantage of spectral correlation may produce better results. N. Ahmed and T. Natarajan (2) have studied an alternate approach involving adaptive three-dimensional transform coding.

1.4 ASSUMPTIONS AND LIMITATIONS

In evaluating the results obtained in this study and considering the utility of the algorithms for a specific application, several ground rules of the study should be kept in mind:

a) By the nature of the study it was impractical to apply the techniques to more than a small set of typical scenes. Performance is scene-dependent, and therefore for any specific application and any particular sensor, results will be somewhat different. In evaluating the results obtained during this study, we have attempted to draw only those conclusions that could be verified in simulations using both scenes.

b) In judging the effects on automatic classification of the various algorithms, it is important to note that even a classifier using ground truth information and uncompressed data will make a significant percentage of errors for some classes. This is due to the fact that representatives of different classes sometimes have similar spectral responses.

c) Specific values for SNR or classification consistency obtained in the simulations should not be interpreted as defining the ultimate performance of these algorithms. The parameters that determine performance of each of the algorithms were optimized based on prior experience and preliminary simulations. Exhaustive simulations varying the parameters over a greater range may lead to somewhat better performance. It is unlikely, however, that the basic conclusions of the study would be affected in a significant way.

2. SURVEY OF ADAPTIVE IMAGE CODING TECHNIQUES

2.1 INTRODUCTION

In image data compression one is concerned with converting an analog picture, often generated by optical sensors, to the samallest set of binary integers such that this set of binary digits can be used to reconstruct a replica of the original image. Although it is conceivable to think of a single reversible operation that would convert the analog two-dimensional imagery to a set of binary digits, existing systems perform this operation with more than one processor. A common approach is to scan and sample the analog data such that it is converted to a set of correlated samples. The correlated data is further processed to eliminate or reduce its correlation prior to quantizing each processed sample individually and optimally. The quantized data can be encoded using entropy coding methods to give a bit rate almost equal to the entropy of the quantized data. A block diagram of the general coding system is shown in Figure 2-1. Adaptive systems are divided into four categories, as illustrated in Figure 2-2. In the following sections we discuss a theoretical system that performs the above operations optimally, before proceeding with the remaining suboptimal systems.

2.1.1 Karhunen-Loeve Transform and Block Quantization of Imagery

The Karhunen-Loeve (KL) transform is a statistical transformation that can be used to generate a set of uncorrelated variates from an analog signal specified by its autocorrelation function. A detailed discussion of one- and two-dimensional KL transform is in References 3 and 4. Here we give a brief description of the two-dimensional KL transform to point out the theoretical and practical problems associated with this method and the assumptions that one must make to overcome these problems.

The two-dimensional KL transform is a method of sampling a two-dimensional analog signal u(x, y) to obtain a set of ordered samples u_i, $i=1, 2, \ldots \infty$ such that for any given n, $u_i (i=1, 2, \ldots, n)$ has a maximum compaction of energy. This is an optimum sampling of u(x, y) if mean square error is used as the criterion of optimality. The KL samples are defined as

$$u_i = \int_0^B \int_0^A u(x, y) \, \phi_i (x, y) \, dx \, dy \qquad (2\text{-}1)$$

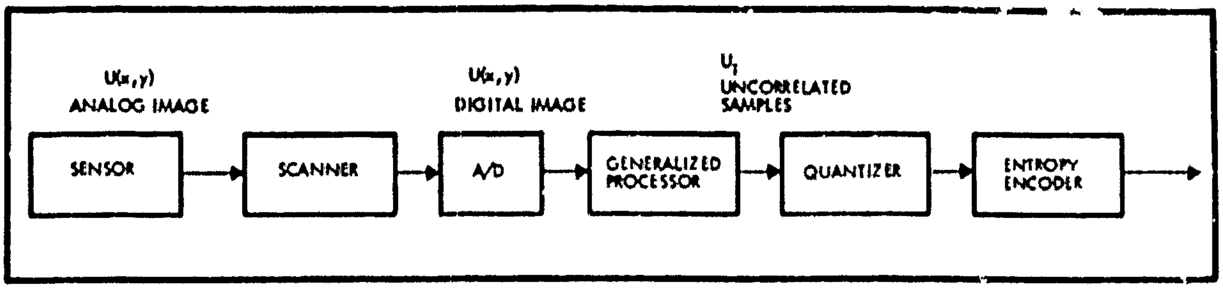


Figure 2-1. Block Diagram of a General Image Coding System

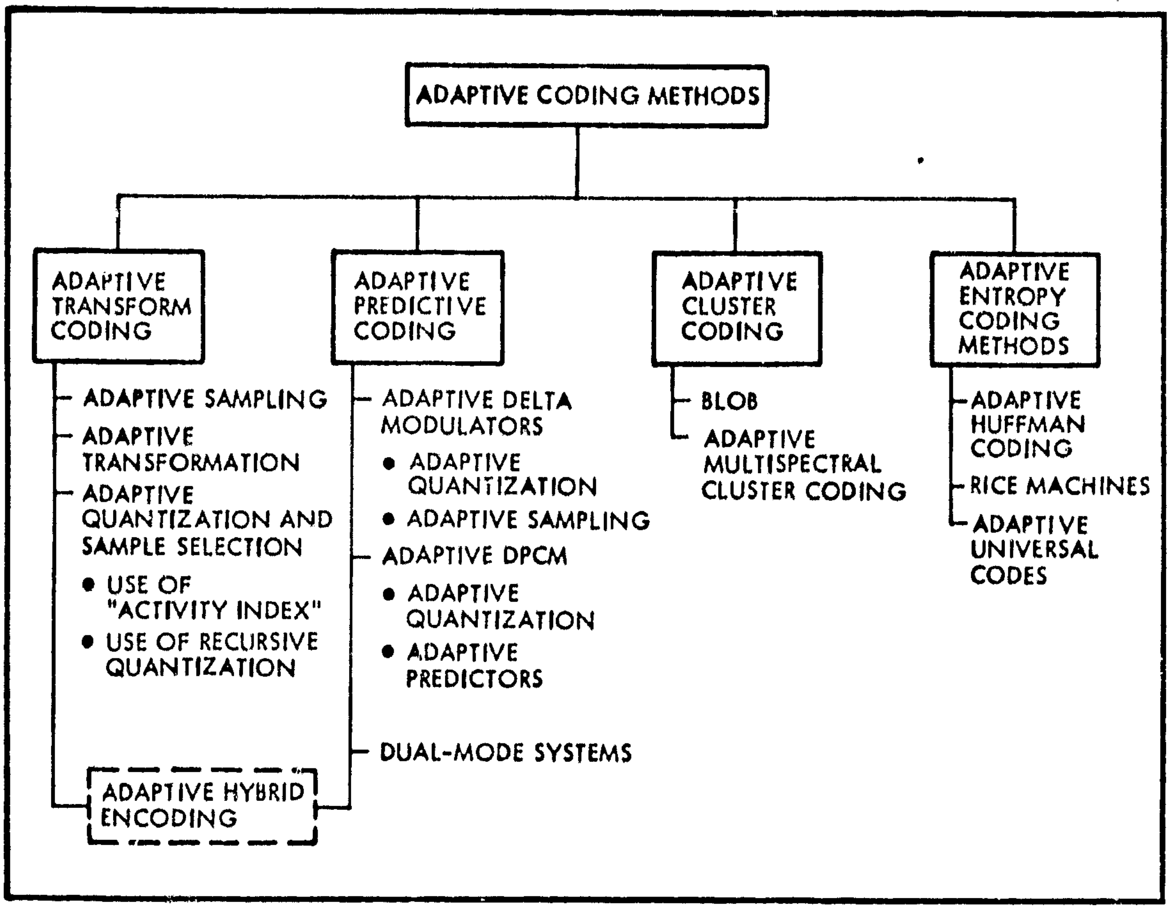


Figure 2-2. Adaptive Methods Discussed in This Literature Survey Divided into Four Categories

$$u(x, y) = \sum_{i=1}^{\infty} u_i \, \phi_i \, (x, y) \qquad (2\text{-}2)$$

where A and B are horizontal and vertical dimensions of the image u(x, y) and ϕ_i (x, y) is related to the correlation of the image by a two-dimensional integral equation.

$$\lambda_i \, \phi_i(x, y) = \int_0^B \int_0^A R(x, \hat{x}, y, \hat{y}) \, \phi_i \, (\hat{x}, \hat{y}) \, d\hat{x} \, d\hat{y} \qquad (2\text{-}3)$$

Note that a continuous KL transform eliminates the need for scanning and sampling the continuous imagery and generates the uncorrelated ordered samples u_i directly from the analog data. This approach, although simple and attractive, is almost impossible to implement. This is because of the following considerations:

1) Solution to the integral equation in (2-3) is only known for specific types of autocorrelation functions

2) Samples u_i can be generated from analog imagery using analog filters with impulse response function ϕ_i (x, y). These filters are very difficult to implement.

Because of these two problems a second approach has been considered that uses already scanned and sampled imagery u(x, y); x, y = 1,...,N to obtain uncorrelated samples u_i. In this approach Equations (2-1) and (2-2) are replaced by their discrete counterpart, i.e.,

$$u_i = \sum_{y=0}^{N-1} \sum_{x=0}^{N-1} u(x, y) \, \phi_i \, (x, y) \qquad (2\text{-}4)$$

$$\lambda_i \, \phi_i \, (x, y) = \sum_{\hat{y}=0}^{N-1} \sum_{\hat{x}=0}^{N-1} R(x, \hat{x}, y, \hat{y}) \, \phi_i \, (\hat{x}, \hat{y}) \qquad (2\text{-}5)$$

This method is known as the discrete KL transform or the Method of Principal Components.[5] The major problem here is solving Equation (2-5) which requires a method for finding the eigen-matrices of a fourth order correlation tensor*

*$R(x, \hat{x}, y, \hat{y})$ is not exactly a tensor since it does not have all tensor properties.

$R(x, \hat{x}, y, \hat{y})$ and such a method does not exist at present. A second problem is the computational complexity of Equation (2-4) which requires N^4 multiplication and addition operations. Although some so called fast KL transforms have been developed to reduce the required number of computations, these methods are only approximations and are only valid for data with a simple exponential correlation.[6,7]

Two other points with regard to the above method are worth noting. The first point is that both continuous and discrete KL transforms require a knowledge of the image correlation. Most imagery data are characterized by sharp edges and areas of high and low details. Therefore, they cannot be regarded as stationary processes over small regions, and an accurate and valid knowledge of the image correlation is not available. The second point is the performance of this system should not be confused with the performance predicted by the Shannon Rate-Distortion function. The Shannon Rate-Distortion function is the absolute lower bound on the performance of any coding algorithm while the method of the two-dimensional KL transform is one specific method and its performance is not the same as that predicted by Shannon Rate-Distortion curve. In fact, for two-dimensional stationary Markov data, the bit rate using the optimum KL transform is about 25 percent more than the bit rate predicted by the Shannon Rate-Distortion function.[4]

2.1.2 Suboptimal Systems

To overcome the above computational and procedural problems, a number of assumptions have been made. The problem of solving for the "eigenmatrices" of the fourth order "tensor" is avoided by the assumption of separability of the correlation function. This simply means that the correlation of data depends on the horizontal and vertical separation of the data and not on the distance separating the data. This model is not valid for most imagery data. The number of computations have been made manageable by using small block sizes. Dividing an image into K^2 blocks reduces the number of computations for an N by N image from N^4 to N^4/K^2 additions and multiplications. This, however, magnifies the stationarity problem. In general, the smaller the block size, the less accurate is the assumption of stationarity.

Other attempts for a simpler processor have been to replace the KL transform with a deterministic fast transformation. This was prompted mainly by the fact that the performance of the Fourier transform is assymptotically identical to that of the KL transform for stationary processes and also the fact that the number of operations required by a Discrete Fourier transform is proportional to $2N^2\log_2 N$. Other unitary transforms such as Hadamard, Cosine, Sine, Slant, and discrete linear basis (DLB)[8] transforms with similar desirable properties have also been used to replace the KL transform. The theoretical performance of these transforms is fairly close to the theoretical performance of the KL transform if the image is assumed to have a stationary separable correlation and the correlation is exponential in both the horizontal and vertical directions. However, since these assumptions are not true for most imagery, particularly on small blocks of data, the simulated results using these transforms are quite inferior to the theoretical results.

2.1.3 Block Quantization

The second step in the processor shown in Figure 2-1 involves quantizing the uncorrelated or almost uncorrelated samples in the transformed domain. Since in general the variances of the transformed samples are different, the number of binary digits assigned to each sample must be different. An optimum block quantization algorithm for assigning binary digits to transform coefficients has been developed. For minimum mean square error, these methods require a knowledge of the sample variances in the transformed domain. An optimum bit assignment simply involves assigning binary digits to individual samples such that the quantization error for each transform coefficient is the same. This requires assigning binary digits to individual samples in proportion to the logarithm of their variances.[3,9]

In developing nonadaptive transform coding systems, generally a deterministic transformation on a fixed block size is used and the block quantization is performed according to some pattern. The problem with this approach is that due to the nonstationary nature of the image data, the information content of the transformed samples changes from one block to the other and this affects the performance of the encoder. The performance of these systems improves by considering adaptive methods. These methods are based on the fact that images are composed of objects which are in turn a combination of edges and areas of varying degrees of detail. Using small block sizes, the degree of picture detail in each block changes drastically. An efficient encoding of these blocks requires more binary digits for areas of high detail and fewer binary digits for areas of low detail. In the following sections we survey the technical literature and discuss the various adaptive methods which have been developed in recent years. In doing this we discuss adaptive transform coding methods separate from adaptive DPCM methods even though DPCM can be considered a special case of a transform coding where a lower triangular transformation is used for transforming the data.[9]

2.2 ADAPTIVE TRANSFORM CODING METHODS

In adaptive methods we assume that the analog picture is raster scanned so that it is converted to a one-dimensional signal and we are concerned with the adaptive operations of sampling, transformation, sample selection, and quantization. Changing the parameters in each of the above operations such as the sampling rate, the size or the type of the transformation and the method of sample selection and quantization affect the overall performance of the encoder. In an optimal adaptive system one would like to have all of the above parameters change in response to the variations in image statistics. In the following sections we consider each of the above parameters individually and discuss the feasibility of their utilization in an adaptive transform coding system.

2.2.1 Adaptive Sampling

In sampling an image, ideally one would want to sample more finely in areas of high detail and sharp change, and coarser in areas of low detail and slow variation. This assigns a larger number of samples (bits) to areas of high details -- improving the picture definition in these areas while still keeping the number of total samples at a relatively low level. In a variable sampling rate system the receiver must be synchronized with the transmitter to know which areas are sampled at what resolutions.

Transform coding systems with variable sampling rates have not been reported in the literature. The only result in this area is contained in unpublished research where a grid of fixed and known dimensions is overlayed on the picture at the transmitter. Different regions are then sampled at different resolutions. This approach requires minimal information to inform the receiver of the particular sampling resolution at each grid location. This system clearly is not desirable for on-board data processing applications and will not be discussed any further.

2.2.2 Adaptive KL Transformation

In KL transform coding methods, implementational constraints require using small block sizes. Nonstationarity of imagery data requires using the proper correlation matrix for each block. This in turn requires one to estimate the covariance of each block of data, then use this covariance to find the proper eigenvectors to be used for transforming the data in that block. This approach presents a number of problems. One is the problem of estimating an N by N covariance matrix using only N points. Such an estimate is grossly inaccurate for large lag values.* Using a small lag value one has to use some model to find the large lag values. A model frequently used to find large lag values from a lag value of unity is the exponential autocorrelation model.** The second problem is the overhead information which is needed to transmit the covariance matrix for each block. Still a third problem is computing a separate set of eigen-vectors for each block. These problems severely constrain the application of such an adaptive KL transform method.

An adaptive KL transform coding method which eliminates these problems at the expense of performance is one suggested by Tasto and Wintz.[11] This system uses a block size of 6 x 6 samples mapped into a string of 36 samples by sequentially scanning each line in the block. The data is then considered as a one-dimensional signal which eliminates the problem of finding the eigen-matrices of a four-dimensional covariance "tensor." Each block (36 sequential picture elements) is classified at the transmitter into one of three possible classes. They use a rather complicated

*A rule of thumb reported by Blackman and Tukey requires at least 10 τ data points to obtain an accurate estimate of covariance for a lag value of τ.[10]

**An exponential autocorrelation model is rather accurate for correlation along the lines or the columns of low contrast imagery. However, it is not very accurate for high contrast imagery.[4]

procedure for classifying each block and the classifier must first be trained on some typical imagery data. For each class a covariance matrix and the corresponding set of eigen-vectors are used to transform that particular class. Each class has its own quantization procedure to block quantize the transformed data. The overhead information is minimal since a maximum of 2 bits per block is needed to specify in which class a given block belongs. They report an improvement of about 30 to 50 percent reduction in bit rate over nonadaptive KL transform methods. Naturally, considering a larger number of classes improves the results but this will complicate the already complex classification procedure.

2.2.3 Adaptive Quantization and Sample Selection in the Transform Domain

Complexity of the adaptive KL transform has forced other researchers to take different approaches to adaptive transform coding problems. These approaches involve using a deterministic transformation such as Hadamard or Fourier transform on a fixed block size and using an adaptive procedure for sample selection and quantization. In the following sections we will discuss various approaches to adaptive sample selection and quantization in the transform domain.

2.2.3.1 Threshold Quantization in the Transform Domain

The simplest system in this category is the method of threshold coding in the transform domain. Using this method one selects a threshold level and transmits only the transform coefficients which are larger than this threshold. The transform coefficients below the threshold level are set to zero at the receiver. This method is adaptive since the number and the location of the samples that are larger than a fixed threshold level change from one block to the other depending upon picture details. However, the system requires a relatively high bit rate for transmitting the addressing information. A simple form of this system that uses a 4 x 4 Hadamard transform was recently reported.[12] Only the dc term and the largest coefficient in the transform domain were transmitted along with the addressing information for the largest ac component. The authors did not report any numerical results but the quality of their encoded images was not as good as the result reported by Landau and Slepian[13] using a similar system with non-adaptive zonal quantization.

Anderson and Huang made a more comprehensive study of the threshold coding method for bandwidth compression of imagery data.[14] Their proposed system used a two-dimensional Fourier transform on a block size of 16 x 16 picture elements. The standard deviation of the samples in each block was measured first. Then amplitude, phase, and position of the L transformed samples with the largest amplitudes were transmitted where L was proportional to the standard deviation of the samples in each block. The system's adaptivity was increased by making the number of quantization levels in each block proportional to the standard deviation of the picture elements in that block. The addressing information and position of the L largest Fourier coefficients were transmitted using a run-length coding algorithm. They reported good results at 1.25 bits per picture element.

2.2.3.2 Adaptive Sample Selection in the Transform Domain

Different adaptive transform coding methods use different measures for sample selection in the transform domain. The most frequently used measures are variances of the elements in the transform domain, the sum of the absolute values of the ac coefficients, and the ac energy of the transform coefficients. The sum of the square of the ac transform coefficients for each block is the same as the sum variances of picture elements in that block for orthogonal transformations. Therefore, the variances and the ac energy of the transform coefficients are measures of the uncertainty or the randomness for the image. The sum of the absolute values of the ac components is also a measure of image activity.

2.2.3.3 Use of the Activity Index for Sample Selection

The sum of the squares of the coefficients in the transform domain or the sum of absolute values in the transform domain, referred to as the activity index,[15] can be used to classify each block to one of M possible classes. This requires M-1 threshold values which can be chosen experimentally. One can choose these threshold values to have approximately a fixed number of bits per sample averaged over a number of blocks. However, in a practical situation, one would use a set of M-1 threshold values that would be controlled by the fullness of the buffer. Each class would use a different sample selection and quantization procedure. The class with high activity index employs more binary digits than the class with low activity index. Gimlett[16] and Claire[15] have recommended use of the activity index with four possible classes. They use a combination of zonal and threshold sampling for each class. Their recommended bit assignment for each class is shown in Figure 2-3 where each 8 x 8 transformed block is broken into two or more regions by dashed lines. Numbers in each sample location indicate the number of binary digits that is used for quantizing that particular coefficient. Zero in a region indicates that the largest element in that particular region should be transmitted with its addressing information. More than one zero implies that the several largest values in that region should be transmitted. Blanks indicate no bits assigned to those coefficients. The authors do not report any results. They state that their particular bit assignment is not unique or optimum but is simply a reasonable bit assignment intuitively suited for the two-dimensional Hadamard transform. The authors conjecture that combining zonal sampling with that of threshold sampling will improve the results significantly. However, Reader[17] has considered mixing threshold sampling with zonal sampling in a situation somewhat different from that of Claire and Gimlett and reports almost no improvements due to the addition of threshold quantization.

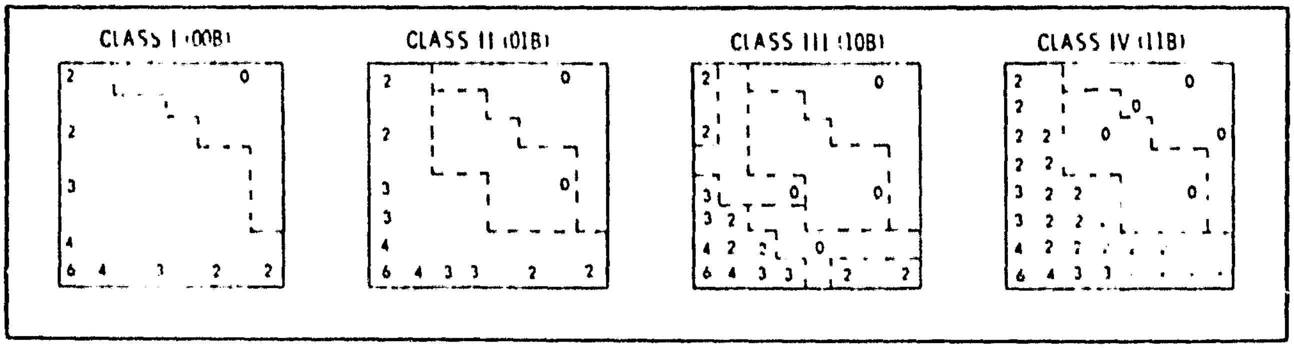


Figure 2-3. Coding Schemes for the Four Classes for 8 x 8 Hadamard Transform

As indicated in the previous paragraph, the sum of the absolute values of the transform coefficients for each block is a measure of the image activity in that block. The relative size of the different transform coefficients indicates the degree of image activity in various directions and frequencies. In a three-dimensional transformation of television signals, relative amplitudes of three coefficients (vectors) adjacent to the dc term reflect the image activity in horizontal, vertical, and the temporal directions. The accuracy of the quantization of vectors corresponding to the image activity in various directions can be used to control the coding fidelity in these directions. Knauer[18] uses this approach adaptively to encode various degrees of movements in television using a 4 x 4 x 4 Hadamard transform. In the adaptive mode of operation, the processor monitors the appropriate vectors to determine the degree of movement for each subpicture. For rapid movements, an option giving high temporal and low spatial fidelity is utilized. For slow movements, an option giving high spatial but low temporal fidelity is utilized. These options improve the subpicture quality of the encoded imagery since human vision is very insensitive to spatial fidelity for rapidly moving objects, but its sensitivity improves as temporal movements slow down.

2.2.3.4 Adaptive Transform Coding Using Recursive Quantization

Instead of the sum of the absolute values of the transform coefficients, one can use the variances of the transform coefficients as an "activity index" to classify each block into a number of classes for subsequent sample selection and quantization. Variances of the coefficients may be used for adaptive sample quantization as well. Tescher et al[18,19,20] make use of the variances of the coefficients in the transformed domain for adaptive bit assignment using two different approaches. In one approach a two-dimensional Fourier transform of a 256 x 256 image is generated. The complex Fourier coefficients are represented in terms of their phases and their amplitudes. The variance of each component in the amplitude plane is estimated and the bit assignment for that particular coefficient is performed in proportion to the logarithm of its estimated variance. The corresponding phase component is quantized using one more binary digit than has been used for amplitude quantization.* The variance of the amplitude of individual coefficients is estimated using a predictor that combines the variances of a number of adjacent quantized elements to predict the variance of a given coefficient. This system requires knowledge of the variances of some initial values to start the process. A number of procedures for estimating the variances of the initial values

*Experiments have shown that in transform coding the reconstructed picture is more sensitive to degradations in phase than it is to degradations in the amplitude.

are discussed in reference 19. Tescher has used this approach to encode both monochrome and color images. His results show an improvement of about 50 percent reduction in the bit rate compared to nonadaptive systems. He has considered using both Fourier and Hadamard transforms with similar results. Although for both transformations he uses an amplitude and phase representation for the transform coefficients, the method would work using other representations as well, i.e., real and imaginary representation for Fourier transform or real numbers for Hadamard transform.

A second approach to the problem suggested by Tescher and Cox[21] involves dividing the image into blocks of 16 x 16 picture elements and scanning of the samples in the transform domain to convert the two-dimensional data to a one-dimensional format. They use the scanning pattern of Figure 2-4 which is preferred over line by line scanning in the sense that it gives a smoother decay in the size of the variances of the transformed coefficients. The monotonic decline of the smoothed energy in the transform coefficients supports the choice of the scanning pattern. Next, they estimate the variance of the one-dimensional data sequence and make a bit assignment in proportion to the logarithm of the estimated variances. When the variance of a coefficient is so small that the number of binary digits assigned for its quantization falls below 0.5 bit, the processor stops and the remaining samples in that block are substituted with zeros at the receiver. The estimate of the variance for the nth transform coefficient $\hat{\sigma}_n^2$ is

$$\hat{\sigma}_n^2 = A_1 \hat{\sigma}_{n-1}^2 + (1 - A_1) \hat{X}_{n-1}^2 \qquad (2\text{-}6)$$

where $\hat{X}_{n-1}$ is the quantized form of the (n-1)th transformed sample (in a one-dimensional sequence) and A_1 is a weighting factor which is chosen rather arbitrarily equal to 0.75 in their experiments. The system is adaptive since more binary digits are assigned to blocks with larger transform coefficients. These blocks are assigned more binary digits since more transform coefficients are selected and the coefficients are quantized more accurately. Tescher and Cox report good results employing this method with Cosine and Slant transforms for a variety of monochrome images. However, they do not report any direct comparison which would show the gain that results due to this method of adaptive sample selection and bit assignment.

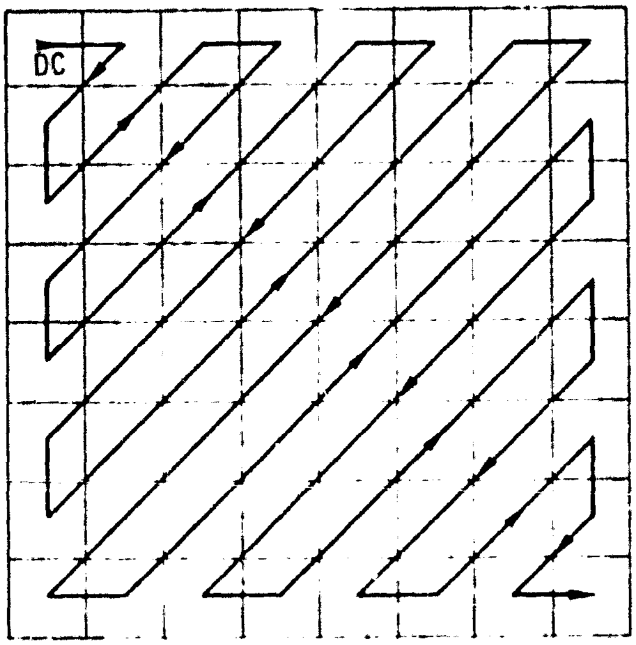
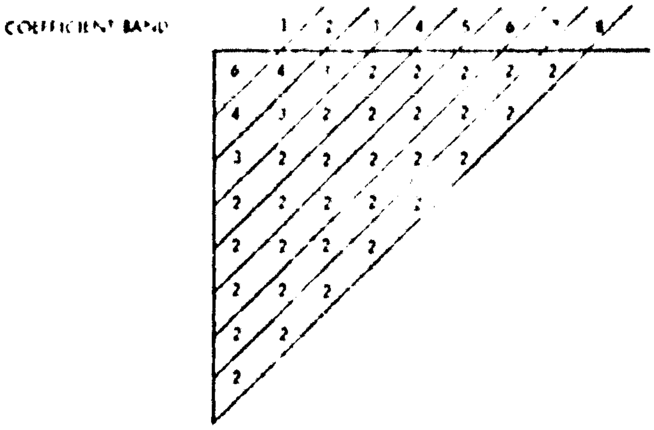


Figure 2-4. Ordering of the Frequency Domain

The adaptive coding methods discussed in previous sections are completely adaptive in that the number of binary digits assigned to each block changes from block to block, resulting in a variable rate system. In addition to these techniques, a number of adaptive methods have been devised that use a fixed number of samples and a fixed number of quantizers in each block. However, a different normalizing constant is used to normalize the samples in that block prior to their quantization. The normalizing constant must also be transmitted for each block. Since these adaptive techniques have a fixed bit rate, they do not require a buffer and buffer control logic to transmit their outputs over a fixed-rate channel. Reader[17] uses a normalizing constant for each block of 16 x 16 samples which is related to the measured variance of the samples in that block. He considers various choices for the normalizing constant and reports good results, but he does not compare his results with that of nonadaptive methods. Schaming[22] has a similar approach except that he considers a different normalizing constant for each band[*] of a subpicture. He claims that the variances of the transform coefficients within each band of the transformed blocks are the same.[**] Thus, he makes the same bit assignment and uses the same normalizing constant for the elements in each band. A sample bit assignment for a block size of 8 x 8 for the Schaming method is shown in Figure 2-5. Schaming also considers an adaptive approach which allows him to change the number of bands that he would transmit for each block. This choice is made by measuring the energy in each band and stopping when the accumulated energy is a certain fraction of the total signal energy for that particular block. Schaming also reports good results but does not indicate the improvements due to using the adaptive approach compared to his nonadaptive method.



6	4	3	2	2	2	2	2
4	3	2	2	2	2	2	
3	2	2	2	2	2		
2	2	2	2	2			
2	2	2	2				
2	2	2					
2	2						
2							

Figure 2-5. Typical Bit Assignment for First 36 Coefficients for Schaming Method

2.3 ADAPTIVE PREDICTIVE CODING SYSTEMS

In predictive coding systems the correlated data is processed to generate a set of uncorrelated signals which is referred to as the differential signal. This signal is quantized by a memoryless quantizer. At the receiver the inverse of the predictor operation is performed to obtain a replica of the original signal from the quantized differential signal. In designing predictive coding systems the predictor and the quantizer are

[*] In a square block each band of samples is separated from the adjacent bands by a straight line running from North-east to South-west.

[**] This corresponds roughly to Tescher's method of scanning an image to form a sequence of one-dimensional signals discussed in Section 2.2.3.4.

optimized individually ignoring the effects of one on the other. This approach is due to the nonlinear nature of the quantizer that makes an overall system optimization impossible. The most commonly used form of predictive coding systems are DPCM and delta modulators. The DPCM system uses a linear predictor which predicts the value of an incoming signal based on a weighted sum of the adjacent elements. The quantizer is either a uniform or nonuniform quantizer that maps the differential signal to one of 2^m possible levels for a system using M bits per sample. A delta modulator is a simple form of a predictive coder where the quantizer is substituted by a comparator and the predictor is substituted by an integrator (summer). Block diagrams of both DPCM and delta modulators are shown in Figure 2-6.

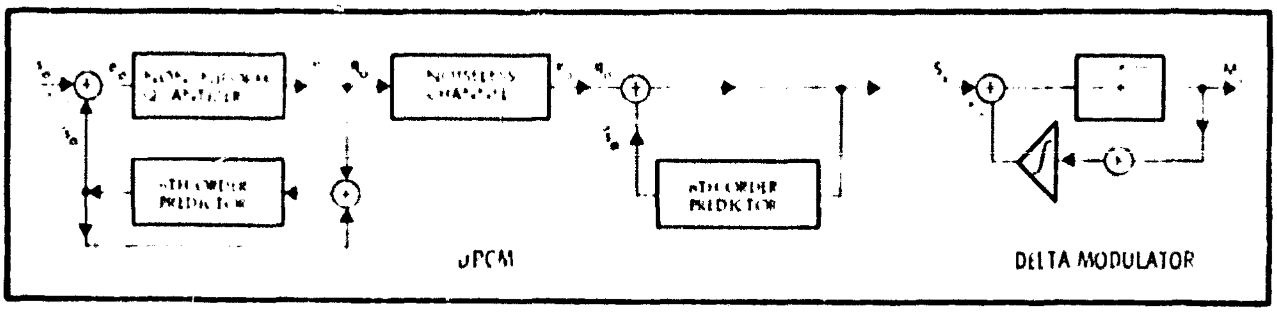


Figure 2-6. Block Diagrams of a DPCM and a Delta Modulator

2.3.1 Adaptive Delta Modulators

Adaptive delta modulators have received more attention than other adaptive systems for coding both speech and video signals. One reason for this may be that the performance of delta modulators improve significantly by making them adapt to signal statistics without introducing much additional complexity to its already simple design.

In a delta modulator each sample is compared to an estimate of it and a positive or negative signal is produced depending upon the comparative value of the incoming sample. The output of the comparator is multiplied by a constant in the feedback loop and is used as an input signal to an integrator whose output is the estimate of the incoming signal. The value of the constant in the feedback loop controls the step size of the delta modulator. A large step size introduces granular noise in the reconstructed signal over the region that the signal is changing gradually while a small step size limits the ability of the encoder to follow large signal variations, thus resulting in slope-overload noise. Typical response of a delta modulator showing both granular and slope-overload noise is shown in Figure 2-7. Both granular and slope-overload noise can be reduced by increasing the number of samples per second. However, this is not desirable since a higher sampling rate is equivalent to a higher bit rate. Instead, the system noise is reduced by using adaptive methods that do not increase the bit rate significantly.

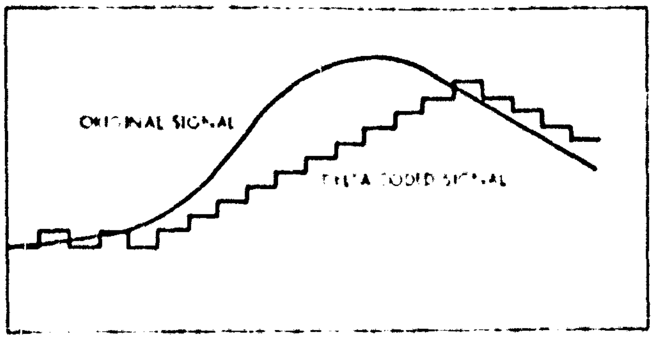


Figure 2-7. A Typical Response of Delta Modulator

Adaptive delta modulators have been considered by a number of authors in recent years. The most widely used approach is to change the step-size of the system according to signal variation. An indication of the type of signal variation is the polarity of the modulator output levels. Output levels of the same polarity indicate large signal variations where output levels of alternating polarities indicate smooth signal variations. Therefore, increasing the step size when output levels of like polarity are present and reducing it for outputs of unlike polarity reduces both granular and slope-overload noise. This type of adaptive system does not require additional overhead information for synchronizing the receiver with the transmitter. This general approach is used to develop a number of different adaptive delta modulators.[23,24,25] In the following sections we will discuss the three most basic systems. Each system has a number of variations which have been developed for various applications.

The first adaptive delta modulator we discuss uses the polarity of the three previous output levels to match the step size to signal variation. The system uses step sizes of ±1, ±2, and ±4 depending upon eight possible combinations for the polarity of the three past output levels. The performance of this delta modulator is studied in encoding monochrome images. Comparisons of the performance of this system with that of a nonadaptive delta modulator and a DPCM system are made[26] for a monochrome image.

A second approach, known as the SONG delta modulator, uses the step size of the past sample to form a step size for the present sample.[27] In this method the current step size Δ_k is generated as

$$\Delta_k = \begin{cases} 2D\, e_{k-1} & \text{for } |\Delta_{k-1}| < 2D \\ |\Delta_{k-1}| \left(e_{k-1} + \frac{1}{2} e_{k-2}\right) & \text{for } |\Delta_{k-1}| > 2D \end{cases} \qquad (2\text{-}7)$$

where D is a constant which is the minimum step size of the system and e_k is the sign of the comparator output for the kth sample. This system performs well for voice signals and its performance for video data is discussed in reference 28.

A third approach to adaptive delta modulators is one suggested by Jayant.[24] This is a modification of a system known as high information delta modulation.[29] This adaptive delta modulator has a 1-bit memory which stores the polarity of the previously transmitted output bit. The step size for the kth sample Δ_k changes for every sample as

$$\begin{aligned} \Delta_k &= P\, \Delta_{k-1} && \text{if } e_k = e_{k-1} \\ \Delta_k &= -\frac{1}{P}\, \Delta_{k-1} && \text{if } e_k \neq e_{k-1} \end{aligned} \qquad (2\text{-}8)$$

where P is a constant and e_k is the polarity of the output bit for the kth sample. The system is completely adaptive as the step size can increase or decrease arbitrarily and the receiver is in complete synchronization with the transmitter. Jayant used this system for coding both speech and pictorial signals. In both cases he found an optimal value of about 1.5 for the constant P using mean square error as the criterion of optimality. Jayant reports an improvement of about 10 dB in signal-to-noise ratio over nonadaptive delta modulators for pictorial data. This gain is larger than the improvements other researchers have reported for adaptive delta modulators.

2.3.2 Delta Modulators with a Variable Sampling Rate

An alternative approach to designing adaptive delta modulators is to use a variable sampling rate. This method is based on the fact that slowly varying signals convey less information than rapidly changing signals, thus fewer samples (or equivalently bits) are needed for their transmission. A delta modulator with variable sampling can be designed that reduces the granular noise due to a small step size and reduces the slope-overload noise due to large number of samples generated over regions of large variation. Hawkes and Simonpieri[30] considered an adaptive delta modulator in which, in addition to variable sampling rate, a variable step size was considered. Both the sampling rate and the step size were controlled by the polarity of the sequential output bit stream. Sequential polarities of the same sign, an indication of large signal variation, increase both sampling rate and step size where sequential polarities of opposite sign, reduces both the sampling rate and the step size. The receiver is synchronized with the transmitter at all times except for the presence of channel errors. The authors' treatment of their proposed system is rather brief and sketchy. The paper does not include either a subjective or an analytical comparison with nonadaptive systems.

2.3.3 Adaptive DPCM Encoders

In a DPCM system, the predictor uses a weighting of adjacent samples to generate an estimate of the incoming signal value. The difference between the predicted and the actual signal value, known as the differential signal, is encoded using a nonlinear quantizer. A nonlinear predictor involves conditional expectation values which are difficult to implement. Instead, in most proposed systems, a linear combination of adjacent samples is included. These weightings are related to the signal correlation by a set of algebraic equations. Experimental results have shown that using the adjacent sample along the same line, the adjacent sample on the previously scanned line, and the diagonal sample of the previously scanned line is sufficient for predicting the incoming sample for most pictorial data. DPCM systems have an advantage over PCM systems because the differential signal has a smaller variance and a well behaved probability density function which allows one to design a nonlinear quantizer matched to the error signal statistics.

DPCM systems using a fixed optimized predictor generate a well behaved stationary differential signal if the original data is stationary. The stationary differential signal can be encoded optimally using a nonlinear quantizer matched to its statistics. However, when the signal is nonstationary and the predictor parameters are fixed, a nonstationary differential signal results. Optimal encoding of the nonstationary differential signal then requires a variable quantizer which would change to accommodate the variations in the differential signal. In designing an adaptive DPCM system one must either use a predictor with variable parameters such that the parameters would change with the variations in the signal, thus generating a stationary differential signal, or one can use a fixed predictor with a variable quantizer to accommodate the resultant nonstationary differential signal. In addition to the two adaptive systems mentioned in the previous paragraph, the adaptivity can be incorporated in the system by using a variable sampling rate and fixing both the predictor and the quantizer.

2.3.3.1 DPCM Systems with Adaptive Predictors

In a DPCM system with an adaptive linear predictor the weightings on the adjacent samples used in predicting an incoming sample can change according to variations in the signal value. Atal and Schroeder[31] studied the performance of such an adaptive system for voice signals. Their proposed system included a 5 msec delay during which the incoming samples were stored in an input buffer and were used to obtain an estimate of the signal covariance matrix. The measured covariance matrix was used to obtain a set of weightings for the predictor. These values were then used for processing the stored signals. The updated values of the predictor coefficients are then transmitted to the receiver once every 5 msec. They used a variable predictor with a two-level quantizer and report good coding results. Although identical systems can be implemented for coding pictorial data, this type of system has not been reported in the open literature. Instead, researchers have used adaptive DPCM systems with a fixed and simple predictor and an adaptive quantizer.

2.3.3.2 DPCM Systems with Adaptive Quantizers

A DPCM system with a fixed predictor will have a nonstationary differential signal for nonstationary data. Using a fixed quantizer, the ronstationarity of the differential signal would cause an abnormal saturation or a frequent utilization of the smallest level in the quantizer. To remedy this situation, the threshold and the reconstruction levels of the quantizer must be made variable to expand and contract according to signal statistics. Adaptation of the quantizer to signal statistics is accomplished using various approaches. Virupaksha and O'Neal[32] suggested an adaptive DPCM system for speech signals that stores 25 samples of the differential signal to obtain an estimate for the local standard deviation of the signal. Then the stored signal is normalized by the estimated standard deviation and is quantized using a fixed quantizer. Naturally the scaling coefficient must be transmitted once for every 25 samples for receiver synchronization. Ready and Spencer[33] use a similar approach in a system called block-adaptive DPCM that they use for bandwidth compression of monochrome images. In block-adaptive DPCM systems, a block of M samples is stored and encoded by N possible quantizers. The total distortion for all M samples using each quantizer is calculated at the transmitter. The normalizing constant giving the smallest distortion is used to scale the samples in the block prior to their quantization and transmission. The system requires $(\log_2 n)/M$ binary digits per sample overhead information for receiver synchronization. Ready and Spencer use a two-dimensional DPCM system employing three adjacent samples in its predictor and use a block of 16 samples with four possible normalizing constants. They report an improvement of 36 percent reduction in bit rate over a similar nonadaptive DPCM system at about 2 bits per sample. The gain becomes smaller at higher bit rates.

A different approach, which has not appeared in the technical literature, is a DPCM system with a variable set of threshold and reconstruction levels. This is the self synchronizing approach described in the adaptive delta modulator discussion where the step size contracts and expands depending upon the polarity of sequential output levels. In a DPCM quantizer the set of threshold and reconstruction levels would contract and expand depending upon the sequential utilization of inner or outer levels of the quantizer. For instance, a variable quantizer can be designed where all reconstruction levels expand by a factor of P (for some optimum value of P) upon two sequential occurrences of the outermost level and they would contract by a factor of 1/P upon opposite sequential happening of the smallest level. This system has the advantage that it is completely adaptive and does not require any overhead information because the receiver is self-synchronizing.

2.3.4 Dual-Mode Predictive Coding Systems

In a dual-mode system two different coding methods are connected in parallel where each method is optimum for a particular type of signal variation. A sensor responds to the input signal variation and switches on the appropriate encoder. A dual-mode system using a DPCM and a delta modulator is proposed by Frei, Schindler, and Vettiger[34] for compressing monochrome imagery.

For regions where the video waveform is relatively smooth the coder operates in the delta mode, producing 1 bit per sample. The video waveform is first encoded by a fast delta coder producing a bit rate of $3f_s$ (f_s is the Nyquist sampling rate). Subsequently, groups of three delta bits are compressed to 1 bit. This compression is performed at the Nyquist rate according to the majority decision rule. Groups of three identical delta bits are treated separately. At the receiving end the compressed delta bits are reconstructed as indicated in the coding table. If the delta coder produces three equal bits within one sampling interval, the coder senses a deviation from a smooth region and switches from the delta mode into the DPCM mode. This mechanism provides for fast sensing of a transient and minimizes slope-overload noise. After the above condition (111 or 000) has been detected, the delta coder is disconnected and further encoding of the video waveform is performed by the DPCM coder.

Switching from the delta to the DPCM mode occurs when a sharp transient occurs in the video waveform. At the start of the transient (three 1s in one sampling interval), the coder switches to the DPCM mode and remains in the DPCM mode until a DPCM idling condition is sensed. The DPCM idling condition is defined by the occurrence of a smallest DPCM word (111 or 000) preceded by a DPCM word of opposite sign. Here the transmitter and receiver each switch to the delta mode of operation.

The state of the mode controller at the transmitter is uniquely determined: stay in delta until a sequence of three equal delta bits indicates a transition, then switch to DPCM and proceed in DPCM until the idling condition appears in the bit stream, then switch back to delta, and so forth. The operation of mode detection at the receiver, however, is ambiguous, since a sequence of three identical bits can also be produced by three compressed delta words. In order to avoid this ambiguity, redundant marker bits M are inserted in the compressed delta bit stream after a sequence of three equal compressed delta bits. After three identical delta bits indicating a mode transition (111 or 000), a marker bit M_1 of equal polarity is inserted to indicate that the mode has to be switched to DPCM. Since there might be up to two equal delta bits preceding the mode transition, an additional marker bit M_2 of opposite sign is inserted into the bit stream, in order to mark the proper position of the end of the transition sequence. The introduction of marker bits slightly increases the bit rate generated by a dual-mode coder.

The authors use the dual-mode system for coding both single frame and television signals. They report good subjective results at about 1.5 bits per picture element for single frames of monochrome imagery. Their result is further supported by an independent simulation of the dual-mode system at the University of Southern California.[35]

In addition to the above methods, other adaptive DPCM systems have been developed for various applications that use modifications of one of the above approaches. These methods are discussed in references 36, 37, and 38.

2.4 ADAPTIVE CLUSTER CODING METHODS

Unlike transform and DPCM coding techniques where the correlated data is processed to generate an uncorrelated signal prior to quantization and transmission, in cluster coding techniques the data is grouped to form a number of clusters. Each cluster is then represented by a number of variates which must be quantized and transmitted. In adaptive cluster coding methods, two processes can be made adaptive. One is the operation of the classifier, and the other is the operation of the quantizer. In the following sections we discuss two cluster coding methods. These are the blob algorithm proposed by Wintz and the multispectral cluster coding method proposed by Hilbert.[39,40,41]

2.4.1 Blob Algorithm

The blob algorithm developed by Wintz examines each four adjacent samples and uses a hypothesis testing method to decide if the samples are close enough to be joined with the adjacent samples or if they belong to a new class. Groups of samples having similar first and second moments are merged into blobs. In this manner the entire imagery is partitioned into blobs such that all picture elements within each blob have similar gray levels and textures.

Since elements in each blob are similar in both gray level and texture, one needs only the boundary information and one representative value for each blob for the reconstruction of that blob at the receiver. The boundary information regarding each blob can be extracted and encoded efficiently using contour tracing algorithms. Efficient contour tracing algorithms have been developed by various researchers in recent years.[42] The efficiency of a contour tracing algorithm depends upon the average number of picture elements in each contour. The larger the average number of pixels per contour, the higher the efficiency of the contour tracing algorithm. Application of the contour tracing algorithm to the blobs of data generated with the blob algorithm results in a more efficient coder because the number of picture elements per contour has been increased.

The blob algorithm is an adaptive classification algorithm because it partitions the entire image rather than image blocks. Using a variable-length coding scheme, such as run-length or Huffman codes for encoding directional information of contours, one would have a fully adaptive data compression system.

2.4.2 Multispectral Cluster Coding Technique

Multispectral clustering has been used for bandwidth compression and classification of multispectral data.[41,42,43] In the case of LANDSAT multispectral data, for each picture element we obtain four measurements: recorded energies in each of four narrow spectral bands. The four measurement values are represented by a 4-tuple in a four-dimensional space which is called the measurement space. Thus, picture elements recording approximately the same energy in corresponding bands will be represented by points which lie close together in the measurement space.

The location in four-dimensional space of the center of the mass of the sample values belonging to a cluster is called the centroid of that cluster. Clusters are generated as follows: initial centroid values are arbitrarily assumed. Then all samples are assigned to the cluster containing the closest centroid. The centroids of each cluster are then replaced by the center of mass of samples in that cluster. This procedure is iterated until a small percentage of samples changes cluster.

Identifying all the picture elements corresponding to one centroid with one class, one obtains a classified image which consists of as many gray levels as there are classes in the measurement space. For n possible classes this image requires $\log_2 n$ binary digits per sample for its transmission using a PCM system. The centroid values require 28 (4 x 7) bits per block for each centroid.

In the adaptive cluster coding algorithm, the multispectral data is first divided into small blocks of fixed size (e.g., 16 x 16 picture elements). Then the elements in each block are clustered into a variable number of classes. Two clusters are grouped into one cluster if the distance in the measurement space between their centroids is less than some threshold. Using a fixed threshold value generates a larger number of classes in blocks with high details and a small number of classes in blocks with low details. For each block the classified image is transmitted, along with the centroids.

The receiver reconstructs each block of the multispectral imagery by generating the individual bands in each block from the classified image and the corresponding centroids of those clusters. The procedure is to examine each point in the classified image and specify to what class it belongs. Then individual bands corresponding to the particular picture location are reconstructed by choosing their values equal to the centroid of that particular class.

Bandwidth compression is achieved if the number of binary digits needed to transmit the classified image and the centroids are smaller than the number of digits required to represent the multispectral data. Experimental results using a fixed number of classes in each block and PCM coding of the classified image and centroids have shown that large compression ratios can be obtained at relatively low distortion levels.[43]

Additional compression can be obtained by using an adaptive method, such as an entropy coding technique, to encode the centroids and the classified imagery. Entropy coding takes advantage of the fact that adjacent elements tend to belong to the same cluster.

