NASA
Technical
Paper
2458

c.2

August 1985

An Image Compression Survey and Algorithm Switching Based on Scene Activity

Michael M. Hart

NASA

NASA
Technical
Paper
2458

1985

An Image Compression Survey and Algorithm Switching Based on Scene Activity

Michael M. Hart

Langley Research Center
Hampton, Virginia

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

ABSTRACT

A comprehensive study of data compression techniques is presented in this paper. A description of these techniques is provided along with a performance evaluation. The complexity of the hardware resulting from their implementation is also addressed. The compression effect on channel distortion and the applicability of these algorithms to real-time processing are presented. Also included is a proposed new direction for an adaptive compression technique for real-time processing.

INTRODUCTION

The increase in resolution and in the number of spectral bands of modern multispectral imaging systems creates a tremendous burden on the down-link channel and the bandwidth required to transmit data to ground stations. In fact, the future imaging system will produce data at rates that will exceed the capability of the down-link channel. Data compression is one of the most powerful tools available to reduce the data volume to be transmitted. Eventually, data compression will be an essential part of modern telemetry systems.

Since most users insist on reversible processes, this paper focuses only on reversible data compression techniques and explores the possibility of their real-time implementation. Various reversible data compression techniques are described, and an evaluation of these techniques in terms of performance, implementation complexity, and immunity to channel noise is presented.

INFORMATION THEORY AND DATA COMPRESSION

The coding of the numerical data is accomplished by means of pulse code modulation (PCM) requiring, in general, a very large bandwidth. In fact, the number of pulses per second to be transmitted is a function both of the number of samples and of the number of bits necessary to represent each sample. To reduce this large number of pulses per second (and consequently the bandwidth), it is necessary to introduce data transformation represented by data or bandwidth compression. Such a transformation can be considered as one which operates on the data given by an information source in such a way as to reduce the amount of nonuseful or redundant data. Since compressed data are, in general, more sensitive than noncompressed data to the channel noise, a channel encoding might be necessary for a noisy channel. An error in the compressed data will generally introduce a considerable amount of distortion.

Entropy

The entropy is defined as the amount of information that is emitted by a data source. The theoretical basis of data compression depends on Shannon's first theorem on the noiseless coding of information. Given a zero memory source S emitting the symbols s_i ($i = 1, 2, \ldots, n$) with the corresponding (independent) probabilities P_i, we can calculate the entropy of the source under the above conditions as

$$H(S) = \sum_{i=1}^{n} P_i \log_2 \frac{1}{P_i} \tag{1}$$

Each of the symbols $s_1, s_2, \ldots, s_n$ can be mapped into a fixed sequence of k symbols taken from a finite alphabet $\mathbf{X} = x_1, x_2, \ldots, x_k$. This procedure corresponds to encoding each symbol s_i into a code word x_i belonging to the set $x_1, x_2, \ldots, x_n$ and having length l_i. We can define the average length of this code $\bar{L}$ as

$$\bar{L} = \sum_{i=1}^{n} P_i l_i \tag{2}$$

Such a code is said to be compact for that source if its average length is less than or equal to that of any uniquely decodable code.

From equations (1) and (2), the following property of $H(S)$ can be proved:

$$H(S) \le \bar{L} \tag{3}$$

Hence, $H(S)$ is a lower bound for code average length. The ratio

$$\eta = \frac{H(S)}{\bar{L}}$$

is defined as the efficiency of the source code, and $1 - \eta$ is the redundancy. The term $H(S)$ can be used to evaluate an upper bound for the mean compression ratio

$$\mathrm{CR} = \frac{\bar{L}_s}{\bar{L}} \tag{4}$$

The symbols $\bar{L}_s$ and $\bar{L}$ represent the source and encoded mean word lengths, respectively.

From equations (3) and (4) we can obtain the maximum compression ratio $\mathrm{CR}_{\mathrm{max}}$ as

$$\mathrm{CR}_{\mathrm{max}} = \frac{\bar{L}_s}{H(S)} \tag{5}$$

Equations (3) through (5) are true if the compression method is perfectly reversible.