2.5 ADAPTIVE ENTROPY CODING AND RELATED TECHNIQUES

To utilize entropy coding methods in compressing the bandwidth of image data, one can measure the histogram of the data and use the optimum code word for the distribution of the gray levels. This method is efficient in reducing the bandwidth of the imagery if the distribution of data samples is peaked at certain gray levels. For a uniform histogram the entropy coding does not offer any advantage over the PCM system. To make an effective use of entropy coding, one may process the data, through a reversible process, such that the histogram.of the output samples has maximum nonuniformity. One such process is the operation of differencing of adjacent elements. Since spatial correlation exists in most imagery, the difference between adjacent elements tends to be small. Hence, the frequency of small difference values is much larger than the frequency of large difference values. Experiments with various types of imagery have shown that the histogram of the difference signal is a highly peaked double-sided exponential function about zero.[26,44] A modified form of the difference entropy coding method is used for compressing the bandwidth of weather pictures resulting in a compression ratio of about 2:1 involving absolutely no degradation. This system uses entropy coding for transmitting differences that are smaller than a prespecified value and uses a PCM code word for transmitting the actual value of the samples if the difference is larger than the prespecified value. Similar results have also been obtained using differential entropy coding and adaptive differential entropy coding at TRW for earth resources multispectral data.[45]

Differencing is not the only process that can be used to increase nonuniformity of the histogram of data samples. Other operations, such as transforming the data using Hadamard or Haar transforms, can be used for this purpose.

Rice[46] uses a sequence of 2 x 2 Hadamard transforms on imagery divided into blocks of 32 x 32 samples. At each stage of transformation the Hadamard components corresponding to the mean of the 2 x 2 samples at the input of the transform stage are passed through the following stage of transformation where the other components are encoded directly. A block size of 64 x 64 samples requires six stages of transformation. The transform coefficients at the output of the final stage of transformation, as well as intermediate transform coefficients, are encoded using the adaptive entropy coding method and are transmitted. Rice reports good coding results at various compression ratios using monochrome imagery.

In adaptive entropy coding one assumes that the statistics of an image change significantly from block to block so that the code words which are optimum for one block do not perform as well for the others. To adapt to the changing statistics one must change the code words accordingly. This is a difficult task since, in addition to measuring the statistics for each block and assigning the code words, one must transmit this information to the receiver so that individual blocks are decoded properly. An alternate approach is to classify each block of data to one of a limited number of classes. Then the proper code word for that class is used to encode the data in that block. Of course, additional information is required to inform the receiver which class of code words have to be used for decoding each individual block This overhead information is small for small number of classes and for large block sizes.

May and Spencer[45] use an adaptive entropy coding method for multispectral data where the spatial-spectral difference signal is encoded using the Huffman code calculated for the statistics of the previous line. This does not require transmitting the code table for each line since it can be calculated at the receiver as well. The authors report an improvement of about 10 percent reduction in the bit rate over nonadaptive Huffman codes.

Rice and Plaunt[47] developed a variable length coding system which is strictly information preserving. Operating on a sequence of source symbols, the Rice machine adapts by selecting one of three coding schemes with computational capability for optimally switching to that one of the three codes which is compatible with the data activity. Code $\overline{FS}$ performs well with low data activity, code FS performs well for data of medium activity, and code CFS performs best with very active data. To adapt to rapid changes in activity, the basic Rice compressor monitors data activity and selects the appropriate code mode based on blocks of 21 data symbols.

The resulting coding system operating on a line-to-line basis produces output rates within 0.3 bit per sample of the one-dimensional entropy of the samples and cannot expand the data by more than 0.1 bit per sample under any circumstance. Rice used this method for coding monochrome images using first order differences between adjacent samples on each line. May and Spencer[45] utilized the Rice machine for coding multispectral data using first order spatial-spectral differences as the input sequence. For multispectral imagery the performance of the Rice machine is the same as that of the adaptive Huffman technique which uses the previous line statistics to generate a new coding table for each line.

A general treatment of universal codes (block codes which adapt to obtain a performance measure arbitrarily close to the signal entropy with increasing block length) is given by Davisson in reference 44.

3. SELECTION OF TECHNIQUES RELEVANT TO MULTISPECTRAL IMAGERY

In this section adaptive compression techniques appropriate for MSS imagery are selected from those surveyed in Section 2. The rationale for rejecting the less appropriate techniques is discussed in Section 3.1. The selected techniques and detailed algorithms are described in the remaining sections. These techniques have been simulated on a computer and their performances on a LANDSAT-A image are evaluated and compared.

3.1 REVIEW AND SELECTION OF ADAPTIVE ALGORITHMS FOR MSS IMAGERY

In the following paragraphs, the adaptive compression techniques surveyed in Section 2 (summarized in Figure 2-2) are reviewed individually, and those most appropriate for on-board compression of multispectral imagery are selected. In selecting the candidate adaptive techniques, we have relied heavily on the characteristics of the multispectral data and our experience on bandwidth compression of the MSS data using nonadaptive techniques. Also, special emphasis is placed on implementation complexity of these techniques and compatibility with Thematic Mapper requirements. Finally, the reasons for rejecting adaptive techniques or a class of adaptive techniques are discussed in the following sections. The surviving candidate adaptive techniques are listed in Figure 3-1.

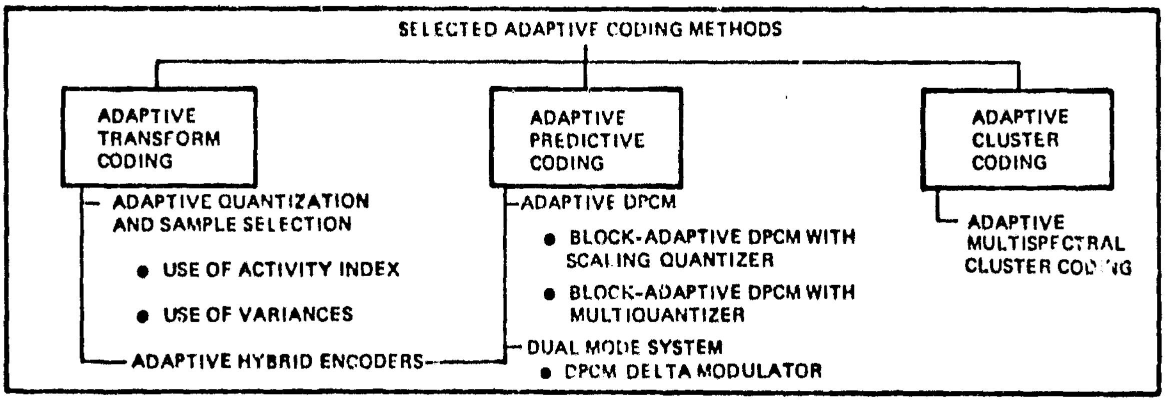

SELECTED ADAPTIVE CODING METHODS
ADAPTIVE TRANSFORM CODING
ADAPTIVE PREDICTIVE CODING
ADAPTIVE CLUSTER CODING
ADAPTIVE QUANTIZATION AND SAMPLE SELECTION
• USE OF ACTIVITY INDEX
• USE OF VARIANCES
ADAPTIVE HYBRID ENCODERS
ADAPTIVE DPCM
• BLOCK-ADAPTIVE DPCM WITH SCALING QUANTIZER
• BLOCK-ADAPTIVE DPCM WITH MULTIQUANTIZER
DUAL MODE SYSTEM
• DPCM DELTA MODULATOR
ADAPTIVE MULTISPECTRAL CLUSTER CODING

Figure 3-1. Surviving Candidate Adaptive Techniques

3.1.1 Adaptive Transform Coding Methods

Among adaptive transform coding methods, we choose the adaptive transform coding methods that use a fixed transformation with a variable sample selec⁺ion and quantization. These are adaptive methods that use an "activit, index" for sample selection and the methods that use recursive quantization in the transform domain.

3.1.1.1 Rejection of Adaptive Methods Using Variable Sampling Rate

The possibility of using a variable sampling rate with transform coding methods is discussed in Section 2.1 of the literature review. It was pointed out that use of adaptive transform coding systems utilizing a variable sampling rate has not been reported in the technical literature. This is partially due to the difficulty of controlling the sampling rate at different regions of the signal and the synchronization problems. In on-board compression of multispectral data, an additional prcblem is that the data is already scanned and sampled. Introducing a variable scanning and sampling mechanism in the system without ample justification is not advisable. For these reasons the idea of variable sampling rate with transform coding systems is rejected.

3.1.1.2 Rejection of the Adaptive KL Transform

In a practical adaptive KL transform, the data is classified into a number of classes and a different set of basis matrices is used for each class. Here the main problem is classification of the incoming set of picture elements to the particular class. Only one such method has been proposed in the literature[11] which is very complicated to implement. In addition, the gains reported using this adaptive method are almost the same as the gains using other adaptive methods we have studied. For these reasons, the adaptive KL transform has not been selected.

3.1.1.3 Rejection of Threshold Quantization Techniques

Due to the complexity of adaptive sampling and adaptive KL transforming of imagery data, researchers have developed adaptive methods where a fixed transformation and a fixed sampling rate are used; however, the operation of sample selection and quantization is made adaptive. One such system uses a fixed threshold value. Only samples larger than this threshold are quantized and transmitted. This coding technique, known as threshold quantization, will not be considered in this study because its performance, as discussed in literature survey, does not offer any improvement over nonadaptive transform coding methods.

3.1.2 Adaptive Predictive Coding

Adaptive predictive coding systems are classified into three groups: adaptive delta modulators, adaptive DPCM, and dual-mode systems.

Adaptive delta modulators are not considered in this study for two reasons:

1) Adaptive delta modulators utilize a binary comparator for the quantizer, and thus operate at a bit rate of 1 bit per sample. At this bit rate the signal fidelity is low. To improve the system performance, one needs to increase the sampling rate. In applications dealing with multispectral earth resources imagery, the sampling rate is fixed. A variable sampling rate increases the system complexity and its recommendation needs ample justification which is not present.

2) The performance of adaptive delta modulators is in general inferior to the performance of the two-dimensional DPCM systems. This is established in reference 48 for the "girl" image.

Adaptive DPCM systems are categorized into two classes; in the first class are systems using adaptive predictors, and in the second class are systems that use adaptive quantizers. Systems using adaptive predictors have not been studied for coding imagery data. This is due to the fact that the quality of coded imagery is more sensitive to the variations in the structure of the quantizer than it is to variations in the weightings used in the predictor. In add tion, it is much simpler to design adaptive quantizers than it is to design adaptive predictors. For these two reasons, we concentrate our efforts on DPCM systems with adaptive quantizers.

A number of DPCM systems with adaptive quantizers were surveyed in Section 2.3. Two block-adaptive DPCM systems were discussed. The system suggested by Ready and Spencer uses multiple prediction loops while the system suggested by Virupaksha and O'Neal uses a single primary loop for encoding and a secondary loop for calculating the gain constant of the quantizer in the primary loop. Both systems have similar performance. The block adaptive system with multiple loops is inherently more complex and is rejected in favor of the block adaptive DPCM system using a single loop.

3.1.3 Adaptive Cluster Coding Methods

Among the adaptive cluster coding methods listed in Figure 2-2, we select adaptive multispectral clustering for further investigation. The BLOB algorithm was rejected because in its present form,[39] it is not optimal for multispectral data; it could be extended to multispectral data but this makes the technique extremely complicated.

3.1.4 Adaptive Entropy Coding

A study of entropy coding and adaptive entropy coding methods for compressing the bandwidth of MSS data was performed at TRW. The salient points of the study were that the entropy coding methods can be used to compress the bandwidth of MSS data by a factor of about 2:1 without any degradation, and the compression technique can be fabricated for on-board processing using an acceptable number of parts and within reasonable weight, power, and size limitations. For details of the study, the reader is referred to reference 39. Because of this comprehensive study, we will not consider entropy coding methods in compressing the bandwidth of MSS data individually. Instead, entropy coding methods will be considered in concatenation with other bandwidth compression techniques. This is because entropy coding further reduces the bit rate of a bandwidth compression technique that generates an uneven distribution of symbols at its output. Here the object is to study the improvements in the performance of each bandwidth compression technique as a result of concatenating it with an entropy coding method. For our application, the Huffman encoder will be used because of its simplicity as compared to other entropy coding techniques.

3.2 DESCRIPTION OF SELECTED TECHNIQUES

In this section the selected techniques are described and their performance on the green band of a sample LANDSAT-A image is evaluated.

3.2.1 Adaptive Transform Coding

Among the many adaptive two-dimensional transform techniques, we have selected two for further study. The first utilizes recursive quantization on the transform coefficients, while the second uses an activity index to sort the transform blocks into various classes.

3.2.1.1 Adaptive Transform Coding Using Recursive Quantization

This method uses a fixed transformation and a fixed block size. The number of coefficients selected in each block for transmission and the number of bits assigned to each coefficient changes from block to block. We propose dividing the image into blocks of 16 x 16 samples. After each block is two-dimensionally transformed using either a Hadamard or Cosine transform, a scanning of the transformed samples is performed to convert the two-dimensional data to a one-dimensional format. We use the same scanning method suggested by Tescher.[21] Next, a first-order recursive relation is used to estimate the variance of each transform coefficient as

$$\hat{\alpha}^2_{i+1} = w\,\hat{\alpha}^2_i + (1 - w)\,\hat{X}^2_i \qquad (3\text{-}1)$$

where $\hat{X}_i$ refers to the ith coefficient in the transform domain after it is quantized, $\hat{\alpha}_i^2$ is the estimated variance of $\hat{X}_i$, and w is the weighting coefficient. Experiments with different values of weighting coefficients have shown that a weighting coefficient of 0.75 can be used with both the Hadamard and Cosine transforms. A block diagram of this technique is shown in Figure 3-2. The system parameters used with this technique are:

a) Type of transformation -- the utility of various types of orthogonal transforms in coding imagery data has been considered by various authors. The results clearly point out that the difference in the performance among various transforms are minimal, and the choice in a particular application should be based on implementational constraints. In this study we use the Hadamard and Cosine transforms.

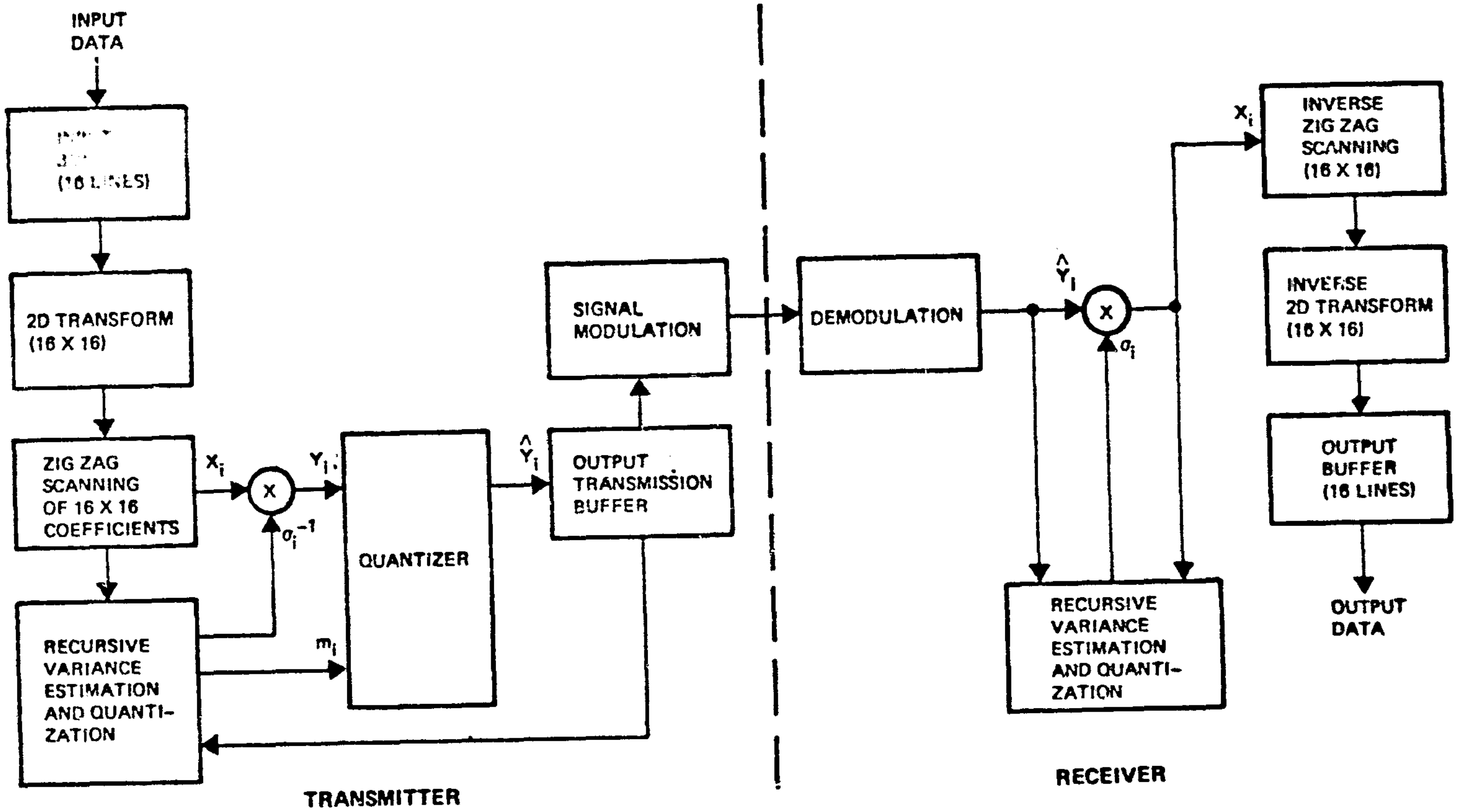

INPUT DATA
(16 LINES)
2D TRANSFORM (16 X 16)
ZIG ZAG SCANNING OF 16 X 16 COEFFICIENTS
RECURSIVE VARIANCE ESTIMATION AND QUANTI-ZATION
X_i
X
σ_i^{-1}
Y_i
m_i
QUANTIZER
$\hat{Y}_i$
OUTPUT TRANSMISSION BUFFER
SIGNAL MODULATION
TRANSMITTER
DEMODULATION
$\hat{Y}_i$
X
σ_i
RECURSIVE VARIANCE ESTIMATION AND QUANTI-ZATION
RECEIVER
X_i
INVERSE ZIG ZAG SCANNING (16 X 16)
INVERSE 2D TRANSFORM (16 X 16)
OUTPUT BUFFER (16 LINES)
OUTPUT DATA

Figure 3-2. Adaptive Transform Coding Using Recursive Quantization

It is anticipated that Cosine transform will give a better result, but the difference should be minimal. The reason for considering these two transforms is that the Hadamard transform is simpler to implement with digital circuitry, while the Cosine transform can be implemented using analog transversal filters utilizing the Chirp-Z transform (CZT) technology.[50] A feeling for the relative complexity of the Hadamard transform and the Cosine transform can be obtained by examining the number of multiplies and adds required for each. A Hadamard transform of N real elements requires $N \log_2 N$ additions. According to W. Chen, in some work that is to be published this summer the corresponding Cosine transform can be implemented with $N \log_2 N - 3N/2 + 4$ real multiplications and $3N/2 (\log_2 N - 1) + 2$ real additions. Thus for N equal to 16, the Hadamard transform requires 64 additions and the Cosine transform requires 44 multiplications and 74 additions.

b) Bit assignment – the number of binary digits assigned to each coefficient is obtained from*

$$m_i = \text{Integer}\left[\frac{1}{2} \log_2 \frac{\hat{\alpha}_i^2}{D} + \frac{1}{2}\right] \qquad (3\text{-}2)$$

where D is the theoretical value of the distortion that results by quantizing the variate with m_i digits. In an actual system, D is controlled by the output buffer. An empty buffer results in small values for D, which in turn increases the output bit rate where a full buffer increases D and reduces the output bit rate. When the variance of a coefficient is so small that the number of binary digits assigned for its quantization falls below 1.0 bit, the processor stops and the remaining coefficients in that block are not transmitted. They are substituted with zeros at the receiver. The system is adaptive since more binary digits are assigned automatically to blocks with larger transform coefficients.

c) Quantization of initial conditions and the dc coefficient – the initial condition for the recursive relation (3-1) is obtained by taking the first four ac coefficients and averaging the square of these quantities to obtain α_2^2. This quantity is then quantized to obtain $\hat{\alpha}_2^2$ using a quantizer with 32 levels, and is

*This bit assignment differs from the optimal allocation of reference 1 because that equation assumes the total number of bits for all coefficients is known. In the adaptive case the total number of bits is now known.

transmitted. At the receiver, $\hat{\alpha}_2^2$ is used to start recursive relation (3-1). The dc level is quantized with a uniform quantizer using 256 levels. This requires a total of 13 bits of overhead information for each block of 16 x 16 samples.

d) Quantization of ac coefficients – the actual quantization of the ac coefficients is performed as shown in Figure 3-2. Each coefficient X_i is normalized by dividing it by $\hat{\alpha}_i$ before quantizing it with a fixed quantizer. At the receiver, each $\hat{X}_i$ is multiplied by $\hat{\alpha}_i$ before inverse scanning and inverse two-dimensional transformation. The fixed quantizer can be a uniform or a nonuniform quantizer. In our simulations, we have used a nonuniform quantizer designed for a Laplacian distribution. Empirical results show that the probability density function of the ac transform coefficients are best modeled by a Laplacian function. This is particularly so for block sizes smaller than 16.* Although Max's quantizer results in minimum quantization error, an instantaneous companding quantizer (ICQ) can be substituted with almost identical results. We have used an ICQ for software simplicity. The operation of the ICQ is shown in Figure 3-3. A mapping $g(\cdot)$ of the transform coefficients results in a signal z with a uniform probability density function which is quantized with a uniform quantizer. The output of the quantizer must pass through a mapping $g^{-1}(\cdot)$ for signal recovery.[51] The mapping $g(\cdot)$ for an exponential probability density function is defined as

$$z = g(x) = \frac{X_o\,[1 - \exp(-Mx/X_o)]}{1 - \exp(-M)} \; ; \; z\,(-e) = -\,z\,(e) \qquad (3\text{-}3)$$

where we choose

$$M = \frac{\sqrt{2}X_o}{3\sigma}$$

and

$$X_o = 3\sigma$$

The quantizer is designed for a unit standard deviation and the input and output of the quantizer is scaled as shown in Figure 3-3.

*For large block sizes, the histogram of the transform coefficients is better fit with a gaussian function.

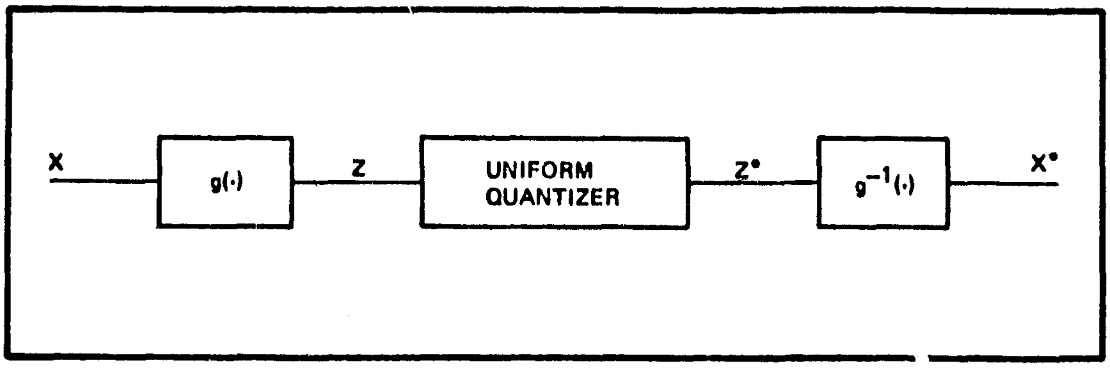


Figure 3-3. Instantaneous Companding Quantizer (ICQ)

The adaptive transform coding technique using recursive quantization is selected as a candidate technique because of its adaptibility and the excellent results reported in the literature using this technique on monochrome imagery.[21]

The adaptive transform coding system suggested by Schaming (discussed in Section 2.2.4 of literature survey) is a special case of the system suggested by Cox and Tescher. Schaming considers transmitting all the elements in each band as a group where in Tescher's method transformed samples in each band are considered individually. Besides, Schaming uses a fixed number of bits for a given coefficient where in this technique it changes depending upon the estimated sample variance.

3.2.1.2 Adaptive Transform Coding Using the Activity Index

This technique uses a fixed transformation such as the Hadamard or Cosine transform to obtain the two-dimensional transformation of each block of imagery. The sum of the squares of the transform coefficients, except for the dc term, is used to classify that particular block into one of M distinct classes. For each class, a particular pattern of sample selection and quantization is specified. This information is also available at the receiver; therefore, only $\log_2 M$ bits of information per block is needed to specify what class each block of data belongs. This adaptive method is simple and seems to offer distinct advantages over the nonadaptive transform coding methods. The block diagram of the encoder for this technique is shown in Figure 3-4.

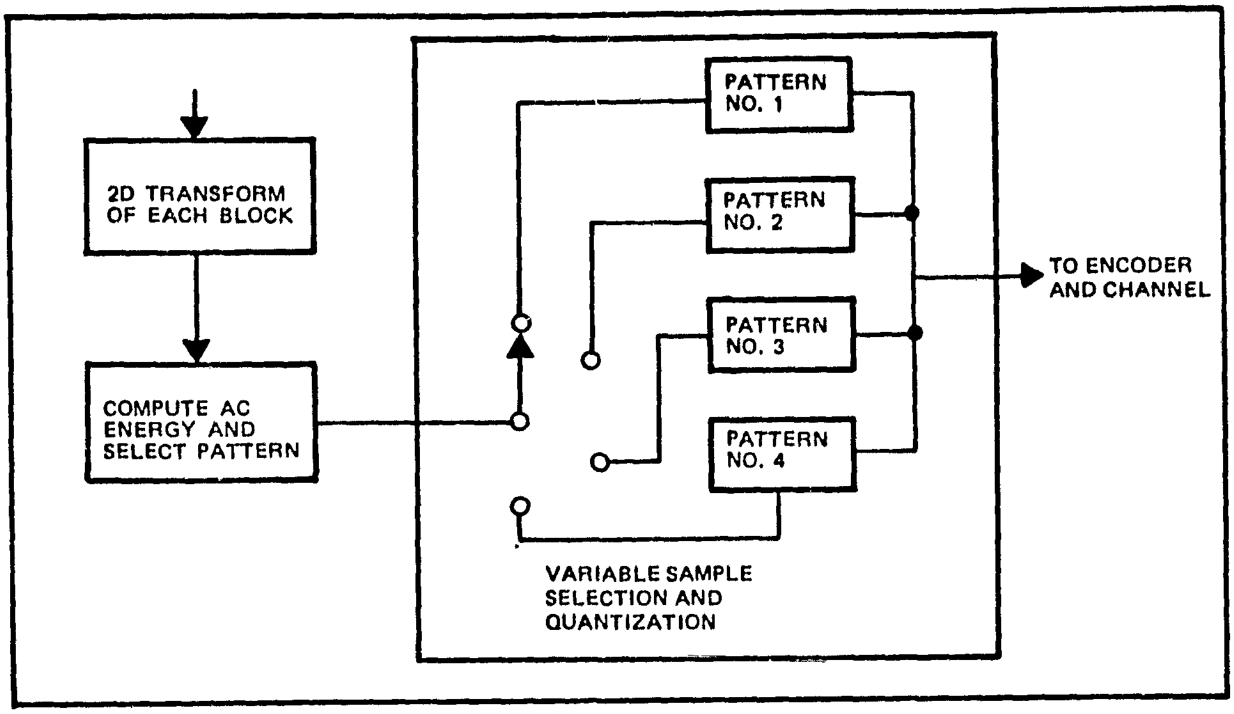

2D TRANSFORM OF EACH BLOCK
COMPUTE AC ENERGY AND SELECT PATTERN
PATTERN NO. 1
PATTERN NO. 2
PATTERN NO. 3
PATTERN NO. 4
TO ENCODER AND CHANNEL
VARIABLE SAMPLE SELECTION AND QUANTIZATION

Figure 3-4. Adaptive Transform Coding Technique Using Four Quantization Classes

The system parameters to be considered with this adaptive method are:

a) Type of transforms – for this adaptive coding technique, we also use Hadamard or Cosine transforms. The reasons for choosing these transforms over other transforms are the same as those discussed in the previous section.

b) Number of classes (M) – the number of distinct classes into which various blocks are classified is an indication of the level of system adaptivity. The larger the number of the classes, the more adaptive the system. However, a large number of classes increases the implementation complexity and increases overhead information. Authors suggesting use of an activity index have proposed using four possible classes for sample selection and quantization.[15,16]

c) <u>Bit assignment of different classes</u> – the optimal bit assignment procedure for one block of imagery is one that gives the same distortion for each coefficient.* When a number of classes are jointly considered, it is desirable to have a bit pattern making the distortion for each coefficient in the block as well as the total distortion in each block uniform. This will result in a uniform distortion over the whole image. Unfortunately, due to the nonstationarity of imagery data, such a bit pattern cannot be specified theoretically. We have used the same bit assignment pattern for both the Hadamard and Cosine transform. These patterns for the four classes are shown in Figure 3-5. These patterns were selected by simulating the system on the computer and experimenting with various bit patterns to minimize the total mean square error.

7	3	2	2	1	0	0	0
3	2	2	1	0	0	0	0
2	2	1	0	0	0	0	0
2	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

CLASS 1

7	4	3	2	2	2	1	1
4	3	2	2	2	1	1	0
3	2	2	2	1	1	0	0
2	2	2	1	1	0	0	0
2	2	1	1	0	0	0	0
2	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

CLASS 2

7	5	4	3	3	2	2	2
5	4	3	3	2	2	2	1
4	3	3	2	2	2	1	1
3	3	2	2	2	1	1	1
3	2	2	2	1	1	1	0
2	2	2	1	1	1	0	0
2	2	1	1	1	0	0	0
2	1	1	1	0	0	0	0

CLASS 3

7	6	5	4	3	3	2	2
6	5	4	3	3	2	2	2
5	4	3	3	2	2	2	2
4	3	3	2	2	2	2	2
3	3	2	2	2	2	2	2
3	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2

CLASS 4

Figure 3-5. Bit Assignment Pattern for Four Quantization Classes

*The coefficients with a variance smaller than or equal to this distortion are not transmitted.

d) Block size — the size of the blocks that the imagery should be divided into before processing is a crucial parameter. In adaptive methods, a small block size utilizes nonstationarity of the data more efficiently than a larger block size, and is simpler to implement. A large block size, on the other hand, utilizes a larger fraction of image correlation. One other problem with small block sizes is that one needs a minimum number of binary digits per block to eliminate the block structure in the reconstructed imagery. Using small block sizes, one needs to assign this minimum number of bits to the less active blocks which in turn results in having less binary digits available for the more active blocks. Based on our experience with nonadaptive methods and the results relating to adaptive methods reported in the literature, a block size of 8 x 8 seems to be the most appropriate choice.

e) Threshold levels for partitioning ac energy — after the ac energy of each block of 8 x 8 coefficients in the transform domain is calculated, one must choose three threshold levels. These levels are then used to classify each block to one of four categories. Histograms of the ac energy of the blocks can be used to choose these threshold values such that the average bit rate of the output data is a prespecified value. However, since the MSS data is grossly nonstationary, one would need an extremely large buffer memory to achieve a fixed output bit rate using prefixed threshold values. We have simplified the problem by choosing the threshold values that are linearly related when one parameter controls the output bit rate. In the actual system, this parameter is controlled by the fullness of the buffer memory. An empty buffer will reduce the size of this parameter, thus increasing the bit rate. A full buffer increases the size of this parameter and results in a lower bit rate.

f) Quantization of coefficients in the transform domain - the bit assignment for different classes was discussed in part c) of this section. Each coefficient is quantized using the number of binary digits that is assigned to it according to its position in a block, the classification of that particular position in a block, and the classification of the particular block. To quantize these coefficients, one must utilize the variance of each coefficient in the structure of the quantizer. This is accomplished by scanning each block in the transform domain using a helical pattern and using the recursive relation of (3-1) to find an estimate for the variance of each coefficient. Then the difference between this method and recursive quantization is that the number of binary digits for each coefficient position for each block is fixed, where in the other technique it varies according to the size of estimated variances. The structure of the simulated quantizer is the same as shown in Figure 3-3.

3.2.1.3 Comparison of Adaptive Transform Coding Techniques

The performance of the two adaptive transform coding techniques is evaluated by encoding the green band of a sample LANDSAT-A image. The simulated results for adaptive transform coding using recursive quantization are shown in Figure 3-6 for a two-dimensional Hadamard transform and a block size of 16 x 16. The performance is in terms of mean square error for only the green band of the MSS data. This figure also shows the performance of the adaptive transform coding technique using the activity index. With this technique, the two-dimensional transform of each image block is obtained, and the ac energy of the transform coefficients is calculated. The ac energy is then compared to three fixed threshold values, and depending on its comparative value, a particular bit assignment pattern and quantization are employed. We have used a linear spacing of the threshold values. The sample selection and the quantization pattern are shown in Figure 3-5. This pattern is used for both Hadamard and Cosine transforms. The performance of the nonadaptive two-dimensional Hadamard transform is shown in Figure 3-6. The adaptive transform encoder using recursive quantization is superior to the other two techniques.

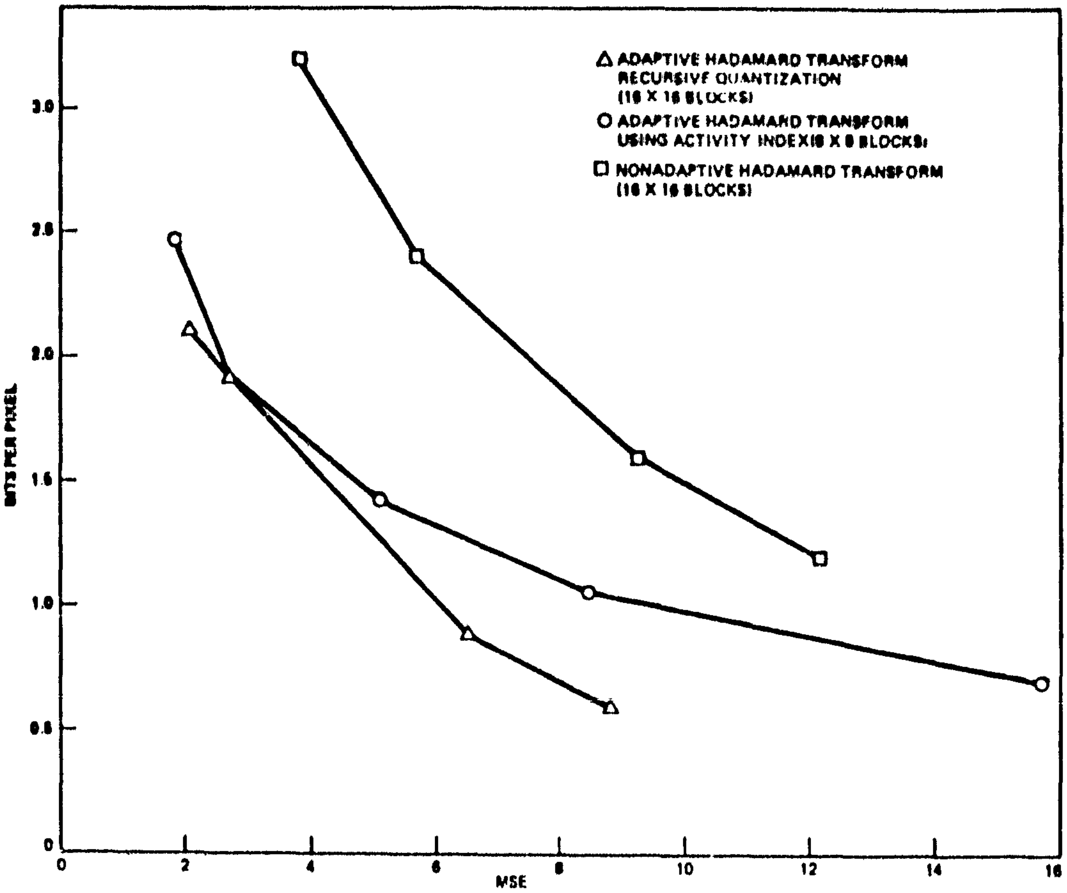


Figure 3-6. Adaptive Transform Coding Comparison

Figure 3-7 compares the performance of the adaptive transform coding techniques using activity index and recursive quantization for Cosine and Hadamard transforms. This figure shows the small improvement at lower bit rates that results by using a Cosine instead of the Hadamard transform. This small gain, however, does not justify the additional complexity of the Cosine transform.

3.2.2 Adaptive Predictive Coding

Among adaptive predictive coding techniques, we have selected block adaptive DPCM, adaptive DPCM with a scaling quantizer, and the dual-mode encoder that combines DPCM with a delta modulator.

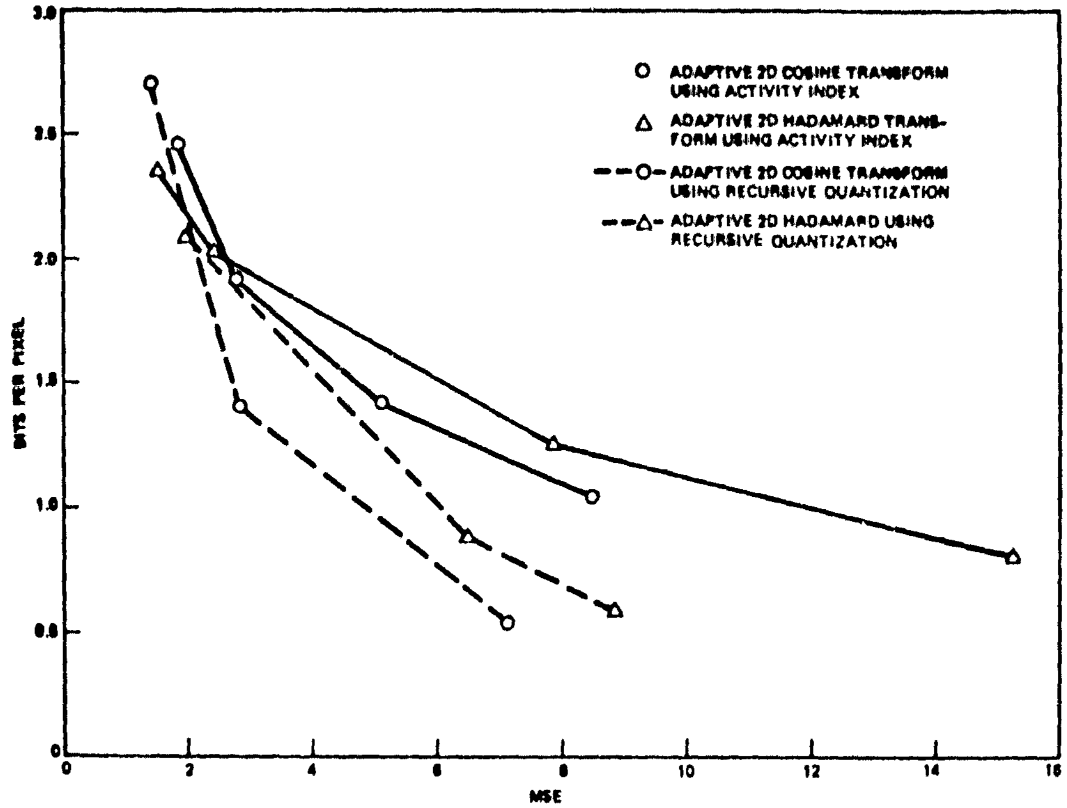


Figure 3-7. Cosine Versus Hadamard Transform Performance

3.2.2.1 Block-Adaptive DPCM System

A block diagram of the encoder is shown in Figure 3-8. The DPCM loop uses a third order predictor that utilizes the adjacent element in the same line, the adjacent element in the same column, and the diagonal element to predict each sample. The quantizer chooses one of M gain values for each block. A block of 16 samples is used in the gain computation loop (without a quantizer) to generate an estimate for the variance of the differential signal as shown in Figure 3-8. Depending upon the value of the variance, one of M gain factors is selected and used to scale the quantizer characteristic in the prediction loop. For M = 8, this requires 3/16th of a bit per sample for transmitting this overhead information.

3.2.2.2 Adaptive DPCM with a Self-Scaling Quantizer

This technique, which has not appeared in the technical literature, employs a DPCM loop with a variable set of quantizer threshold and reconstruction levels. It is a self-synchronizing system where the step size contracts and expands depending upon the polarity of sequential output levels. In a DPCM quantizer, the set of threshold and reconstruction levels would contract or expand depending upon the sequential utilization of the inner or outer levels of the quantizer. In this study, a variable quantizer is designed where all reconstruction levels expand by a factor of P upon two sequential happenings of the outermost level and contract by a factor of 1/P upon two sequential happenings of the smallest levels. This system has the advantage that it is completely adaptive and does not require overhead information because the receiver is self synchronizing. A P value of 1.5 is used in the simulation because it gives the optimum result for adaptive delta modulators.

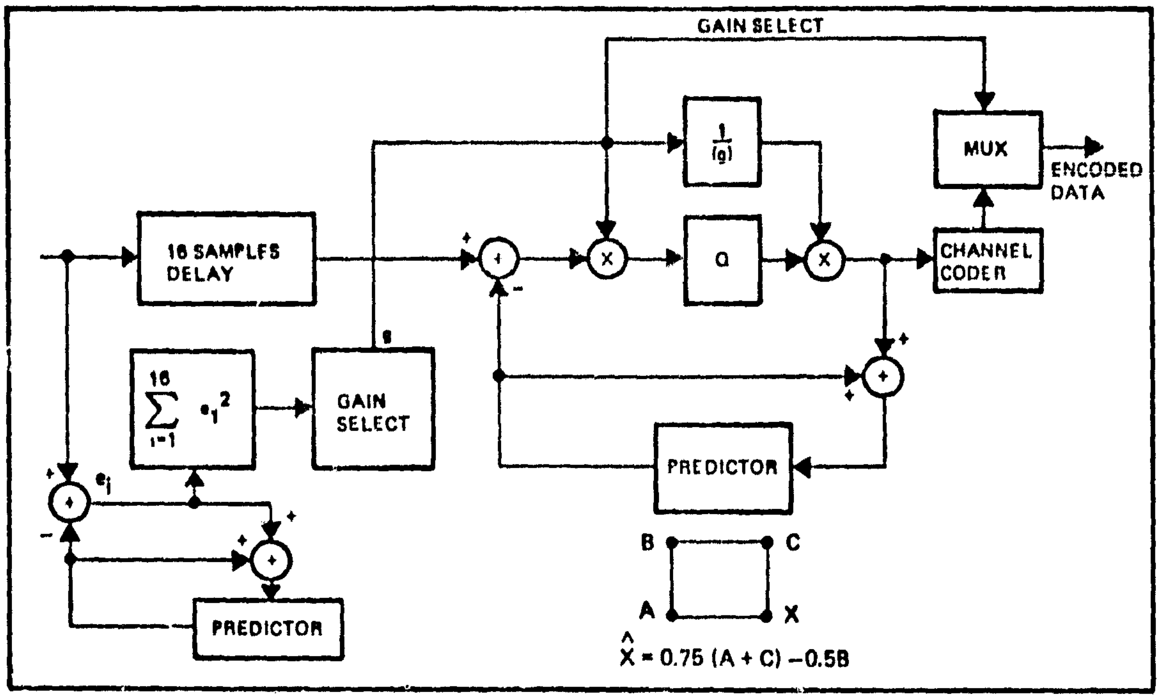


Figure 3-8. Block Adaptive DPCM

3.2.2.3 Dual-Mode Predictive Coding System

The dual-mode system uses a DPCM loop and a delta modulator. For regions where the video waveform is relatively smooth, the dual-mode system operates in the delta mode, producing 1 bit per sample. The video waveform is first encoded by a fast delta coder producing a bit rate of $3f_s$* (f_s is the Nyquist sampling rate). Subsequently, groups of three delta bits are compressed to 1 bit. This compression is performed at the Nyquist rate according to a majority decision, as can be seen from the coding table in Figure 3-9. Groups of three identical delta bits are treated separately. At the receiving end the compressed delta bits are reconstructed as indicated in the coding table. If the delta coder produces three equal bits within one sampling interval, the coder senses a deviation from a smooth region and switches from the delta mode into the DPCM mode. This mechanism provides for fast sensing of a transient and avoids slope overload noise. After the above condition (111 or 000) has been detected, the delta coder is disconnected and further encoding of the video waveform is performed by the DPCM coder.

DELTA CODE AT TRANSMITTER		MODE	CODE FOR TRANSMISSION	DELTA CODE AT RECEIVER
0 0 1		DELTA	0	0 1 0
0 1 0				0 1 0
1 0 0				0 1 0
1 1 0		DELTA	1	1 0 1
1 0 1				1 0 1
0 1 1				1 0 1
0 0 0	DELTA → DPCM TRANSITION		0 0 1 →	0 0 0
1 1 1			1 1 1 →	1 1 1

Figure 3-9. Coding Table for Delta Mode

Switching from the delta mode to the DPCM mode occurs when a sharp transient occurs in the video waveform. At the start of the transient (three 1s in one sampling inverval), the coder switches to the DPCM mode and remains in the DPCM mode until a DPCM idling condition is sensed. The DPCM idling condition is defined by the occurrence of the smallest DPCM word (111 or 000) preceded by a DPCM word of opposite sign; then both the transmitter and receiver switch to the delta mode of operation.

*In the simulated system the number of samples is tripled by linear interpolation.

The state of the mode controller at the transmitter is uniquely determined: stay in the delta mode until a sequence of three equal delta bits indicates a transition, then switch to DPCM and proceed in the DPCM mode until the idling condition appears in the bit stream; then switch back to delta, and so forth. The operation of mode detection at the receiver, however, is ambiguous since a sequence of three identical bits can also be produced by three compressed delta words. To avoid ambiguity, redundant marker bits (M) are inserted in the compressed delta bit stream after a sequence of three compressed delta bits as shown in Figure 3-10. After three identical delta bits indicating a mode transition (111 or 000), a marker bit M_1 of equal polarity is inserted to indicate that the mode has to be switched to DPCM. Since there might be up to two equal delta bits preceding the mode transition, an addition marker bit M_2 of opposite sign is inserted into the bit stream, in order to mark the proper position at the end of the transition sequence. The introduction of marker bits slightly increases the bit rate generated by a dual-mode coder.

3.2.2.4 Comparison of Adaptive Predictive Coding Techniques

We have simulated block adaptive DPCM, adaptive DPCM with a self-scaling quantizer, and the dual-mode encoder, and have used them to compress the bandwidth of the green band of a sample LANDSAT-A image. The block adaptive DPCM system stores a block of 16 samples and uses these samples in a predictor loop to find the standard deviation of the differential signal for each block. The standard deviation of the differential signal in each block is quantized to eight levels using a uniform quantizer and is used in optimizing the quantizer for the particular block. This requires overhead information of 3/16 bits per sample for a block length of 16.

The second adaptive DPCM encoder uses a fixed quantizer with a one-word memory. The system keeps track of the output levels. A sequential occurrence of the inner reconstruction levels of opposite sign causes a contraction of the reconstruction levels. Repeated occurrence of the outermost levels of the same sign causes an expansion of the reconstruction values. The contraction and the expansion of the reconstructed levels is achieved by dividing and multiplying the reconstruction levels by a constant factor of 1.5, respectively.

← COMPRESSED DELTA-MODE →

				MARKER						MARKER				
0	1	1	1	(0)	1	1	0	0	0	(1)	1	0	1	1
1.	1.	2.	3.	(M)	1.	2.	1.	2.	3.	(M)	1.	1.	1.	2.

COMPRESSED DELTA-MODE				MARKER M		TRANSITION SEQUENCE				MARKERS M1, M2			DPCM-MODE				
0	0	0	. . .	(1)	. . .		1	1	. . .	(1)	(0)	. . 1	0	1	1	1	1
1	0	0	· · · · · · · · ·			1	1	1	· ·	(1)	(0)	· · 1	0	1	1	1	1
1	0	1				1	1	1		(1)	(0)	1	0	1	1	1	1
0	1	1				1	1	1		(1)	(0)	1	0	1	1	1	1
1	1	1		(0)		1	1	1		(1)	(0)	1	0	1	1	1	1

AND VICE VERSA FOR A NEGATIVE TRANSITION

Figure 3-10. Marker Insertion Scheme

The performance of the two adaptive DPCM systems and the dual-mode encoder, as well as the nonadaptive DPCM encoder, are shown in Figure 3-11. As one can see, the use of adaptive methods offers significant improvements over the nonadaptive methods at low bit rates. However, the difference between the adaptive and nonadaptive methods is rather insignificant at high bit rates. This figure clearly shows that the performance of the block adaptive DPCM encoder is superior to other adaptive DPCM techniques we have considered at all bit rates.