A higher value for CR than that given by equation (5) can only be obtained by introducing a certain amount of distortion in the reconstructed signal. In the latter case, the process is said to be irreversible. In this paper, only reversible compression methods are considered.

Channel Capacity

The entropy (eq. (1)) can be considered as the average information associated with the emission of a source symbol. Let the output alphabet reproducing the source be B with r symbols. Then $B = \{b_j\}$ where $j = 1, 2, \ldots, r$, and $P(b_j)$ is the probability of b_j. Mutual information can be defined as a function of the source symbols $\{S_i\} \in S$ and of the received symbols $\{b_j\} \in B$ by

$$I(S,B) = \sum_{S,B} P(s_i, b_j) \log_2 \frac{P(s_i, b_j)}{P(s_i)P(b_j)} \qquad (6)$$

and it represents the average information obtained from the emission of a symbol s_i when b_j is known. The mutual information $I(S,B)$ is a nonnegative convex function of the probabilities $P(s_i)$ and always admits a maximum. This maximum, taken over all the possible choices of the source probability distribution $P(s_i)$, is called the channel capacity C, where

$$C = \max_{\{P(s_i)\}} I(S,B) \qquad (7)$$

In fact, if $H(S) < C$, it is always possible to find a channel coding method for transmission on a noisy channel, such that the error probability at the receiver is lower than an arbitrary small quantity. However, this could imply the use of a prohibitively long code, which is not of practical usefulness.

Rate Distortion Function

Let a vector $\mathbf{X}$ with n components of the source alphabet $\{x_1, x_2, \ldots, x_n\}$, $x_i \in S$, be encoded in a vector $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$ with $y_i \in B$. We denote the word distortion measure by $D_n(\mathbf{X}, \mathbf{Y})$, which could be expressed as the cost of sending x_i and receiving y_j where $i \neq j$.

The function $D_n(\mathbf{X}, \mathbf{Y})$ is defined by the user. An often-used measure of distortion is the single-letter fidelity criterion, where $D_n(\mathbf{X}, \mathbf{Y})$ is the mean of the single distortions introduced by representing x_i with y_i; i.e.,

$$D_n(\mathbf{X}, \mathbf{Y}) = \frac{1}{n} \sum_{i=1}^{n} D(x_i, y_i) \qquad (8)$$

For channels with memory, more complex definitions are needed to measure distortion, and in general, these are very difficult to deal with.

In many cases, equation (8) is used as a first approximation for systems with memory. From equation (8), the overall average distortion $\bar{D}$ will depend on the conditional probability $P(y_i|x_i)$ and is given by

$$\bar{D} = \sum_{S,B} P(x_i)P(y_i|x_i)D(x_i, y_i) \qquad (9)$$

when $\bar{D}$ turns out to be less than a preset quantity D, $P(y_i|x_i)$ is called D-admissible. Now we can define the rate distortion function $R(D)$ as the minimum of the average mutual information

$$R(D) = \min_{\left\{ \substack{P(y_i|x_i) \\ D\text{-admissible}} \right\}} \sum_{S,B} P(x_i)P(y_i|x_i) \log \frac{P(y_i|x_i)}{P(y_i)} \qquad (10)$$

where the minimum is taken over all the possible $P(y_i|x_i)$ values that are D-admissible.

REVERSIBLE DATA COMPRESSION TECHNIQUES

Reversible data compression techniques include redundancy reduction, differential pulse code modulation (DPCM), and linear transformation (refs. 1 - 4). The various algorithms used in these techniques are summarized in figure 1.