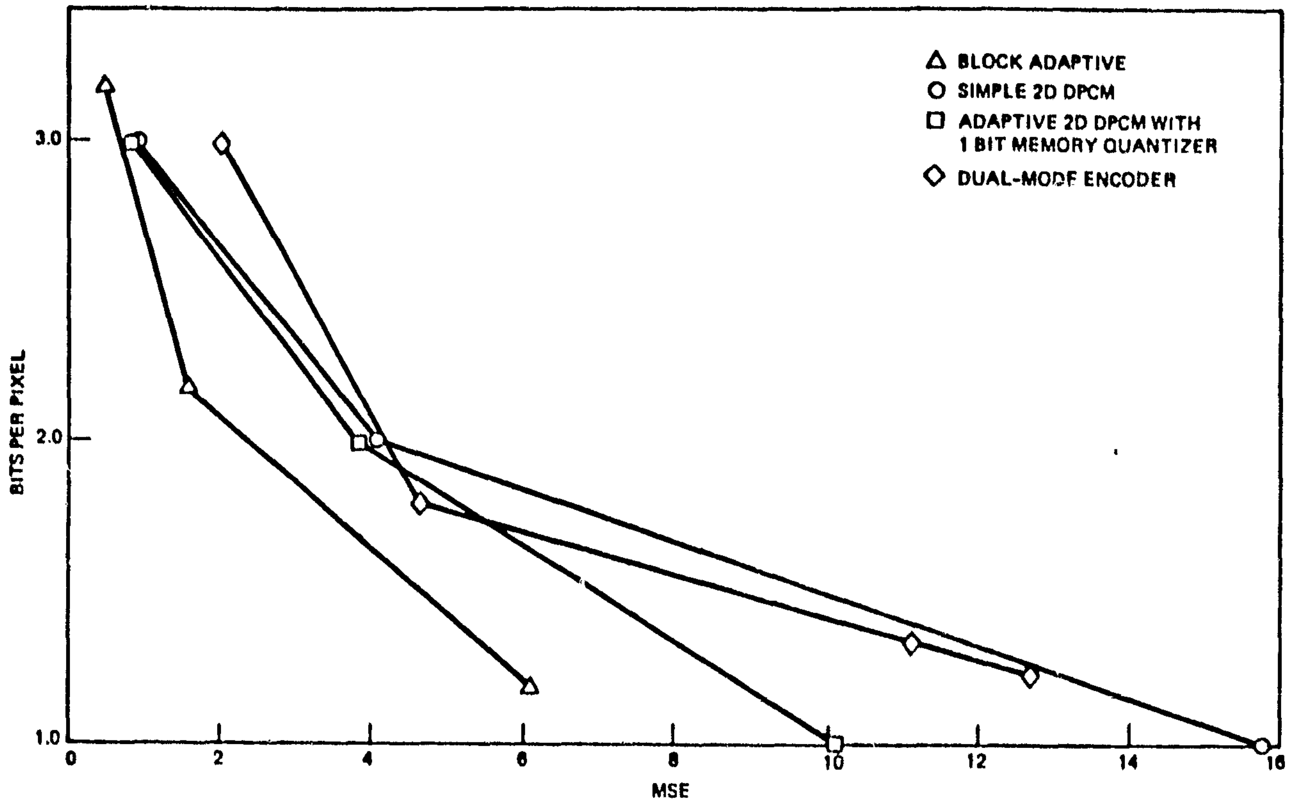


Figure 3-11. Performance Comparison of Predictive Coding Techniques

3.2.3 Adaptive Hybrid Coding

Hybrid encoders use a cancatenation of a unitary transform and DPCM encoders. The hybrid technique uses a block size of 16 or 32 in the direction of the orthogonal transform and the transform coefficients are encoded with DPCM loops. Due to the large block size required in the DPCM direction, adaptive sample selection cannot be employed to improve system performance.

The proposed adaptive hybrid encoder uses a single timeshared block-adaptive DPCM loop to encode the transform coefficients. A block diagram of the proposed encoder is shown in Figure 3-12. The DPCM encoder uses a one element delay in the prediction loop. The gain in the prediction loop and the bit assignment among the various transform coefficient encoders is shown in Table 3-1. The gain values and bit assignments were chosen for the best performance using typical MSS imagery. In an on-board bandwidth compression system using this technique, one should be able to change the bit assignment among various coefficients for maximum flexibility. The DPCM loops use the same quantizer as employed in single-loop block-adaptive DPCM. For details on the quantizer, refer to Section 3.2.1.1.

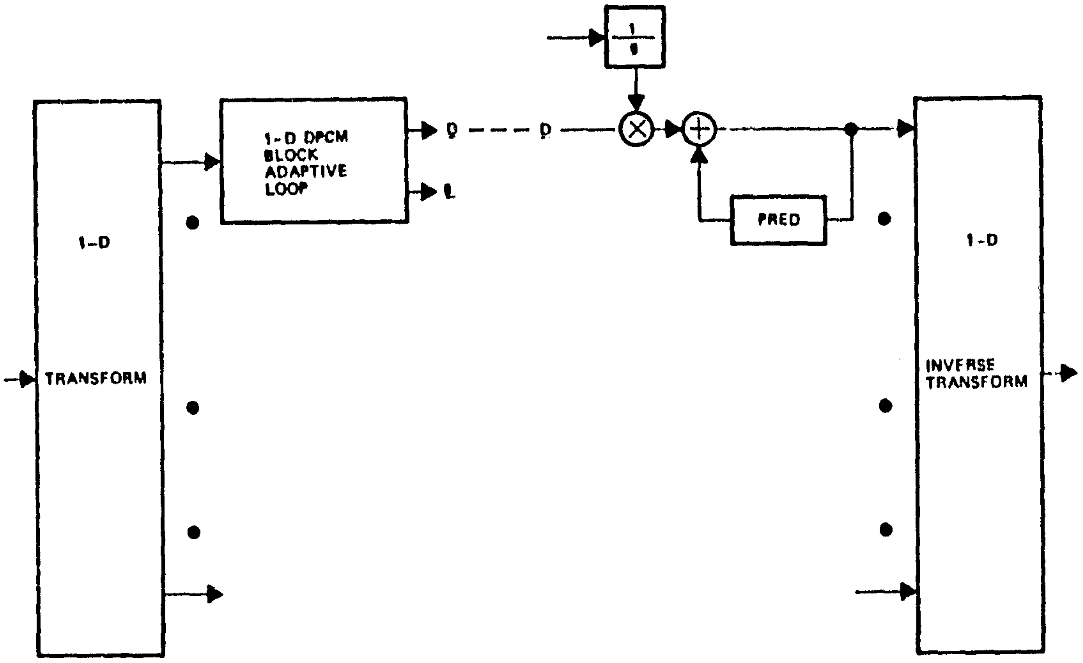


Figure 3-12. Block Diagram of Adaptive Hybrid System

Table 3-1. Adaptive Hybrid Encoder Loop Parameters for Individual MSS Bands

Coefficient No.		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Gain Value in DPCM Loops		$\frac{15}{16}$	$\frac{15}{16}$	$\frac{15}{16}$	$\frac{15}{16}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
2 Bits Per Sample Per Band	MSS-4	5	3	3	3	2	2	2	2	1	1	1	1	1	1	1	0
	MSS-5	5	3	3	3	3	3	2	2	2	2	2	2	2	2	2	1
	MSS-6	5	3	3	3	3	3	3	2	2	2	2	2	1	1	1	1
	MSS-7	5	3	3	2	2	2	2	1	1	1	1	0	0	0	0	0
1 Bit Per Sample Per Band	MSS-4	4	3	2	2	2	1	1	1	0	0	0	0	0	0	0	0
	MSS-5	4	3	3	2	2	2	2	1	1	1	0	0	0	0	0	0
	MSS-6	4	3	2	2	2	1	1	1	0	0	0	0	0	0	0	0
	MSS-7	4	2	2	1	1	1	0	0	0	0	0	0	0	0	0	0
0.5 Bit Per Sample Per Band	MSS-4	3	2	1	1	[illegible]	0	0	0	[illegible]	0	0	0	0	0	0	0
	MSS-5	3	2	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	MSS-6	3	2	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	MSS-7	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0

The performance of the hybrid encoder using both the Hadamard and Cosine transform is shown in Figure 3-13. The performance of the hybrid encoder using the Cosine transform is superior to the encoder that uses Hadamard transform. However, the difference is fairly small and the technique using the Hadamard transform may be more desirable due to less hardware complexity.

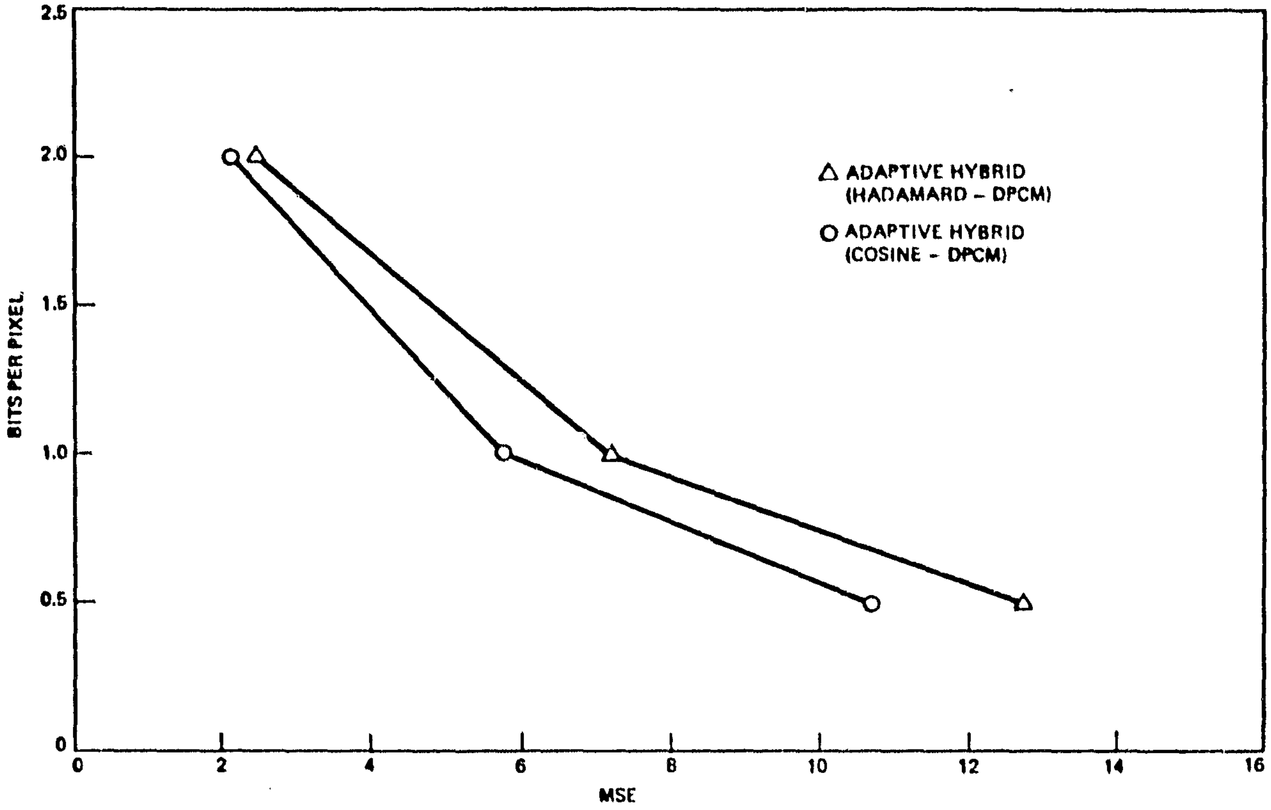


Figure 3-13. Hybrid Encoder Performance

3.2.4 Adaptive Clus*er Coding

In the adaptive multispectral cluster coding method, the data is first divided into small blocks of fixed size (i.e., 16 x 16 picture elements). Then the elements in each block are clustered into a variable number of classes. Using a fixed threshold value, a large number of classes are generated in blocks with high detail and a small number of classes in blocks with low detail. For each block, the classified image is transmitted along with the centroids.

The receiver reconstructs each block of the multispectral imagery by generating the individual bands in each block from the classified image and the corresponding centroids of those clusters. The procedure is to examine each point in the classified image and specify to what class it belongs. The individual bands corresponding to the particular picture location are reconstructed by choosing their values equal to the centroids of that particular class.

The method is made adaptive by generating a variable number of clusters in each block and by using an entropy coding technique to encode the classified image.

The adaptive clustering method we have simulated uses a block size of 16 x 16. It starts with a maximum of 32 classes per block and reduces the number of classes by merging the most similar of the clusters. This continues until the minimum distance bewteen two distinct clusters is larger than a prespecified thre hold value. Figure 3-14 shows the resulting number of clusters for the sample LANDSAT-A image using a threshold value of 0.8. The bit rate (per sample) for the (i,j)th block is

$$R_{ij} = \frac{\log_2 \left|C_{ij}\right|}{B} + \frac{PC_{ij}}{N^2} \quad . \qquad (3\text{-}4)$$

where C_{ij} is the number of clusters per block, B is the number of bands in multispectral data, P is the number of bits used to quantize each centroid value, and N^2 is the number of samples per block. Equation (3-4) assumes the classified image in PCM encoded. The performance of the adaptive cluster coding method is shown in Figure 3-15 in terms of MSE versus bits per pixel for threshold values ranging from 0.8 to 0.2. Figure 3-15 also shows the performance of the nonadaptive cluster coding method which is included here for comparison. Figure 3-16 shows a comparison of adaptive and nonadaptive cluster coding methods in terms of classification consistency. In obtaining classification consistency, the original and the encoded imagery are classified into nine classes. A point-by-point comparison of the two classified images indicates what percentage of the samples is classified consistently. This figure also contains the classification consistency of the nonadaptive technique.

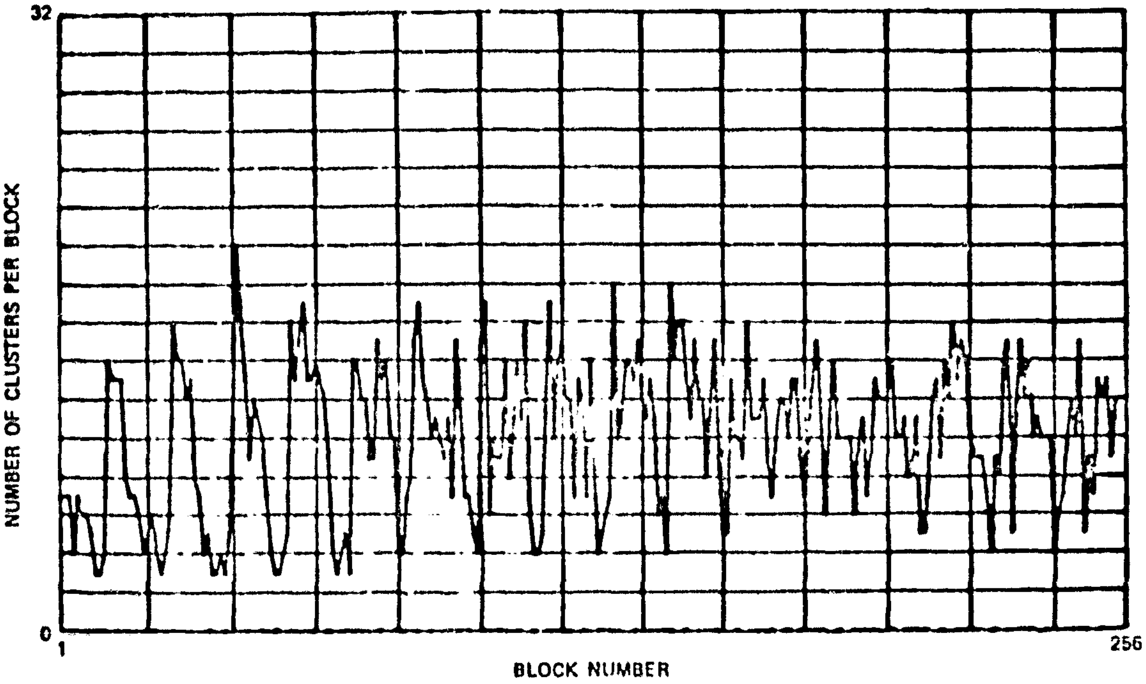


Figure 3-14. Number of Clusters Per Block

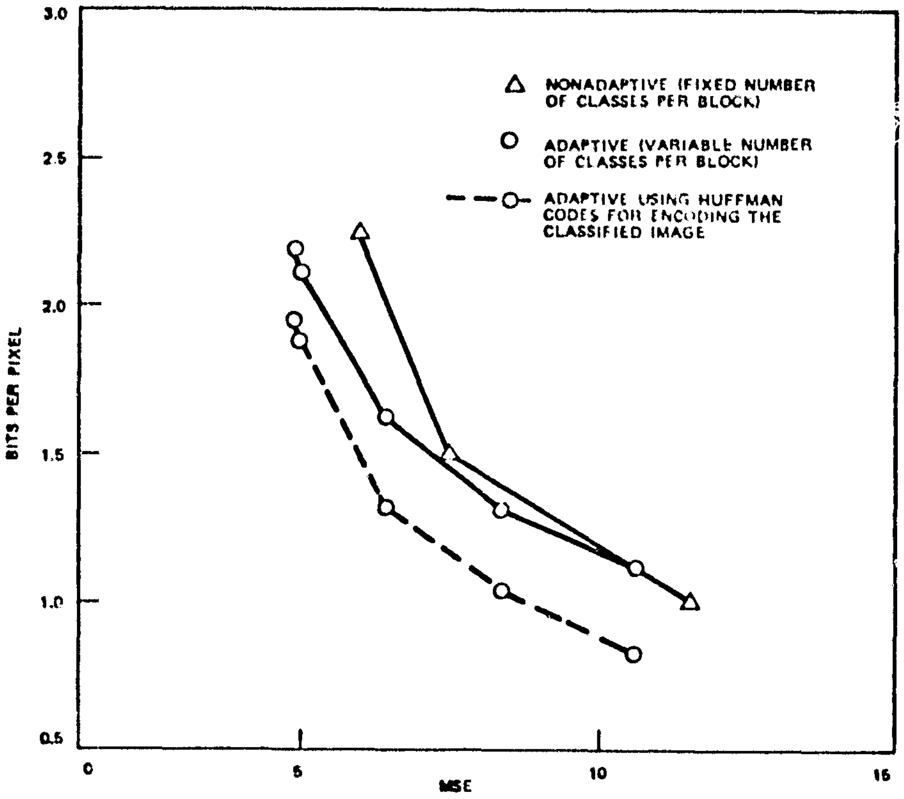


Figure 3-15. Performance of Adaptive and Nonadaptive Cluster Coding Methods

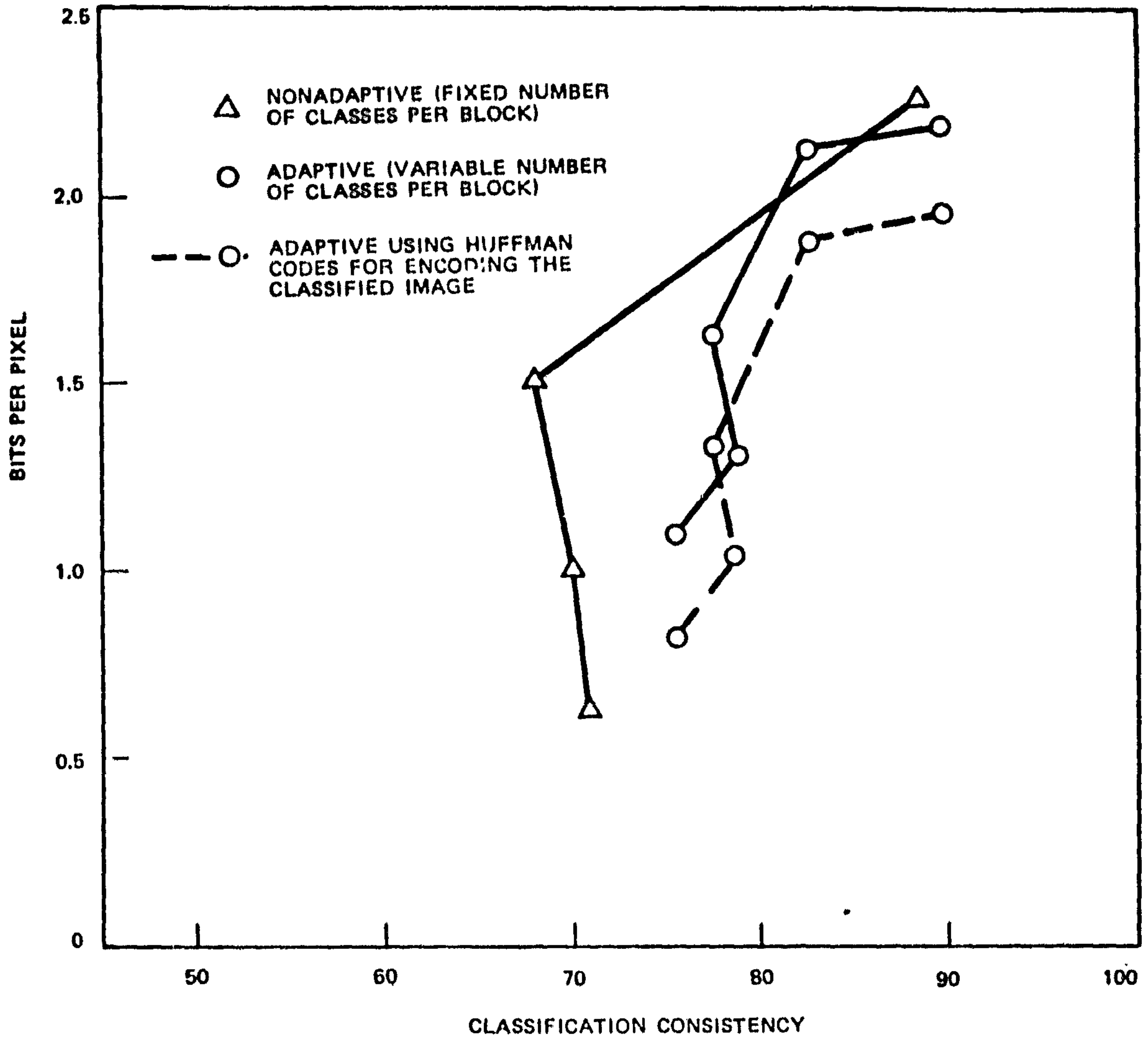


Figure 3-16. Performance of Adaptive and Nonadapt Cluster Coding Methods

The adaptive cluster coding method previously discussed uses a PCM technique to transmit the classified picture for each block. The performance of the system improves if a more efficient encoder is used to transmit the image. One such system can be designed by generating a difference signal, then using a Huffman encoder to transmit the difference signal. The difference signals $Y(K,\ell)$; $k, \ell = 1, \ldots, 16$ for each block is generated as

$$\begin{aligned} Y(k,\ell) &= X(k,\ell) && \text{for } \ell = 1 \\ Y(k,\ell) &= X(k,\ell) - X(k,\ell-1) && \text{for } k = 1, \ldots, 16 \\ & && \ell = 2, \ldots, 16 \end{aligned} \qquad (3\text{-}5)$$

where $X(k,\ell)$ denotes the values of the classified image. The difference signal $Y(k,\ell)$ has a smaller entropy than the classified signal $X(k,\ell)$; therefore, it can be encoded using less binary digits. For a maximum of 32 clusters per block, the entropy of the difference signal $Y(k,\ell)$ for the block indexed by i and j is

$$H_{ij} = \sum_{m=0}^{31} P\left[Y(k,\ell) = m\right] \log_2 P\left[Y(k,\ell) = m\right] \qquad (3\text{-}6)$$

Note that m does not assume any negative values since mod 32 arithmetic is used in generating $Y(k,\ell)$ from $X(k,\ell)$.

The corresponding bit rate using a Huffman encoder is

$$\hat{R}_{ij} = \frac{H_{ij}}{B} + \frac{PC_{ij}}{N^2} \qquad (3\text{-}7)$$

Figure 3-17 shows the variation of H_{ij} and $\hat{R}_{ij}$ for various blocks (16 x 16 samples in four bands) of the standard MSS data. The improvement in bit rate versus MSE and classification accuracy is shown in Figures 3-15 and 3-16 by dashed lines. It is worth noting that the Huffman encoder improves the performance of the adaptive clustering technique. The improvement due to adaptive clustering is larger at higher bit rates when the additional improvement due to Huffman encoder is more significant than at lower bit rates.

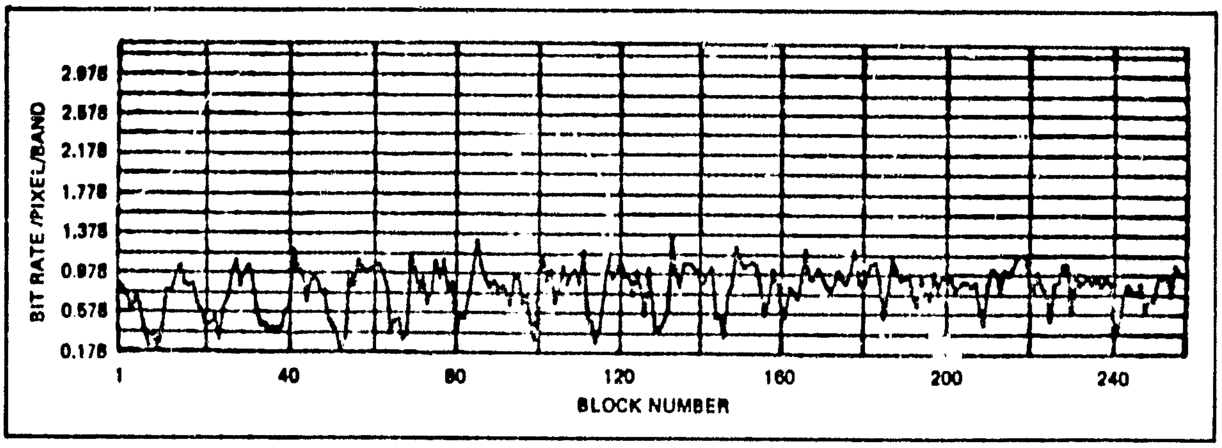


Figure 3-17a. Bit Rate Variation Per Pixel Per Band for 16 x 16 Blocks

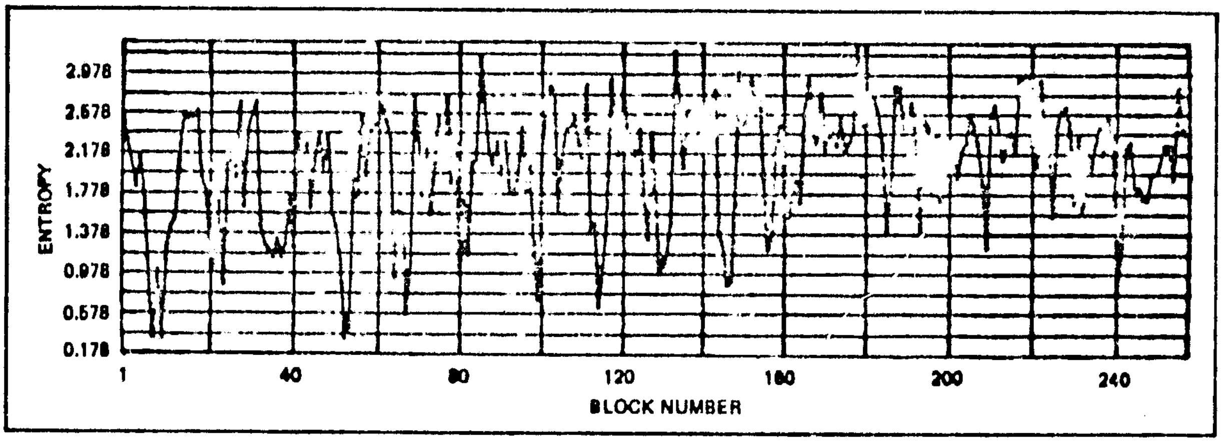


Figure 3-17b. Classified Image Entropy Variation for 16 x 16 Blocks (Threshold = 0.8)

Figure 3-17. Cluster Coding Parameters

In implementing the clustering algorithm, emphasis was placed on evaluating the algorithm's potential, rather than on speed of operation. The number of computations required for each 16 x 16 block is variable, depending on the data in that block. However, a first approximation is based on the number of calculations required to assign each pixel to the closest centroid. The squared distance from each pixel to each centroid is calculated. Since there are four spectral bands, each distance measurement requires four squaring operations and three summing operations. Thus for 32 classes, one iteration of the clustering algorithm requires 32 x 256 x 4 multiplies and 32 x 256 x 7 adds (32,768 multiplies and 57,344 adds). Significant speedup of this approach can be obtained by defining distance as the sum of the absolute distances in each spectral band, rather than using Euclidian distance. This would eliminate all the multiplies but would add 65,536 adds. Further reductions could be achieved by reducing the number of iterations. This is done by increasing the permissible number of pixels changing cluster at each iteration. Finally, by starting the clustering procedure with less than 32 centroids, a speed improvement can be realized.

4. DATA SETS PROCESSED

4.1 DESCRIPTION OF SELECTED IMAGERY

Two sets of multispectral scanner data generated by NASA's LANDSAT were used to test the performance of the various adaptive compression algorithms on typical multispectral images. The four bands of each of the images are shown in Figures 4-1 and 4-2. For convenience we call the first scene the "agricultural scene" and the second scene the "Bald Knob Scene." Results based on these scenes provide a good measure of performance of the algorithm for several reasons. First, the scenes cover quite different land uses, one being primarily agricultural and desert, the other primarily vegetated mountainous terrain with relatively little human influence. Thus, the agricultural scene has a great deal of regular rectangular structure, while the Bald Knob scene has virtually none. Second, the reflected energy is distributed among the spectral bands very differently in the two scenes. This implies that the number of bits assigned to each band should vary between scenes. Third, the agricultural scene was used in the study of nonadaptive techniques and thus the results can be compared with the previous study. Finally, the Bald Knob scene is desirable because ground truth data exists at NASA/ARC corresponding with this scene.

4.2 STATISTICAL MODELING OF THE DATA

To have the best possible compression results, it is necessary to take advantage of the statistics of the data. The adaptive techniques described previously adapt to the statistics of a single band in various different ways. In each case, however, some statistics are computed in a small block of the data and the compression technique uses this statistic in the encoding process. For each successive block, the statistic is recomputed. However, these statistics do not determine how much of the total bit allotment should be used for each band (except with the cluster coding algorithm). The approach we have taken to this problem has been to transform the original spectral bands into a new set of bands in which the number of bits assigned to each (transform) band is roughly constant for each scene and need not adapt. To this end this section describes the statistical properties of the two scenes used in this study.

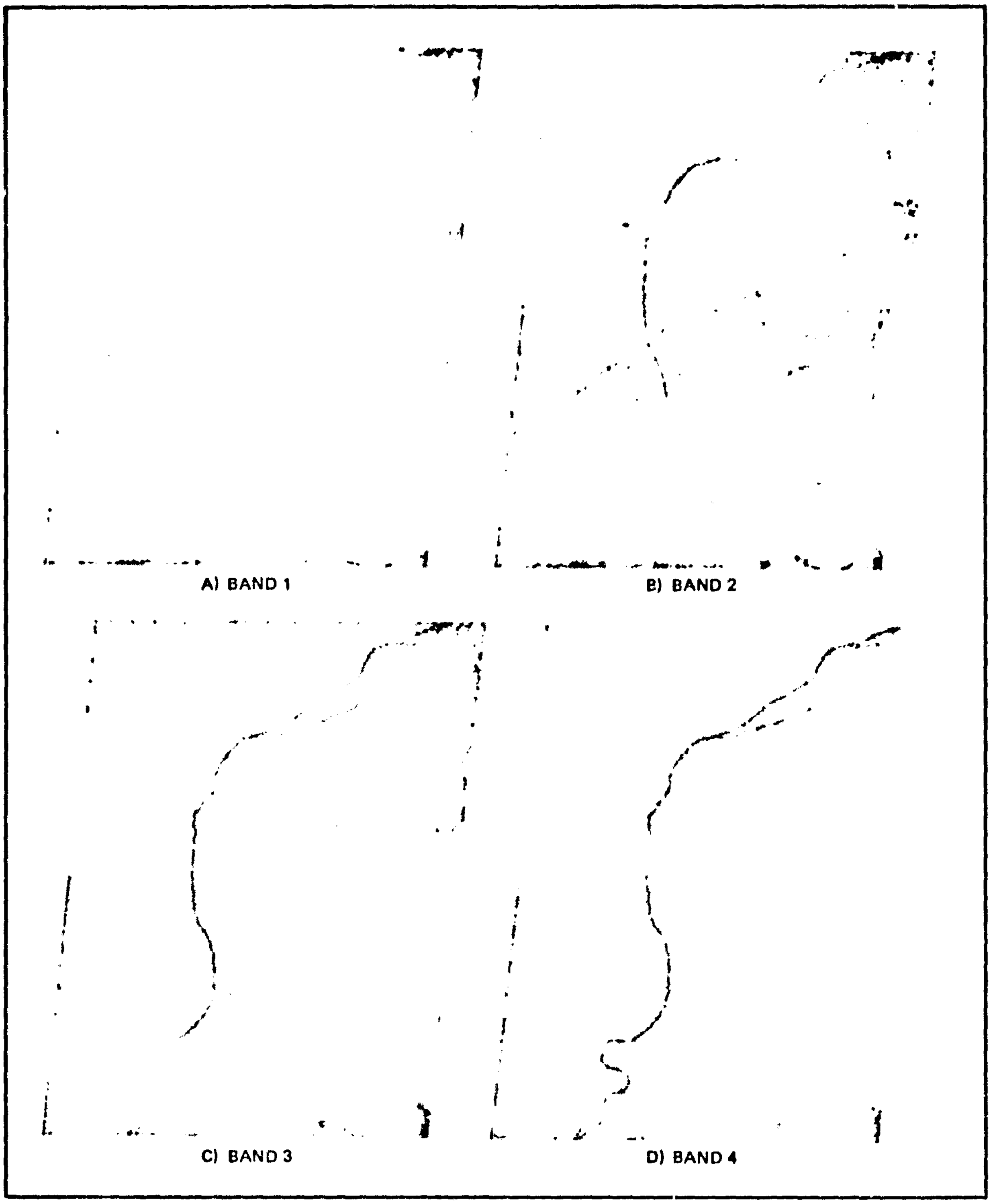


Figure 4-1. Agricultural Scene

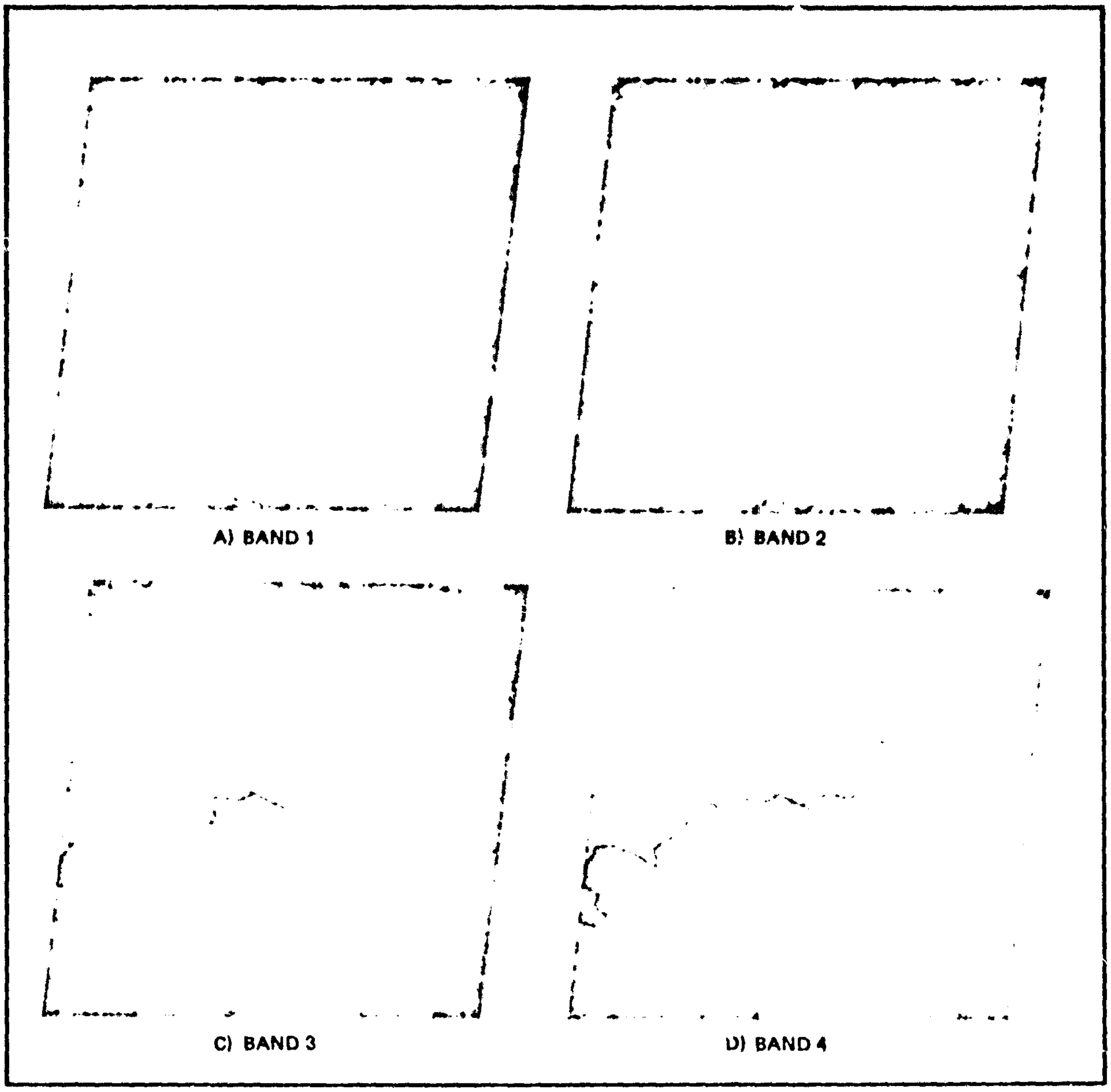


Figure 4-2. Bald Knob Scene

For each band of the agricultural scene, the dimensions are 256 x 256 pixels. For the Bald Knob scene, the horizontal dimension is 256 pixels, but the vertical dimension is 192 pixels. Statistics were gathered for each band of both scenes. These results are tabulated in Table 4-1. Comparing the variances of the data, it is apparent that the amount of variation is distributed very differently among the bands in the two scenes. One should also note that the fourth band of scanner data provided by LANDSAT is always 6 bit data while the other three bands are 7 bit data.

Table 4-1. Scene Statistics

Statistic	Scene							
	Agricultural Scene				Bald Knob Scene			
	Band 1	Band 2	Band 3	Band 4	Band 1	Band 2	Band 3	Band 4
Minimum	24	15	10	2	21	14	9	2
Maximum	84	103	96	52	59	70	82	48
Mean	47.326	52.17	59.579	26.826	30.302	26.818	39.377	22.513
Variance	130.90	333.65	121.61	41.083	11.841	22.86	103.58	42.905
Standard Deviation	11.441	18.266	11.028	6.4096	3.441	4.7812	10.42	6.55

To illustrate the correlation between the various spectral bands, we have made scatter plots of various pairs of bands. These are shown in Figures 4-3 and 4-4. Because the fourth band has half the number of bits of the other bands, it has been scaled by a factor of 2 in these plots. The primary point of interest in Figures 4-3 and 4-4 is the fact that, for both scenes, bands 1 and 2 and bands 3 and 4 are highly correlated, whereas the remaining pairs of bands have relatively low correlation. This is true even through the data variance is distributed differently among the bands for the two scenes. The implication of this finding is that the difference between bands 1 and 2 and the difference between bands 3 and 4 (band 4 must be doubled first) contain relatively little information. Thus, if we create four transformed bands, two of which are the difference between bands 1 and 2 and the difference between bands 3 and 4, bits can be distributed such that these two different bands get relatively few bits and the other two bands get most of the bits.

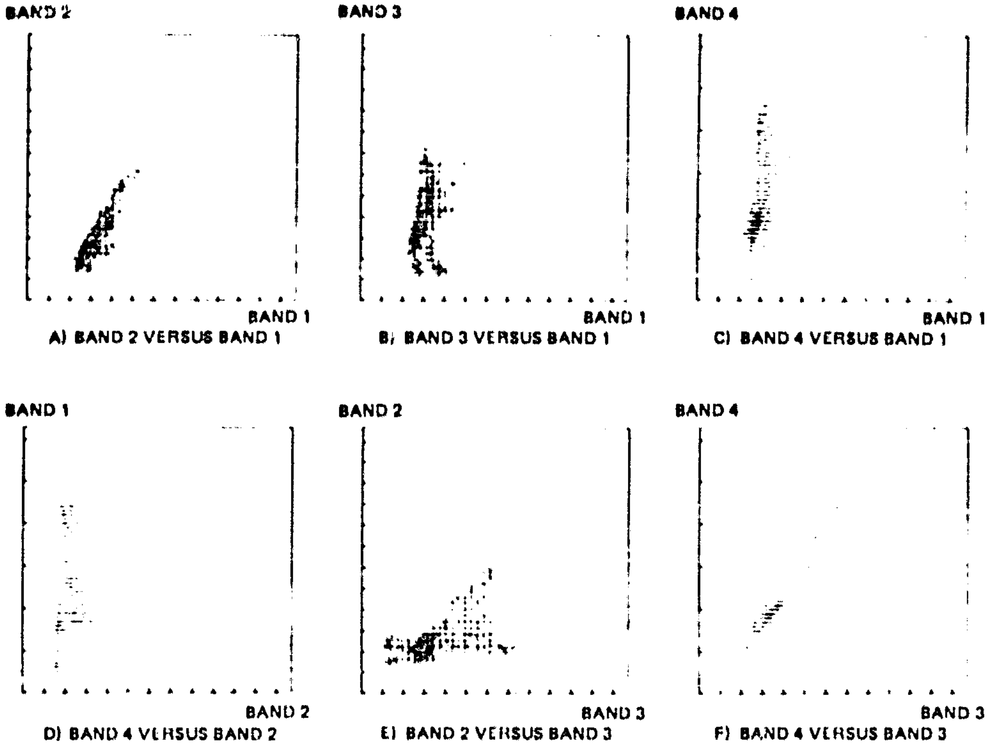


Figure 4-3. Correlation Between Bands for Bald Knob Scene

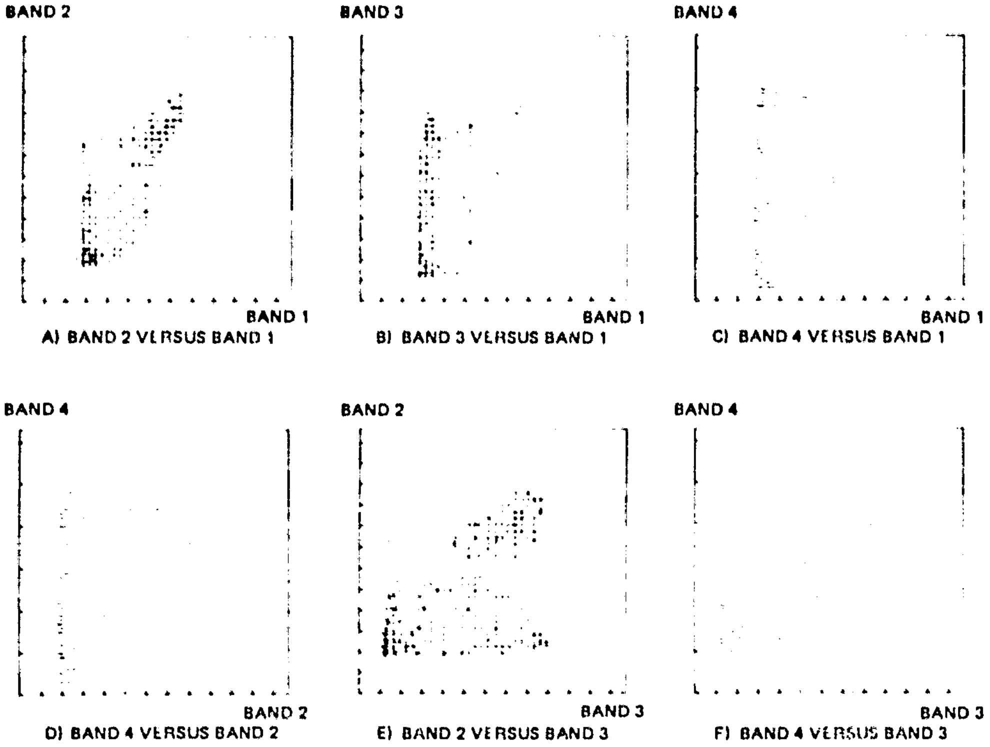


Figure 4-4. Correlation Between Bands for Agricultural Scene

A transform that creates four independent bands, two of which are the desired bands, is the Haar transform. The matrix that specifies the transform for the 4 x 4 case we are interested in is

$$\frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

Thus, the first transform band is the average of the four bands, the last two transform bands are scaled versions of the difference between bands 1 and 2 and the difference between bands 3 and 4, and the second transform band is the difference between the average of bands 1 and 2 and bands 3 and 4. Figure 4-5 shows that the variance of the first two transform bands is much higher than the variance of the other two. Each of the four diagrams shows the relative distribution of variance in four bands: 4a) for the agricultural scene, 4b) for the Bald Knob scene, 4c) for the transformed agricultural scene, and 4d) for the transformed Bald Knob scene. In each case the sum of the variances has been set to 1. The variances are presented in Table 4-2. The fourth MSS band has been prescaled by 2.

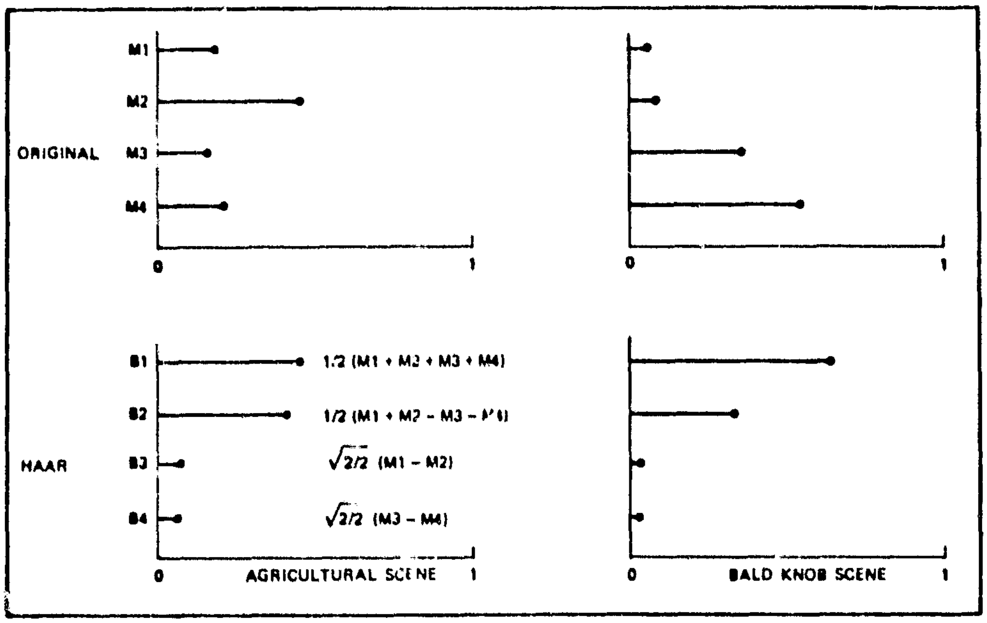


Figure 4-5. Haar Compaction of Variance

Table 4-2. Variances of Original and Transformed Bands

Variance	Agricultural Scene				Bold Knob Scene			
	Band 1	Band 2	Band 3	Band 4	Band 1	Band 2	Band 3	Band 4
Original	130.9	333.65	121.61	162.332	11.841	22.86	108.58	171.62
Haar	339.57	302.22	48.66	48.14	201.02	102.58	3.77	6.70

In both cases more than 87 percent of the data variance is in the first two bands. For one scene the first two bands had roughly equal variance and for the other scene band 1 had almost twice as much variance as band 2 A reasonable bit assignment for the Haar bands can be expected to be 50 percent of the bits for band 1, 40 percent for band 2, and 5 percent each for bands 3 and 4.

5. COMPARISON OF SELECTED COMPRESSION TECHNIQUES

In this section we compare, according to several different criteria, the most promising techniques described in Section 3. The selected algorithms were simulated on a computer and applied to both test scenes at bit rates ranging from 0.5 to 2 bits per pixel. Error as defined by the various criteria was measured in each case and plotted. Specific bit assignments used for the runs are presented in Section 5.1. The error measures used are then defined and the performance of the algorithms plotted and interpreted in Section 5.2. Finally, Section 5.3 covers the system considerations that affect selection of a particular algorithm for real world applications.

5.1 COMPRESSION SIMULATIONS PERFORMED

The bit rate for each of the selected algorithms is determined in a different way. For the adaptive DPCM technique, all that is specified is the bit rate for each band. This algorithm operates at a fixed rate, and therefore the number of bits per pixel specified applies to each pixel. For the hybrid techniques, each coefficient of the 1D transform is coded using DPCM. The rate used for each coefficient is selected independently and is fixed. Thus, to specify the overall bit rate, the rate for each coefficient in each band must be specified. For a block size of 16 and four spectral bands, 64 rates must be specified. The resulting coder operates at a fixed rate. For the 2D transform coder, only the approximate average bit rate is specified. The bit rate used for a particular coefficient is one-half the log (base 2) of the estimated variance of that coefficient minus the sum of the approximate bit rate and the log (base 2) of the maximum data value scaled down by 128. The average bit rate corresponding to a particular distortion varies dpending on the scene properties. Thus, the simultaneous rate produced by the encoder is variable from pixel to pixel and scene to scene. The last selected algorithm, cluster coding, also operates at a variable rate. For cluster coding, only one parameter is specified to determine the compression ratio. This parameter is the minimum "distance" between clusters. Any two clusters closer together than the minimum distance are merged into one cluster. This reduces the number of cluster centroids that have to be transmitted and the number of bits required to specify the cluster number of each pixel. The distance measure used is described in Section 3.

In selecting parameters for each of the compression algorithms, optimization for each scene was not attempted. Instead, reasonable parameter values based on experience, judgment, and preliminary simulations were used. Bit allocations to the various bands were kept the same for both scenes since this is the most likely situation to arise in practice. Bit allocations to the four bands were kept the same when the Haar transform was not used. When the Haar transform was used, initial processing was done allocating the same number of bits to Haar bands 1 and 2 and the same number to Haar bands 3 and 4. The ratio between the bands 1 and 2 allocation and bands 3 and 4 allocation was chosen to approximately assign 75 percent of the bits to the high variance bands and 25 percent to the low variance bands. This corresponds to most of the energy being in Haar bands 1 and 2 as shown in Section 4. Modifications of this basic bit assignment were made to achieve the desired total bit rate. After running these simulations, the bit assignment to the different bands was varied slightly if some improvement could be anticipated. In the hybrid transform case, bit assignments had to be made for each transform coefficient. Here, decreasing coefficient variance as a function of frequency was assumed. Specific coefficient allocations were determined iteratively. Table 5-1 summarizes the runs made and the parameters used.

5.2 COMPRESSION PERFORMANCE

Evaluation of data compression schemes is accomplished by determining for a fixed bit rate the fidelity between an original image and a reconstruction using the compressed data. Various ways of measuring fidelity have been developed. The applicability of one or more of these depends on the specific applications for which the imagery is intended. We have used several measures in this study. The same measures were used in the Study of On-Board Compression of Earth Resources Data (NAS2-8394) — the predecessor of this study which looked at nonadaptive compression techniques. To make this final report self-contained, we will briefly describe each of the fidelity measures. For greater detail, refer to the previous final report.