Redundancy Reduction

The redundancy reduction method is based on whether a data point could be successfully determined within a preset tolerance of the actual point. Predictions and interpolations are carried out according to the following difference polynomial:

$$Y_t' = Y_{t-1} + \Delta Y_{t-1} + \Delta^2 Y_{t-1} + \ldots + \Delta^n Y_{t-1}$$

where

Y_t'	predicted data sample at time t
Y_{t-1}	sample one period prior to t
ΔY_{t-1}	$= Y_{t-1} - Y_{t-2}$
$\Delta^2 Y_{t-1}$	$= \Delta Y_{t-1} - \Delta Y_{t-2}$
$\Delta^n Y_{t-1}$	$= \Delta^{n-1} Y_{t-1} - \Delta^{n-1} Y_{t-2}$

The basic difference between predictors and interpolators is that predictors use only past samples to predict the present one, whereas interpolators use both past and future samples. Comparison of various degrees of difference polynomials has shown that above a third-degree polynomial, there is little or no improvement in performance (ref. 5). It is the author's opinion that the improvement from first degree to second or third degree does not justify the added complexity. (See fig. 2.) Hence, the evaluation in this paper is done on zero- and first-order polynomials.

Prediction Algorithms

The following compression algorithms predict the present sample by using a difference polynomial. If

$$|Y_t - \hat{Y}_t| \leq K \tag{11}$$

where

Y_t actual sample value

$\hat{Y}_t$ predicted value

K tolerance band

then the sample is not transmitted. The process continues until the condition in equation (11) fails, then the actual sample is transmitted with a code appended to inform the ground station of the number of samples that were not transmitted. At the ground station, the decompressor fills in the samples that were not transmitted by using the same polynomial that was used at the compressor. The compressor and decompressor can use one of the following polynomials: (1) zero-order predictor, (2) zero-order predictor with an offset, (3) first-order predictor, (4) first-order predictor with a slope correction, or (5) optimum linear predictor. These polynomials are discussed in the sections which follow.

Zero-order predictor. In the zero-order predictor (ZOP) algorithm, it is always assumed that

$$\hat{Y}_t = Y'_{t-1} \tag{12}$$

where

$\hat{Y}_t$ sample to be predicted at time t

Y'_{t-1} actual transmitted sample or previous successfully predicted sample

A graph illustrating this algorithm is shown in figure 3. At time t, we can see that there will be no transmission, since the tolerance band (the two dashed lines) contains the previous sample point. Whereas, at $t+2$ the actual data point is transmitted, since it falls outside the tolerance band placed on the predicted sample.

Zero-order predictor with offset. The zero-order predictor with an offset is basically the same as the ZOP. The prediction polynomial is $\hat{Y}_t = Y'_{t-1}$ as long as a sample is not transmitted (redundant). Once a sample is transmitted, the first point in the next interval is offset as follows:

$$\hat{Y}_t = Y_{t-1} + |\delta|\text{sgn}(Y_{t-1} - Y'_{t-2}) \tag{13}$$

where

Y_{t-1} sample at $t-1$

Y'_{t-2} sample at $t-2$ (If the sample is transmitted, the actual sample is used. If not, the predicted value is used.)

$|\delta|$ magnitude of offset

These two algorithms are relatively simple to implement; however, they perform best when the actual data vary very slowly.

First-order predictor. The first-order predictor (FOP) algorithm is similar to the above algorithms except that the predictor uses a first-order polynomial

$$Y_t = Y'_{t-1} + \Delta Y_{t-1} = 2Y'_{t-1} - Y'_{t-2} \tag{14}$$

A graph illustrating the FOP is shown in figure 4. The implementation complexity for this algorithm is still low, and it performs well with data that vary at a medium rate.

First-order predictor with slope correction. The difference between the first-order predictor with a slope correction and the algorithms previously described is in the prediction polynomial. In this algorithm, as long as $|\sum| < K$ (the tolerance band), then

$$\hat{Y}_t = Y'_{t-1} + \Delta Y_\tau \tag{15}$$

where

$\sum \quad = \hat{Y}_{t-1} - Y_{t-1}$

ΔY_τ increment in Y, computed τ sample periods prior to $t-1$

τ number of sample periods between $t-1$ and time of prior transmission

If $|\sum| \geq K$, then

$$\hat{Y}_t = Y'_{t-1} + \Delta Y_0 \tag{16}$$

where

$$\Delta Y_0 = \Delta Y_\tau + \frac{K\text{sgn}\sum}{\tau} + \frac{\sum - K\text{sgn}\sum}{c}$$

and $c = 2$ if $\tau > 1$; $c = 1$ if $\tau = 1$. The implementation complexity of this algorithm is medium. This algorithm can handle data that are more active, and it follows the data slope faster than the previously discussed predictors.

Optimum linear predictor. The optimum linear predictor algorithm predicts the present sample by using a linear combination of past samples

$$\hat{Y}_t = \sum_{k=1}^{N} a_k Y_{t-k} \tag{17}$$

The coefficients a_k are chosen to minimize the mean square error between the predicted and actual values. These coefficients are found by solving N linear equations involving the autocorrelation function

$$\sum_{k=1}^{N} a_k R_y[(\tau - k)T] = R_y[(\tau + h)T] \quad (\tau = 1,\ 2,\ \ldots,\ N)$$

where

$R_y[(\tau - k)T]$ autocorrelation function of signal for lag of $(\tau - k)T$

h number of sample periods since last transmitted sample

T time between sampling

The implementation complexity for this algorithm is quite high.

Interpolation Algorithm

The interpolation algorithm approximates the data with a zero- or first-order polynomial curve. The best way to describe this method is by an example. First transmit Y_0 and Y_1, then approximate Y_2 by using equation (12) or (14). Determine if Y_2 is within $\pm K$ units of Y_0 and Y_1. If true, then approximate Y_3 as above and determine if Y_3 is within $\pm K$ of Y_0, Y_1, and Y_2. Keep repeating this process until the above condition fails; that is, Y_n is not within $\pm K$ of all the previous samples. When this happens, a starting point and an ending point of a line segment are transmitted. This line segment represents the points Y_0, Y_1, $\ldots$, Y_{n-1}. This process then continues with Y_n considered as Y_0 for the new line segment. Several methods exist for representing redundant samples by a straight-line segment. To achieve the largest compression, it is necessary to select a line segment within K units of as many samples as possible (where K is the desired tolerance). This optimum algorithm requires freedom of both the starting and ending points of the line and results in four degrees of freedom. Since the four-degree-of-freedom algorithm is an extremely complex process to implement, anchoring the starting point of the line segment to an actual or computed sample greatly simplifies the implementation. One way is to anchor the starting point of a new line to the end of the previous line. (This is called a joined-line segment.) Another way is to anchor the starting point of the line to the actual out-of-tolerance sample.

Differential Pulse Code Modulation (DPCM)

The general block diagram of a DPCM system is shown in figure 5. In this technique, a predicted sample $\hat{Y}_n$ is evaluated by using any of the prediction algorithms. The difference e_n between the actual sample and the predicted one is quantized. In basic DPCM, the uniform quantization of the e_n values may cause an edge degradation. However, if the correlation of the input signal is high and the prediction algorithm is efficient, DPCM generally offers a higher efficiency than PCM. In general, with an equal number of bits, the signal-to-noise ratio (SNR) is higher for DPCM than for PCM. With an equal SNR, DPCM requires a lower number of bits than PCM. The gain G in the SNR of DPCM with respect to PCM can be expressed by

$$G = \frac{E\{Y_n^2\}}{E\{e_n^2\}} = \frac{E\{Y_n^2\}}{E\{(Y_n - \hat{Y}_n)^2\}} \tag{18}$$

where $E\{Y_n^2\}$ and $E\{e_n^2\}$ are the variances of Y_n and e_n, respectively.

Minimizing the denominator of equation (18) will improve the gain. Basic DPCM does offer a better performance than PCM. Nevertheless, when nonstationary signals are processed, there may be large peak errors in the reconstructed data. To avoid these errors, many adaptive DPCM techniques were studied (refs. 6 9). In the adaptive DPCM, the step amplitude of the quantization interval changes to follow the signal evolution. The step value becomes small when the signal is quiescent and large when the signal is more active. However, with adaptive DPCM, the improvement obtained may become apparent only when large variations of the signal follow quiescent periods. In this case, the step amplitude can assume a very low value, and before it has time to become comparable with the difference signal, large errors can arise.