Table 5-1. Simulations Performed

ALGORITHM	BIT RATE (BITS/ PIXEL/ BAND)	BIT PATE BY BANDS		PARAMETER THAT DETERMINES PATE		SCENE	DISCUSSION
		BAND	RATE	PARAMETER	VALUE		
HAAR – 2D ADAPTIVE DPCM	2.19	HAAR 1 HAAR 2 HAAR 3 HAAR 4	3.19 2.19 2.19 1.19	RATE HAAR BAND 1 RATE HAAR BAND 2 RATE HAAR BAND 3 RATE HAAR BAND 4	3 2 2 1	AGRICULTURAL BALD KNOB	OVERHEAD IS INCLUDED. IT EQUALS 0.19 BIT/ PIXEL/BAND. FOR A ZERO BIT ALLOCATION, NO OVERHEAD IS REQUIRED
	1.095	HAAR 1 HAAR 2 HAAR 3 HAAR 4	2.19 2.19 0 0	RATE HAAR BAND 1 RATE HAAR BAND 2 RATE HAAR BAND 3 RATE HAAR PAND 4	2 2 0 0	AGRICULTURAL BAND KNOB	PREDICTOR USED WAS A = -0.5, B = 0.75, C = 0.75 FOR
	0.595	HAAR 1 HAAR 2 HAAR 3 HAAR 4	1.19 1.19 0 0	RATE HAAR BAND 1 RATE HAAR BAND 2 RATE HAAR BAND 3 RATE HAAR BAND 4	1 1 0 0	AGRICULTURAL BALD KNOB	$\hat{s}_1$ · · $\hat{s}_2$ $\hat{s}_3$ · · x $\hat{x} = A\hat{s}_1 + B\hat{s}_2 + C\hat{s}_3$
HAAR – HYBRID (HADAMARD)	2.008	HAAR 1 HAAR 2 HAAR 3 HAAR 4	3.012 3.012 1.004 1.004	HAAR BAND 1 HAAP BAND 2 HAAR PAND 3 HAAR BAND 4	BITS FOR EACH COEFFICIENT 5,5,5,4,4,4,3,3,3,3,2,2,2,1,1,1 5,5,5,4,4,4,3,3,3,3,2,2,2,1,1,1 4,3,3,2,2,2,0,0,0,0,0,0,0,0,0,0 4,3,3,2,2,2,0,0,0,0,0,0,0,0,0,0	AGRICULTURAL BALD KNOB	COEFFICIENTS GIVEN IN ORDER OF INCREASING SEQUENCE
	1.005	HAAR 1 HAAR 2 HAAR 3 HAAR 4	1.506 1.506 0.504 0.504	HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	5,4,4,3,3,2,2,1,0,0,0,0,0,0,0,0 5,4,4,3,3,2,2,1,0,0,0,0,0,0,0,0 2,2,1,1,1,1,0,0,0,0,0,0,0,0,0,0 2,2,1,1,1,1,0,0,0,0,0,0,0,0,0,0	AGRICULTURAL BALD KNOB	OVERHEAD IS INCLUDED. IT EQUALS 0.012 BIT/PIXEL/BAND IF ALL COEFFICIENTS HAVE NONZERO BIT ALLOCATIONS. COEFFICIENTS ASSIGNED ZERO BITS PEQUIRE NO OVERHEAD. THUS THE OVERHEAD IS 0.012 X FRACTION OF COEFFICIENTS WITH NONZERO BIT ALLOCATIONS. PREDICTOR COEFFICIENTS ARE, 15/16, 15/16, 15/16, 15/16, 3/4, 3/4, 3/4, 3/4, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2, 1/2
	0.503	HAAR 1 HAAR 2 HAAR 3 HAAR 4	0.754 0.754 0.252 0.252	HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	3,3,2,2,1,1,0,0,0,0,0,0,0,0,0,0 3,3,2,2,1,1,0,0,0,0,0,0,0,0,0,0 2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0 2,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0	AGRICULTURAL BALD KNOB	
HAAR – 2D ADAPTIVE (HADAMARD)	1.95	HAAR 1 HAAR 2 HAAR 3 HAAR 4	2.816 3.071 0.987 0.945	APPROXIMATE RATE HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	 2.4 1.5 0.5 0.5	AGRICULTURAL	OVERHEAD IS INCLUDED MAX VALUE ASSUMED (FOR PURPOSES OF BIT ASSIGNMENT):
	1.01	HAAR 1 HAAR 2 HAAR 3 HAAR 4	1.508 1.479 0.547 0.519	HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	1.65 0.65 0.1 0.1	AGRICULTURAL	HAAR BAND 1: 511 HAAR BAND 2: 256 HAAR BAND 3: 180 HAAR BAND 4: 180
	0.62	HAAR 1 HAAR 2 HAAR 3 HAAR 4	0.766 0.945 0.386 0.375	HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	0.95 0.3 -0.15 -0.15	AGRICULTURAL	

Table 5-1. Simulations Performed (Continued)

ALGORITHM	BIT RATE (BITS/ PIXEL/ BAND)	BIT RATE BY BANDS		PARAMETER THAT DETERMINES RATE		SCENE	DISCUSSION
		BAND	RATE	PARAMETER	VALUE		
HAAR - 2D ADAPTIVE (HADAMARD) (CONTINUED)	1.83	HAAR 1 HAAR 2 HAAR 3 HAAR 4	2.91 2.75 0.857 0.820	HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	3 3 1 1	BALD KNOB	
	0.854	HAAR 1 HAAR 2 HAAR 3 HAAR 4	1.411 1.304 0.33 0.37	HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	1.5 1.5 0.5 0.5	BALD KNOB	
	0.404	HAAR 1 HAAR 2 HAAR 3 HAAR 4	0.61 0.61 0.19 0.21	HAAR BAND 1 HAAR BAND 2 HAAR BAND 3 HAAR BAND 4	0.75 0.75 0.25 0.25	BALD KNOB	
CLUSTER CODING	1.282	ALL BANDS	1.282	SWAIN FU DISTANCE	0.7	AGRICULTURAL	OTHER PARAMETERS OF THE CLUSTERING ARE THE INITIAL NUMBER OF CLASSES (16), THE MINIMUM NUMBER OF CLASSES (7), THE PERCENTAGE CONVERSION IN EACH CLUSTERING ITERATION (98)
	0.8735	ALL BANDS	0.8735	SWAIN FU DISTANCE	0.9	AGRICULTURAL	
	1.041	ALL BANDS	1.041	SWAIN FU DISTANCE	0.7	BALD KNOB	
	0.7641	ALL BANDS	0.7641	SWAIN FU DISTANCE	0.5	BALD KNOB	
	0.4235	ALL BANDS	0.4235	SWAIN FU DISTANCE	1	BALD KNOB	BIT RATE IS PREDICTED USING THE FIRST ORDER ENTROPY OF THE CLUSTERED IMAGE OVERHEAD IS INCLUDED
ADAPTIVE 2D (COSINE)	1.81	MSS-4 MSS-5 MSS-6 MSS-7	1.397 2.23 1.799 1.61	APPROXIMATE B.R. MSS-4 MSS-5 MSS-6 MSS-7	2.0 2.0 2.0 2.0	AGRICULTURAL	NO SPECTRAL TRANSFORM.
	0.67	MSS-4 MSS-5 MSS-6 MSS-7	0.521 0.821 0.641 0.686	APPROXIMATE B.R. MSS-4 MSS-5 MSS-6 MSS-7	0.5 0.5 0.5 0.5	AGRICULTURAL	
HYBRID (HADAMARD)	2.011	MSS-4 MSS-5 MSS-6 MSS-7	1.82 2.45 2.32 1.45	MSS-4 MSS-5 MSS-6 MSS-7	BITS FOR EACH COEFFICIENT 5,3,3,3,2,2,2,2,1,1,1,1,1,1,1,0 5,3,3,3,3,3,2,2,2,2,2,2,2,2,2,1 5,3,3,3,3,3,2,2,2,2,2,1,1,1,1,1 5,3,3,2,2,2,2,1,1,1,1,1,1,0,0,0	AGRICULTURAL	
	1.04	MSS-4 MSS-5 MSS-6 MSS-7	1.04 1.35 1.04 0.729	MSS-4 MSS-5 MSS-6 MSS-7	4,3,2,2,2,1,1,1,0,0,0,0,0,0,0,0 4,3,3,2,2,2,2,1,1,1,0,0,0,0,0,0 4,3,2,2,2,1,1,1,0,0,0,0,0,0,0,0 4,2,2,1,1,1,0,0,0,0,0,0,0,0,0,0	AGRICULTURAL	
	0.52	MSS-4 MSS-5 MSS-6 MSS-7	0.520 0.582 0.520 0.457	MSS-4 MSS-5 MSS-6 MSS-7	3,2,1,1,1,0,0,0,0,0,0,0,0,0,0,0 3,2,1,1,1,1,0,0,0,0,0,0,0,0,0,0 3,2,1,1,1,0,0,0,0,0,0,0,0,0,0,0 3,2,1,1,0,0,0,0,0,0,0,0,0,0,0,0	AGRICULTURAL	

5.2.1 Mean Square Error

Mean square error (MSE) is the most frequently used criterion of optimality in data compression as well as in most other estimation and filtering problems. This is due partly to the inherent simplicity of this criterion which allows for closed-form analytical solutions, and partly to the fact that many sensing systems respond directly to the energy contained in the stimulus and that energy and MSF are closely related.

Experiments with the MSE have shown that it does have some correlation with the subjective quality of the reconstructed imagery. Human vision is more sensitive to error in the darker portions of the image than in lighter areas. However, because many users will have the capability to computer enhance, spatially expand, and contrast stretch Thematic Mapper type imagery, specific properties of the eye, such as sensitivity to particular spatial frequencies or different sensitivity to errors at different illumination levels, are not as critical to include in performance measures as they would be in video applications. For multiband data, we take as our measure of MSE the average of the MSE in the several bands. Thus

$$\varepsilon^2 = \frac{1}{MN} \sum_{j=1}^{M} \sum_{i=1}^{N} (x_{ij} - \hat{x}_{ij})^2 \quad . \tag{5-1}$$

where

x_{ij} = sample i in band j of original imagery

$\hat{x}_{ij}$ = sample i in band j of reconstructed imagery

M = number of bands

N = number of samples per band

ε = RMS error

Figures 5-1 and 5-2 are plots of the MSE performance of the four selected algorithms for the two different LANDSAT scenes. For the three nonclustering techniques, the Haar spectral transform was applied before compressing the data. Selected runs were also made without using the Haar transform.

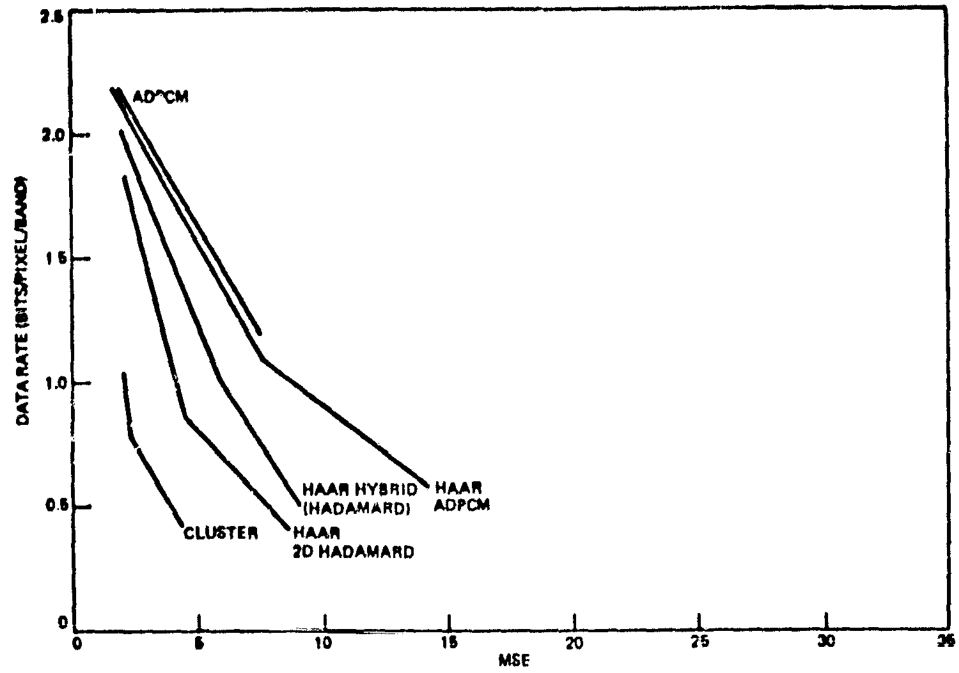


Figure 5-1. Data Rate Versus Mean Square Error — Bald Knob Scene

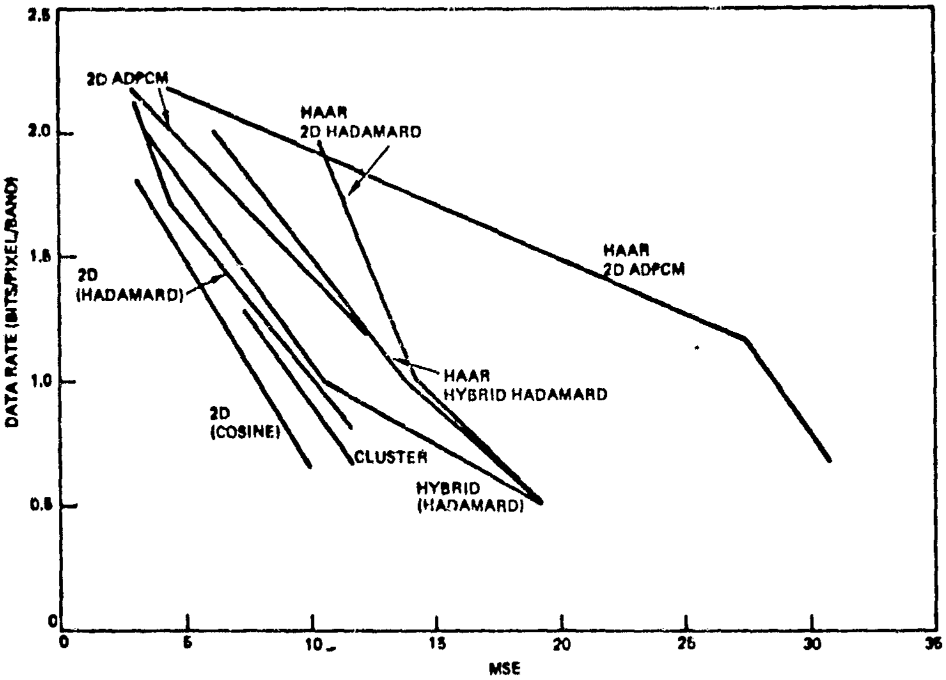


Figure 5-2. Data Rate Versus Mean Square Error — Agricultural Scene

The number of bits assigned to each resulting band was the same for both scenes. While this does not yield the optimal compression for either scene, in a real application a fixed bit allocation between bands is most likely.

The plots of Figures 5-1 and 5-2 lead to several interesting conclusions:

a) At a rate of 2 bits per pixel, performance of the adaptive DPCM technique is close to that of the best technique. This corresponds with results obtained by other researchers. The good DPCM performance applied both with and without the Haar transform. This basically says that the predictor error within any 16 pixel block can be well represented by only four levels.

b) DPCM performance degrades faster than that of any of the other techniques as the bit rate goes below 2 bits. This is partially due to the fact that bit rates assigned to each band must go in discrete increments of 1 bit per pixel. Thus, one band cannot be assigned 1.5 bits and another band 0.5 bit. This means that the bit allocation between bands can only be roughly fit to the data statistics. Another contributor to performance degradation below 2 bits per pixel is that at 1 bit per pixel the predictor has just one positive and one negative level. Even the relatively small changes in one 16-pixel block cannot be represented by one level. Consequently, slope overload and graininess become significant, resulting in substantially increased MSE.

c) The Haar spectral transform produces mixed results. On the Bald Knob Scene, a small performance improvement was produced by introduction of the Haar transform. However, on the agricultural scene, introduction of the Haar led to an even greater loss in performance. This loss was particularly pronounced for the DPCM technique. One would initially expect that by decorrelating the spectral bands and compacting most of the energy into just two bands, some compression improvements could be obtained. This was in fact observed in the study of nonadaptive techniques (an improvement of about 1 dB in SNR) that preceded this study. Several factors prevent the Haar from realizing its potential. The three that are likely to be most important are the following. First, by taking the spectral transformation, the block-to-block nonstationarity of data is reduced. Since adaptive techniques are designed to utilize this nonstationarity the lack of it impairs their performance. Second, the third and fourth transform bands have a greater percentage of their energy in the higher frequencies than the original bands. This means that the statistical model that the compression algorithms depend on is not as appropriate for the transform bands as for the orig'nal bands. At the heart of all the techniques is the assumption that, on the average, data changes are gradual. Third, since the spatial frequency content varies between transform bands, to take full advantage of the energy compaction achieved, the compressors need to have different parameters in each band. For example, the predictor in the DPCM algorithm should have different weights for the different bands. At the same time, the set of gains used to make the DPCM adaptable should be adjusted to the ranges of each of the transform bands. In conclusion, the fact that the Haar spectral transform yielded mixed results does not imply that effective use cannot be made of spectral correlation, but rather that to take advantage of this correlation requires modification of the spatial techniques according to the scene statistics of each transform band.

d) Results are very scene-dependent. For a fixed algorithm and bit rate, absolute MSE is considerably less for the Bald Knob scene than for the agricultural scene. On the other hand, the relative performance of the algorithms is approximately the same for both scenes. In both cases DPCM performance degrades rapidly below 2 bits per pixel, cluster coding is one of the best performers, and the 2D Hadamard and hybrid Hadamard/DPCM perform comparably.

5.2.2 Signal-to-Noise Ratio

A criterion closely related to mean square error is the signal-to-noise ratio (SNR). Indeed this criterion can be considered a normalized form for the MSE. Peak-to-peak signal to root mean square (rms) value of the noise as well as rms signal-to-noise ratio are widely used by the television industry as a measure of television signal quality. The advantage of this measure is that it is independent of scale – if the data is scaled by a factor of 2 and the MSE is scaled by a factor of 2, then the SNR remains the same. It also gives a feel for the number of bits of meaningful data there are. For example, an increase in SNR of 10 dB corresponds to roughly 1.7 additional bits of useful data. The definition used in the previous study for SNR is

$$\mathrm{SNR} = 10 \log_{10} \frac{p^2}{\varepsilon^2} \qquad (5\text{-}2)$$

where ε is as defined in Equation (5-1) and p is the peak-to-peak possible range of the data. P was taken to be 127. The same measure is used in this study. However, because band 4 of the LANDSAT data is only 6 bit data rather than 7 bit as are the other bands, a slightly more appropriate measure of SNR would be obtained by scaling the error in each band according to its own range. Then

$$SNR = 10 \log_{10} \frac{1}{M} \sum_{j=1}^{M} \frac{p_j^2}{\varepsilon_j^2} \tag{5-3}$$

where

p_j is the data range in band j

ε_j is the mean square error in band j

M is the number of bands

For purposes of comparison with the previous study results we use that definition. The difference in results between the two definitions is less than a dB.

Figures 5-3 and 5-4 are plots of the SNR performance of the four selected algorithms. Since SNR is basically a way of normalizing the mean square error, conclusions based on the MSE can also be derived from the plots of SNR. Several additional conclusions can be best described using the SNR plots:

a) The two-dimensional Cosine transform encoder performs about 2 dB better than the two-dimensional Hadamard transform encoder in the region of 0.5 to 2 bits per pixel.

Better performance from the Cosine is to be expected, since the basis functions are smoother and less sensitive to orientation. This comparison was performed only for the agricultural scene.

b) The clustering algorithm and the 2D Cosine transform are the best performers. Their results are within 1.5 dB of one another. As one would expect, these are also the most complex algorithms to implement.

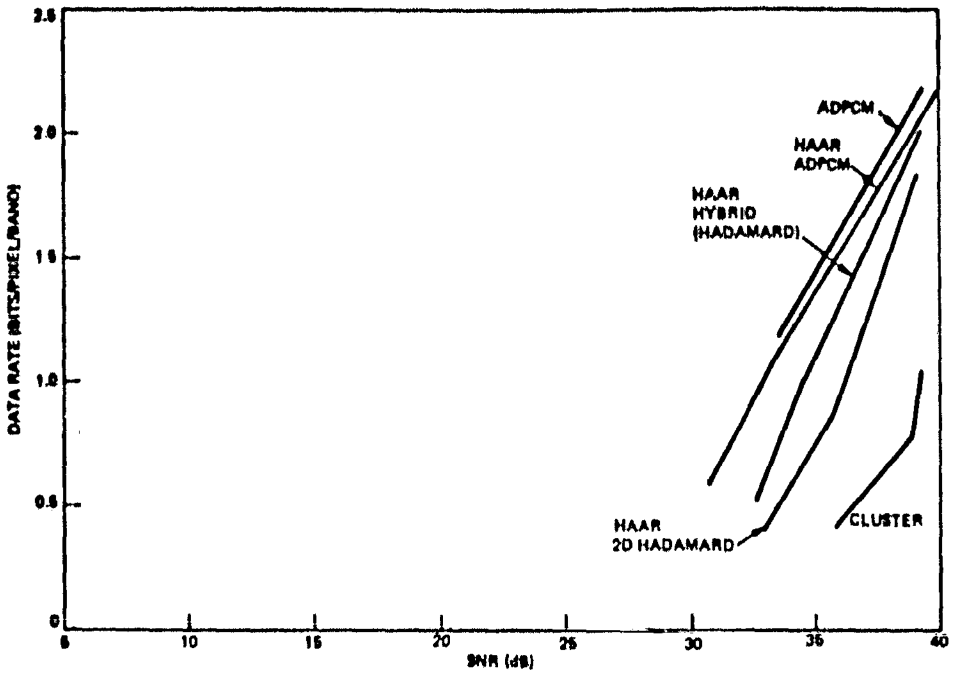


Figure 5-3. Data Rate Versus Signal-to-Noise Ratio — Bald Knob Scene

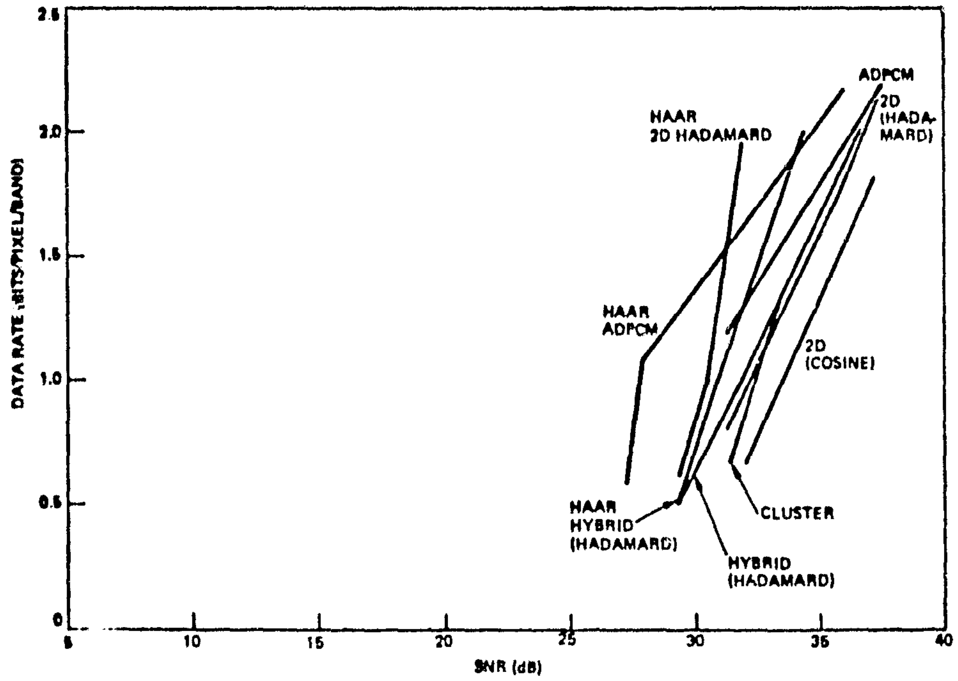


Figure 5-4. Data Rate Versus Signal-to-Noise Ratio -- Agricultural Scene

c) The hybrid (Hadamard) technique provides higher SNR than any of the recommended techniques of the previous study, On-Board Compression of Earth Resources Data (NAS2-8394). See Figure 9-2 of the previous study. The significance of this finding is that the complexity of the hybrid transform is considerably less than that of the previously recommended algorithm. It does not require calculating correlation matrices and their eigenvectors, as does the KL transform, nor does it require the numerous iterations of the clustering algorithm.

5.2.3 Classification Consistency

Many important uses of the earth resources data rely heavily on the use of computerized pattern classification and pattern recognition of the multispectral earth resources imagery. For pattern classification applications, the multispectral data is frequently first used to obtain a clustered image. This clustered image is then used for image data extraction and classification. Thus, it is of importance that a particular bandwidth compression method not result in significant changes in the resulting clustered imagery, and a criterion of performance which can be employed for evaluating various coding algorithms is the performance of various encoders in maintaining clustering consistency. That is, the clustered image obtained from the encoded multispectral data should be identical to the clustered picture obtained from the original set of multispectral data. The degree to which the clusters differ is a measure of compression-induced change in classification.

To test the degree of classification consistency, a set of programs was developed for TRW's Interdata 80 Image Processing Facility. These programs, which are described in the previous study (NAS2-8394), make it possible to do classification on multispectral imagery; to compare results obtained for compressed and uncompressed data; and to display the image itself, the clustered imagery, and the difference between pairs of clustered images. A flow diagram showing the sequence in which these programs are used in obtaining a measure of preservation of classification accuracy is shown in Figure 5-5. The clustering programs' flow is shown in Figure 5-6.

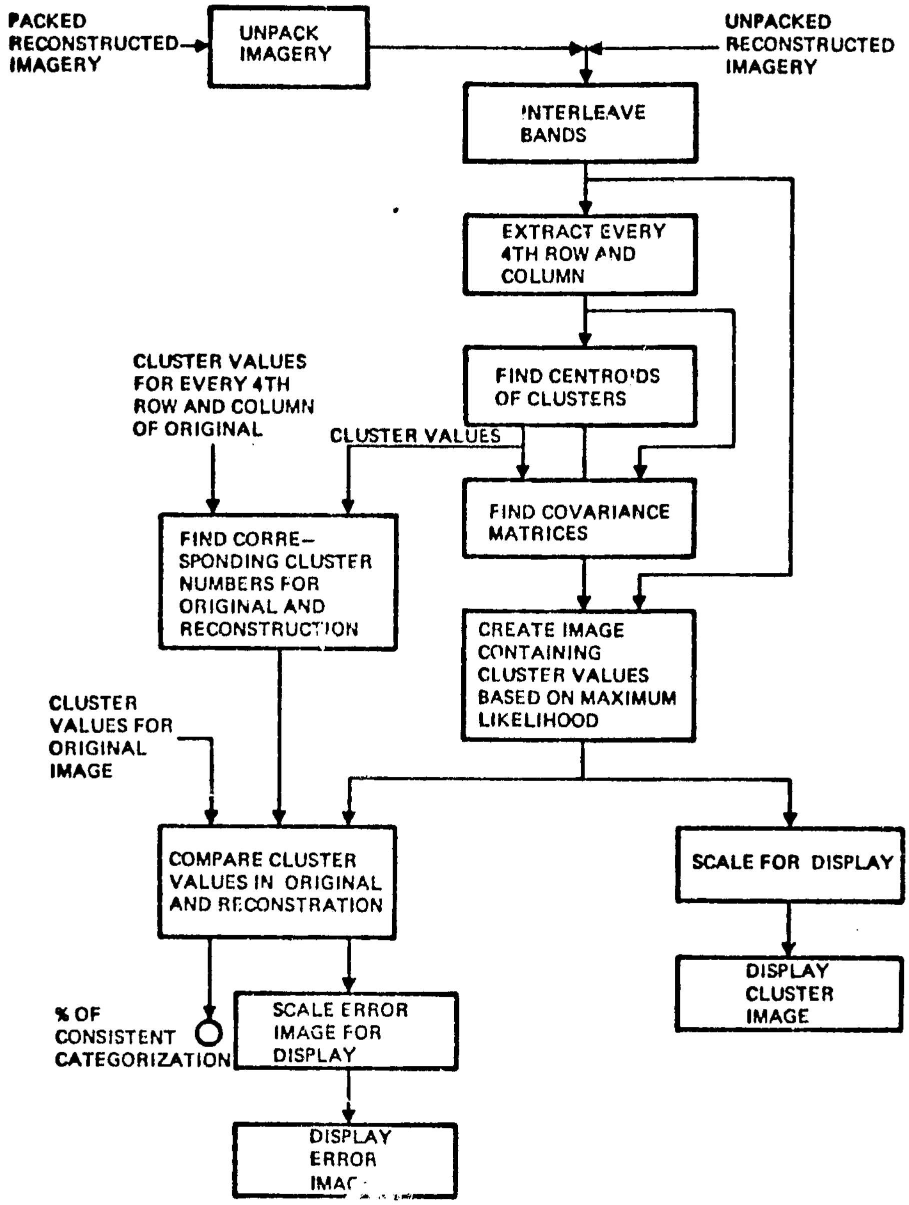

PACKED RECONSTRUCTED IMAGERY
UNPACK IMAGERY
UNPACKED RECONSTRUCTED IMAGERY
INTERLEAVE BANDS
EXTRACT EVERY 4TH ROW AND COLUMN
FIND CENTROIDS OF CLUSTERS
CLUSTER VALUES FOR EVERY 4TH ROW AND COLUMN OF ORIGINAL
CLUSTER VALUES
FIND COVARIANCE MATRICES
FIND CORRE-SPONDING CLUSTER NUMBERS FOR ORIGINAL AND RECONSTRUCTION
CREATE IMAGE CONTAINING CLUSTER VALUES BASED ON MAXIMUM LIKELIHOOD
CLUSTER VALUES FOR ORIGINAL IMAGE
COMPARE CLUSTER VALUES IN ORIGINAL AND RECONSTRATION
SCALE FOR DISPLAY
DISPLAY CLUSTER IMAGE
% OF CONSISTENT CATEGORIZATION
SCALE ERROR IMAGE FOR DISPLAY
DISPLAY ERROR IMAGE

Figure 5-5. Measuring Preservation of Classification Accuracy

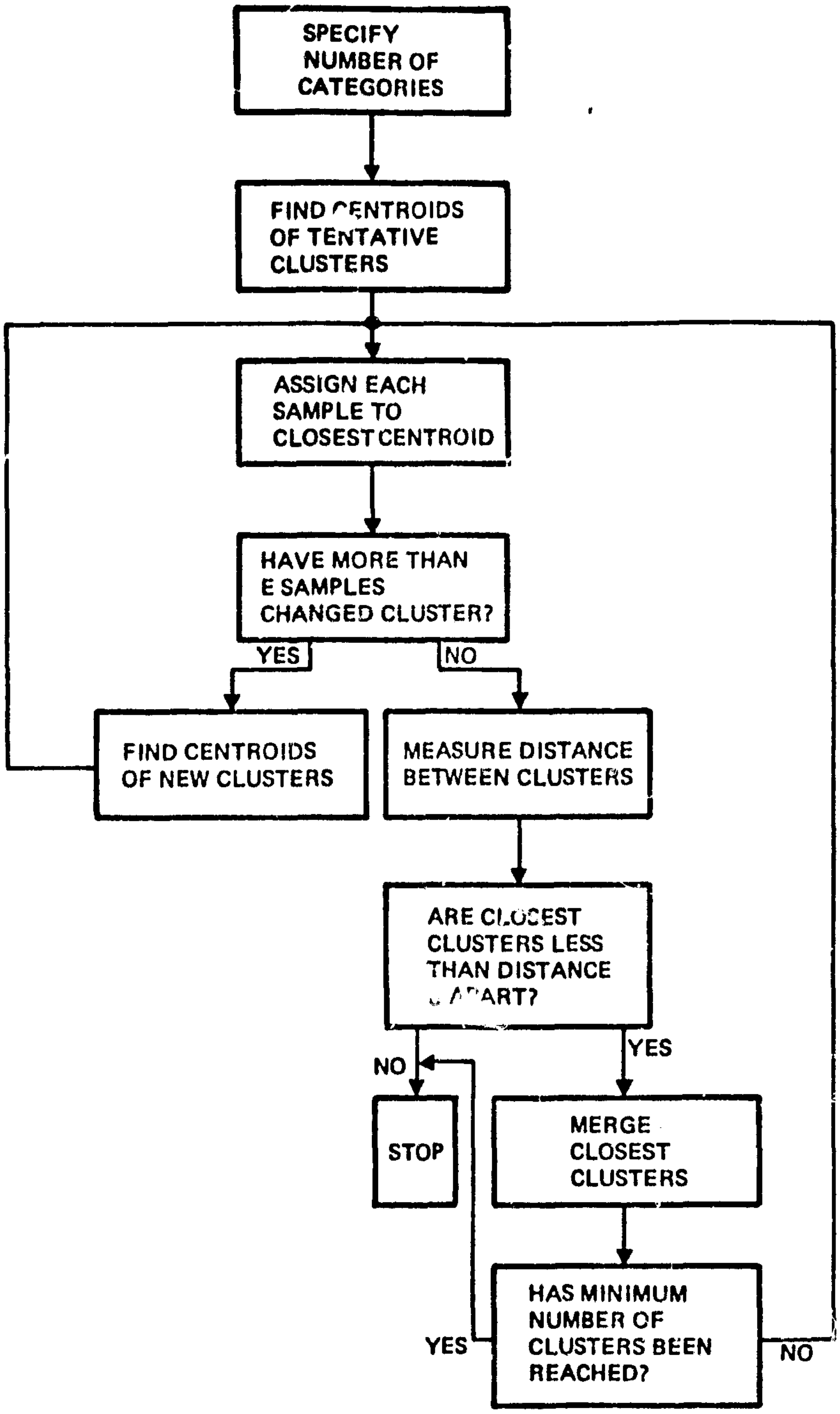

SPECIFY NUMBER OF CATEGORIES
FIND CENTROIDS OF TENTATIVE CLUSTERS
ASSIGN EACH SAMPLE TO CLOSEST CENTROID
HAVE MORE THAN E SAMPLES CHANGED CLUSTER?
YES
NO
FIND CENTROIDS OF NEW CLUSTERS
MEASURE DISTANCE BETWEEN CLUSTERS
ARE CLOSEST CLUSTERS LESS THAN DISTANCE ∪ APART?
NO
YES
STOP
MERGE CLOSEST CLUSTERS
HAS MINIMUM NUMBER OF CLUSTERS BEEN REACHED?
YES
NO

Figure 5-6. Clustering Technique Used

Figures 5-7 and 5-8 are plots of the classification consistency provided by the four selected algorithms. Results for two variations of the cluster coding technique are given in Figure 5-8. For all other simulations the minimum number of clusters allowed in each block was seven. In Figure 5-8 we have plotted results for a minimum number of clusters per block of seven and of one. Implications of the plots are:

a) Small changes in intensity can make a significant difference in how data clusters together. For example, at 2 bits per pixel per band, MSE was less than 5 for the bald knob scene, yet the classification consistency varied between 60 and 70 percent. This indicates that the intensity distinguishing the various clusters was very small for the Bald Knob scene. Inspection of the images corroborates this conclusion.

b) Classification consistency is not strongly correlated with MSE. Classification consistency for the agricultural scene is generally higher than that for the Bald Knob Scene. This may be due to the cluster centroids in the Bald Knob scene being closer together than in the agricultural scene.

c) The effect of a particular compression algorithm on classification is very scene-dependent. Hence, evaluation of effects should be done on an application-by-application basis. It should be noted that, even with noncompressed data, automatic classification leads to mixed results — depending on training samples, the classes to be distinguished, and overall scene content.

5.2.4 Subjective Quality

To determine the general qualitative effects of using a particular compression algorithm, photographic prints were made of the compression results. These prints were also valuable in making sure that the software was performing correctly.

Mean square error, SNR, and classification consistency are all measures that provide an average value over an entire data set. Subjective examination of photographs reveals local error structures introduced by compression algorithms. For example, the 2D Hadamard compression algorithm at low bit rates introduces some rectangular structure in the image. This structure is not revealed by the quantitative measures described above, but is readily visible in photographs of the reconstructions.

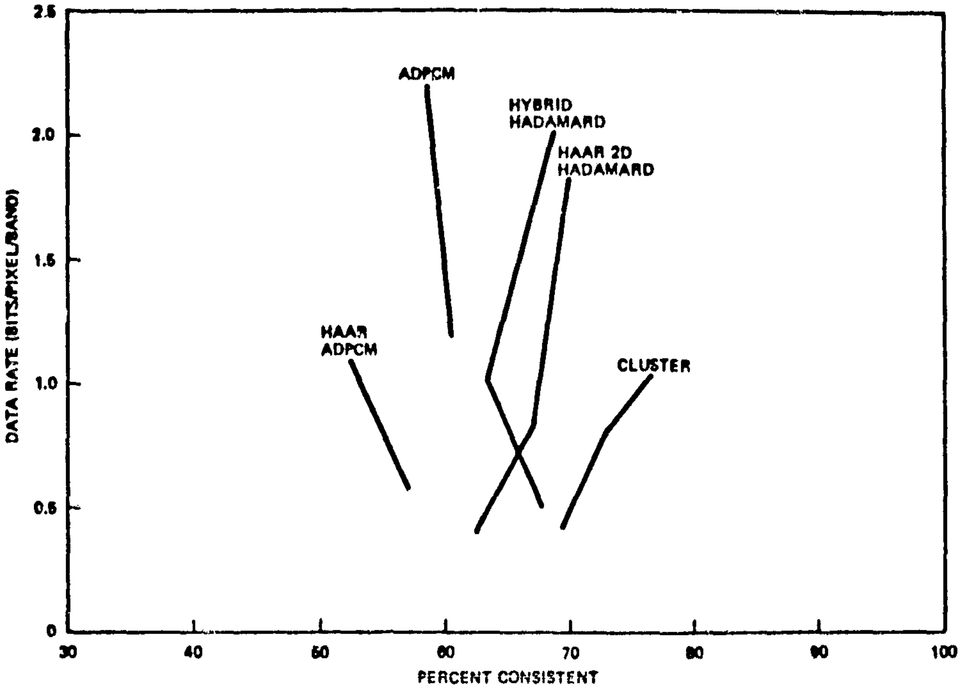


Figure 5-7. Data Rate Versus Classification Consistency Bald Knob Scene

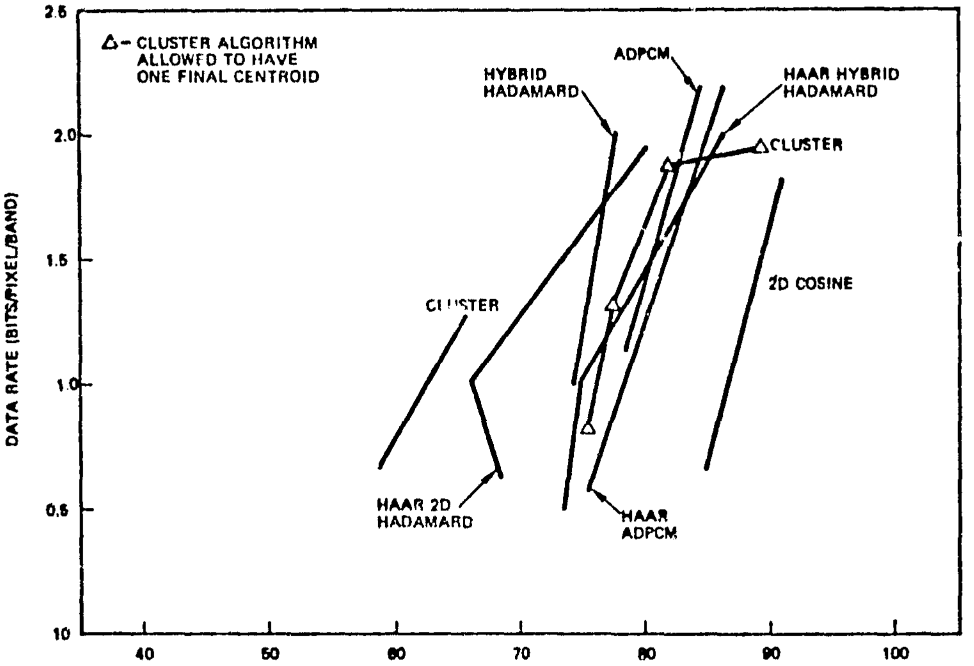


Figure 5-8. Data Rate Versus Classification Consistency Agricultural Scene

Figures 5-9 through 5-19 allow one to judge subjectively the performance of the algorithms at different bit rates for both scenes processed. The photos correspond to the following algorithms:

- Haar/ADPCM
- Haar/hybrid (Hadamard)
- Haar/2D Hadamard
- Clustering.

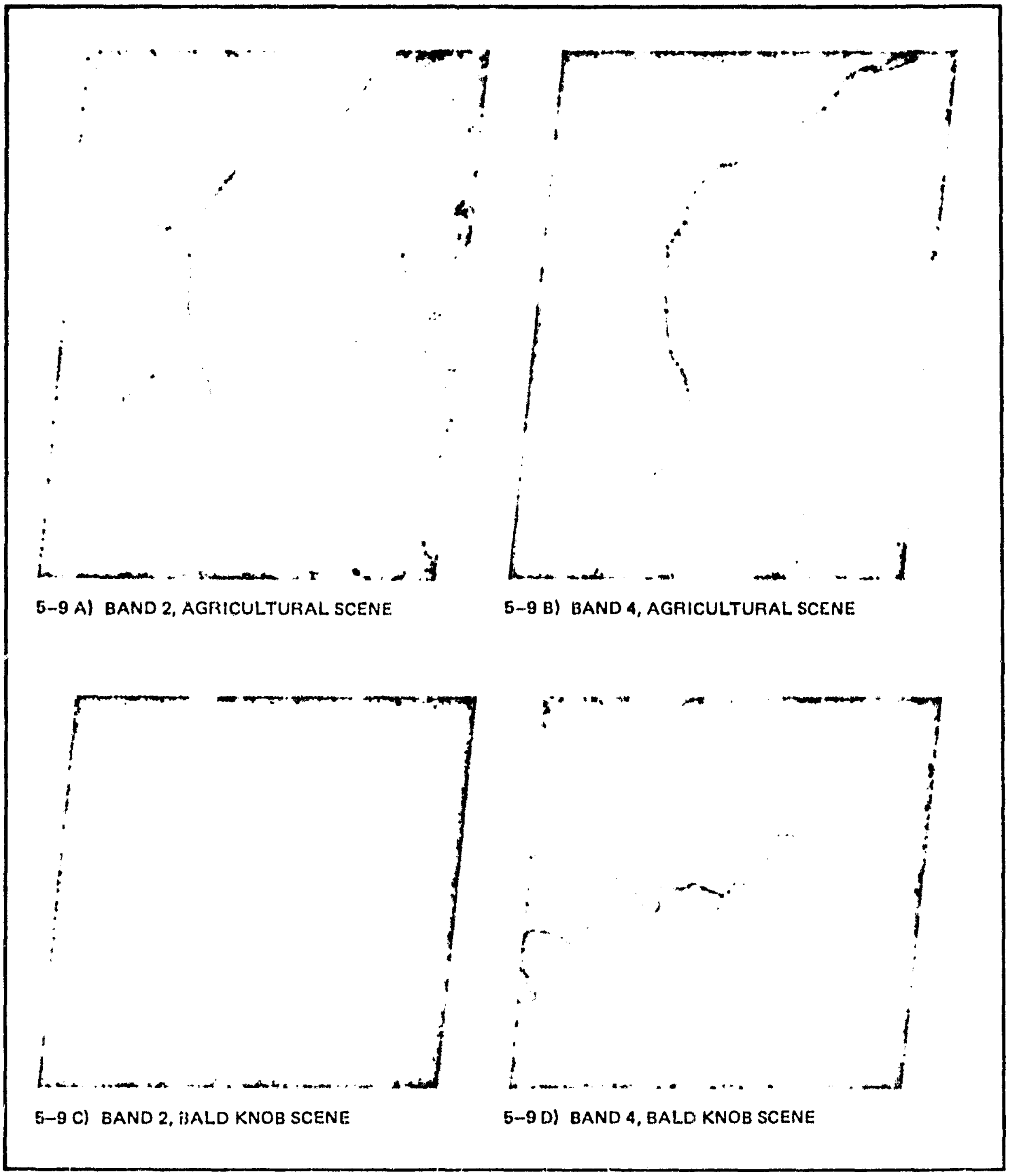
The rates are as indicated in the figures. Only two bands are provided because they are enough to illustrate the essential behavior of each algorithm. Qualitative evaluation of the photographic results leads to the following conclusions:

a) At 2 bits per pixel per band, all the algorithms yield excellent results with virtually unnoticeable artifacts.

b) At bit rates below 2 bits per pixel per band, adaptive DPCM produces a blurred appearance — edges are less sharp and overall contrast is reduced. These effects are particularly pronounced on the agricultural scene because of its many edges. All these effects can be attributed to slope overload. The predictor error cannot be quantized accurately to follow sharp edges.

c) At bit rates below 2 bits per pixel per band, the hybrid (Hadamard) algorithm yields blocky results. These blocks are due to the fact that few of the transform coefficients have nonzero bit allocations.

d) The clustering algorithm leads to crisp images at all bit rates. However, contouring is evident somewhat at 1 bit and a significant amount at 0.5 bit. The crispness and contouring are both due to the same feature of the algorithm. The clustering algorithm does not use the small difference between adjacent radiance values to obtain compression. Instead, it groups elements into classes using an entire 16 x 16 block of data. Thus, edges can be very sharp. On the other hand, all elements in the same classes are represented by the same radiance. This means that at low bit rates, where there are very few classes, large regions of constant intensity (contours) result.



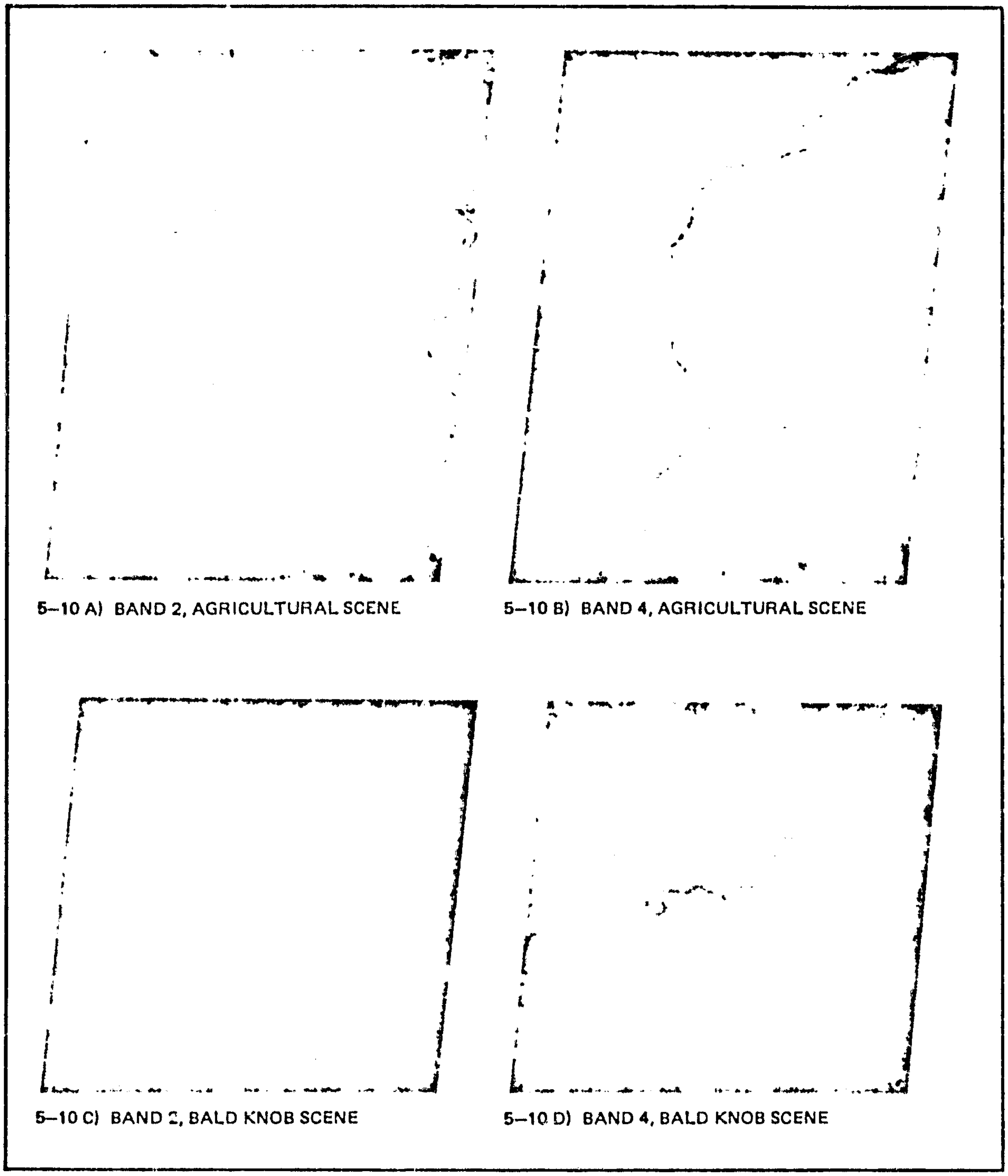
5-9 A) BAND 2, AGRICULTURAL SCENE

5-9 B) BAND 4, AGRICULTURAL SCENE

5-9 C) BAND 2, BALD KNOB SCENE

5-9 D) BAND 4, BALD KNOB SCENE

Figure 5-9. Haar/Adaptive DPCM Performance at 2.19 Bits/Pixel/Band



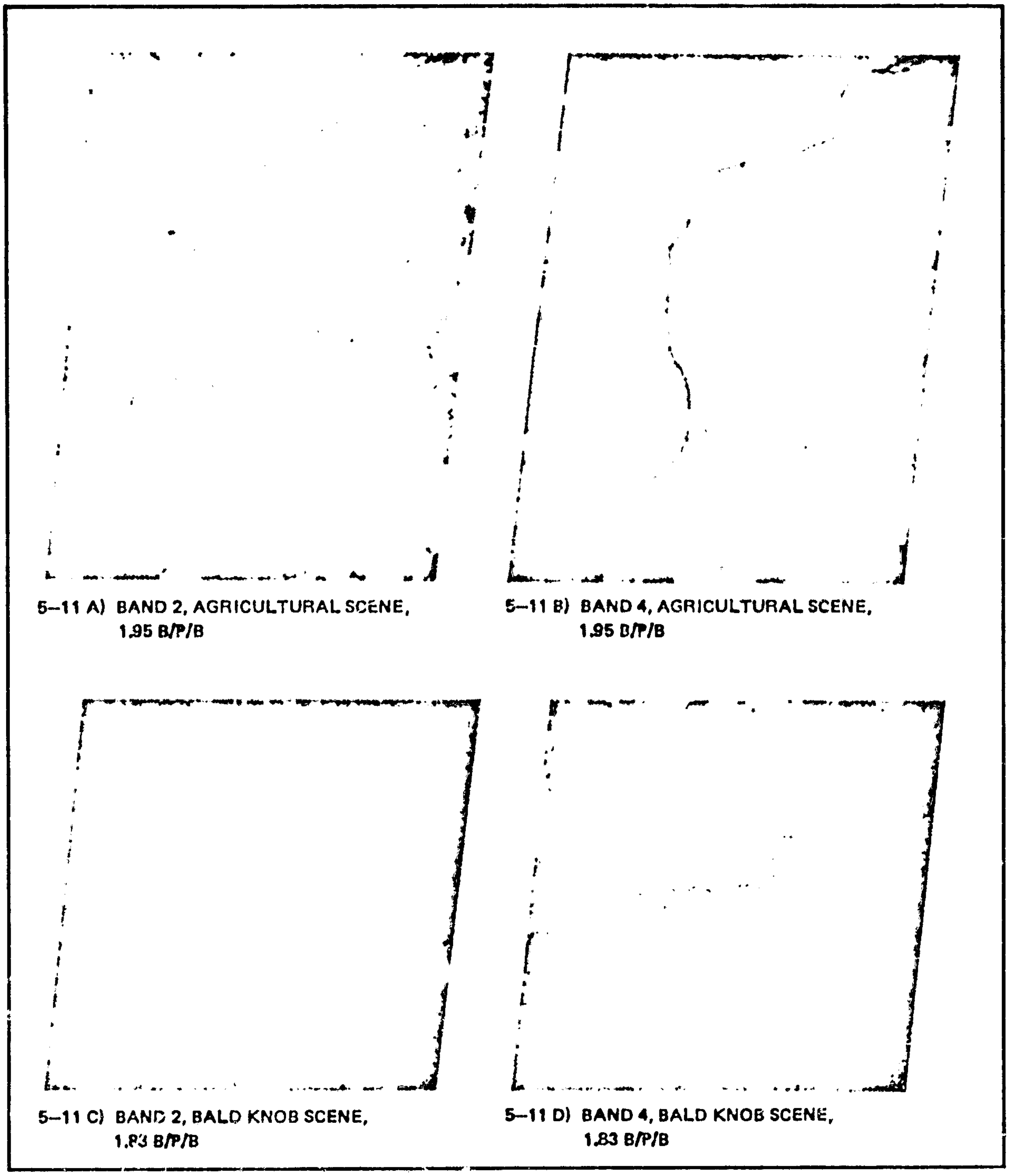
5–10 A) BAND 2, AGRICULTURAL SCENE

5–10 B) BAND 4, AGRICULTURAL SCENE

5–10 C) BAND 2, BALD KNOB SCENE

5–10 D) BAND 4, BALD KNOB SCENE

Figure 5-10. Haar/Hybrid Hadamard Performance at 2 Bits/Pixel/Band



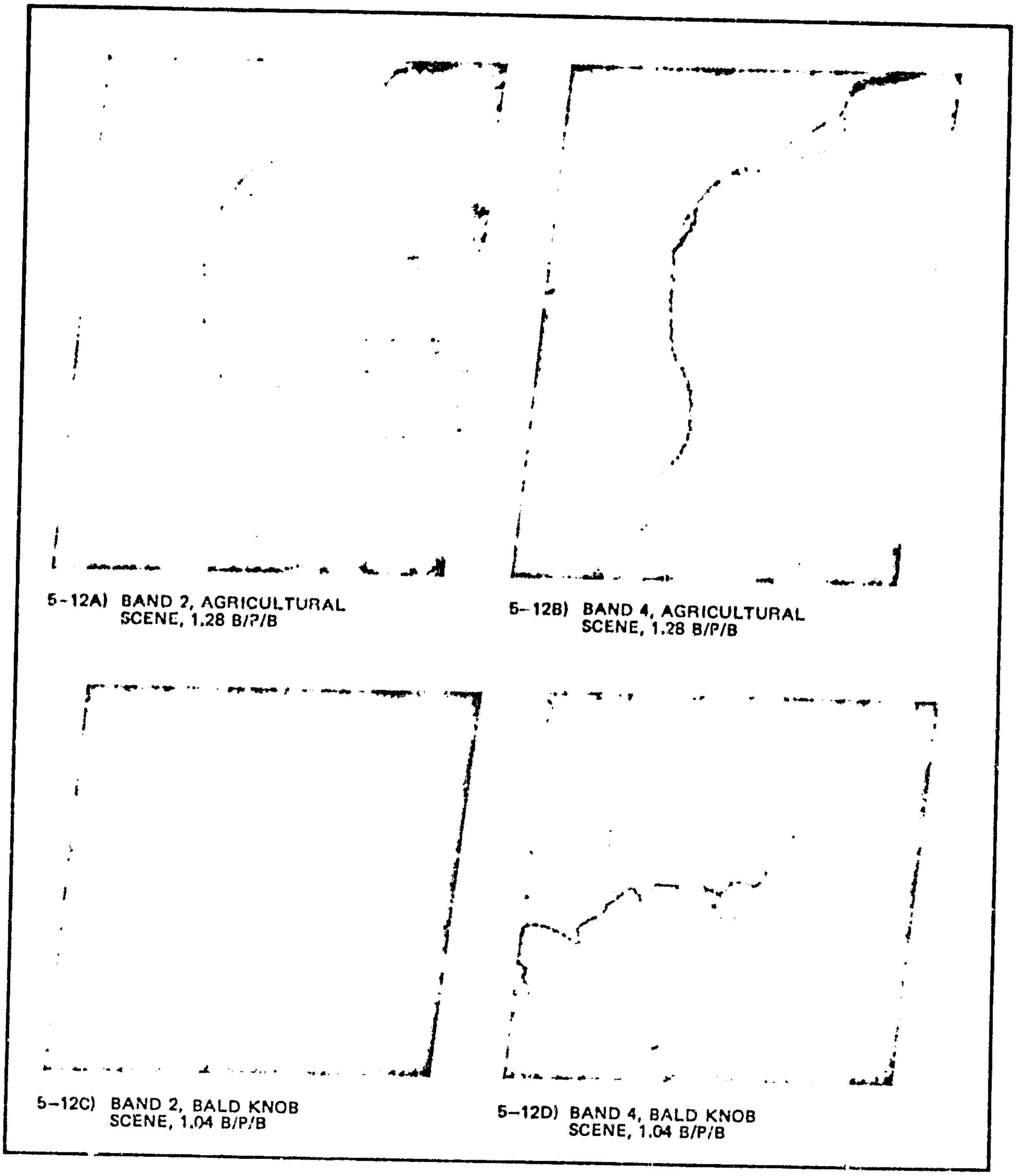
5–11 A) BAND 2, AGRICULTURAL SCENE, 1.95 B/P/B

5–11 B) BAND 4, AGRICULTURAL SCENE, 1.95 B/P/B

5–11 C) BAND 2, BALD KNOB SCENE, 1.83 B/P/B

5–11 D) BAND 4, BALD KNOB SCENE, 1.83 B/P/B

Figure 5-11. Haar/2D Hadamard Performance at Approximately 2 Bits/Pixel/Band



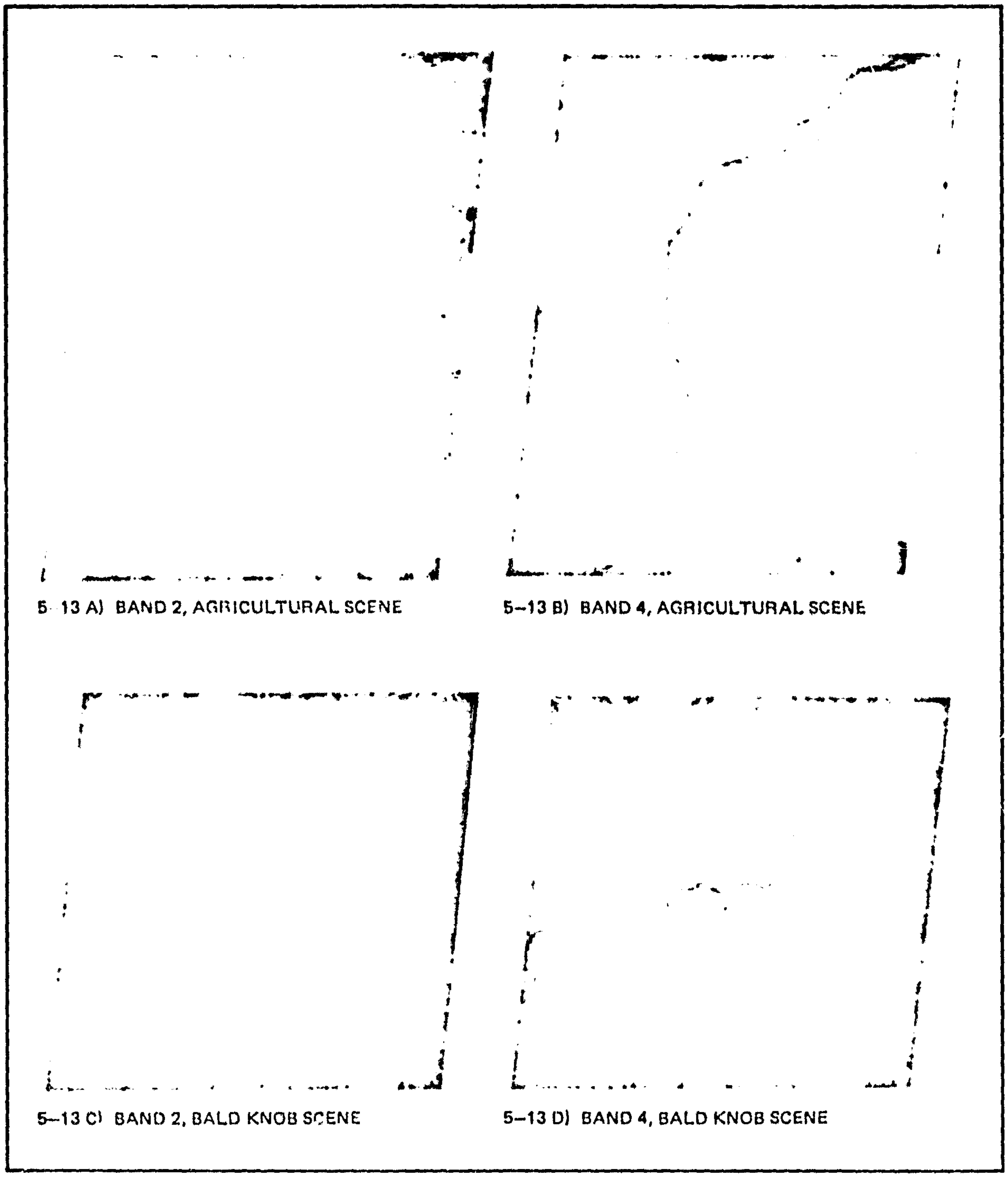
5-12A) BAND 2, AGRICULTURAL SCENE, 1.28 B/P/B

5-12B) BAND 4, AGRICULTURAL SCENE, 1.28 B/P/B

5-12C) BAND 2, BALD KNOB SCENE, 1.04 B/P/B

5-12D) BAND 4, BALD KNOB SCENE, 1.04 B/P/B

Figure 5-12. Adaptive Clustering Performance at Approximately 1 Bit/Pixel/Band



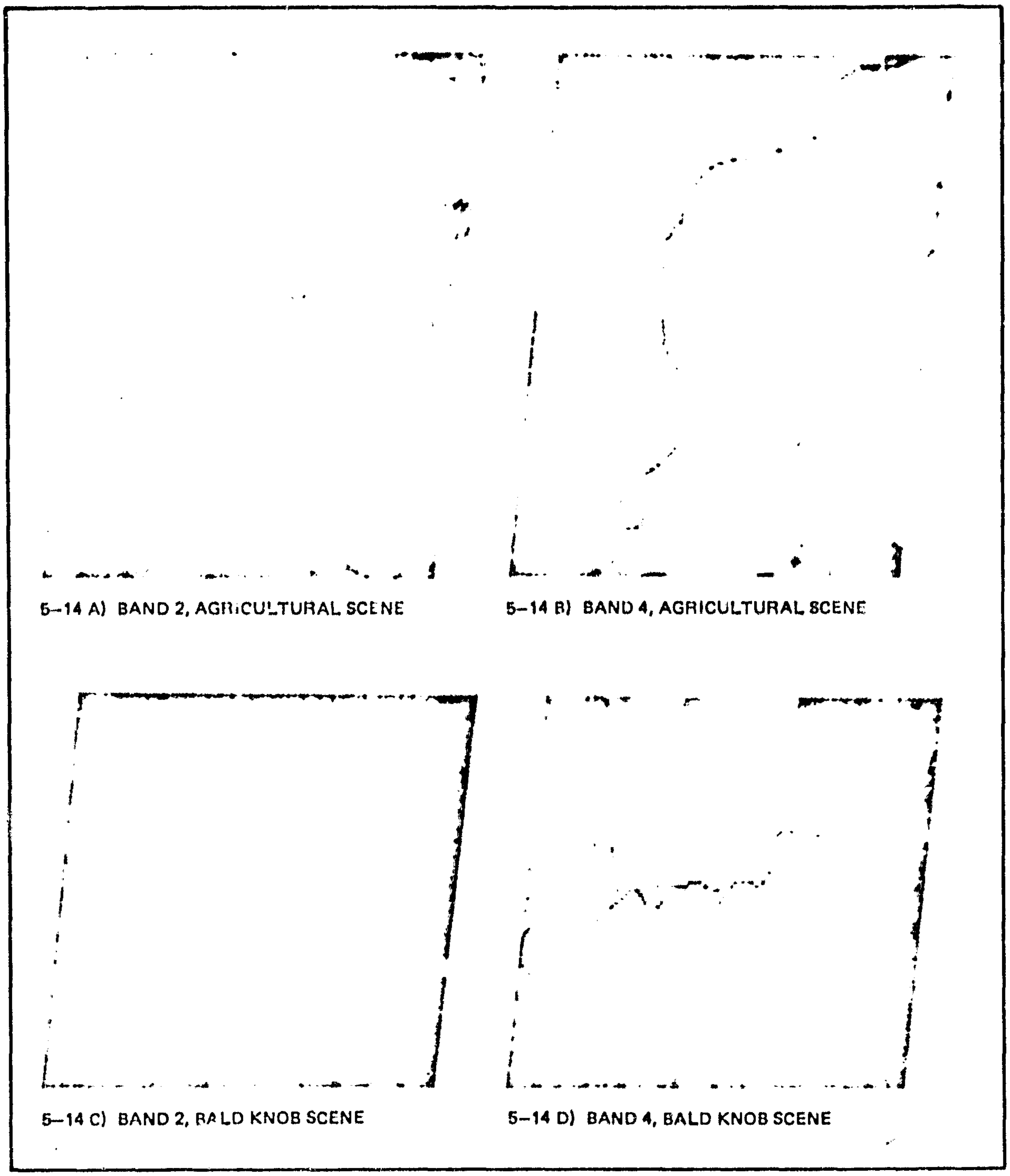
5–13 A) BAND 2, AGRICULTURAL SCENE

5–13 B) BAND 4, AGRICULTURAL SCENE

5–13 C) BAND 2, BALD KNOB SCENE

5–13 D) BAND 4, BALD KNOB SCENE

Figure 5-13. Haar/Adaptive DPCM Performance at 1.09 Bits/Pixel/Band



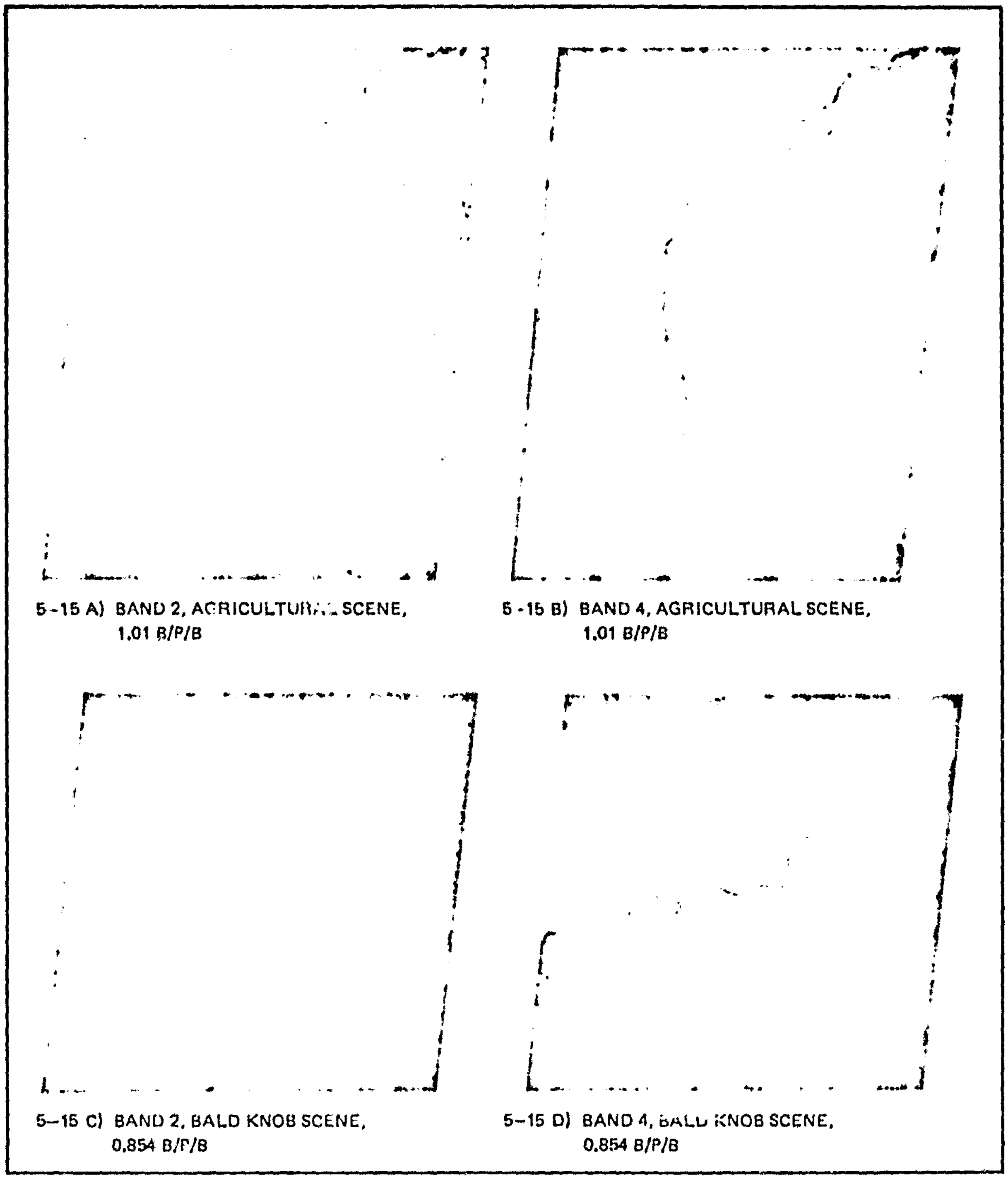
5–14 A) BAND 2, AGRICULTURAL SCENE

5–14 B) BAND 4, AGRICULTURAL SCENE

5–14 C) BAND 2, BALD KNOB SCENE

5–14 D) BAND 4, BALD KNOB SCENE

Figure 5-14. Haar/Hybrid Hadamard Performance at 1 Bit/Pixel/Band



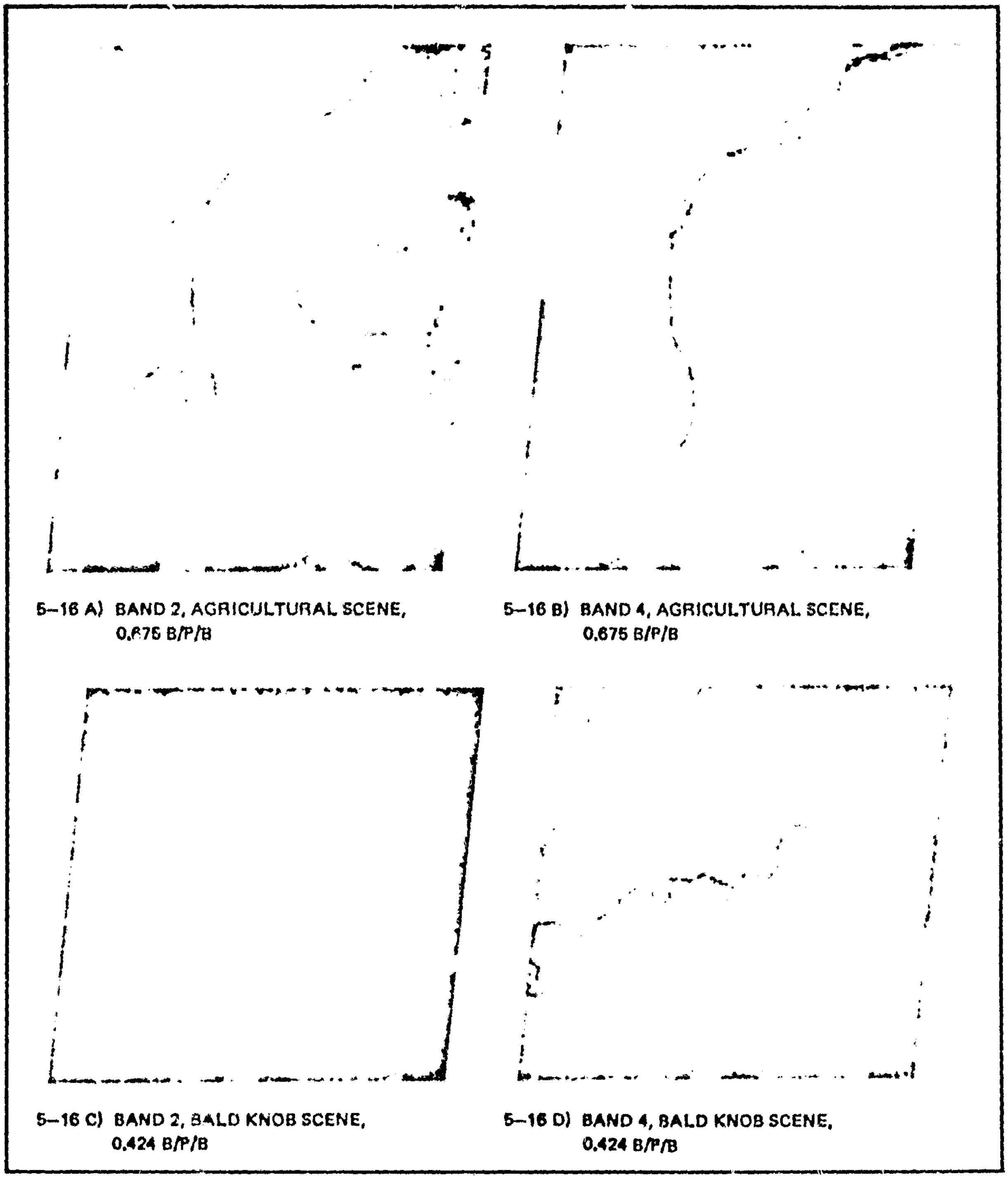
5-15 A) BAND 2, AGRICULTURAL SCENE, 1.01 B/P/B

5-15 B) BAND 4, AGRICULTURAL SCENE, 1.01 B/P/B

5-15 C) BAND 2, BALD KNOB SCENE, 0.854 B/P/B

5-15 D) BAND 4, BALD KNOB SCENE, 0.854 B/P/B

Figure 5-15. Haar/2D Hadamard Performance at Approximately 1 Bit/Pixel/Band



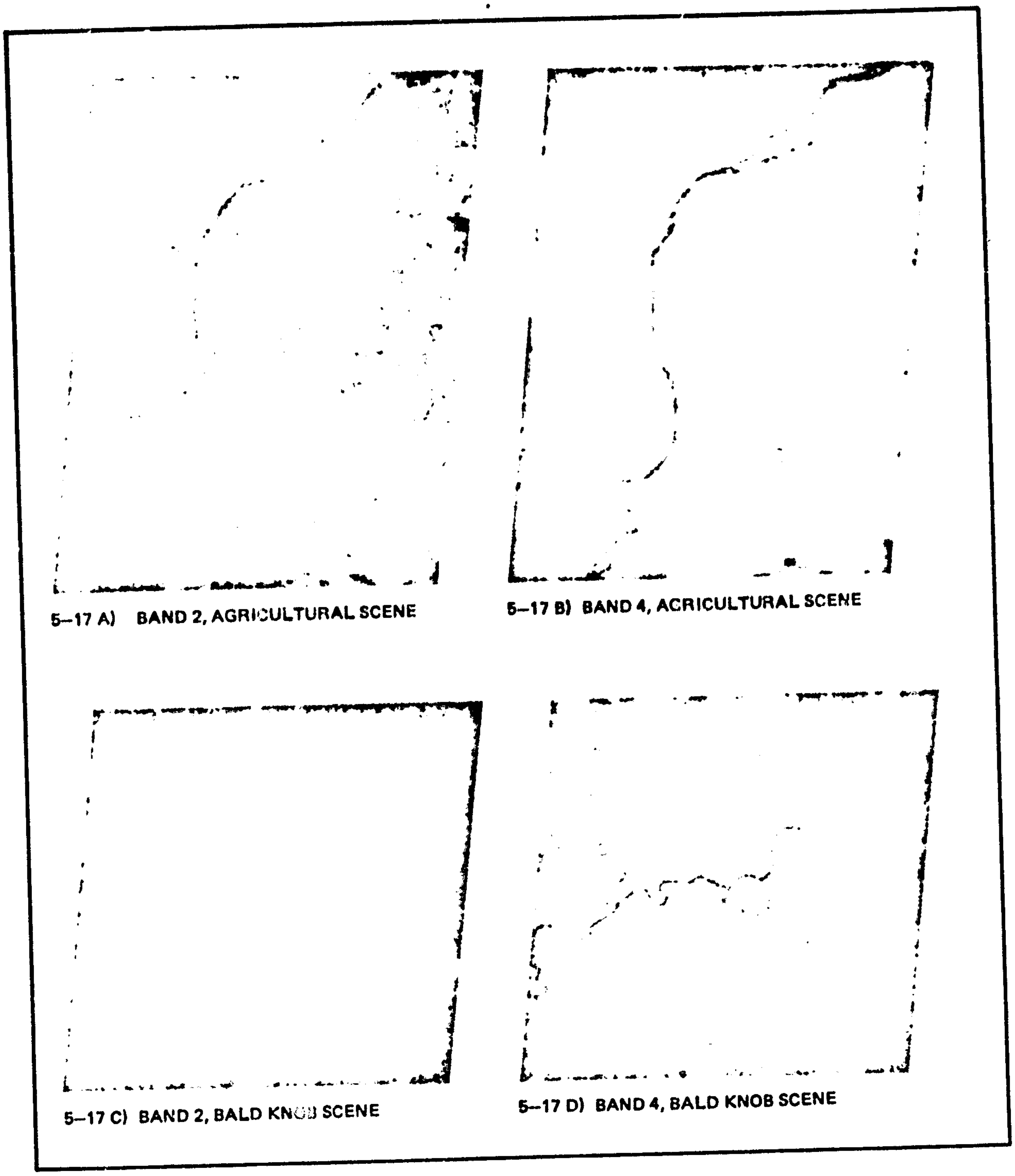
5–16 A) BAND 2, AGRICULTURAL SCENE, 0.675 B/P/B

5–16 B) BAND 4, AGRICULTURAL SCENE, 0.675 B/P/B

5–16 C) BAND 2, BALD KNOB SCENE, 0.424 B/P/B

5–16 D) BAND 4, BALD KNOB SCENE, 0.424 B/P/B

Figure 5-16. Adaptive Clustering Performance at Approximately 0.5 Bit/Pixel/Band



5–17 A) BAND 2, AGRICULTURAL SCENE

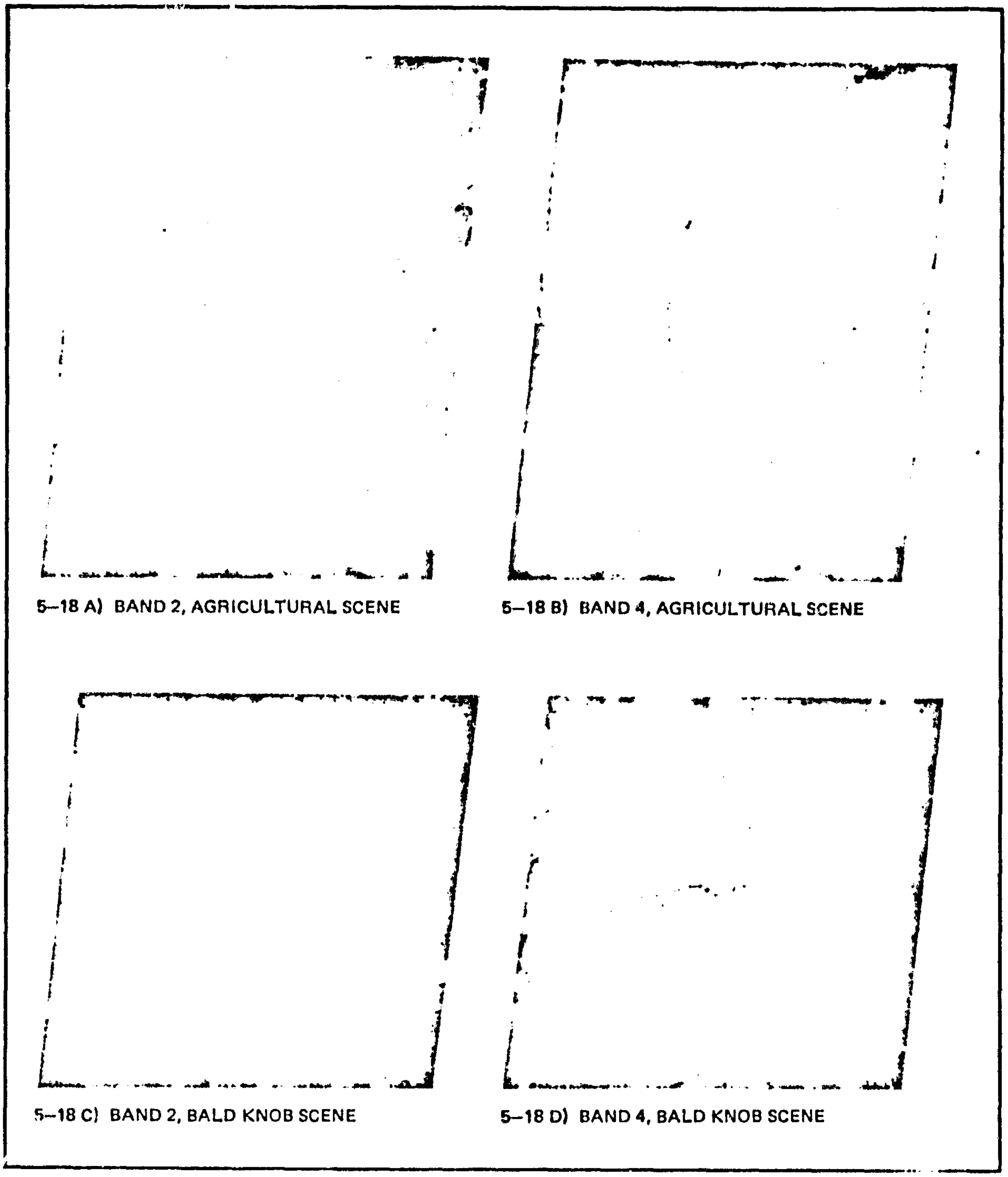
5–17 B) BAND 4, ACRICULTURAL SCENE

5–17 C) BAND 2, BALD KNOB SCENE

5–17 D) BAND 4, BALD KNOB SCENE

Figure 5-17. Haar/Adaptive DPCM Performance at 0.59 Bit/Pixel/Band

C-2



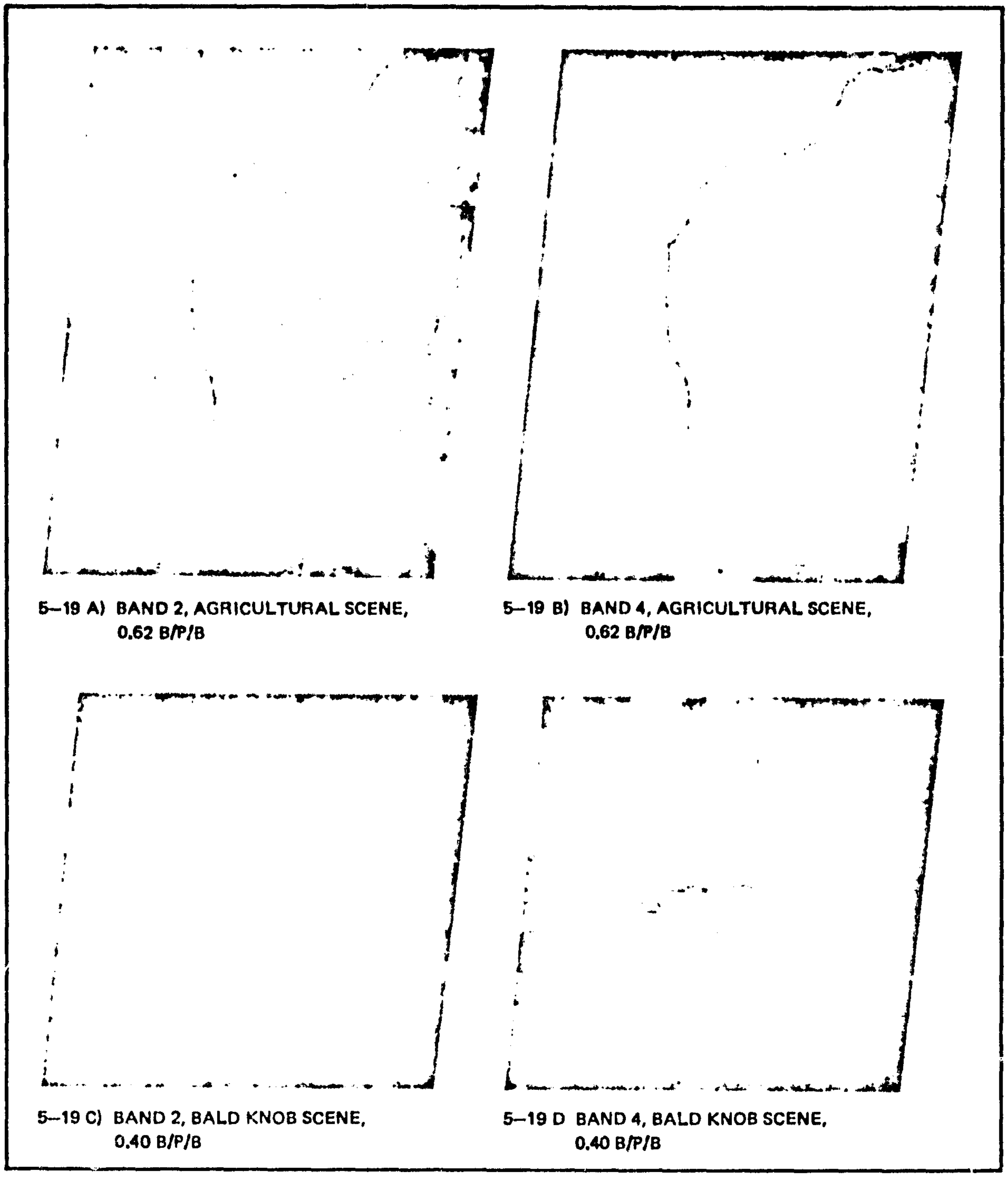
5–18 A) BAND 2, AGRICULTURAL SCENE

5–18 B) BAND 4, AGRICULTURAL SCENE

5–18 C) BAND 2, BALD KNOB SCENE

5–18 D) BAND 4, BALD KNOB SCENE

Figure 5-18. Haar/Hybrid Hadamard Performance at 0.5 Bit/Pixel/Band



5–19 A) BAND 2, AGRICULTURAL SCENE, 0.62 B/P/B

5–19 B) BAND 4, AGRICULTURAL SCENE, 0.62 B/P/B

5–19 C) BAND 2, BALD KNOB SCENE, 0.40 B/P/B

5–19 D BAND 4, BALD KNOB SCENE, 0.40 B/P/B

Figura 5-19. Haar/2D Hadamard Performance at Approximately 0.5 Bit/Pixel/Band

5.3 SYSTEM CONSIDERATIONS

In addition to the criteria of optimality which are used to evaluate and compare the performance of various techniques, there exists a different set of criteria which deals with the systems aspects of the various techniques. This set of criteria is particularly important in design and operation of the system under the imperfect conditions imposed by the real world. These criteria are discussed in the following sections.

5.3.1 Computation and Implementation Complexity

The complexity of any technique is eventually measured in terms of the total number of parts, weight, power, and the volume required. However, before one can specify the above design parameters, one must specify the number of operations, the memory which is required for implementing a particular bandwidth compression technique, and the environment in which the compressor will operate. In Section 7, several of the algorithms are sized for a Thematic Mapper type application. Considerations such as data format and scanning properties are included in the evaluation.

5.3.2 Sensor Imperfections

A number of different sensor phenomena contribute to degraded compression performance compared with that expected for an ideal sensor. To explain those sensor properties which adversely affect compression, first two types of multispectral sensors which are most likely to be used for future satellite-based gathering of earth resources data are examined. They are the Thematic Mapper and High Resolution Pointable Imager. The most important sensor parameters which may affect the bandwidth compression performance of the selected techniques are:

- Photodetector nonuniformity
- Signal-to-noise ratio
- Radiometric nonlinearity
- Spectral misregistration
- Geometric distortion due to satellite attitude variations
- Geometric distortion due to scan pattern
- Data rate.

The affect of these parameters on various bandwidth compression methods is analyzed by measuring the impact of the above imperfections on the correlation of the data and relating that to the expected performance of the selected bandwidth compression algorithms. Based on results of the On-Board Compression of Earth Resources Data Study, these effects are likely to have little impact on compressor performance.

5.3.3 Channel-Error Effect

A different type of imperfection present in most communication systems is the channel error. This, in general, includes perturbation of the transmitted signal due to atmospheric turbulence, natural and man-made interference, and thermal noise present in the transmitter and receiver of the system. In digital communication systems, the overall effect of the above imperfections is expressed in terms of bit error rate (BER) which is the percentage of binary integers which are detected erroneously. Naturally, this depends on the type of receiver and the modulation technique that is used in a particular communications system. A fixed bit-error rate affects some bandwidth compression methods more severely than others. For instance, in a transform coding system the channel error occurring at a particular transform component distorts a specific frequency component of the image. This degradation appears at all points in the image having a contribution from that particular frequency component. As such it has a different effect on a human observer than an equal perturbation occurring in the spatial domain directly, as happens in DPCM systems. For adaptive techniques the overhead information is particularly important. The key to obtaining good results is that the overhead be protected against errors.

To adequately study these effects, one must complete an overall system design, including transmission channel formatting, before the total effect of channel errors on data reconstruction can be evaluated.

5.3.4 Effect of Scene Dynamic Range

Haze and other atmospheric conditions can cause the effective scene contrast to be reduced at the sensor. It is important to know the effect that such contrast attenuation has on data compression performance. To get a feeling for compressor contrast sensitivity, we made selected additional simulations.

For the Bald Knob Scene we reduced the contrast in each band by a factor of 2 by scaling each band by 0.5. Then we performed compression at 1 bit per pixel (not including overhead) using the adaptive Haar-2D Hadamard, Haar-Hybrid Hadamard, and Haar-ADPCM techniques. The mean square error between the original data and twice the reconstruction was then calculated. Table 5-2 compares the results with the corresponding results for the unattenuated data.

Table 5-2. Corresponding Mean Square Errors for Bald Knob Scene

Algorithm	Mean Square Error	
	Original Data	Attenuated Data
Haar-2D Hadamard	4.5	7.3
Haar-ADPCM	7.6	8.0
Haar-Hybrid Hadamard	5.8	7.1

The DPCM algorithm is least sensitive to attenuation of the data and the 2D Hadamard technique is most sensitive. Approximately 0.5 is added to the mean square error because of quantization of odd valued samples when they are scaled down by a factor of 2 and the reconstruction is doubled. Thus, the DPCM technique itself introduces no significant additional error in the case of a factor of 2 attenuated signal. This is to be expected since the quantizer is adjusted in proportion to the data standard deviation in each block. On the other hand, the two-dimensional transform techniques determine how many bits to assign to successive coefficients based on the quantized value of the previous coefficient. This recursive bit assignment means that any error introduced in the first coefficient can propagate to the other coefficients, leading to increased overall error.

6. ALGORITHM APPLICABILITY

There are several sequences of steps that data can take in going from the sensor to the ultimate user. Data compression may be applicable in several of these steps. The type of compression that is most appropriate for any particular link in the data dissemination chain depends on a variety of criteria such as encoding complexity, decoding complexity, sensitivity to channel errors, channel capacity, and compression ratio. In this section we specify the characteristics that define each link in the chain and propose specific forms of compression best suited for each link.

Figure 6-1 summarizes the data flow for LANDSAT. For each step the salient properties are listed.

The key elements in the chain are the data gathering platform, in this case called LANDSAT; the direct readout station (an end user receiving data directly from the sensor); the central processing facility that collects, archives, and disseminates data to users; the intermediate user then can, for example, transmit data to a remote user in a van; and the end user who actually analyzes and draws conclusions from the data. Having a remote user in a van allows an image processing facility to be taken to a variety of users while maintaining access to a large data base through a communications link. The parameters that constrain each element in the data chain are indicated in Figure 6-1.

Table 6-1 indicates the suitability of each compression algorithm for each task in the data flow. The rationale for recommendations made in Table 6-1 are provided in Table 6-2. Since detailed analysis of algorithm implementation complexity was beyond the scope of the study with the exception of the Thematic Mapper application described in Chapter 7, the conclusions are based primarily on compression performance, computational complexity, and the function of each element in the chain. Applicability of each of the adaptive algorithms is considered for high, medium, and low (2, 1, 0.5 bits per pixel per band) data rates.