Linear Transformations

The transform coding technique uses a mathematical operator to transform the input image data into another domain, where the closely correlated input data are transformed, ideally, into uncorrelated data. The basic block diagram of a transform system is shown in figure 6. To reduce the data throughput, the image is divided into blocks of data for subsequent processing. Any errors that occur are averaged over a single block, and thus error propagation is reduced. Until the coder stage (see fig. 6), the process is completely reversible with no loss of information. The coder provides the data compression by selecting the various coefficients according to their significance until a preset threshold is met. Beyond this threshold, the remaining coefficients are discarded. This method uses a major property of the transform, by which the input image energy is compacted into a few coefficients. This enables the least important coefficients to be deleted without a large increase in the error of the reconstructed image. Ideally, the thresholding should be adaptive by sending more information when there is higher activity in a block of data (e.g., at an edge of an object). The inverse transformation uses the same information as the forward transformation to reconstruct the image. The advantages of this method over other methods are less sensitivity to variations between differing images and superior coding at the lower bit rates. The main disadvantages are some blurring at the edges and a certain loss of details in the image caused by the loss of high spatial frequencies. In addition, the hardware implementation of these transforms is complex, and the computations are time consuming.

Mathematically, the transform can be expressed as a summation over the dimensions considered. For an $N \times M$ image array $f(x,y)$, the two-dimensional transformed array $P(u,v)$ is given by

$$P(u,v) = \sum_{x=1}^{N} \sum_{y=1}^{M} f(x,y) O(x,y:u,v) \qquad (19)$$

where the operator kernel $O(x,y:u,v)$ represents a weighting constant, which is, in general, a function of both input and output image coordinates.

Similarly, the inverse transformation is given by

$$f(x,y) = \sum_{u=1}^{N} \sum_{v=1}^{M} P(u,v) O^{-1}(x,y:u,v) \qquad (20)$$

Fast algorithms do exist, and the implementation of these algorithms can greatly enhance the throughput of the compression system. Some of the most common operators and their properties are discussed in the following sections.

Karhunen-Loeve Transform

In the Karhunen-Loeve transform (K-LT), the matrix is found by first evaluating the covariance matrix of the image, which is of $N^2 \times N^2$ dimensionality for an $N \times N$ image. Then the eigenvectors of the covariance matrix are computed and used as the basis for the transformation matrix. These bases are unique for each data block. Because of the tremendous amount of computation, the K-LT is only used as a universal reference in the comparison of other transforms.

Discrete Fourier Transform

The main advantage of the discrete Fourier transform is the fast Fourier transform (FFT) introduced by Cooley and Tukey in 1965, which reduces the computation involved. Typically, the number of complex operations for an $N \times 1$ FFT is $N \log_2 N$ as compared with N^2 computations required in the conventional approach. The main disadvantage with this transform is that complex arithmetic is involved. The performance of this method is shown in figure 6 along with that of the other transforms discussed. The discrete Fourier transform is not very efficient at the lower values of block size but does improve as the block size is increased.

Discrete Cosine Transform (DCT)

The discrete cosine transform (DCT) provides the most promising performance of all the techniques because of its near-optimum mean-square-error performance. (See fig. 7.) The DCT is derived from the Fourier expression by taking the real parts of its exponential form. The two-dimensional forward transformation $F(u,v)$ for an $N \times N$ image $f(j,k)$ is given by

$$\begin{aligned} F(u,v) = {} & \frac{4C(u,v)}{N_2} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \\ & \times \frac{\cos(2j+1)u\pi \cos(2k+1)v\pi}{2N} \\ & (u,v = 0,\ 1,\ \ldots,\ N-1) \qquad (21) \end{aligned}$$

where $C(u,v) = 1/2$ for $u = v = 0$ and $C(u,v) = 1$ for $u,v = 1,\ 2,\ \ldots,\ N-1$. The DCT may be implemented by using a double-sized FFT or directly using a fast cosine transform devised recently by Chen et al. (ref. 10). As the block size increases to $N \geq 16$, the basis vectors of the transformation matrix approach the eigenvectors of a first-order Markov process correlation matrix, and hence the performance of the DCT approaches that of the K-LT.