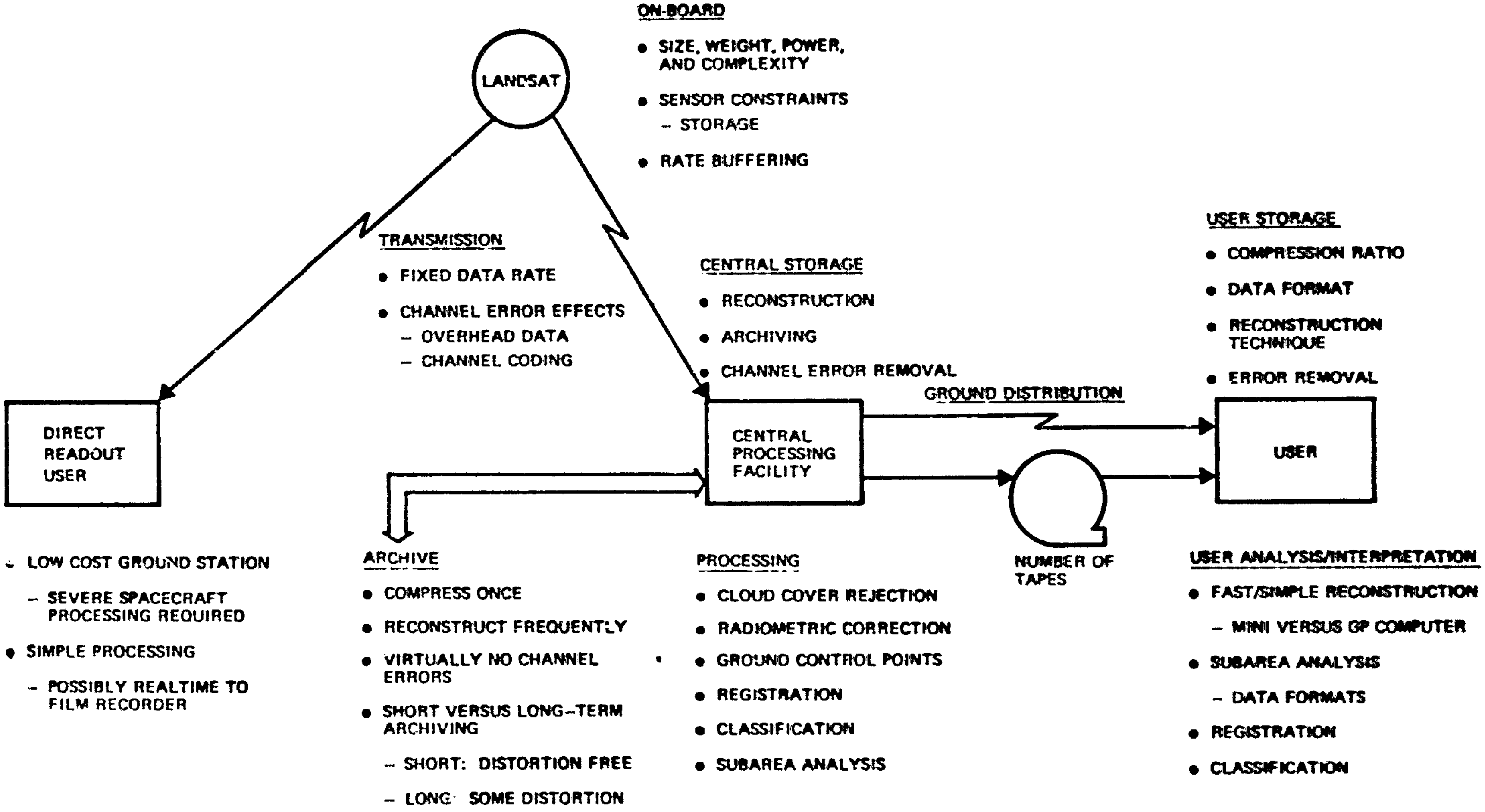


Figure 6-1. Data Processing Flow for LANDSAT

Table 6-1. Compression Applications – Suitability
(✓ Indicates Suitability)

APPLICATION / TECHNIQUE	SENSOR TO USER DIRECT	SENSOR TO CENTRAL PROCESSING FACILITY	ARCHIVING		CENTRAL FACILITY TO USER	INTERMEDIATE USER TO END	USER STORAGE	USER ANALYSIS
			SHORT TERM	LONG TERM				
EDITING	✓				✓	✓	✓	✓
DISTORTION FREE (ENTROPY CODING)		✓	✓	✓	✓			
PATTERN CLASSIFICATION	✓							✓
CLUSTERING								
HIGH RATE 2 BITS				✓				
MEDIUM RATE 1 BIT				✓				
LOW RATE 0.5 BIT				✓				
2D ADAPTIVE TRANSFORM								
HIGH RATE 2 BITS								
MEDIUM RATE 1 BIT								
LOW RATE 0.5 BIT	✓				✓			
ADAPTIVE HYBRID								
HIGH RATE 2 BITS			✓					
MEDIUM RATE 1 BIT	✓				✓	✓		
LOW RATE 0.5 BIT	✓				✓	✓		
ADAPTIVE 2D DPCM								
HIGH RATE 2 BITS	✓	✓	✓	✓	✓	✓		
MEDIUM RATE 1 BIT								
LOW RATE 0.5 BIT								
ADDITION OF SPECTRAL TRANSFORM	✓		✓		✓	✓		✓

Table 6-2. Compression Application — Rationale

APPLICATION / TECHNIQUE	SENSOR TO USER DIRECT	SENSOR TO CENTRAL PROCESSING FACILITY	ARCHIVING		CENTRAL FACILITY TO USER	INTERMEDIATE USER TO END USER	USER STORAGE	USER ANALYSIS
			SHORT TERM	LONG TERM				
EDITING	HIGH COMPRESSION SIMPLE TO IMPLEMENT	DISCARDS TOO MUCH DATA	DISCARDS TOO MUCH DATA	DISCARDS TOO MUCH DATA	HIGH COMPRESSION SIMPLE TO IMPLEMENT	HIGH COMPRESSION SIMPLE TO IMPLEMENT	HIGH COMPRESSION SIMPLE TO IMPLEMENT	HIGH COMPRESSION SIMPLE TO IMPLEMENT
DISTORTION FREE (ENTROPY CODING)	VARIABLE RATE	FACTOR OF TWO COMPRESSION, NO DISTORTION	FACTOR OF TWO COMPRESSION, NO DISTORTION	FACTOR OF TWO COMPRESSION, NO DISTORTION	FACTOR OF TWO COMPRESSION, NO DISTORTION RELATIVELY SIMPLE TO DECODE	VARIABLE RATE FAIRLY COMPLEX TO ENCODE	FAIRLY COMPLEX TO ENCODE	MUST BE DECODED TO BE INTERPRETED
PATTERN CLASSIFICATION	EXTREMELY HIGH COMPRESSION FOR A VERY LIMITED SET OF CLASSES	DISCARDS TOO MUCH DATA	DISCARDS TOO MUCH DATA	DISCARDS TOO MUCH DATA	NOT ACCURATE ENOUGH	EXTREMELY HIGH COMPRESSION FOR A VERY LIMITED SET OF CLASSES	DISCARDS TOO MUCH DATA UNLESS CLASSIFIED DATA IS FINAL RESULT	SUMMARIZES DATA
CLUSTERING								
HIGH RATE 2 BITS	TOO COMPLEX TO ENCODE, VARIABLE RATE	TOO COMPLEX TO ENCODE	TOO COMPLEX TO ENCODE, SMALL DISTORTION	EASY TO DECODE	MUCH MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	MORE COMPLEX TO ENCODE THAN COMPARABLY PERFORMING ALGORITHMS	TOO COMPLEX TO ENCODE	DOES NOT PROVIDE ANY AID IN ANALYSIS
MEDIUM RATE 1 BIT	TOO COMPLEX TO ENCODE	TOO COMPLEX TO ENCODE	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	TOO MUCH DISTORTION	MORE COMPLEX TO ENCODE THAN COMPARABLY PERFORMING ALGORITHMS	TOO COMPLEX TO ENCODE	DOES NOT PROVIDE ANY AID IN ANALYSIS
LOW RATE 0.5 BIT	TOO COMPLEX TO ENCODE	TOO COMPLEX ENCODE	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	TOO COMPLEX	TOO COMPLEX	TOO COMPLEX TO ENCODE	DOES NOT PROVIDE ANY AID IN ANALYSIS
2D ADAPTIVE TRANSFORM								
HIGH RATE 2 BITS	VARIABLE RATE, MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	VARIABLE RATE, MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	TOO COMPLEX TO ENCODE, SMALL DISTORTION	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	VARIABLE RATE MUCH MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	DOES NOT PROVIDE ANY AID IN ANALYSIS
MEDIUM RATE 1 BIT	VARIABLE RATE LARGE COMPLEXITY AND NOT SIGNIFICANTLY BETTER THAN HYBRID	VARIABLE RATE, LARGE COMPLEXITY AND NOT SIGNIFICANTLY BETTER THAN HYBRID	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	VARIABLE RATE, MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHMS	DECODING TOO COMPLEX	DOES NOT WARRANT THE EFFORT	DOES NOT PROVIDE ANY AID IN ANALYSIS
LOW RATE 0.5 BIT	VARIABLE RATE NOT SIGNIFICANTLY BETTER PERFORMANCE THAN HYBRID	WITH COSINE TRANSFORM BEST SNR	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	WITH COSINE TRANSFORM BEST SNR	DECODING TOO COMPLEX	DOES NOT WARRANT THE EFFORT	DOES NOT PROVIDE ANY AID IN ANALYSIS
ADAPTIVE HYBRID								
HIGH RATE 2 BITS	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHM DPCM	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHM DPCM	SMALL DISTORTION	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHM DPCM	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHM DPCM	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHM DPCM	MORE COMPLEX THAN COMPARABLY PERFORMING ALGORITHM DPCM	DOES NOT PROVIDE ANY AID IN ANALYSIS
MEDIUM RATE 1 BIT	BEST PERFORMANCE, FIXED RATE	TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	BEST PERFORMANCE FOR ACCEPTABLY COMPLEX ALGORITHMS ONLY USE IF DATA IS TRANSMITTED, NOT ON CCT	GOOD PERFORMANCE, ACCEPTABLE COMPLEXITY	DOES NOT WARRANT THE EFFORT	DOES NOT PROVIDE ANY AID IN ANALYSIS
LOW RATE 0.5 BIT	BETTER PERFORMANCE THAN DPCM, MUCH LESS COMPLEX THAN 2D HADAMARD OR CLUSTERING	TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	BEST PERFORMANCE FOR ACCEPTABLY COMPLEX ALGORITHMS ONLY USE IF DATA IS TRANSMITTED, NOT ON CCT	BETTER PERFORMANCE THAN DPCM, MUCH LESS COMPLEX THAN 2D HADAMARD OR CLUSTERING	DOES NOT WARRANT THE EFFORT	DOES NOT PROVIDE ANY AID IN ANALYSIS
ADAPTIVE 2D DPCM								
HIGH RATE 2 BITS	GOOD PERFORMANCE RELATIVELY SIMPLE TO IMPLEMENT	GOOD PERFORMANCE RELATIVELY SIMPLE TO IMPLEMENT	SMALL DISTORTION	GOOD PERFORMANCE RELATIVELY SIMPLE TO IMPLEMENT	GOOD PERFORMANCE RELATIVELY SIMPLE TO IMPLEMENT	GOOD PERFORMANCE RELATIVELY SIMPLE TO IMPLEMENT	NOT HIGH ENOUGH COMPRESSION TO WARRANT THE EFFORT	DOES NOT PROVIDE ANY AID IN ANALYSIS
MEDIUM RATE 1 BIT	HYBRID PERFORMS BETTER	TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	HYBRID PERFORMS BETTER WITH ACCEPTABLY INCREASED COMPLEXITY	HYBRID PERFORMS BETTER WITH ACCEPTABLY INCREASED COMPLEXITY	DOES NOT WARRANT THE EFFORT	DOES NOT PROVIDE ANY AID IN ANALYSIS
LOW RATE 0.5 BIT	PERFORMS WORSE THAN HYBRID REQUIRES SOME SPECTRAL TRANSFORM BANDS NOT TO BE TRANSMITTED	PERFORMS WORSE THAN HYBRID REQUIRES SOME SPECTRAL TRANSFORM BANDS NOT TO BE TRANSMITTED	INTRODUCES TOO MUCH DISTORTION	INTRODUCES TOO MUCH DISTORTION	HYBRID PERFORMS BETTER WITH ACCEPTABLY INCREASED COMPLEXITY	HYBRID PERFORMS BETTER	DOES NOT WARRANT THE EFFORT	DOES NOT PROVIDE ANY AID IN ANALYSIS
ADDITION OF SPECTRAL TRANSFORM	MAXIMUM INFORMATION AT FULL RESOLUTION USING 1 HAAR BAND	PERFORMANCE WORSE THAN CORRESPONDING SYSTEMS NOT UTILIZING SPECTRAL TRANSFORM FOR ADAPTIVE TECHNIQUES	TOGETHER WITH ENTROPY CODING SHOULD PROVIDE HIGHER COMPRESSION	MAKES BIT ALLOCATION TO VARIOUS BANDS ALMOST CONSTANT	MAXIMUM INFORMATION AT FULL RESOLUTION USING 1 HAAR BAND	MAXIMUM INFORMATION AT FULL RESOLUTION USING 1 HAAR BAND	DOES NOT WARRANT THE ADDITIONAL EFFORT	MAXIMUM INFORMATION AT FULL RESOLUTION USING 1 HAAR BAND

7. HARDWARE IMPLEMENTATION CONSIDERATIONS FOR THEMATIC MAPPER IMAGERY

In this section emphasis is placed on data compression techniques as applied to Thematic Mapper (TM) imagery. Basic premises are: 1) the TM is implemented with the full-up eight-band growth capability, and 2) the communications link has a fixed 15 Mbps capacity, thus requiring an average 8:1 compression ratio.

Two techniques are examined with increasing levels of flexibility and performance. The first approach is a 2D-DPCM predictive technique which can be either nonadaptive or adaptive. The 2D-DPCM approach is easily implemented and can yield excellent results if made adaptive. It suffers somewhat, however, in that it is not as flexible as may be desired. This is because the bits per pixel ratio per band is limited to integer values.

The second approach is a hybrid Hadamard-DPCM transform/predictive technique that is extremely flexible and reprogrammable, and timeshares equipment to reduce parts and power. A cosine transform can be easily substituted for the Hadamard transform with a subsequent parts and power penalty.

Total parts and power summaries for these techniques are shown in the following table.

	Integrated Circuit Count	Power (Watts)
<u>2D-DPCM</u>		
Nonadaptive	115	24.1
Adaptive	220	36.5
Reconfigurable and Adaptive	300	43.8
<u>Hadamard-DPCM</u>		
Nonadaptive	347	32.4
Adaptive	441	42.6
Reconfigurable and Adaptive	521	49.9

All of these techniques can be adopted with components available or announced in vendors' catalogs. Total parts count and power consumption are practical for spacecraft use, even for the most flexible configuration listed in the table.

The data compression hardware design presented in this section is based upon interfacing with a TM instrument concept proposed by TRW. Subsequently, our instrument lost in the competitive procurement. However, the major instrument concepts have not changed drastically and, in general, would not affect the majority of the data compression hardware design.

7.1 THEMATIC MAPPER CONCEPT

The TM is an advanced space sensor designed to obtain high resolution multispectral imagery.

The TRW proposed TM uses a multifaceted mirror rotating at a constant rate to sweep an optical field of view past eight detector arrays. Two of the detector arrays are offered as growth potential. Each sweep scans a ground swath 185 km wide and 0.96 km along-track at a rate of seven scans per second from an orbital altitude of 705 km. Each detector array is optimized for a different spectral band, ranging through the visible to the infrared. Except for band 6, each array has 32 detectors, each of which has a resolution field of view (RFOV) of 30 x 30 meters. Band 6, which is optimized for the 10.4 to 12.5 micron infrared region, has an RFOV of 120 x 120 meters with growth capability of 60 x 60 meters. Figure 7-1 illustrates the TM multiplexer design. Each color band is treated separately from the detector to the final multiplexer on the right side of the figure. The optics are such that the detector RFOVs form an along-track column or footprint, which is swept in a cross-track direction by the scanning mirror (Figure 7-2).

Detectors within each array simultaneously generate analog outputs for each RFOV advance of the scan mirror. Each output is sampled and digitized to the 8 bit level before being multiplexed with spacecraft telemetry for transmission to a ground terminal.

Key characteristics of TM imagery and output data rates are tabulated in Figure 7-2. Because of the swath area being mapped, and the excellent spatial and radiometric resolution, the total output data rate is 118.24 Mbps for the full-up 8 band system, or 88.68 Mbps with only six bands.

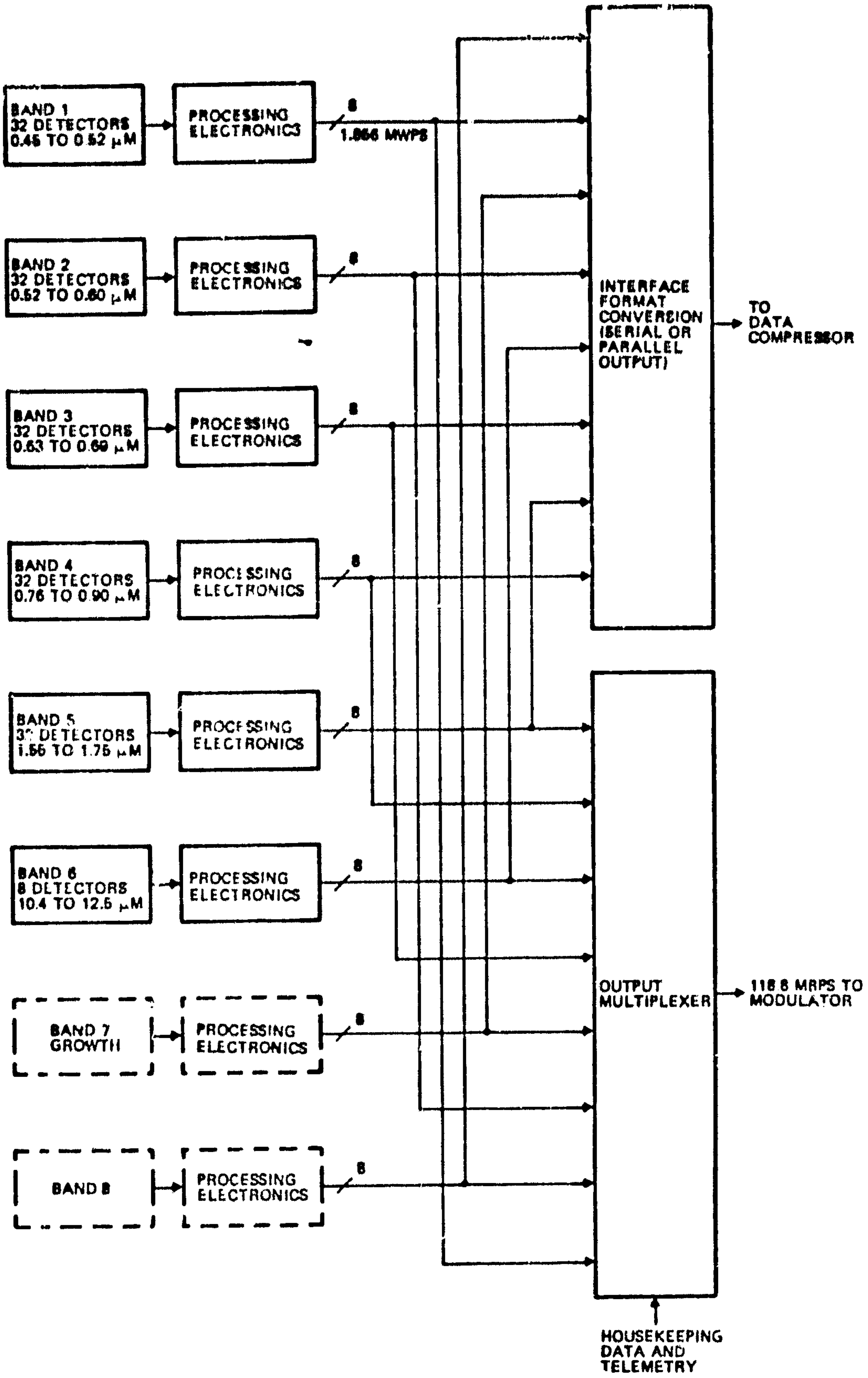
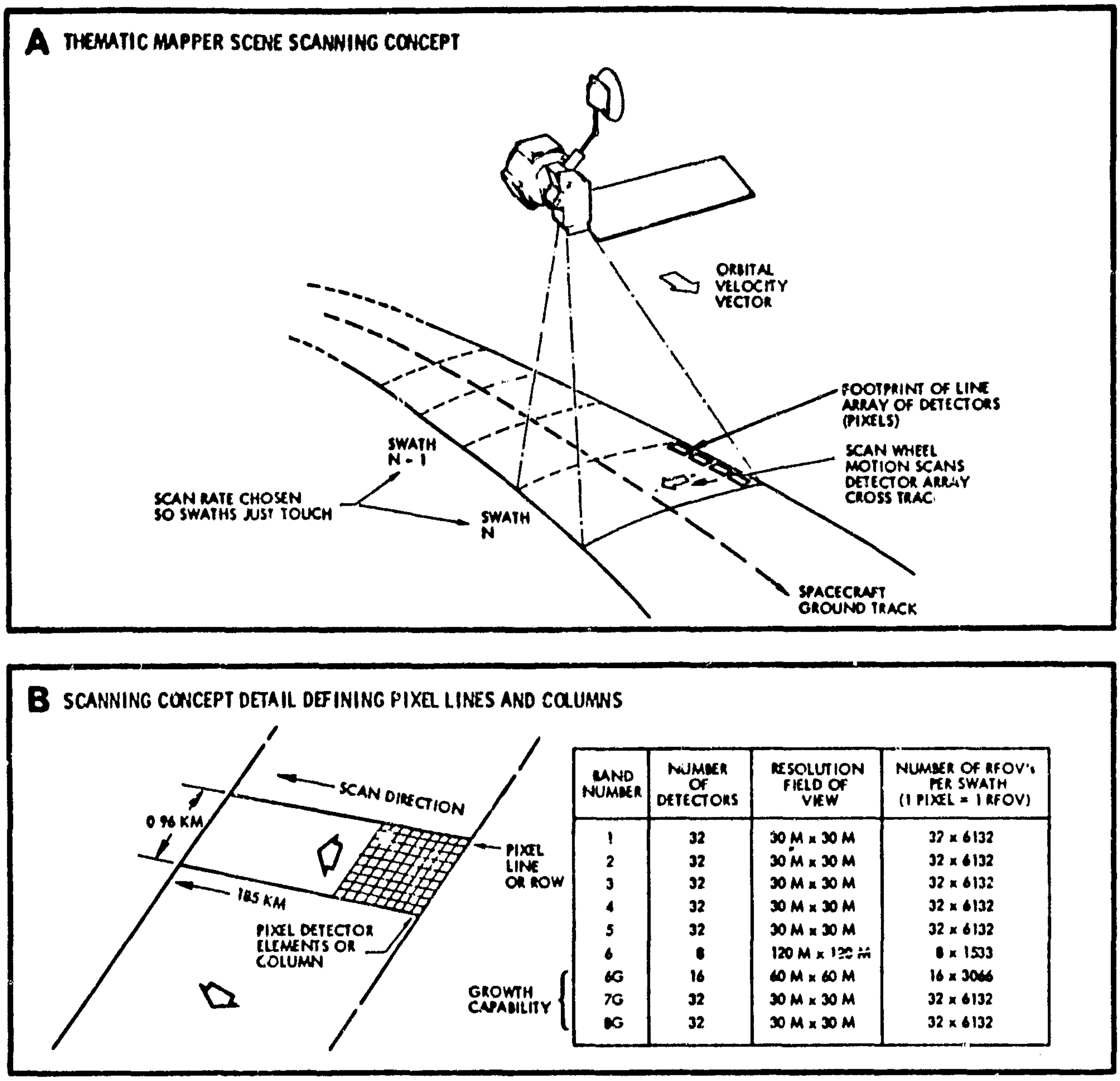


Figure 7-1. Thematic Mapper Signal Processing Concept


A THEMATIC MAPPER SCENE SCANNING CONCEPT
ORBITAL VELOCITY VECTOR
FOOTPRINT OF LINE ARRAY OF DETECTORS (PIXELS)
SCAN WHEEL MOTION SCANS DETECTOR ARRAY CROSS TRAC
SWATH N - 1
SCAN RATE CHOSEN SO SWATHS JUST TOUCH
SWATH N
SPACECRAFT GROUND TRACK
B SCANNING CONCEPT DETAIL DEFINING PIXEL LINES AND COLUMNS
SCAN DIRECTION
0 96 KM
185 KM
PIXEL LINE OR ROW
PIXEL DETECTOR ELEMENTS OR COLUMN
GROWTH CAPABILITY

BAND NUMBER	NUMBER OF DETECTORS	RESOLUTION FIELD OF VIEW	NUMBER OF RFOV's PER SWATH (1 PIXEL = 1 RFOV)
1	32	30 M x 30 M	32 x 6132
2	32	30 M x 30 M	32 x 6132
3	32	30 M x 30 M	32 x 6132
4	32	30 M x 30 M	32 x 6132
5	32	30 M x 30 M	32 x 6132
6	8	120 M x 120 M	8 x 1533
6G	16	60 M x 60 M	16 x 3066
7G	32	30 M x 30 M	32 x 6132
8G	32	30 M x 30 M	32 x 6132

C KEY THEMATIC MAPPER IMAGERY PARAMETERS

	ACCURATE	APPROXIMATE
ALTITUDE	705.3 KM	705 KM
OUTPUT DATA RATE (SERIAL)	118.24751 MBPS	118 MBPS
OUTPUT WORD RATE (PARALLEL)	14 780939 MWPS	15 MWPS
COLOR BAND WORD RATE	1.8476174 MWPS	1.85 MWPS
A/D CONVERTER SAMPLE RATE	3.695238 MSPS	3.70 MSPS
DETECTOR SAMPLE INTERVAL	17.319603 μSEC	17.32 μSEC
DETECTOR EXPOSURE TIME	12.989702 μSEC	13 μSEC
DETECTOR CLOCK RATE	7.3904696 MHz	7.4 MHz

	ACCURATE	APPROXIMATE
SCAN RATE	7.0356152 STRIPE/SEC	7 STRIPE/SEC
SCAN TOTAL	142.13398 MSEC	140 MSEC
SCAN TIME ACTIVE	106.19810 MSEC	105 MSEC
SAMPLES PER SCAN ACTIVE	6131.671 SAMPLES	6100 SAMPLES
SAMPLES PER SCAN TOTAL	8206.538 SAMPLES	8200 SAMPLES
SCAN EFFICIENCY	74.716901 PERCENT	75 PERCENT
EARTH TRACK	6.7541905 KM/SEC	6.75 KM/SEC
A/D CLOCK RATE	47.299204 MHz	47 3 MHz

Figure 7-2. Thematic Mapper Scan Characteristics

7.2 SELECTION OF CANDIDATE DATA COMPRESSION APPROACHES

The high output rate from the TM instrument (88 to 120 Mbps) must be compressed by the spacecraft to <15.062 Mbps for compatibility with S-band ground stations. Therefore, the on-board data compression system must provide a maximum data compression ratio of 8:1, which depends upon the number of spectral bands implemented in the TM instrument or the number of bands actually selected for compression.

From the results of our study, we have narrowed the set of applicable TM compression algorithms to three:

1) Adaptive hybrid Hadamard-DPCM

2) Adaptive hybrid Cosine-DPCM

3) Adaptive spatial 2D-DPCM.

A spectral transformation preceding the spatial encoder [illegible] rejected due to its complexity and small, if any, performance improvement when used in conjunction with the three selected algorithms. Implementation complexity was due to corresponding pixels in differing spectral bands being generated at different times. In fact, the spatial separation between bands was as much as 146 pixels in the TRW-proposed design.

We recommend using 2D-DPCM on each spectral band at 2 bits per pixel or greater, and a hybrid approach on each spectral band at 1 bit per pixel.

7.3 DATA COMPRESSOR DESIGN CONSIDERATIONS

The goal of a data compressor designer is to create a system which provides good performance, is flexible, and minimizes parts and power consumption. These requirements are somewhat at odds with each other, but we have created such a design through emphasis on adaptivity, reprogrammability, and timesharing.

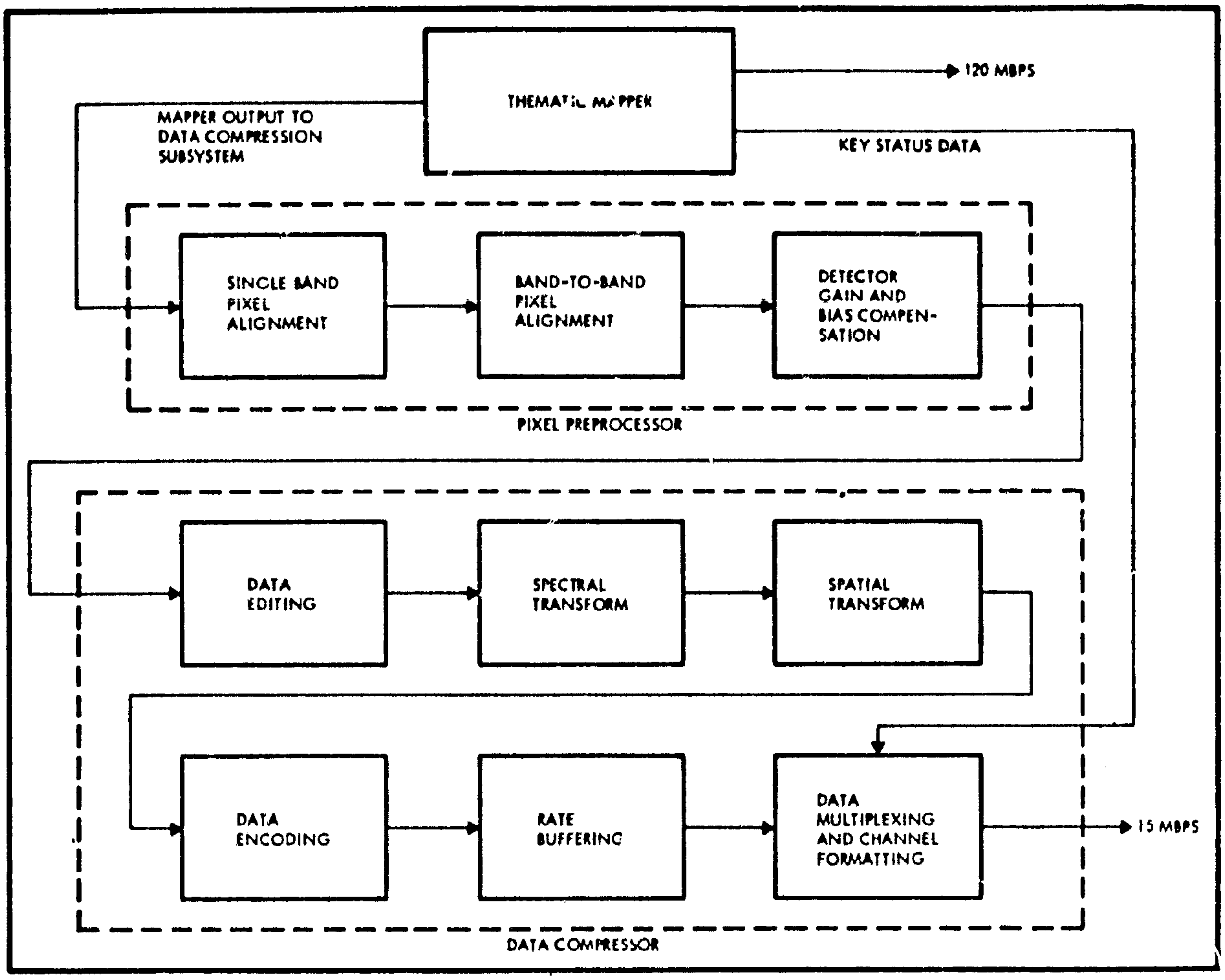
This section reviews the preliminary considerations appropriate to the hardware design of a data compressor, including:

- Interfaces between the TM and the data compressor
- Reviewing candidate approaches from a sizing and complexity viewpoint

- Describing the elements of the data compressor which make the system adaptive

- Reviewing timesharing and reprogramming consideration.

Figure 7-3 shows a generic data compression system, consisting of concatenated elements. Since it has been shown that spectral encoding offers little help in reducing the data rate, we have elected to omit spectral encoding.



THEMATIC MAPPER
120 MBPS
MAPPER OUTPUT TO DATA COMPRESSION SUBSYSTEM
KEY STATUS DATA
SINGLE BAND PIXEL ALIGNMENT
BAND-TO-BAND PIXEL ALIGNMENT
DETECTOR GAIN AND BIAS COMPEN-SATION
PIXEL PREPROCESSOR
DATA EDITING
SPECTRAL TRANSFORM
SPATIAL TRANSFORM
DATA ENCODING
RATE BUFFERING
DATA MULTIPLEXING AND CHANNEL FORMATTING
15 MBPS
DATA COMPRESSOR

Figure 7-3. On Board Data Compaction Processing Flow

Both 2D-DPCM and hybrid spatial encoding techniques are described, along with the problems of allocating and assigning the bits available for a total transmission rate of 15 Mbps. It is shown that it is useful to group detectors in groups of 16, corresponding to the top and bottom halves of the seven 32-detector bands, and to the complete 16 detectors of band 6. These 15 groups can then be accommodated by timesharing a single 16-point transform element.

7.3.1 TM Interfaces

Figure 7-1 illustrates the TM multiplexer design. Each color band is treated separately from the detector to the final multiplexer on the right side of the figure. The final multiplexer has an interface which provides band-separated 8 bit parallel word outputs for the data compressor. Also shown in Figure 7-1 is a parallel-to-serial interface format converter for each band to reduce the number of wires in the interconnect cable to the data compressor. With the parallel-to-serial converter, the interface consists of clock and word sync lines plus eight lines carrying serial image data at 1.856 Mwps, except for band 6 which generates 0.926 Mwps for the 16-detector growth configuration.

The format of the data received at the data compressor is indicated in Figure 7-4a. The data is received out of detector sequence, primarily because of the detector chip design which places odd and even detectors in adjacent interleaved rows. A pixel alignment circuit, illustrated in Figure 7-4b, is used to resequence the words to appear in numerical order.

7.3.2 The Bit Allocation Problem for Fixed Channel Rates

Figure 7-5a indicates TM source data rates in each of the eight spectral bands, and the associated telemetry and synchronization data. Seven of the eight bands have 32 detectors. The eighth band (band 6) has eight detectors in the basic design, and 16 in the growth version. For the balance of this discussion, band 6 will be assumed to have 16 detectors, as in the growth configuration. With 16 detectors, band 6 has 1/4 the resolution of the remaining bands, instead of 1/16 as in the basic system; accordingly, it generates data at a rate that is 1/4 the rate of the other bands.

It is convenient to partition the 32-detector bands into two 16-detector groups, as illustrated in Figure 7-5b. Some compression techniques work very well with data from 32 parallel sources, for example 2D-DPCM.

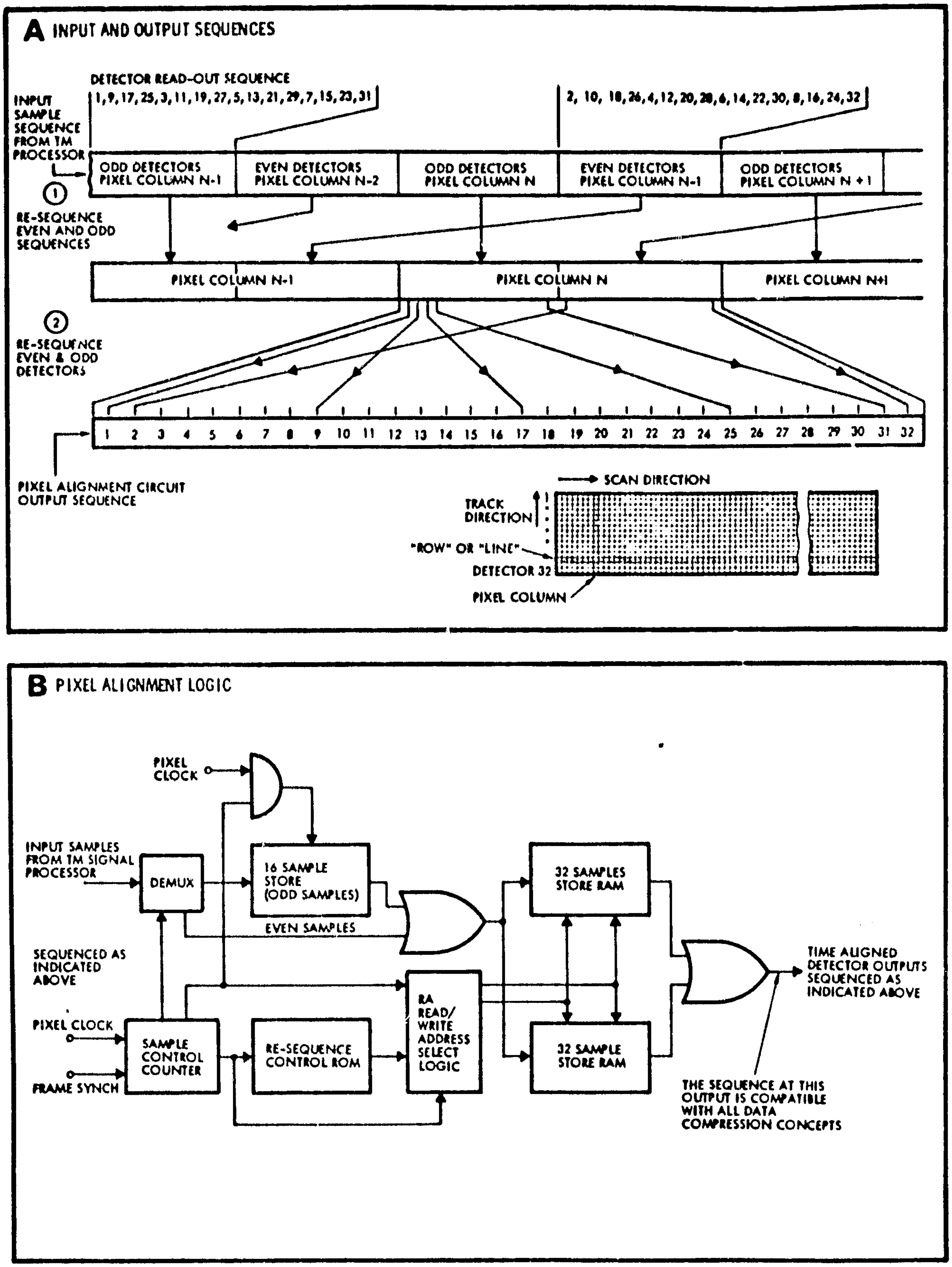

A INPUT AND OUTPUT SEQUENCES
DETECTOR READ-OUT SEQUENCE
INPUT SAMPLE SEQUENCE FROM TM PROCESSOR
1,9,17,25,3,11,19,27,5,13,21,29,7,15,23,31
2, 10, 18,26,4,12,20,28,6,14,22,30,8,16,24,32
ODD DETECTORS PIXEL COLUMN N-1
EVEN DETECTORS PIXEL COLUMN N-2
ODD DETECTORS PIXEL COLUMN N
EVEN DETECTORS PIXEL COLUMN N-1
ODD DETECTORS PIXEL COLUMN N + 1
1
RE-SEQUENCE EVEN AND ODD SEQUENCES
PIXEL COLUMN N-1
PIXEL COLUMN N
PIXEL COLUMN N+1
2
RE-SEQUENCE EVEN & ODD DETECTORS
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
PIXEL ALIGNMENT CIRCUIT OUTPUT SEQUENCE
SCAN DIRECTION
TRACK DIRECTION
"ROW" OR "LINE"
DETECTOR 32
PIXEL COLUMN
B PIXEL ALIGNMENT LOGIC
PIXEL CLOCK
INPUT SAMPLES FROM TM SIGNAL PROCESSOR
DEMUX
16 SAMPLE STORE (ODD SAMPLES)
EVEN SAMPLES
32 SAMPLES STORE RAM
SEQUENCED AS INDICATED ABOVE
PIXEL CLOCK
FRAME SYNCH
SAMPLE CONTROL COUNTER
RE-SEQUENCE CONTROL ROM
RA READ/ WRITE ADDRESS SELECT LOGIC
32 SAMPLE STORE RAM
TIME ALIGNED DETECTOR OUTPUTS SEQUENCED AS INDICATED ABOVE
THE SEQUENCE AT THIS OUTPUT IS COMPATIBLE WITH ALL DATA COMPRESSION CONCEPTS

Figure 7-4. Pixel Alignment Logic Sequences Pixels to be Compatible With all Data Compression Concepts

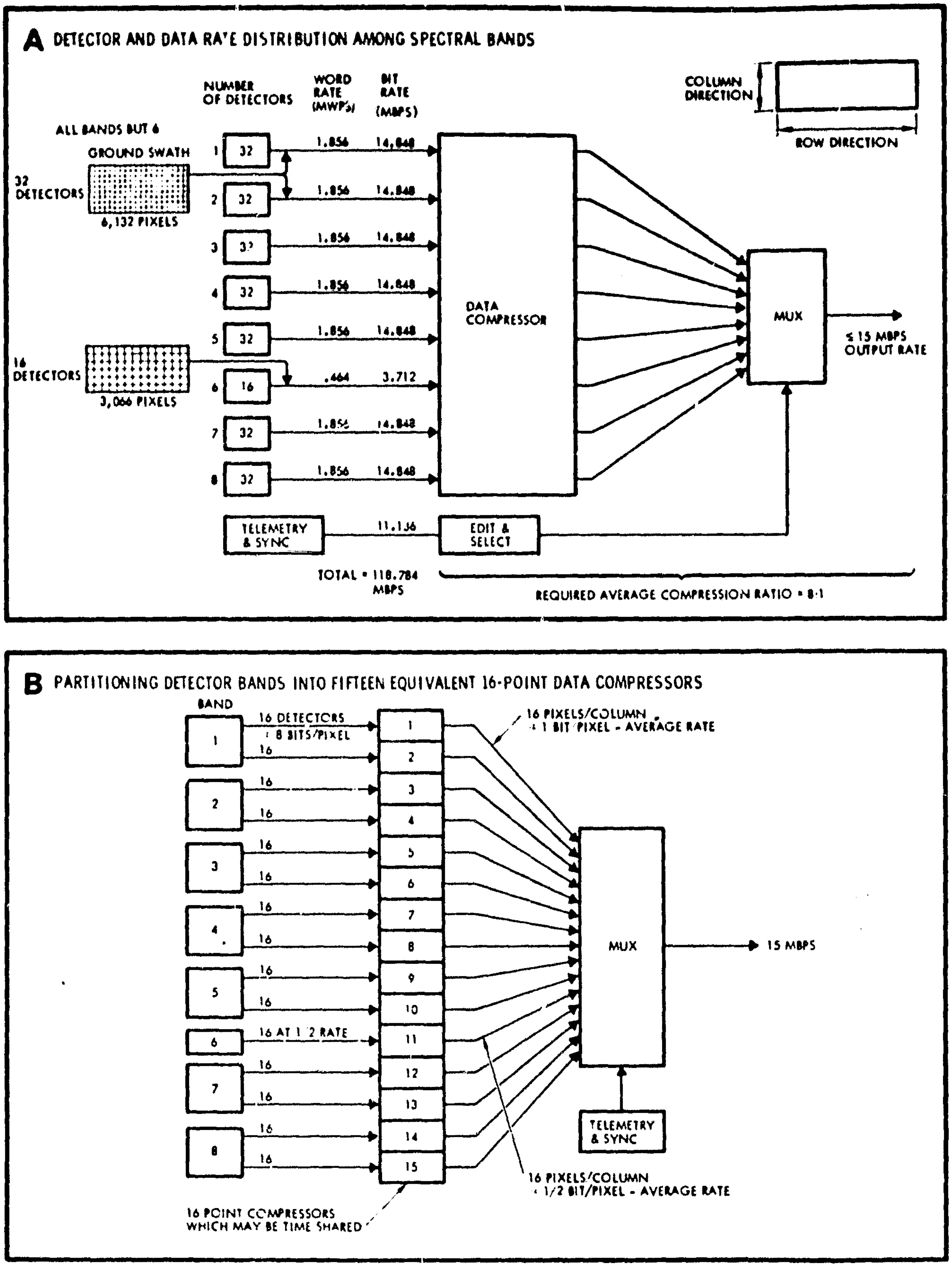


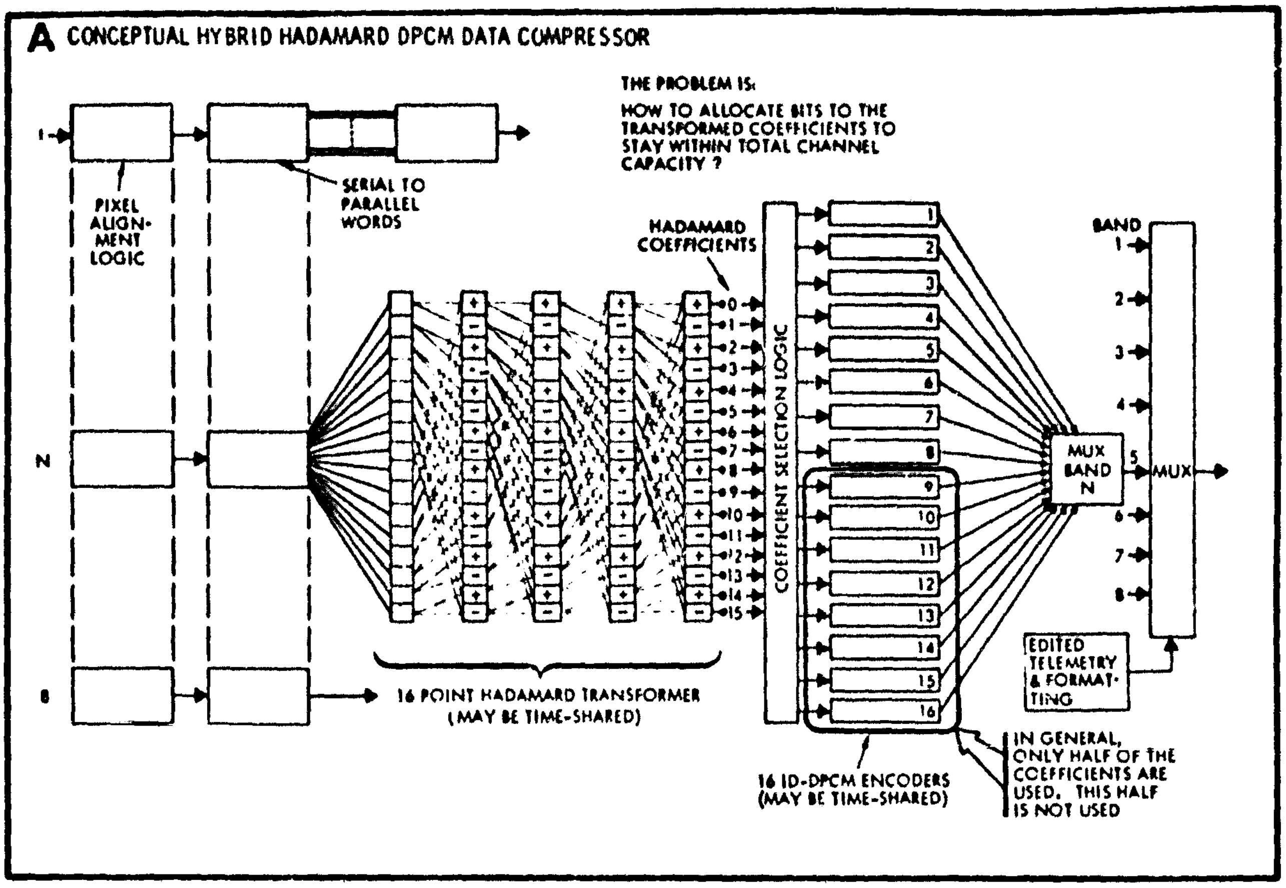
Figure 7-5. Assignment of Detector Bands and Data Compressors

The hybrid Hadamard-DPCM approach also works very well with 32-point inputs, but the Hadamard transform requires four times as much power and parts. Since the Hadamard-DPCM approach is more flexible, it is convenient to adopt a basic building block sized for 16-detector channels. This grouping yields a total of fifteen 16-point compressors with an average input rate of 8 bits per pixel and an output rate of 1 bit per pixel for an 8 band system. Band 6 is the one exception, since it receives one set of data for every two sets received by the other bands due to the different resolution.

Conceptually, the data compressor consists of fifteen 16-point compressors, or a reduced number of compressors which are timeshared. Each compressor receives data in pipeline form from 16 detectors at 8 bits per detector and outputs 16 bits that represent the imagery from all 16 detectors. Conceptually, there need not be a one-for-one relationship between the 16 output bits and the 16 input detectors, as long as the required compression ratio is achieved. Due to the character of the algorithm, 2D-DPCM encoders are limited in the assignment of encoded bits per pixel, such that each pixel must be allocated 1, 2 or more bits. It is not easy to obtain fractional bits for DPCM systems. This necessarily requires eliminating some of the bands before encoding.

Figure 7-6a illustrates a Hadamard-DPCM compressor with a 16-point Hadamard pipeline transformer, followed by 1D-DPCM encoders. The output of the transformer represents 16 coefficients in the transform domain, with the magnitude of each coefficient related to the frequency content of a single column of input pixels. In general, the lower frequency components are larger and have more information content, whereas the higher frequency components are often very small and carry little information, except for very busy scenes such as a city.

Because of the varying coefficient amplitudes, it is appropriate to allocate the available 16 bits in a manner which assigns more bits to the larger coefficients and fewer to the smaller coefficients. Figure 7-6b illustrates several possible bit assignment patterns. Four cases are illustrated in which 16, 18, 20, or 24 bits are allocated to the 16 coefficients. Considering the 16 bit case first, it is evident that there is a large variety of possible bit assignment patterns, with most of them having some


A CONCEPTUAL HYBRID HADAMARD DPCM DATA COMPRESSOR
THE PROBLEM IS:
HOW TO ALLOCATE BITS TO THE TRANSFORMED COEFFICIENTS TO STAY WITHIN TOTAL CHANNEL CAPACITY ?
1
N
8
PIXEL ALIGN-MENT LOGIC
SERIAL TO PARALLEL WORDS
HADAMARD COEFFICIENTS
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
COEFFICIENT SELECTION LOGIC
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
16 POINT HADAMARD TRANSFORMER (MAY BE TIME-SHARED)
16 ID-DPCM ENCODERS (MAY BE TIME-SHARED)
MUX BAND N
BAND
1
2
3
4
5
6
7
8
MUX
EDITED TELEMETRY & FORMAT-TING
IN GENERAL, ONLY HALF OF THE COEFFICIENTS ARE USED. THIS HALF IS NOT USED

B REPRESENTATIVE BIT ALLOCATION PATTERNS

AVERAGE SIT RATE	BITS PER 16 COEFFICIENTS	COEFFICIENT NUMBER															
		1	2	3	4	5	6	7	8	9	10	1'	12	13	14	15	16
1.0	16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
		3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
		3	2	1	1	1	1	1	1	1	1	1	1	1	0	0	0
		4	3	2	2	2	1	1	1	0	0	0	0	0	0	0	0
		5	4	3	2	1	1	0	0	0	0	0	0	0	0	0	0
1.125	18	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		4	3	1	1	1	1	1	1	1	1	1	1	1	0	0	0
		5	4	3	2	1	1	1	1	0	0	0	0	0	0	0	0
		6	4	3	2	1	1	1	0	0	0	0	0	0	0	0	0
1.25	20	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
		5	3	2	1	1	1	1	1	1	1	1	1	1	0	0	0
		7	5	3	2	1	1	1	0	0	0	0	0	0	0	0	0
1.5	24	4	3	2	2	2	1	1	1	1	1	1	1	1	1	1	1
		5	4	3	2	1	1	1	1	1	1	1	1	1	1	0	0
		6	5	4	3	2	1	1	1	1	0	0	0	0	0	0	0
		7	6	5	3	1	1	1	0	0	0	0	0	0	0	0	0

SOME OF THE POSSIBLE BIT ALLOCATION PATTERNS WHICH YIELD AN AVERAGE OF 1 BIT/PIXEL

THESE ALLOCATIONS AFFECT THE NUMBER OF QUANTIZING LEVELS IN EACH DPCM ENCODER. SPECIFIC LEVEL "CUT POINTS" MUST ALSO BE DETERMINED

Figure 7-6. Allocation of Available Bits to Hadamard Coefficients

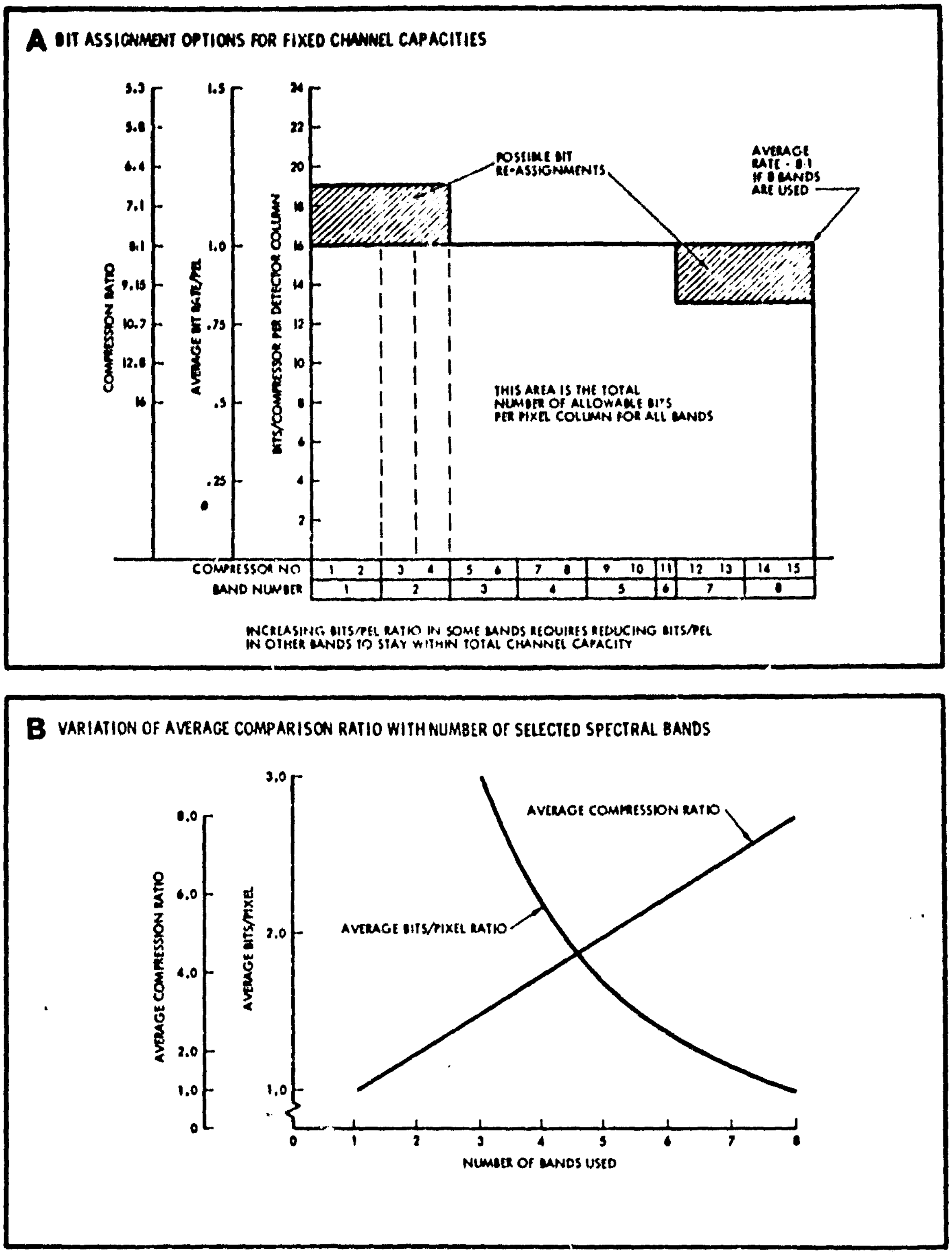
coefficients that are allocated "zero" bits. Experience has shown that on the average, the best reproduced imagery is obtained for patterns such as the 4, 3, 2, 2, 2, 1, 1, 1, 0,... pattern in which half of the coefficients receive zero bits. This permits discarding approximately half of the Hadamard coefficients, while obtaining improved fidelity with the remaining coefficients.

In the more general case, it is not necessary to assign 16 bits per channel, as long as the overall channel average is 16. Thus, it is a simple matter to allocate 18, 20, or more bits to the 16 coefficients, and to use fewer than 16 bits in other cases. It is therefore fairly easy to obtain a noninteger average bit per pixel rate for a Hadamard-DPCM encoder.

The preceding example is based on an average 8:1 compression ratio. If only six bands are implemented in the mapper, then only a 6:1 ratio is required, which makes 4/3 x 16 ∿ 21 bits available to each coefficient set.

Figure 7-7 addresses the total number of bits from all 15 16-point compressors for each new pixel in the scan direction. Part A is a histogram which indicates the number of bits assigned to each of the 15 compressors. As presented, the area under the histogram envelope is the total output bit rate per each column of pixels in the scan direction. For an average 8:1 compression ratio, the area is 232 bits.

In the event there is more interest in certain detector bands, they may be allocated more bits at the expense of a less interesting band, as long as the area is constant. In general, this increases the bits per pixel ratio and reduces the compression ratio in the bands of interest. This process is quite straightforward for the hybrid encoders, with a large variety of options available. Conversely, with 2D-DPCM, only integer bit per pixel ratios are allowed, and it is not possible to improve fidelity in some bands without jumping from 1 bit per pel to 2 bits per pel. This steals bits from some other band to the point where it is completely eliminated. Of course, if some bands are of no interest, with either approach it is possible to eliminate them and to redistribute their bits among the remaining bands, as illustrated in Figure 7-7b.



A BIT ASSIGNMENT OPTIONS FOR FIXED CHANNEL CAPACITIES

INCREASING BITS/PEL RATIO IN SOME BANDS REQUIRES REDUCING BITS/PEL IN OTHER BANDS TO STAY WITHIN TOTAL CHANNEL CAPACITY

B VARIATION OF AVERAGE COMPARISON RATIO WITH NUMBER OF SELECTED SPECTRAL BANDS

Figure 7-7. Options for Distributing Available Bits Among Bands

7.4 ADAPTIVE HYBRID COMPRESSOR

Figure 7-8 shows the functional block diagram of the reconfigurable adaptive Hadamard-DPCM data compressor system. Figure 7-9 shows a functional logic diagram of this system. This system performs a Hadamard transform on pixel column data, followed by one-dimensional DPCM compression on transform coefficients of adjacent pixel columns.

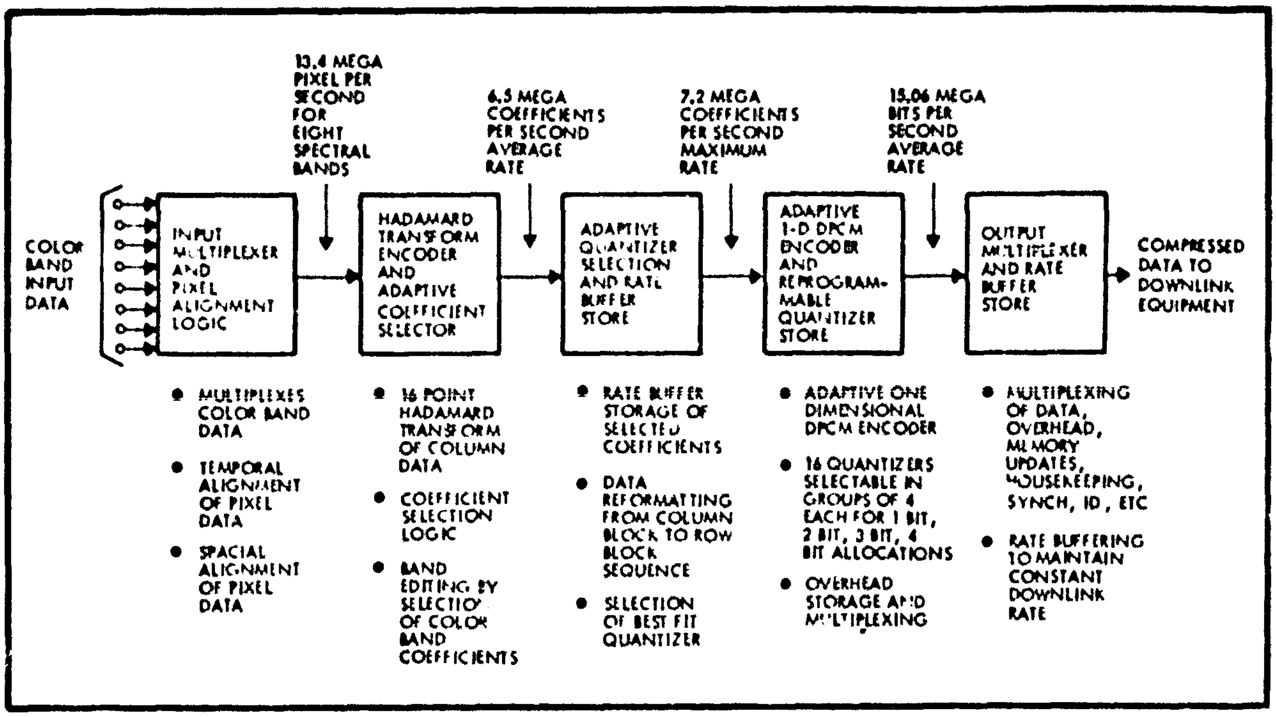

13.4 MEGA PIXEL PER SECOND FOR EIGHT SPECTRAL BANDS
6.5 MEGA COEFFICIENTS PER SECOND AVERAGE RATE
7.2 MEGA COEFFICIENTS PER SECOND MAXIMUM RATE
15.06 MEGA BITS PER SECOND AVERAGE RATE
COLOR BAND INPUT DATA
INPUT MULTIPLEXER AND PIXEL ALIGNMENT LOGIC
HADAMARD TRANSFORM ENCODER AND ADAPTIVE COEFFICIENT SELECTOR
ADAPTIVE QUANTIZER SELECTION AND RATE BUFFER STORE
ADAPTIVE 1-D DPCM ENCODER AND REPROGRAM-MABLE QUANTIZER STORE
OUTPUT MULTIPLEXER AND RATE BUFFER STORE
COMPRESSED DATA TO DOWNLINK EQUIPMENT
• MULTIPLEXES COLOR BAND DATA
• TEMPORAL ALIGNMENT OF PIXEL DATA
• SPACIAL ALIGNMENT OF PIXEL DATA
• 16 POINT HADAMARD TRANSFORM OF COLUMN DATA
• COEFFICIENT SELECTION LOGIC
• BAND EDITING BY SELECTION OF COLOR BAND COEFFICIENTS
• RATE BUFFER STORAGE OF SELECTED COEFFICIENTS
• DATA REFORMATTING FROM COLUMN BLOCK TO ROW BLOCK SEQUENCE
• SELECTION OF BEST FIT QUANTIZER
• ADAPTIVE ONE DIMENSIONAL DPCM ENCODER
• 16 QUANTIZERS SELECTABLE IN GROUPS OF 4 EACH FOR 1 BIT, 2 BIT, 3 BIT, 4 BIT ALLOCATIONS
• OVERHEAD STORAGE AND MULTIPLEXING
• MULTIPLEXING OF DATA, OVERHEAD, MEMORY UPDATES, HOUSEKEEPING, SYNCH, ID, ETC
• RATE BUFFERING TO MAINTAIN CONSTANT DOWNLINK RATE

Figure 7-8. Adaptive Hadamard-DPCM Data Compressor Functional Block Diagram

Data compression is obtained with the Hadamard transform encoder (HTE) by eliminating some of the higher order coefficients at the maximum compression ratio of 8:1. The remaining compression is obtained from the DPCM encoder. The adaptive Hadamard-DPCM compressor accepts a total of 15 million 8 bit words per second from all of the color and processors, outputs 15 Mbps, and achieves a maximum compression ratio of 8:1.

Flexibility is provided by uplink programming of the number of bands selected for processing, the number of HTE coefficients processed, and the selection and content of the DPCM quantization patterns. A feature of this system is the automatic selection of the "best fit" available quantizer on a block-by-block basis.

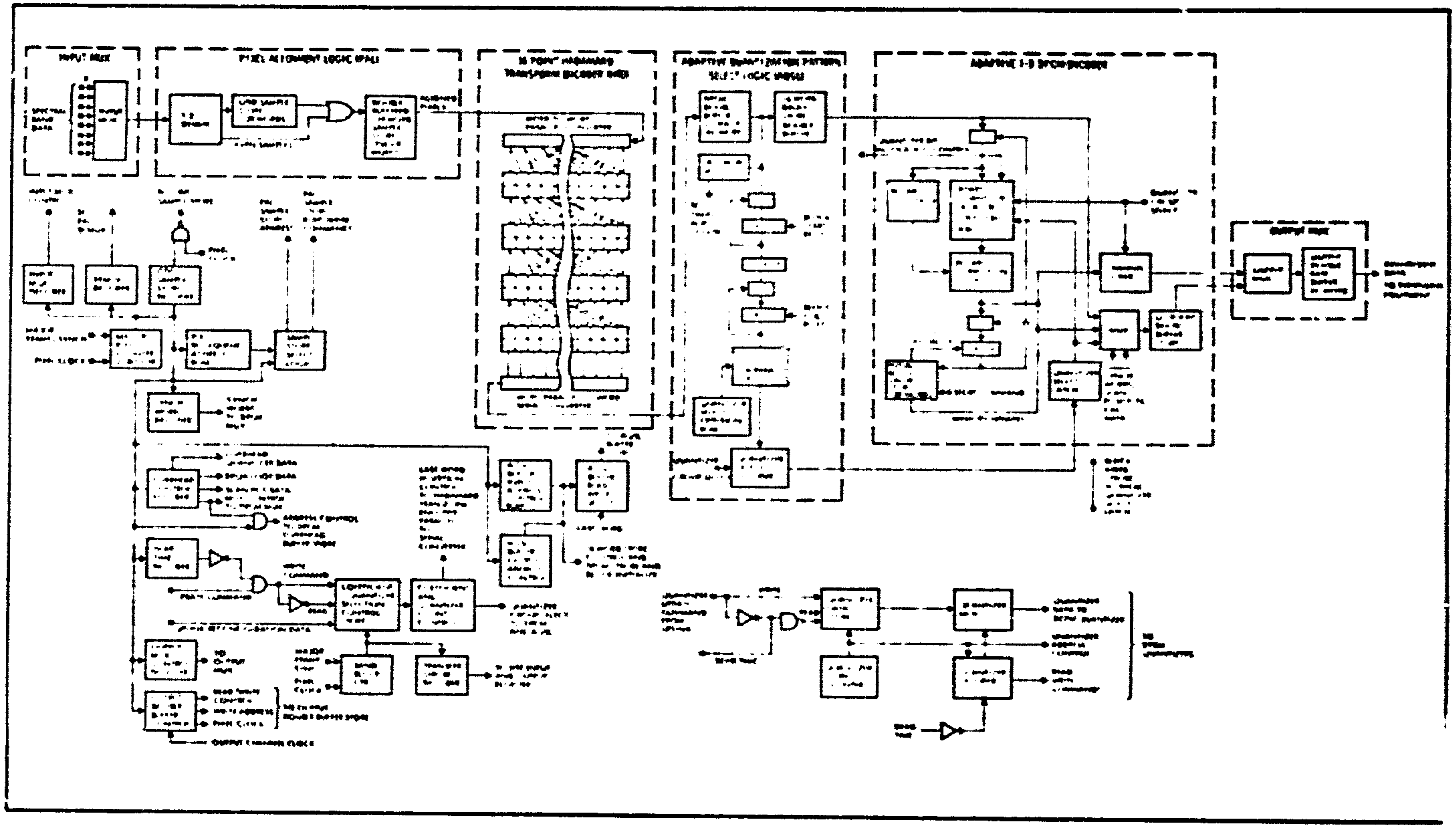

INPUT MUX
SPECTRAL BAND DATA
PIXEL ALIGNMENT LOGIC (PAL)
ALIGNED PIXELS
16 POINT HADAMARD TRANSFORM ENCODER (HED)
ADAPTIVE QUANTIZATION PATTERN SELECT LOGIC (AQPSL)
ADAPTIVE 1-D DPCM ENCODER
OUTPUT MUX

Figure 7-9. Adaptive Hybrid Data Compressor Functional Logic Diagram

7.4.1 Hybrid Compressor Functional Blocks

This system is comprised of the following blocks, each of which is discussed below:

- Input multiplexer and pixel alignment logic
- Hadamard transform encoder and adaptive coefficient selector
- Adaptive quantizer selector and rate buffer
- Adaptive 1D-DPCM encoder and reprogrammable quantizer
- Output multiplexer and rate buffer.

7.4.1.1 Input Multiplexer and Pixel Alignment Logic

The input multiplexer serves to sequentially sample each of the eight TM color band outputs and to reformat this data into a single bit parallel, word serial data stream. The input multiplexer output data rate is approximately 15 Mwps. The multiplexer samples all bands, whether the data is to be used or not. Band 6 is sampled at the same rate as all other bands The timing and control logic, in conjunction with the HTE coefficient selection logic, removes the redundancy and unwanted bands. This technique permits band selection with minimal multiplexer complexity.

As previously discussed in Section 7.3, the pixel data is received from each color band processor in a spatially and temporally skewed format. Figure 7-10a illustrates the input format for all eight bands. The action of the input multiplexer further mixes the data by alternately sequencing pixels from each color band. The function of the pixel alignment logic (PAL) is to separate the data stream into detector sequential order on a per-band-column-oriented basis, as illustrated in Figure 7-10b.

A detailed description of the operation of this logic is presented in Section 7.4.2 for the timeshared version of the PAL which must operate on all eight color bands.

7.4.1.2 Hadamard Transform Encoder and Adaptive Coefficient Selector

The Hadamard Transform Encoder parallel processes 16 adjacent pixels in a pipeline fashion. Word serial pixel data is shifted into the HTE from the PAL. Logic located at the input to the HTE converts the word serial data to a word parallel data format. Thus, the data progresses

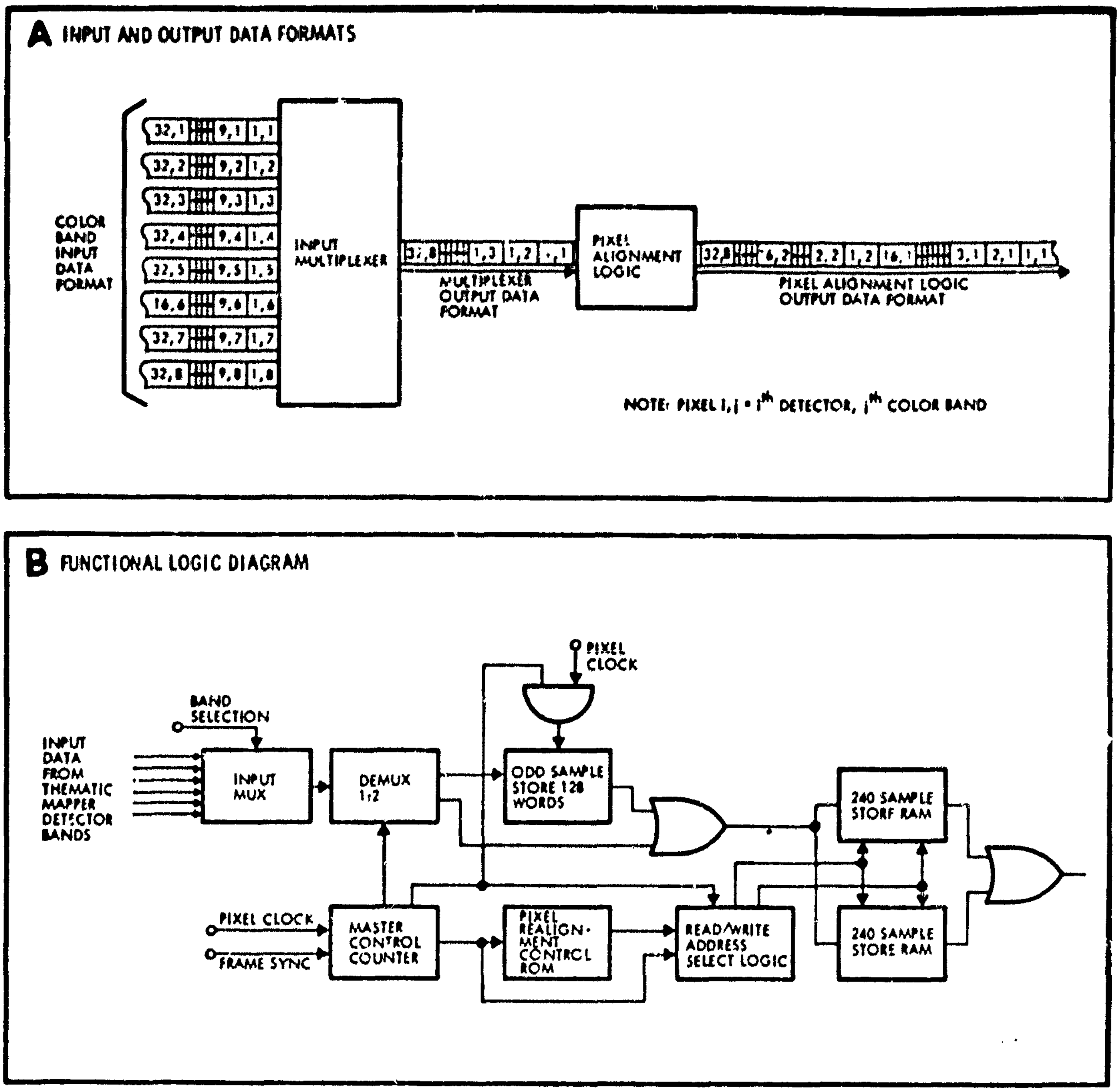

A INPUT AND OUTPUT DATA FORMATS
COLOR BAND INPUT DATA FORMAT
32,1 9,1 1,1
32,2 9,2 1,2
32,3 9,3 1,3
32,4 9,4 1,4
32,5 9,5 1,5
16,6 9,6 1,6
32,7 9,7 1,7
32,8 9,8 1,8
INPUT MULTIPLEXER
32,8 1,3 1,2 1,1
MULTIPLEXER OUTPUT DATA FORMAT
PIXEL ALIGNMENT LOGIC
32,8 16,2 2,2 1,2 16,1 3,1 2,1 1,1
PIXEL ALIGNMENT LOGIC OUTPUT DATA FORMAT
NOTE: PIXEL I,J = Ith DETECTOR, Jth COLOR BAND
B FUNCTIONAL LOGIC DIAGRAM
INPUT DATA FROM THEMATIC MAPPER DETECTOR BANDS
BAND SELECTION
INPUT MUX
DEMUX 1:2
PIXEL CLOCK
ODD SAMPLE STORE 128 WORDS
240 SAMPLE STORF RAM
240 SAMPLE STORE RAM
PIXEL CLOCK
FRAME SYNC
MASTER CONTROL COUNTER
PIXEL REALIGNMENT CONTRCL ROM
READ/WRITE ADDRESS SELECT LOGIC

Figure 7-10. Inpuι Multiplexer and Pixel Alignment Logic

through the HTE at 1/16 (for a 16-point HTE) of the input 15 Mwps rate. At the output, word-parallel-to-word-serial conversion logic is provided. It is at this point that coefficient selection is performed in accordance with uplink intormation stored within the timing and control logic. If none of the HTE output coefficients are selected for further processing, that band is effectively eliminated.

7.4.1.3 Adaptive Quantizer Selector and Rate Buffer

Three functions are performed by the Adaptive Quantizer Selection Logic and Rate Buffer (AQSL) element. These are:

- Column to row format conversion
- Rate buffering of the transform coefficients
- Selection of the best-fit DPCM quantization pattern for each pixel row data block.

The coefficient data output from the HTE is formatted in the same manner as the original detector pixel data; that is, in column sequence. Compression within the HTE takes place in this column/vertical dimension. The DPCM encoder, on the other hand, compresses in the other (row/horizontal) dimension. The column-to-row conversion function is performed within the AQSL rate buffer memory by the manner in which it is written into, and read from, this memory. The operation of the AQSL is discussed in Section 7.4.4.1.

As previously mentioned, not all of the HTE output coefficients are necessarily processed. Indeed, a color band may be completely eliminated by not processing any of its coefficients. This coefficient elimination process is performed within the rate buffer memory under ground-commandable program control. However, the elimination of coefficients leaves holes in the data stream. It is also the function of the rate buffer memory to smooth the data flow to eliminate these holes.

A block adaptive DPCM encoder follows the HTE. The selected block size for the 1D-DPCM configuration is 16 pixels long. The AQSL contains logic which measures the block's data activity and selects the quantizer with the best quantization pattern for that data block. To accommodate the time necessary for this decision process, the output data block (from the buffer memory) is delayed within the AQSL for one additional block time.

7.4.1.4 Adaptive 1D-DPCM Encoder and Reprogrammable Quantizer Store

The adaptive 1D-DPCM encoder receives two inputs from the AQSL, the input row block coefficient data, and a code word indicating the quantizer to be employed for processing the block of data.

The DPCM encoder loop is capable of processing 8 Msps; in this application it operates at approximately 7.5 Msps.

Contained within the DPCM encoder are 16 quantizers with unique, reprogrammable quantization patterns. The quantizer may be programmed, via uplink command, during the active scan time, but the actual update will not take place until the dead time at the end of the scan. Buffer storage for this purpose is contained within the timing and control logic. The quantizers are organized into four groups of four each. One group outputs 1 bit per pixel, the next group 2 bits per pixel, the third group 3 bits per pixel, and the fourth group 4 bits per pixel. Within each group are four quantizers, each containing different cut point data. The quantizer group to be used in processing a particular coefficient is based on bit allocations that may be operator-selected and reprogrammed via the command link. The AQSL selects a quantizer cut point set from within the group.

The DPCM outputs the compressed pixel data, overhead data (the particular quantizer selected for each block), memory updates (the actual value of the predictor memory), and certain housekeeping data (sync words, scan position, calibration data, etc). This information is sent to the output multiplexer and rate buffer.

7.4.1.5 Output Multiplexer and Rate Buffer Store

The output multiplexer combines the output from the DPCM encoder with housekeeping data into a bit serial output data stream for transmission. Rate buffering is provided to smooth the data gathered during the active scan time throughout the full swath period. Additionally, drivers are provided to buffer the data compressor electrically to the downlink transmission equipment.

7.4.2 Input Multiplexer and Pixel Alignment Logic

As described in the previous section and illustrated in Figure 7-10a, the color band data is input to the data compression system in a misaligned sequence. The input multiplexer and PAL are required to temporally and spatially align the column pixel data on a color band basis.

Figure 7-10b illustrates the functional lopic diagram of the input multiplexer and the PAL.

The input multiplexer consists of eight, one-out-of-eight digital multiplexer circuits, one for each input word of eight data bits. The outputs of these circuits are wired to form an 8 bit bus. Selection of the input signal to be passed through to the output is by internal decoders driven in parallel by the master counter in the timing and control logic.

To remove the temporal delay between odd and even detectors, the pixel data is demultiplexed onto two lines. This demultiplexer changes state in accordance with a decoded output from the master counter to segregate the odd-numbered pixels from the even-numbered ones. The odd-numbered pixels are delayed 128 pixel intervals (16 odd pixels per band for each of 8 bands) until their temporally-associated even-numbered pixels are input. A read-only-memory (ROM), programmed with the misaligned sequence data is addressed by the master counter (straight sequence). The ROM output addresses the pixels into the correct addresses within a pair (double-buffered) of storage random access memories (RAMs). The storage allocation within the RAMs is on a per-band-column basis.

When one RAM is filled, writing progresses into the other. Data is read out of the RAM not being filled, under the control of addresses generated in straight sequence by the master counter. The control of the reading and writing is performed by the read/write address select logic.

7.4.3 Hadamard Transform Encoder

Although many configurations of the Hadamard transform encoder are possible, processing timing considerations have limited the HTE configurations considered to either the 16- or 32-point pipeline configurations. Figure 7-11a illustrates this configuration. Not shown in Figure 7-11a are the interstage latches required to buffer and hold the data for inputting into the succeeding arithmetic elements. Discussion with personnel within TRW's Microelectronics Center has indicated that it is feasible to build an LSI chip containing much of the HTE within a single package, thereby greatly reducing the number of required parts.

The HTE receives input data from the PAL word serial (bit parallel) per-color-band column format. This data is shifted into the serial-to-parallel converter. Upon receipt of a transfer command from the timing

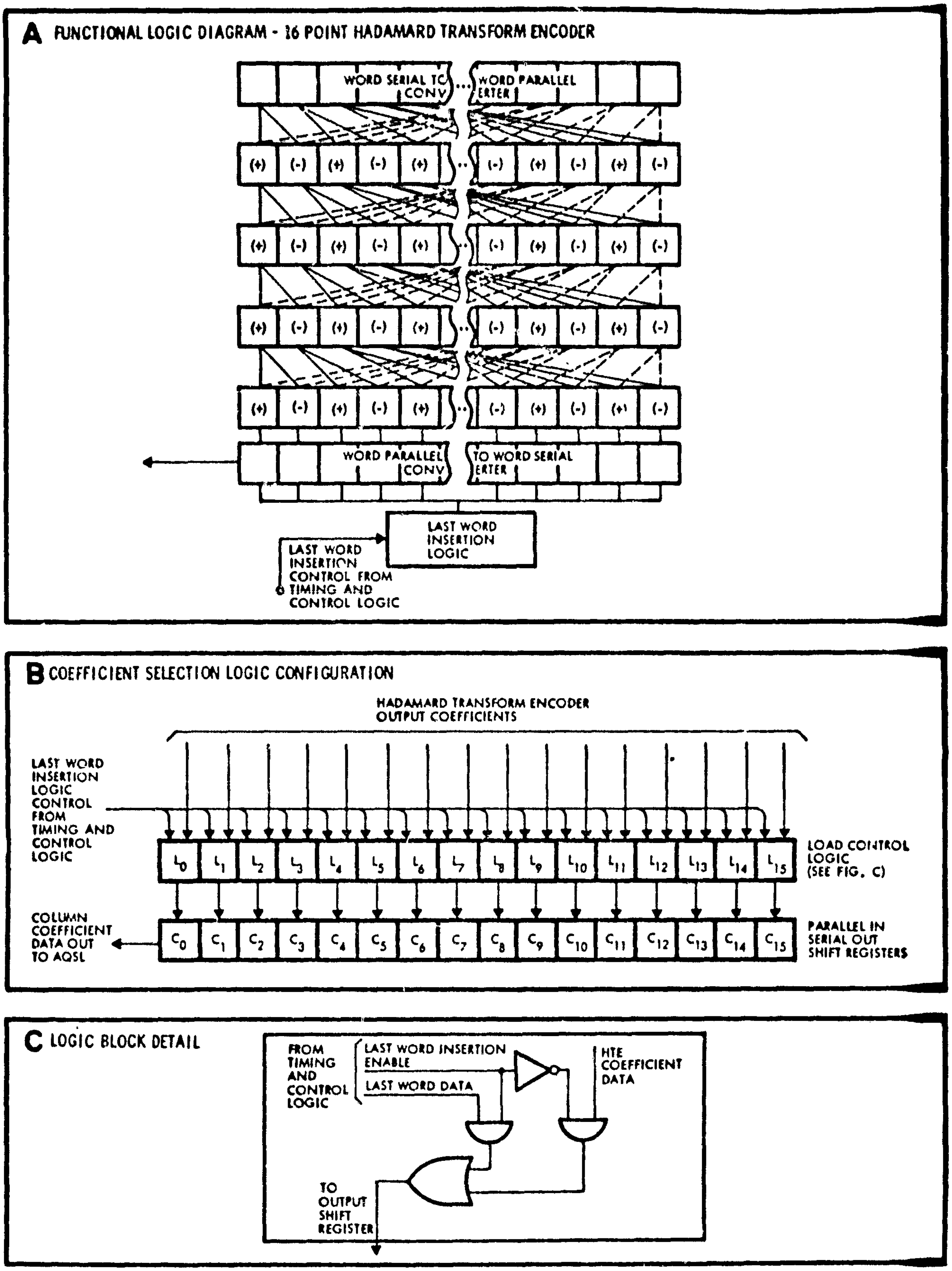

A FUNCTIONAL LOGIC DIAGRAM - 16 POINT HADAMARD TRANSFORM ENCODER
WORD SERIAL TO WORD PARALLEL CONVERTER
(+) (-) (+) (-) (+) (-) (+) (-) (+) (-)
WORD PARALLEL TO WORD SERIAL CONVERTER
LAST WORD INSERTION LOGIC
LAST WORD INSERTION CONTROL FROM TIMING AND CONTROL LOGIC
B COEFFICIENT SELECTION LOGIC CONFIGURATION
HADAMARD TRANSFORM ENCODER OUTPUT COEFFICIENTS
LAST WORD INSERTION LOGIC CONTROL FROM TIMING AND CONTROL LOGIC
L0 L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15
LOAD CONTROL LOGIC (SEE FIG. C)
COLUMN COEFFICIENT DATA OUT TO AQSL
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C12 C13 C14 C15
PARALLEL IN SERIAL OUT SHIFT REGISTERS
C LOGIC BLOCK DETAIL
FROM TIMING AND CONTROL LOGIC
LAST WORD INSERTION ENABLE
LAST WORD DATA
HTE COEFFICIENT DATA
TO OUTPUT SHIFT REGISTER

Figure 7-11. Hadamard Transform Encoder

and control logic, the data is parallel transferred to the arithmetic elements. Data progresses down through the HTE in a manner similar to a number of parallel serial shift registers; each shift being controlled by the transfer strobe.

Figure 7-11b illustrates the manner in which data is output from the HTE. The HTE coefficient data passes through the bank of logic blocks and is strobed into parallel-to-serial converter elements. The coefficient selection function is performed within the bank of logic blocks, using the circuit of Figure 7-11c.

Uplink commands containing information detailing the number and identification of HTE coefficients, per color band, to be selected are stored within the timing and control logic. This information is decoded within the timing and control logic and supplied to the coefficient selection logic elements. Here it serves to cause the insertion of a readily recognizable word following the last selected coefficient to be processed further. The selected coefficients are counted from the lowest to the highest order coefficients. Indeed if a whole color band coefficient is to be eliminated, the last word is inserted into the lowest order coefficient slot.

The coefficients are shifted out with the lowest order coefficients first.

7.4.4 Adaptive Quantizer Select Logic and Rate Buffer Store

The adapter quantizer selection logic and rate buffer (AQSL) perform three distinct functions: 1) column-to-line sequence conversion, 2) rate buffering, and 3) quantizer selection. Each of these functions is discussed in the following paragraphs.

7.4.4.1 Overall AQSL Operation

Figure 7-12a shows a functional logic diagram of the AQSL block. Column sequential coefficient data (containing the selected coefficients and the last words) is input into the rate buffer memory. This memory is implemented as a double buffer, thereby permitting simultaneous input and output of data. Both the column-to-row conversion and rate buffering functions are performed within the rate buffer.

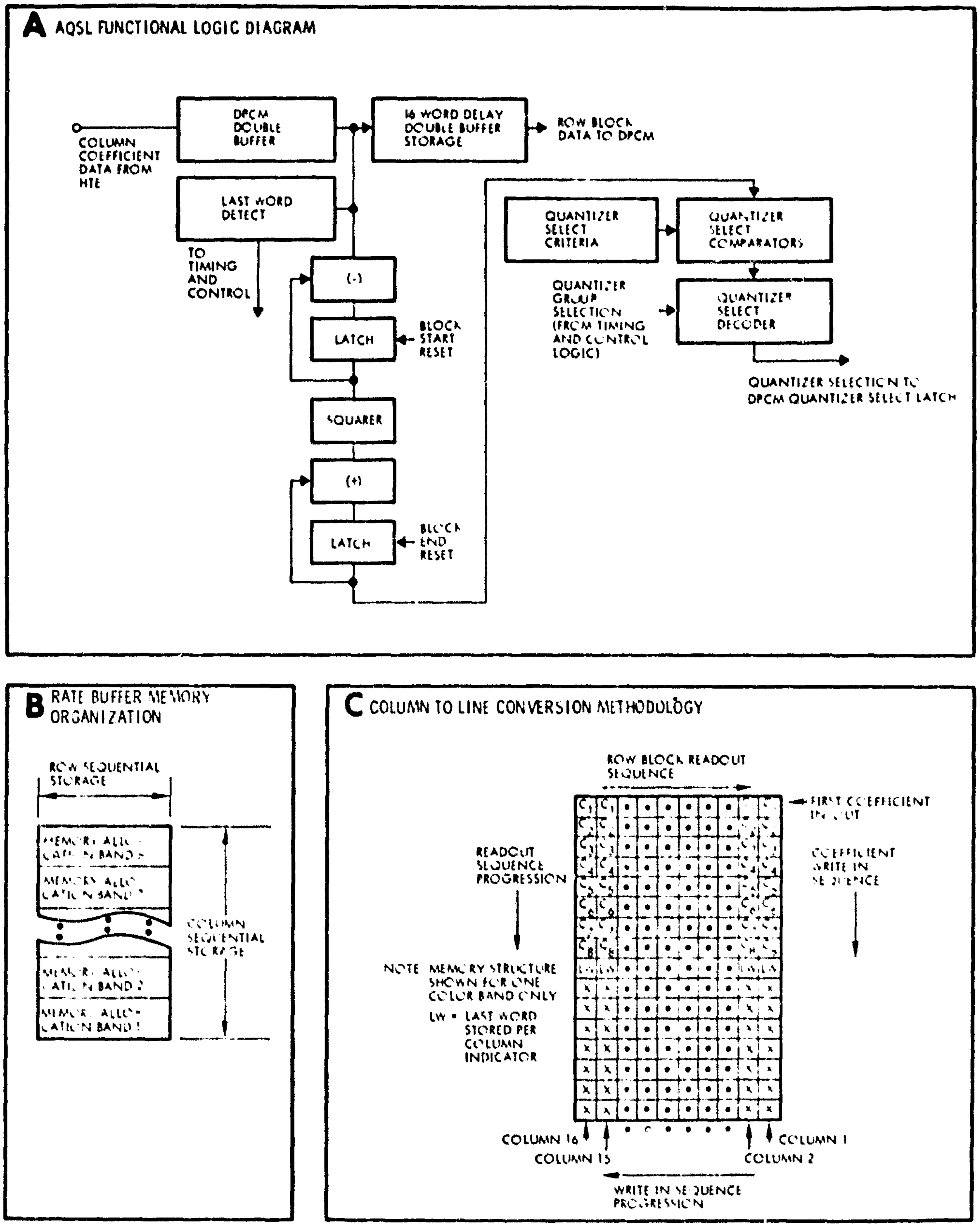

A AQSL FUNCTIONAL LOGIC DIAGRAM
COLUMN COEFFICIENT DATA FROM HTE
DPCM DOUBLE BUFFER
16 WORD DELAY DOUBLE BUFFER STORAGE
ROW BLOCK DATA TO DPCM
LAST WORD DETECT
TO TIMING AND CONTROL
(-)
LATCH
BLOCK START RESET
SQUARER
(+)
LATCH
BLOCK END RESET
QUANTIZER SELECT CRITERIA
QUANTIZER SELECT COMPARATORS
QUANTIZER GROUP SELECTION (FROM TIMING AND CONTROL LOGIC)
QUANTIZER SELECT DECODER
QUANTIZER SELECTION TO DPCM QUANTIZER SELECT LATCH
B RATE BUFFER MEMORY ORGANIZATION
ROW SEQUENTIAL STORAGE
MEMORY ALLOCATION BAND 8
MEMORY ALLOCATION BAND 7
MEMORY ALLOCATION BAND 2
MEMORY ALLOCATION BAND 1
COLUMN SEQUENTIAL STORAGE
C COLUMN TO LINE CONVERSION METHODOLOGY
ROW BLOCK READOUT SEQUENCE
FIRST COEFFICIENT IN/OUT
READOUT SEQUENCE PROGRESSION
COEFFICIENT WRITE IN SEQUENCE
NOTE MEMORY STRUCTURE SHOWN FOR ONE COLOR BAND ONLY
LW = LAST WORD STORED PER COLUMN INDICATOR
C1 C2 C3 C4 C5 C6 C7 C8
LW
X
COLUMN 16
COLUMN 15
COLUMN 2
COLUMN 1
WRITE IN SEQUENCE PROGRESSION

Figure 7-12. Adaptive Quantizer Selection Logic and Rate Buffer

As data is read from the rate buffer, it is operated upon by the quantizer select logic. This logic performs a sum of squares operation upon the data. Upon completion of 16 cycles of arithmetic operations (one complete row data block), the resultant sum is compared against stored quantizer selection criteria, decoded, and transmitted to the DPCM encoder logic as a quantizer selection command code.

While the arithmetic operations proceed, the row data block is held within a 16-word serial shift register bank to delay the data to enter the DPCM encoder in phase with the quantizer selection code.

7.4.4.2 Column-to-Line Conversion and Rate Buffering

Figure 7-12b illustrates the basic rate buffer memory organization. This figure illustrates that a block of memory is reserved for each color band. Figure 7-12c indicates the methodology of the memory organization within each memory block. Referring to this figure it can be seen that the coefficient data is written into the memory in a columnwise fashion (last words included). The write-in sequence progresses top to bottom, right to left. The readout sequence, however, proceeds on a row-by-row basis. Stored coefficients are read out sequentially right to left, top to bottom, thus completing the column-to-row conversion.

The memory is sized to accommodate a row block of 16 words of data. To do this, 16 columns of data must be stored for each color band. Twice this much memory must be provided to accommodate the requirement to double buffer to permit simultaneous read/write operations.

Addressing the read/write control for this memory is provided by the timing and control logic.

As the memory is read row by row, in time the last words are encountered. These special words are decoded by the last word detection logic and serve to signal the timing and control logic that no more coefficients will be forthcoming from that color band. This, in turn, causes the read address control logic within the timing and control logic to discontinue addressing that block of color band data and to skip on to the next band. By this means, holes in the data stream are skipped over and automatic rate buffering is accomplished.

7.4.4.3 Quantizer Selection Logic

Figure 7-13 illustrates the functional logic diagram of the quantizer selection logic. Row coefficient data exiting from the rate buffer store is input into both the squaring logic (table look-up ROM) and a 16-word delay register. The squared data is iterated through a digital accumulator consisting of an adder and a storage latch. The latch accumulates the cumulative sum of squares value. After 16 iterations (sufficient for a DPCM row data block), the sum of squares value is compared against programmed selection criteria by the digital comparator array. These devices each output one of three signals corresponding to whether one input is numerically greater than, equal to, or less than, the other input. By this means, the actual cumulative sum of squares value can be located to within one of the four segments of its total possible range. The comparator outputs are decoded into one of four quantizer selection codes and are transmitted to the DPCM encoder.

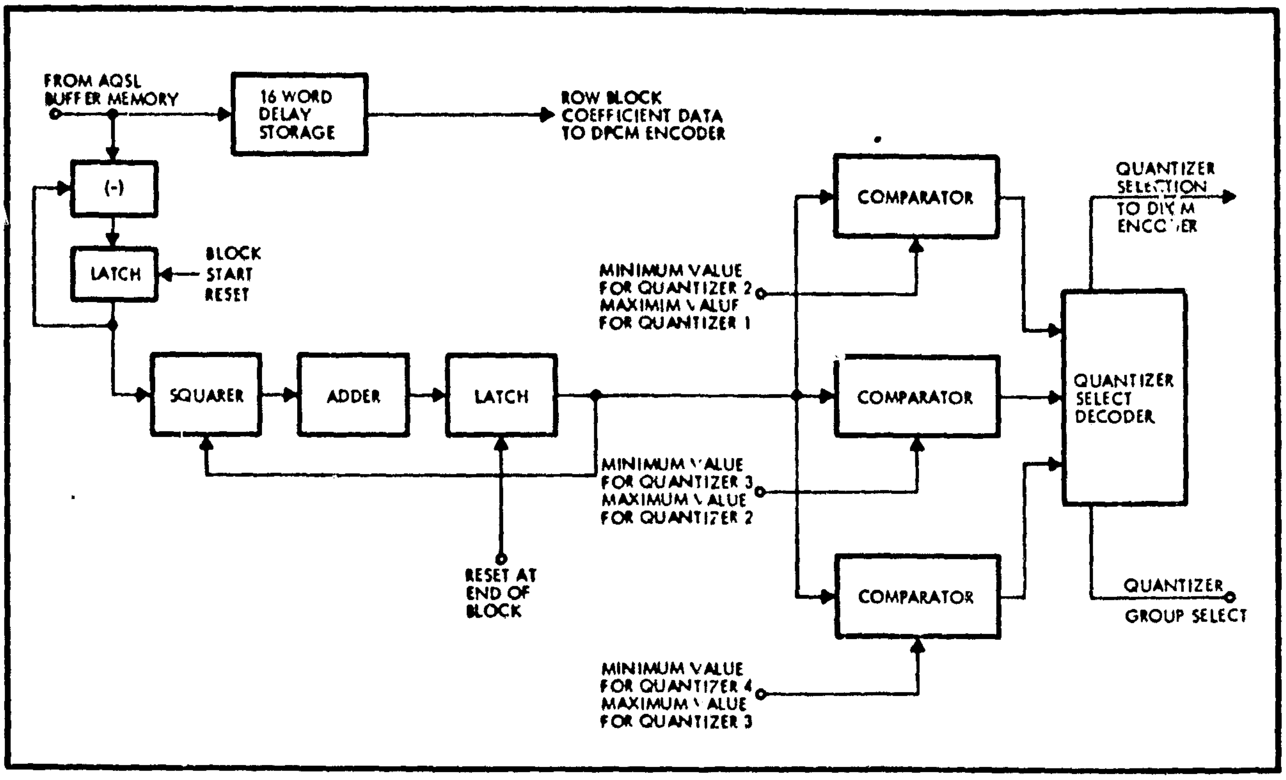

FROM AQSL BUFFER MEMORY
16 WORD DELAY STORAGE
ROW BLOCK COEFFICIENT DATA TO DPCM ENCODER
(-)
LATCH
BLOCK START RESET
SQUARER
ADDER
LATCH
RESET AT END OF BLOCK
MINIMUM VALUE FOR QUANTIZER 2 MAXIMIM VALUE FOR QUANTIZER 1
MINIMUM VALUE FOR QUANTIZER 3 MAXIMUM VALUE FOR QUANTIZER 2
MINIMUM VALUE FOR QUANTIZER 4 MAXIMUM VALUE FOR QUANTIZER 3
COMPARATOR
COMPARATOR
COMPARATOR
QUANTIZER SELECT DECODER
QUANTIZER SELECTION TO DPCM ENCODER
QUANTIZER GROUP SELECT

Figure 7-13. Quantizer Selection Logic

It should be noted that the uplinked stored program selects which one of four groups of quantizers is used to encode a particular color band, and the AQSL selects a particular quantizer from that group.

Simultaneous with the transmission of the quantizer selection code to the DPCM encoder, the row coefficient data also becomes available to the DPCM encoder's subtractor.

7.4.5 Adaptive 1D-DPCM Encoder

The theory and operation of a DPCM encoder loop are discussed in Section 3.2.2. Its mechanization is covered in this section, with particular attention paid to the quantizer and channel coding elements.

The arithmetic and operational sequences within a 1D-DPCM loop, such as in Figure 7-14a, are straightforward, except for the polarity detection and correction and the quantizer and channel coding circuitry.

The quantizer selection code is applied to the quantizer select latch, where it is held for the duration of a row block of data (16 coefficients). Figure 7-14b illustrates how this is accomplished. The quantizer group selection is accomplished by the timing and control logic in accordance with the stored uplink commands. The quantizer select signals select a quantizer from within the selected group.

Stored within the quantizers are mean values corresponding to a particular cut-point value. The cut-point values are represented by the addresses of each quantizer RAM, and the mean value by the contents stored at that address. Thus for all input memory addresses between two cut-points, the memory contents are identical and equal to the mean value for that cut-point bin. The difference between the current coefficient and the previously processed coefficient is taken from the subtractor and applied to the quantizer as an input read address, and the quantizer outputs the mean value output corresponding to the cut-point bin of the difference signal.

To conserve space within the quantizer RAMs, the sign bit of the difference signal is stripped from the difference prior to quantization and recombined with the quantizer output.

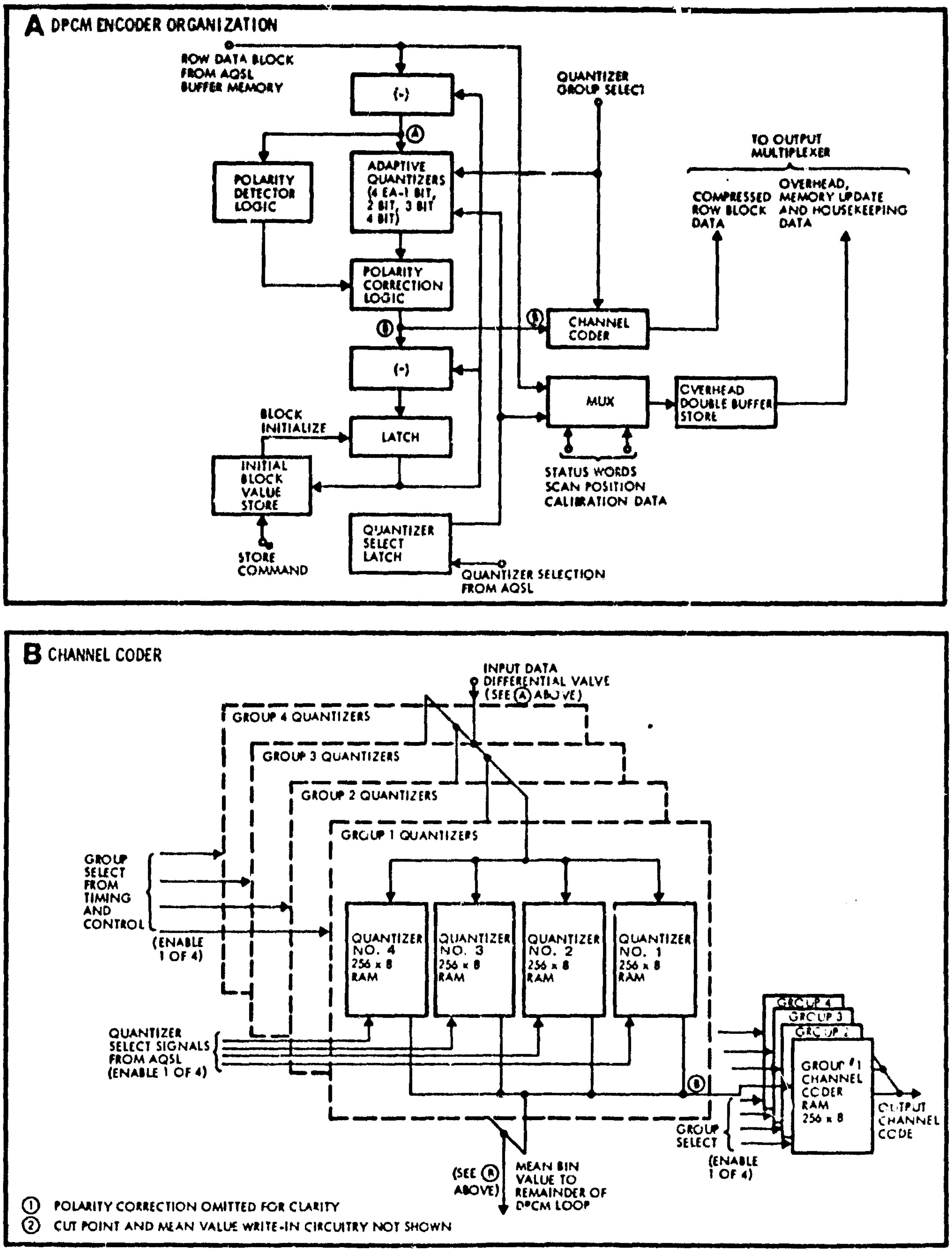

A DPCM ENCODER ORGANIZATION
ROW DATA BLOCK FROM AQSL BUFFER MEMORY
(-)
QUANTIZER GROUP SELECT
Ⓐ
POLARITY DETECTOR LOGIC
ADAPTIVE QUANTIZERS (4 EA-1 BIT, 2 BIT, 3 BIT 4 BIT)
TO OUTPUT MULTIPLEXER
COMPRESSED ROW BLOCK DATA
OVERHEAD, MEMORY UPDATE AND HOUSEKEEPING DATA
POLARITY CORRECTION LOGIC
Ⓑ
①
CHANNEL CODER
(-)
MUX
OVERHEAD DOUBLE BUFFER STORE
BLOCK INITIALIZE
LATCH
INITIAL BLOCK VALUE STORE
STATUS WORDS SCAN POSITION CALIBRATION DATA
STORE COMMAND
QUANTIZER SELECT LATCH
QUANTIZER SELECTION FROM AQSL
B CHANNEL CODER
INPUT DATA DIFFERENTIAL VALVE (SEE Ⓐ ABOVE)
GROUP 4 QUANTIZERS
GROUP 3 QUANTIZERS
GROUP 2 QUANTIZERS
GROUP 1 QUANTIZERS
GROUP SELECT FROM TIMING AND CONTROL
(ENABLE 1 OF 4)
QUANTIZER NO. 4 256 x 8 RAM
QUANTIZER NO. 3 256 x 8 RAM
QUANTIZER NO. 2 256 x 8 RAM
QUANTIZER NO. 1 256 x 8 RAM
QUANTIZER SELECT SIGNALS FROM AQSL (ENABLE 1 OF 4)
②
GROUP 4
GROUP 3
GROUP 2
GROUP #1 CHANNEL CODER RAM 256 x 8
OUTPUT CHANNEL CODE
GROUP SELECT
(ENABLE 1 OF 4)
(SEE Ⓑ ABOVE)
MEAN BIN VALUE TO REMAINDER OF DPCM LOOP
① POLARITY CORRECTION OMITTED FOR CLARITY
② CUT POINT AND MEAN VALUE WRITE-IN CIRCUITRY NOT SHOWN

Figure 7-14. DPCM Quantizer and Channel Coder Organization

The corrected mean value is similarly applied to the address inputs of the channel coder RAM, group selected to match the quantizer group selection. Depending upon the group selection, the channel coder RAMs will output one of four unique 1 bit, 2 bit, 3 bit, or 4 bit codes for the polarity corrected mean value input.

The use of tristate output RAMs permits the wired-ORing of outputs without additional multiplexing.

The mean value outputs from the quantizers are supplied to the remainder of the DPCM feedback loop, as well as to the input of the channel coder. The channel coder output is provided to the output multiplexer.

System operating constraints require that the DPCM encoder must sequentially process data blocks from each selected color band. As many as 112 coefficient blocks of data (16 coefficient blocks for seven bands plus eight coefficient blocks for the IR band) from each of seven color bands may be interposed between contiguous coefficient data blocks from the same color band. To reinitialize the DPCM loop, the feedback loop value remaining within the latch logic at the end of each block is stored within the initial block value store. At the time of quantizer selection, this value is restored to the latch. Thus, whenever a new coefficient block of data is processed by the DPCM encoder, it is initialized by the final value which existed upon termination of processing the previous corresponding coefficient block within that color band.

Along with the channel code, the downlink must also be provided with the identification of the particular quantizer used for each data block. Further, to prevent cumulative systematic errors from building up due to transmission channel errors, words from the initial block value store (memory update) must be provided every eight blocks. The quantizer identification, memory update overhead, and assorted housekeeping data such as frame sync, selected telemetry measurements, and scan position are multiplexed together, buffered, and supplied to the output multiplexer.

7.4.6 Output Multiplexer and Rate Buffer

The output multiplexer and rate buffer assembly illustrated in Figure 7-15 serves to combine the channel data, and all overhead, calibration, and housekeeping data into a formatted bit serial data stream. Rate buffering is provided to smooth the output data stream throughout the active and inactive scan interval.

The rate buffer output is parallel-to-serial converted via a parallel in/serial out shift register not shown on Figure 7-15 and electrically buffered to the downlink transmission equipment.

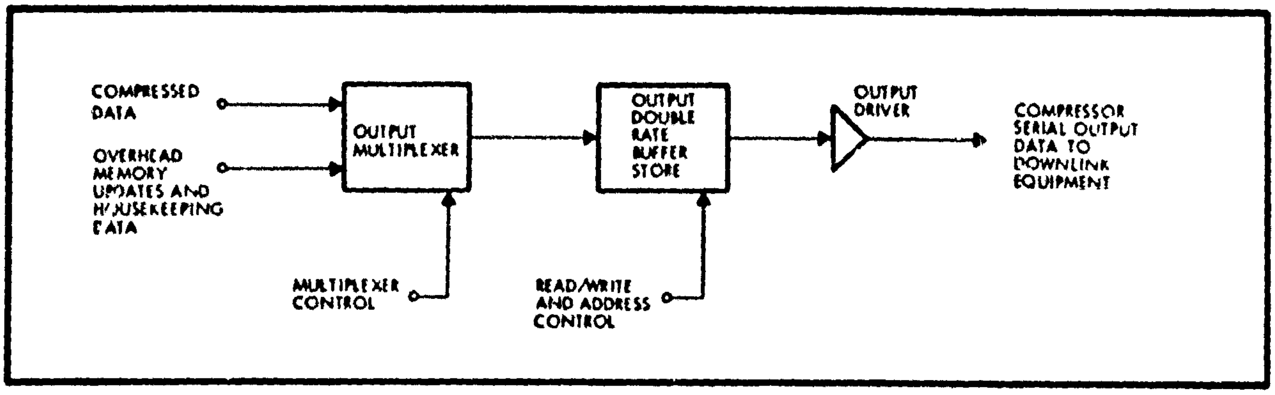

COMPRESSED DATA
OVERHEAD MEMORY UPDATES AND HOUSEKEEPING DATA
OUTPUT MULTIPLEXER
MULTIPLEXER CONTROL
OUTPUT DOUBLE RATE BUFFER STORE
READ/WRITE AND ADDRESS CONTROL
OUTPUT DRIVER
COMPRESSOR SERIAL OUTPUT DATA TO DOWNLINK EQUIPMENT

Figure 7-15. Output Multiplexer and Rate Buffer Functional Block Diagram

7.4.7 Timing and Control Logic

The timing and control logic serves to coordinate and synchronize all data movement within the data compression system. Further, it provides the storage for the uplinked system reconfiguration commands and the means for executing these commands.

As illustrated in Figure 7-16a, the timing and control assembly consists of the following three major blocks:

1) Master counter and decoder

2) Coefficient and quantizer selection

3) Quantizer store and control.

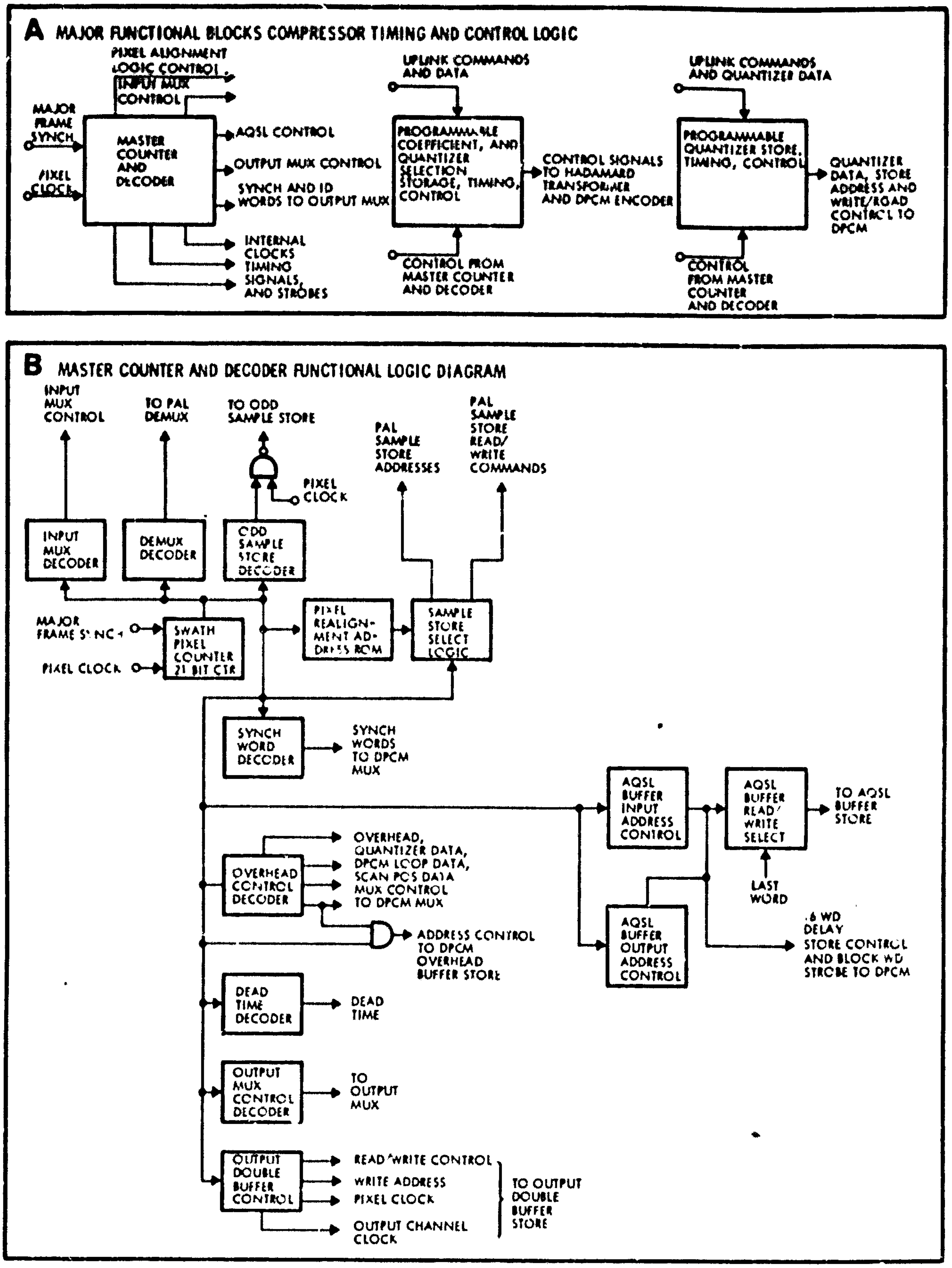

A MAJOR FUNCTIONAL BLOCKS COMPRESSOR TIMING AND CONTROL LOGIC
PIXEL ALIGNMENT LOGIC CONTROL
INPUT MUX CONTROL
MAJOR FRAME SYNCH
PIXEL CLOCK
MASTER COUNTER AND DECODER
AQSL CONTROL
OUTPUT MUX CONTROL
SYNCH AND ID WORDS TO OUTPUT MUX
INTERNAL CLOCKS TIMING SIGNALS, AND STROBES
UPLINK COMMANDS AND DATA
PROGRAMMABLE COEFFICIENT, AND QUANTIZER SELECTION STORAGE, TIMING, CONTROL
CONTROL FROM MASTER COUNTER AND DECODER
CONTROL SIGNALS TO HADAMARD TRANSFORMER AND DPCM ENCODER
UPLINK COMMANDS AND QUANTIZER DATA
PROGRAMMABLE QUANTIZER STORE, TIMING, CONTROL
CONTROL FROM MASTER COUNTER AND DECODER
QUANTIZER DATA, STORE ADDRESS AND WRITE/READ CONTROL TO DPCM
B MASTER COUNTER AND DECODER FUNCTIONAL LOGIC DIAGRAM
INPUT MUX CONTROL
TO PAL DEMUX
TO ODD SAMPLE STORE
PIXEL CLOCK
PAL SAMPLE STORE ADDRESSES
PAL SAMPLE STORE READ/ WRITE COMMANDS
INPUT MUX DECODER
DEMUX DECODER
ODD SAMPLE STORE DECODER
MAJOR FRAME SYNC
PIXEL CLOCK
SWATH PIXEL COUNTER 21 BIT CTR
PIXEL REALIGNMENT ADDRESS ROM
SAMPLE STORE SELECT LOGIC
SYNCH WORD DECODER
SYNCH WORDS TO DPCM MUX
AQSL BUFFER INPUT ADDRESS CONTROL
AQSL BUFFER READ/ WRITE SELECT
TO AQSL BUFFER STORE
LAST WORD
AQSL BUFFER OUTPUT ADDRESS CONTROL
16 WD DELAY STORE CONTROL AND BLOCK WD STROBE TO DPCM
OVERHEAD CONTROL DECODER
OVERHEAD, QUANTIZER DATA, DPCM LOOP DATA, SCAN POS DATA MUX CONTROL TO DPCM MUX
ADDRESS CONTROL TO DPCM OVERHEAD BUFFER STORE
DEAD TIME DECODER
DEAD TIME
OUTPUT MUX CONTROL DECODER
TO OUTPUT MUX
OUTPUT DOUBLE BUFFER CONTROL
READ/WRITE CONTROL
WRITE ADDRESS
PIXEL CLOCK
OUTPUT CHANNEL CLOCK
TO OUTPUT DOUBLE BUFFER STORE

Figure 7-16. Timing and Control Logic for the Reconfigurable Adaptive Hadamard-DPCM Encoder

7.4.7.1 Master Counter and Decoder

Figure 7-16b presents the functional logic diagram of the master counter and decoder. The heart of this logic is the swath pixel counter. It is reset each major frame at the time of the start of the active scan. The swath pixel counter is driven by the pixel clock. This clock operates at eight times the rate the data is clocked out of each color band. This counter is decoded by the various decoders shown in Figure 7-16b to supply the correct control signals to the various logic elements. Also included within this block are the pixel realignment ROM and the sync word generators.

7.4.7.2 Coefficient and Quantizer Selection

Figure 7-17a illustrates the coefficient and quantizer selection timing and control logic. Within this element, storage is provided for the uplinked coefficient selection and quantizer selection commands. These commands may be uplinked at any time during the frame, but they will not take effect until after the next succeeding swath scan begins. Thereafter, they will be continuously executed until receipt of another update.

The coefficient selection data specifies the HTE output coefficients to be further processed for each color band. This data is used to control the insertion of the last word command within the HTE.output parallel-to-serial conversion logic.

Stored with the coefficient select data is the quantizer group select data which controls the selection of the proper group of four quantizers for the color band being processed by the DPCM encoder. The band block counter acts as a slave unit to the swath pixel counter in the master counter and decoder logic.

7.4.7.3 Quantizer Store and Control

Figure 7-17b shows the functional logic diagram of the quantizer store and control. As this data is accessed at a high rate within the DPCM encoder, local storage is provided within the encoder loop. Buffer storage is provided within the quantizer store and control timing and control logic to allow updating of the quantizer and channel code data at any time during the scan frame. The encoder loops are updated only during each swath time. This control logic block provides all necessary control signals to perform this function. The quantizer store counter acts as a slave to the swath pixel counter in the master counter and decoder logic.

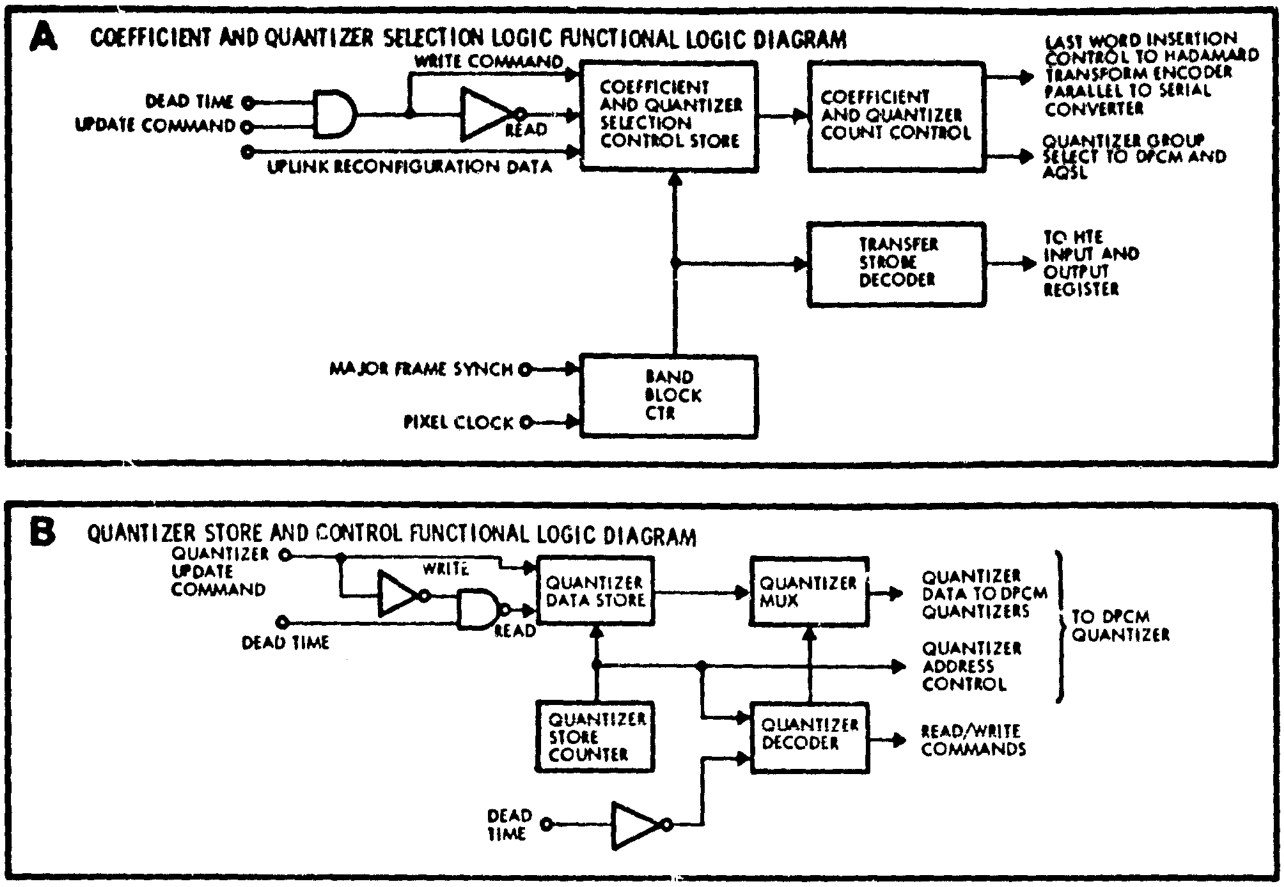

A
COEFFICIENT AND QUANTIZER SELECTION LOGIC FUNCTIONAL LOGIC DIAGRAM
WRITE COMMAND
DEAD TIME
UPDATE COMMAND
READ
UPLINK RECONFIGURATION DATA
COEFFICIENT AND QUANTIZER SELECTION CONTROL STORE
COEFFICIENT AND QUANTIZER COUNT CONTROL
LAST WORD INSERTION CONTROL TO HADAMARD TRANSFORM ENCODER PARALLEL TO SERIAL CONVERTER
QUANTIZER GROUP SELECT TO DPCM AND AQSL
TRANSFER STROBE DECODER
TO HTE INPUT AND OUTPUT REGISTER
MAJOR FRAME SYNCH
PIXEL CLOCK
BAND BLOCK CTR
B
QUANTIZER STORE AND CONTROL FUNCTIONAL LOGIC DIAGRAM
QUANTIZER UPDATE COMMAND
WRITE
DEAD TIME
READ
QUANTIZER DATA STORE
QUANTIZER MUX
QUANTIZER DATA TO DPCM QUANTIZERS
QUANTIZER ADDRESS CONTROL
TO DPCM QUANTIZER
QUANTIZER STORE COUNTER
QUANTIZER DECODER
READ/WRITE COMMANDS
DEAD TIME

Figure 7-17. Timing and Control Elements of the Reconfigurable Hadamard-DPCM Encoder

7.4.8 Hybrid Compressor Hardware Summary

The reconfigurable adaptive Hadamard-DPCM encoder is just one of a number of possible configurations such as nonadaptive or adaptive but not reconfigurable. Table 7-1 shows the results of the hardware tradeoff study in terms of Integrated Circuit (IC) package counts and power consumption for three configurations.

Table 7-1. Hadamard-DPCM Compressor Sizing

System Configuration/ Functional Blocks	Nonadaptive Hybrid (Watts)	Adaptive Hybrid (Watts)	Reconfigurable Adaptive Hybrid (Watts)
Input multiplexer and pixel alignment logic	25 IC/1.3	25 IC/1.3	25 IC/1.3
16-point Hadamard transform encoder	256 IC/13.1	256 IC/13.1	256 IC/13.1
DPCM loop	12 IC/3.1	56 IC/8.7	56 IC/8.7
Adaptive quantizer select logic	—	50 IC/4.6	50 IC/4.6
Output multiplexer and buffer	25 IC/4.6	25 IC/4.6	25 IC/4.6
Timing and control logic	29 IC/7.3	29 IC/7.3	109 IC/14.6
Total	347 IC/32.4	441 IC/42.6	521 IC/49.9

7.5 2D-DPCM ADAPTIVE DATA COMPRESSION SYSTEM

An adaptive 2D-DPCM encoder is an approach to compressing Thematic Mapper image data that has less complexity and flexibility as compared to the Hadamard-DPCM approach previously described. The 2D-DPCM encoder can be made adaptive to accommodate local scene statistics, and can be made reprogrammable to accept revised quantization patterns. The 2D-DPCM encoder is not as flexible, however, in that (in general) only integer bit per pixel ratios can be obtained by the basic encoder. This necessarily leads to a requirement to eliminate or edit out some bands since we must allocate one, two, or more bits per pixel per band encoded.

7.5.1 Operation of the Adaptive 2D-DPCM Compressor

Figure 7-18 illustrates the functional logic diagram of the adaptive 2D-DPCM data compression system. This configuration consists of five major functional blocks:

1) Input multiplexer and pixel alignment logic

2) Adaptive quantizer select logic and rate buffer

3) 2D-DPCM encoder

4) Output multiplexer and rate buffer

5) Timing and control logic.

7.5.1.1 Input Multiplexer and Pixel Alignment Logic

The input multiplexer and pixel alignment logic for the adaptive 2D-DPCM compressor is identical in function and operation to the one used in the adaptive Hadamard-DPCM encoder. For a functional and operational description of this block, the reader is referred to Paragraphs 7.4.1.1 and 7.4.2.

7.5.1.2 Adaptive Quantizer Select Logic and Rate Buffer

The function and operation of this block is similar to that of the Hadamard-DPCM encoder as described in Paragraphs 7.4.1.3 and 7.4.4, with the following exceptions:

a) Column-to-row conversion is not required

b) The AQSL quantizer selection decision is based on both present and previous column information and uses a feedback path from the 2D-DPCM encoder

c) The data block size is a single column of 32 pixels, except for band 6 which uses 16 pixels.

Within the Hadamard-DPCM compression approach, the Hadamard transform encoder processes column data, and the DPCM encoder processes row aligned data. This establishes a requirement for column-to-row conversion. The operation of a 2D-DPCM encoder is such that column data is processed and stored in a manner to make it available for inclusion in the processing of the next successive column. The column storage within the DPCM encoder is organized to ensure that row sequential pixel data is aligned column by column.

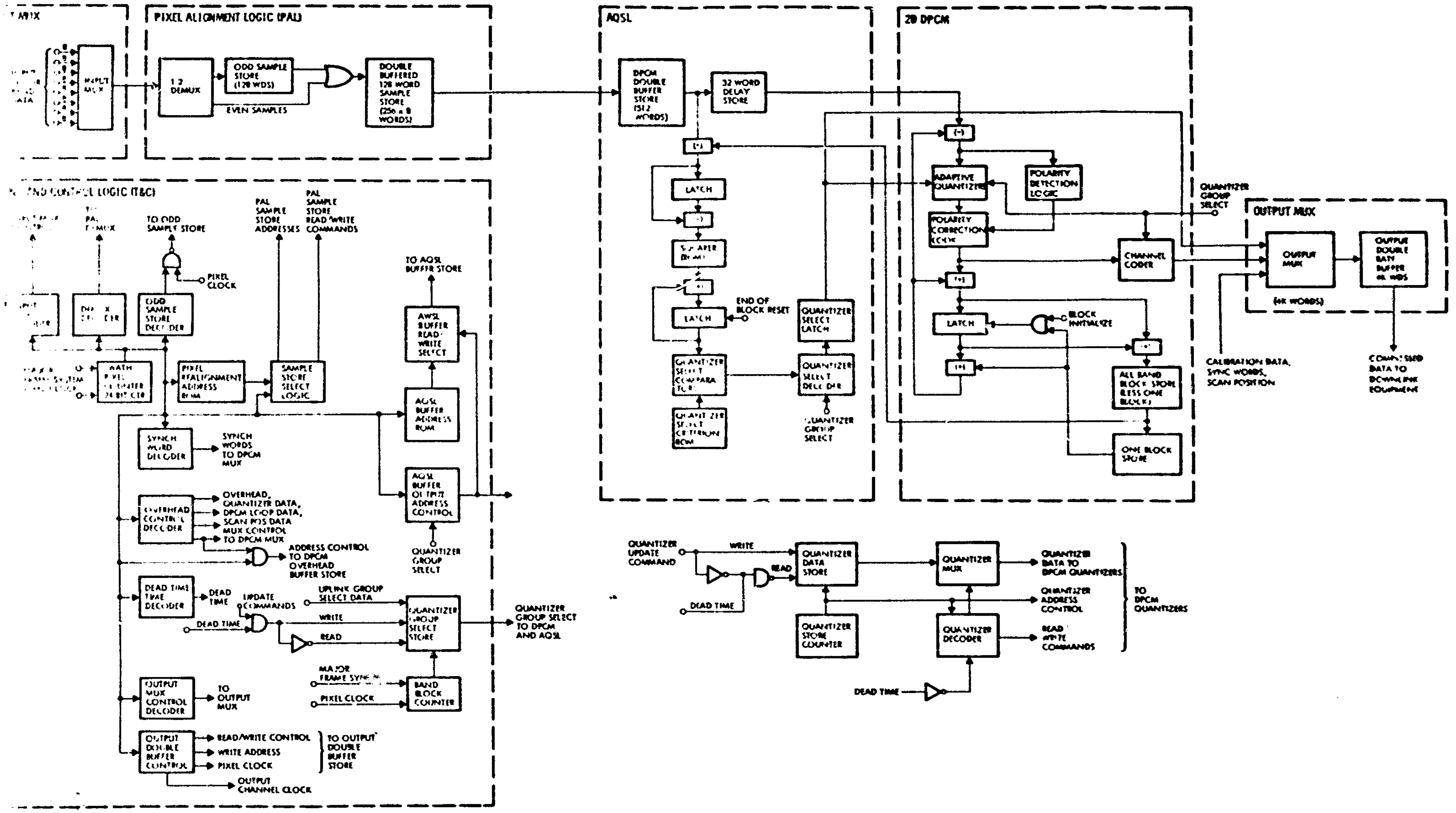

MUX
INPUT MUX
PIXEL ALIGNMENT LOGIC (PAL)
1:2 DEMUX
ODD SAMPLE STORE (120 WDS)
EVEN SAMPLES
DOUBLE BUFFERED 128 WORD SAMPLE STORE (256 x 8 WORDS)
AQSL
DPCM DOUBLE BUFFER STORE (512 WORDS)
32 WORD DELAY STORE
LATCH
SQUARER
LATCH
END OF BLOCK RESET
QUANTIZER SELECT COMPARATOR
QUANTIZER SELECT CRITERION ROM
QUANTIZER SELECT LATCH
QUANTIZER SELECT DECODER
QUANTIZER GROUP SELECT
2D DPCM
ADAPTIVE QUANTIZER
POLARITY DETECTION LOGIC
POLARITY CORRECTION LOGIC
LATCH
BLOCK INITIALIZE
ALL BAND BLOCK STORE (LESS ONE BLOCK)
ONE BLOCK STORE
CHANNEL CODER
QUANTIZER GROUP SELECT
OUTPUT MUX
OUTPUT MUX
OUTPUT DOUBLE BATH BUFFER 4K WDS
(4K WORDS)
CALIBRATION DATA, SYNC WORDS, SCAN POSITION
COMPRESSED DATA TO DOWNLINK EQUIPMENT
TIMING AND CONTROL LOGIC (T&C)
TO PAL DEMUX
TO ODD SAMPLE STORE
PIXEL CLOCK
PAL SAMPLE STORE ADDRESSES
PAL SAMPLE STORE READ/WRITE COMMANDS
DEMUX CONTROL DECODER
ODD SAMPLE STORE DECODER
MATH PIXEL COUNTER 24 BIT CTR
PIXEL REALIGNMENT ADDRESS ROM
SAMPLE STORE SELECT LOGIC
TO AQSL BUFFER STORE
AQSL BUFFER READ/WRITE SELECT
AQSL BUFFER ADDRESS ROM
AQSL BUFFER OUTPUT ADDRESS CONTROL
QUANTIZER GROUP SELECT
SYNCH WORD DECODER
SYNCH WORDS TO DPCM MUX
OVERHEAD CONTROL DECODER
OVERHEAD, QUANTIZER DATA, DPCM LOOP DATA, SCAN POS DATA MUX CONTROL TO DPCM MUX
ADDRESS CONTROL TO DPCM OVERHEAD BUFFER STORE
DEAD TIME TIME DECODER
DEAD TIME
UPDATE COMMANDS
DEAD TIME
UPLINK GROUP SELECT DATA
WRITE
READ
QUANTIZER GROUP SELECT STORE
QUANTIZER GROUP SELECT TO DPCM AND AQSL
MAJOR FRAME SYNC
PIXEL CLOCK
BAND BLOCK COUNTER
OUTPUT MUX CONTROL DECODER
TO OUTPUT MUX
OUTPUT DOUBLE BUFFER CONTROL
READ/WRITE CONTROL
WRITE ADDRESS
PIXEL CLOCK
TO OUTPUT DOUBLE BUFFER STORE
OUTPUT CHANNEL CLOCK
QUANTIZER UPDATE COMMAND
WRITE
READ
DEAD TIME
QUANTIZER DATA STORE
QUANTIZER STORE COUNTER
QUANTIZER MUX
QUANTIZER DECODER
DEAD TIME
QUANTIZER DATA TO DPCM QUANTIZERS
QUANTIZER ADDRESS CONTROL
READ WRITE COMMANDS
TO DPCM QUANTIZERS

Figure 7-18. Adaptive 2D-DPCM Encod
Functional Logic Diagr

Data for all bands is always written into the buffer memory; however, some of this data may be edited out. As with the Hadamard DPCM encoder, the AQSL performs the rate buffering function by skipping over designated color band data. This function is accomplished within the timing and control logic by the quantizer group select storage logic. If no quantizer group is selected for a particular color band, the logic assumes the color band has been edited out and skips over readout of the stored data for that band.

Because the 2D-DPCM encoder processes both row and column sequential pixel data, the quantizer selection must be performed in like manner. Previous column pixel data is subtracted from present column pixel data for processing within the AQSL loop.

7.5.1.3 2D-DPCM Encoder

The function and operation of the 2D-DPCM encoder is similar to that of the 1D-DPCM encoder described in Paragraphs 7.4.1.4 and 7.4.5, with the following exceptions:

a) Column sequential data from the present and previous column is used

b) Column storage for the previous column for each selected color band is required

c) The block size is one column of 32 pixels

d) Block initialization data is stored within the previous column storage.

With the exception of the storage and subsequent arithmetic processing of the mean values derived from the previous column, the arithmetic and logic operation of this encoder is similar to the previously discussed 1D-DPCM encoder. Fetching the appropriate quantizer is also the same as for the 1D-DPCM encoder.

7.5.1.4 Output Multiplexer and Rate Buffer

With the exception of the output data format, the function and operation of the adaptive 2D-DPCM output multiplexer and rate buffer is identical to its counterpart in the adaptive Hadamard-DPCM compression system as described in Paragraphs 7.4.1.5 and 7.4.6.

7.5.1.5 Timing and Control Logic

The concept, function, and operation of the 2D-DPCM compressor timing and control logic is similar to that of the adaptive Hadamard-DPCM compressor as described in Paragraph 7.4.7, except that:

a) Coefficient selection logic is not required

b) The logic for the row to column conversion is not required

c) The Hadamard transform encoder control logic is not required.

Quantizer group selection for a particular color band is an uplink programmable feature of both data compression systems described herein. In the adaptive 2D-DPCM system, editing out a particular color band is accomplished by not providing quantizer group selection data for that band. This suppresses the readout of the stored pixel data for that band from the AQSL buffer memory.

With the aforementioned exceptions, all of the reprogrammable flexibility in the adaptive Hadamard-DPCM data compression system is also in the adaptive 2D-DPCM data compressor.

7.5.2 2D-DPCM Compressor Hardware Summary

The reconfigurable adaptive 2D-DPCM encoder is just one of a number of possible configurations such as nonadaptive or adaptive but not reconfigurable. Table 7-2 shows the results of the hardware tradeoff study in terms of Integrated Circuit (IC) package counts and power consumption for three configurations.

Table 7-2. 2D-DPCM Compressor Sizing

System Configuration/ Functional Blocks	Nonadaptive 2D-DPCM (Watts)	Adaptive 2D-DPCM (Watts)	Reconfigurable and Adaptive 2D-DPCM (Watts)
Input multiplexer and pixel alignment logic	25 IC/1.3	25 IC/1.3	25 IC/1.3
DPCM loop	36 IC/12.4	83 IC/18.2	83 IC/18.2
Adaptive quantizer select logic	–	58 IC/6.6	58 IC/6.6
Output multiplexer and buffer	25 IC/3.1	25 IC/3.1	25 IC/3.1
Timing and control logic	29 IC/7.3	29 IC/7.3	109 IC/14.6
Total	115 IC/24.1	220/36.5	300/43.8

8. CONCLUSIONS AND RECOMMENDATIONS

8.1 CONCLUSIONS

This study follows two TRW studies on bandwidth compression of MSS data. The first study, completed April 74,[45] considered compressing the bandwidth of MSS data using distortion-free coding techniques. The salient features of that study are:

a) Compressed bit rates, averaged over the scene, vary from a minimum of 1.22 bits/sample to a maximum of 3.747 bits/sample for the strictly information preserving algorithms.

b) The strictly information preserving algorithms can compress four full 100 x 100 nmi scenes to occupy the same number of magnetic tapes currently required to store one full scene. An even greater reduction is possible with the essentially information-preserving algorithms.

c) The effect of channel errors is minimal if the channel bit error rate is less than 10^{-6}. Channels with higher error rates can be used if frequency memory updates are included.

d) An implementation of one of the candidate techniques (SSDI/Rice algorithm) was developed to illustrate the feasibility of operation at rates above 100 Megabits/second with moderate complexity. Parallel data compressor units operating on blocks of data permit operation at several hundred Mbps.

The main conclusion of the "Study of On-Board Compression of Earth Resources Data,"[1] completed September 1975, was that compression ratios of 3 or 4 to 1 are achievable with very little distortion on multispectral earth resources imagery data. We also concluded that the compression ratio may be increased to 5 to 1 by making the systems operate at a variable bit rate with a Huffman encoder. The problem of utilizing spectral redundancy of the multispectral data was considered and it was concluded that utilizing spectral redundancy by using a spectral transformation improves the signal-to-noise ratio (SNR) by about 1 dB.

The conclusions of the current study of adaptive techniques for compressing the bandwidth of MSS imagery are:

1) At a compression ratio corresponding to 2 bits per pixel, all adaptive coding techniques produce good results. The reconstructed images are subjectively only slightly distinguishable from originals, the SNR

after coding is as high as 38 dB, and recognition accuracies of over 85 percent are achievable. This performance is possible using adaptive bandwidth compression techniques that operate at a fixed bit rate. This is a significant conclusion of this study since reducing the bandwidth of data generated by the TM to a bandwidth that can be transmitted over S-band corresponds to a compression ratio equivalent to 1.84 bits per pixel. Performance of adaptive techniques is approximately 2 to 5 dB better than that of the best nonadaptive techniques as shown in Figure 8-1.

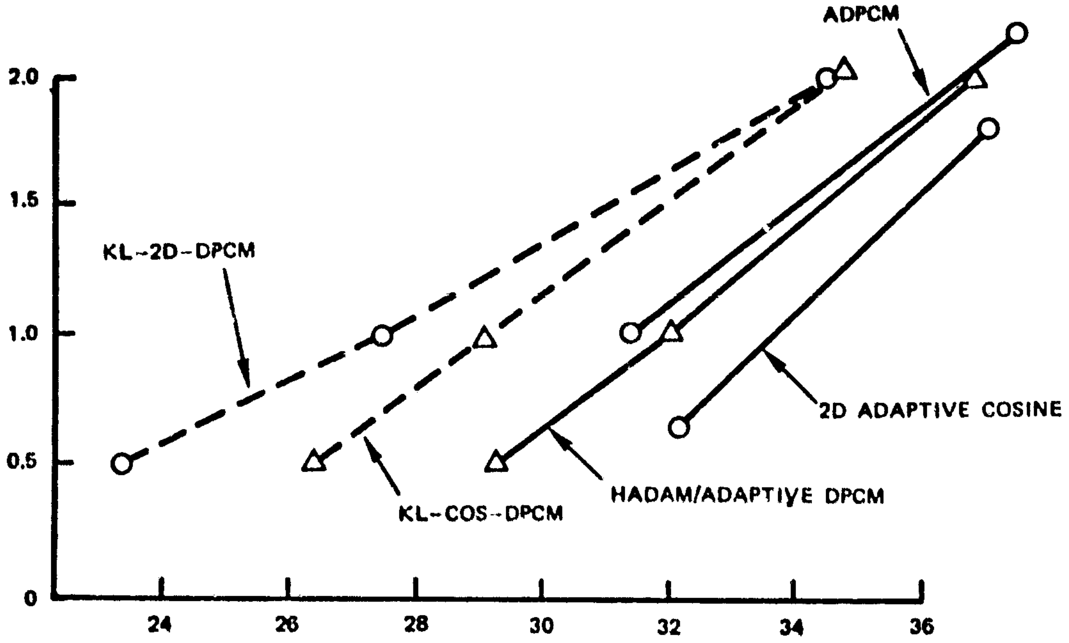


Figure 8-1. Comparison of the Best Adaptive and Nonadaptive Techniques

2) At lower bit rates, the difference between the four selected adaptive techniques becomes larger. The 2D adaptive DPCM system degrades faster than others. At 1 bit per pixel this difference grows to a few dB. At this bit rate, adaptive cluster coding and adaptive 2D transform coding using the Cosine transform give the best results. However, these techniques have large implementation complexity and operate at a variable bit rate which requires rate buffering and buffer control logic which further increases the implementation complexity. The hybrid encoder is slightly inferior to the

best techniques at this bit rate, out due to its much lower implementation complexity it may be most appropriate for LANDSAT-D applications. At this compression ratio the hybrid encoder using the Hadamard transform shows some blockiness that disappears by utilizing the Cosine transform instead of the Hadamard transform.

3) The use of the Haar as a spectral transformation results in smaller SNR for the agriculture scene. This loss is different for various techniques and also depends on the compression ratio. However, it is fairly large as shown in Figure 5-6. The reasons for this loss is explained in Section 5.2. For the Bald Knob scene, the results improve very slightly due to use of the spectral transform. Since the data format in the TM is such that spectral transformation is fairly complex to implement, we conclude that spectral transformation should not be used in the on-board implementation of any bandwidth compression system using one of the techniques proposed in this study.

4) Although the four selected techniques give rather similar results as measured by MSE and SNR, the subjective effects of the degradation for various techniques is quite different. The degradation in the reconstructed pictures using the 2D DPCM system is in the form of blurring. The hybrid encoder and 2D transform coding techniques using the Hadamard transform introduce blockiness at low bit rates while the cluster coding technique gives crisp pictures with a contouring effect at bit rates lower than 1 bit per pixel.

5) The classification consistency performance of the selected techniques is very scene dependent. Where most techniques give good classification consistency (strongly correlated with SNR) using the agricultural scene (see Figure 5-8), they produce rather poor results using the Bald Knob scene. This is primarily due to the fact that the spectral signatures of the classes in the Bald Knob scene are less distinct than those of the classes in the agricultural scene. Therefore, a small change in the gray level of a particular pixel causes a shift in class for the Bald Knob scene where a similar change does not affect the classification of the corresponding pixel in the agricultural scene. We conclude that the automatic classification performance of compression techniques is highly application and scene dependent.

6) In adaptive techniques, the crucial parameters of the encoder are optimized for individual picture blocks as opposed to nonadaptive techniques that use a fixed set of parameters. It has two distinct impacts on the performance of the adaptive system. The first is that this makes the performance of the techniques less scene dependent than nonadaptive techniques, making the results more generally appplicable. Second, the adaptive techniques handle abnormal situations such as variations in the sensor gain and bias and visibility variations more effectively than the nonadaptive techniques. Indeed, the effect of haze was simulated in this study and was shown that it has minimal effect on the performance of the selected adaptive techniques.

8.2 RECOMMENDATIONS

The fundamental conclusion of this study is that the bandwidth of imagery generated by scanners such as the TM can be reduced without introducing significant degradation such that the data can be transmitted over an S-band channel. This corresponds to a compression ratio equivalent to 1.84 bits per pixel. The study further shows that this can be achieved using at least two fairly simple techniques with weight-power requirements well within the constraints of the LANDSAT-D satellite. These are the adaptive 2D DPCM and adaptive hybrid techniques. The specific suggestions for further NASA activities in the data compression area are:

1) A prototype data compression unit is recommended to demonstrate the concept and provide a body of test data for evaluation by various users. One such experimental project can compress the bandwidth of MSS data using perhaps two selected compression ratios and provide this data to various users for their evaluation.

2) The issue of supervised and non-supervised classification accuracy is of paramount importance in machine processing of earth resources imagery. The non-supervised classifier employed in this study is not a suitable measure for evaluating the performance of the bandwidth compression techniques for MSS data. In fact, experiments with a Bayes-supervised classifier using a set of training samples corresponding to six known geological features shows that the recognition accuracy actually improves when the bandwidth of the MSS data is compressed by a factor of 6 to 1.[52] We

recommend further studies of supervised and non-supervised classification techniques to develop a classifier that is meaningful in terms of incorporating user requirements as well as serving as a meaningful tool in measuring performance of various bandwidth compression techniques.

3) Further study of the cluster coding technique is recommended. This is the only bandwidth compression technique which uses clustering and classification of the MSS data for its bandwidth compression. However, this technique in its present form is very complex for hardware or software implementation. Its performance, on the other hand, is compatible with the performance of the best suggested techniques. Further study of this technique for reducing its hardware and software complexity as well as use of a supervised classification algorithm for the clustering procedure is recommended.

REFERENCES

1. A. Habibi, Study of On Board Compression of Earth Resources Data, TRW Final Report, NAS2-8394, September 1975.

2. N. Ahmed, T. Natarajan, "An Adaptive Transform Coding Approach for Multispectral Scanner Data," Final Report for NASA ARC, NCA2-OR363-601.

3. T.T.Y. Huang, and P.M. Schulthesiss, "Block Quantization of Correlated Gaussian Random Variables," IRE Transactions on Communication Systems, Vol. CS-11, No. 3, pp. 289-296, September 1963.

4. A. Habibi, and P.A. Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Transactions on Communication Technology, Vol. COM-19, No. 1, pp. 50-63, February 1971.

5. H. Hotelling, "Analysis of Complex of Statistical Variables into Principal Components," Journal of Educational Psychology, Vol. 24, pp. 417-441, 498-520, 1933.

6. A.K. Jain, "A Fast Karhunen-Loeve Transform for Finite Discrete Images," Proc. National Electronics Conference, Chicago, Illinois, October 1974.

7. R.M. Haralick, N. Grisworld, N. Kattiyabulwanich, "A Fast Two-Dimensional Karhunen-Loeve Transform," Proceedings of SPIE, Vol. 66, pp. 144-159, August 1975.

8. R.M. Haralick, K. Shanmugam, "Comparative Study of Discrete Linear Basis for Image Data Compression," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 4, January 1974, pp. 16-27.

9. A. Habibi, R.S. Hershel, "A Unified Representation of Differential Pulse-Code Modulation (DPCM) and Transform Coding Systems," IEEE Transactions on Communications, Vol. COM-22, No. 5, pp. 692-696, May 1974.

10. R.B. Blackman, and J.W. Tukey, "Measurement of Power Spectra from the Point of View of Communications Engineering," Dover, New York, 1959.

11. M. Tasto, P.A. Wintz, "Image Coding by Adaptive Block Quantization," IEEE Transactions on Communication Technology, Vol. COM-19, No. 6, pp. 957-971, December 1971.

12. G.M. Dillard, "Application of Ranking Techniques to Data Compression for Image Transmission," NTC 75 Conference Record, Vol. 1, pp. 22-18 to 22-22.

13. H.J. Landau, D. Slepian, "Some Computer Experiments in Picture Processing for Bandwidth Reduction," Bell System Technical Journal, Vol. 50, No. 5, pp. 1525-1540, May-June 1971.

14. G.B. Anderson, T.S. Huang, "Piecewise Fourier Transformation for Picture Bandwidth Compression," IEEE Transactions on Communication Technology, Vol. COM-19, pp. 133-140, April 1971.

15. E.J. Claire, "Bandwidth Reduction in Image Transmission," ICC'72 Conference Record, pp. 39-8 to 39-13.

16. J.I. Gimlett, "Use of Activity Classes in Adaptive Transform Image Coding," IEEE Transactions on Communication, Vol. COM-23, No. 7, pp. 785-786, July 1975.

17. C. Reader, "Intraframe and Interframe Adaptive Transform Coding," Proceedings of SPIE, Vol. 66, pp. 108-118, August 1975.

18. S.C. Knauer, "Real Time Video Compression Algorithm for Hadamard Transform Processing," Proceedings of SPIE, Vol. 66, pp. 58-69, August 1975.

19. A.G. Tescher, "The Role of Phase in Adaptive Image Coding," Ph.D. Thesis, University of Southern California, Electrical Engineering Department, January 1974.

20. A.G. Tescher, H.C. Andrews, and A. Habibi, "Adaptive Phase Coding in Two and Three-Dimensional Fourier and Walsh Image Compression," 1974 Picture Coding Symposium, Goslar, Germany, August 26-28.

21. R.V. Cox, A.G. Tescher, "Generalized Adaptive Transform Coding," 1976 Picture Coding Symposium, Asiloma, California, January 28-30.

22. W.B. Schaming, "Digital Image Transform Coding," PE-622, Internal Memorandum, RCA Corporation, 1974.

23. J.E. Abate, "Linear and Adaptive Delta Modulation," Proceedings of IEEE (special issue on Redundancy Reduction), Vol. 55, No. 3, pp. 298-308, March 1967.

24. N.S. Jayant, "Adaptive Delta Modulation with a One-Bit Memory," BSTJ, Vol. 49, No. 3, pp. 321-342, March 1970.

25. R.H. Bosworth, and J.C. Candy, "A Companded One-Bit Coder for PICTUREPHONE Transmission," BSTJ, Vol. 48, No. 5, pp. 1459-1479. May 1969.

26. A. Habibi, "Delta Modulation and DPCM Coding of Color Signals," Proceedings of the International Telemetry Conference, Vol. 8, Los Angeles, pp. 333-343, October 10-12, 1972.

27. C.L. Song, J. Garodnick, D.L. Schilling, "A Variable Step Size Robust Delta-Modulator," IEEE Transactions on Communication, Vol. COM-19, pp. 1033-1099, December 1971.

28. N.R. Scheinberg, D.L. Schilling, M.Z. Ali, I. Paz, "A Technique for Correcting Transmission Errors in Video Delta Modulation Channels," ICC'75 Conference Record, Vol. 2, pp. 27-21 to 27-25.

29. M.R. Winkler, "High Information Delta Modulation," IEEE International Conv. Rec., pt. 8, pp. 260-265, 1963.

30. T.A. Hawkes, P.A. Simonpieri, "Signal Coding Using Asynchronous Delta Modulation," IEEE Transactions on Communication, Vol. COM-22, No. 3, pp. 346-348, March 1974.

31. B.S. Atal, M.R. Schroeder, "Adaptive Predictive Coding of Speech Signals," BDTJ, Vol. 49, pp. 1968-1973, 1970.

32. K. Virupaksha, J.B. O'Neal, Jr., "Entropy-Coded Adaptive Differential Pulse-Code Modulation (DPCM) for Speech," IEEE Trans. on Comm., Vol. COM-22, No. 6, pp. 777-787, June 1974.

33. P.J. Ready, D.J. Spencer, "Block Adaptive DPCM Transmission of Images," NTC'75 Conference Record, Vol. 2, pp. 22-10 to 22-17.

34. A.H. Frie, H.R. Schindler, P. Vettiger, "An Adaptive Dual-Mode Coder/Decoder for Television Signals," IEEE Trans. on Comm. Tech., Vol. COM-19, No. 6, pp. 933-943, December 1971.

35. W.K. Pratt, "Dual Mode DPCM/Delta Modulation Image Coding System for Real Time Television Transmission," 1974 Picture Coding Symposium, Goslar, Germany, August 26-28.

36. C.C. Cutler, "Delayed Encoding: Stabilizer for Adaptive Coders," IEEE Trans. on Comm. Tech., Vol. COM-19, No. 6, pp. 898-907, December 1971.

37. W. Kaminski, E.F. Brown, "An Edge Adaptive Three-Bit Ten-Level Differential PCM Coder for Television," IEEE Trans. on Comm. Tech., Vol. COM-19, No. 6, pp. 944-947, December 1971.

38. L.H. Zetterberg, "Adaptive Schemes with Delayed Decision," IEEE Trans. on Comm., Vol. COM-22, No. 9, pp. 1195-1198, September 1974.

39. J.N. Gupta, P.A. Wintz, "A Boundary Finding Algorithm and its Applications," IEEE Trans. on Circuits and Systems, Vol. CAS-22, No. 4, pp. 351-362, April 1975.

40. E.E. Hilbert, "Joint Classification and Data Compression of Multi-dimensional Information Source Application to ERTS," ICC'75, Conference Record, Vol. II, pp. 27-1 to 27-6.

41. E.E. Hilbert, "Joint Pattern Recognition Data Compression Concept for ERTS Multispectral Imaging," Proceedings of SPIE, Vol. 66, pp. 122-137, August 1975.

42. L.C. Wilkins, P.A. Wintz, "A Contour Tracing Algorithm for Data Compression for Two-Dimensional Data," School of Elect. Engr., Purdue University, W. Lafayette, Indiana, Tech. Report TR-EE 69-14, 1969.

43. A. Habibi, A.S. Samulon, "Bandwidth Compression of Multispectral Data," Proceedings of SPIE, Vol. 66, pp. 23-35, August 1975.

44. L.D. Davisson, "Universal Noiseless Coding," IEEE Trans. on Information Theory, Vol. 19, pp. 783-795, November 1973.

45. C.L. May, D.J. Spencer, "ERTS Image Data Compression Technique Evaluation," Final Report for NASA Contract NAS5-21746, April 1974.

46. R.F. Rice, "An Advanced Imaging Communication System for Planetary Exploration," Proceedings of SPIE, Vol. 66, pp. 70-89, August 1975.

47. R.F. Rice, J.R. Plaunt, "Adaptive Variable-Length Coding for Efficient Compression of Space Craft Television Data," IEEE Trans. on Comm. Tech., Vol. COM-19, No. 6, pp. 889-897, December 1971.

48. A. Habibi, "Delta Modulation and DPCM Coding of Color Signals," Proceedings of 1972 International Telemetering Conference, October 1972, pp. 333-343.

49. C.L. May, D.J. Spencer, ERTS Image Data Compression Technique Evaluation, TRW Final Report, NAS5-21746, April 1974.

50. H.J. Whitehouse, J.M. Speiser, and R.W. Means, "High-Speed Serial Access Linear Transform Implementation," Presented at the All Applications Digital Computer (AADC) Symposium, Orlando, Florida, January 1973 (reprinted as NUC TN 1026).

51. A. Habibi, "A Note on the Performance of Memoryless Quantizers," 1975 NTC Records, Vol. 1, pp. 36 (16-22), December 1975.

52. A. Habibi, A.Y. Huang, "Classification Consistency of Bandwidth Compressed MSS Images Using Bayes Supervised Classifier," TRW IOC 77-7130-69817.09.