Hadamard Transform

The basis vectors of the Hadamard transform are a series of rectangular waveforms taking the values of +1 or −1 only. This simplifies the hardware implementation, since the Hadamard transform does not require any multiplication.

The two-dimensional Hadamard transform for an $N \times N$ $f(x, y)$ array can be written in series form as

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)(-1)^{P(x,y:u,v)} \tag{22}$$

where

$$P(x, y : u, v) = \sum_{i=0}^{N-1} [u_i(x_i) + v_i(y_i)]$$

The terms u_i, v_i, x_i, and y_i are the binary representation of u, v, x, and y, respectively, for example

$$(u)_{\text{decimal}} = (u_{n-1} \quad u_{n-2} \quad \cdots \quad u_1 u_0)_{\text{binary}}$$

where $u_i \in \{0, 1\}$. The summation in the exponent in equation (22) is performed modulo two. The main disadvantage of this transform is that it is not as efficient in energy compaction as the previous transforms; thus, the compression ratio is degraded.

SOFTWARE EVALUATION OF COMPRESSION ALGORITHMS

The performance of each compression algorithm was assessed from the following criteria:

1. The scene activity that gives the best results in terms of lowest mean square error in reconstruction.
2. Compression efficiency.
3. Implementation complexity (the suitability for real-time implementation).
4. Immunity to channel noise. The comparison was done according to equation (9), and $D(\mathbf{X}, \mathbf{Y})$ was evaluated by using equation (8). The channel was assumed to be a binary symmetric channel.
5. Energy compaction property.
6. Additional information required for linear transformation.

Since the performance of most of the algorithms is sensitive to the scene correlation, Gaussian white noise data were generated and passed through a filter to introduce the desired correlation to the data. The filter equation is given by

$$Y_n = \alpha Y_{n-1} + (1 - \alpha) X_n \tag{23}$$

where

Y_n output of filter

Y_{n-1} previous output of filter

X_n input (Gaussian white noise data)

α parameter < 1 for stability criterion

Hence

$$H(\omega) = \frac{1 - \alpha}{1 - \alpha e^{-j\omega}} \tag{24}$$

where ω is the radian frequency and

$$|H(\omega)|^2 = \frac{(1 - \alpha)^2}{1 - 2\alpha(\cos\omega) + \alpha^2} \tag{25}$$

where $H(\omega)$ is the Fourier transform of the transfer function of the filter. Then the power spectra of the filter output are given by

$$S_y(\omega) = |H(\omega)|^2 S_x(\omega)$$

where $S_x(\omega)$ are the input power spectra. Since the input is white Gaussian, then

$$S_x(\omega) = \sigma_x^2$$

where σ^2 is the variance and

$$S_y(\omega) = \frac{(1 - \alpha)^2 \sigma_x^2}{1 - 2\alpha(\cos\omega) + \alpha^2} \tag{26}$$

Taking the inverse Fourier transform of equation (26) yields

$$R_y(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{(1 - \alpha)^2 \sigma_x^2}{1 - 2\alpha(\cos\omega) + \alpha^2} e^{-jn\omega} d\omega \tag{27}$$

Evaluating equation (27) by contour integration yields

$$R_y(n) = \frac{\alpha^n (1 - 2\alpha + \alpha^2)}{1 - \alpha^2} \sigma_x^2 \tag{28}$$

where $R_y(n)$ is the autocovariance of the filter output. Notice that $R_y(n)$ is a function of the filter parameter α. In evaluating the compression algorithms, we increased α from 0.1 to 0.9 in increments of 0.1, then from 0.9 to 0.99 in increments of 0.01.

The results of the software simulation are summarized in table I. The K-LT and the DCT transforms are best in accommodating all types of scene activity with the best efficiency in terms of mean square error and compression ratio. However, their implementation is complex. The Hadamard algorithm and (if bursts are ignored, since they occur infrequently) the adaptive DPCM algorithm can handle all types of scene activity; however, their efficiencies are not as good as those of the K-LT and the DCT. The predictors and interpolators, which are simple to implement, were best suited for a particular type of scene activity, as seen in table I.

ALGORITHM SWITCHING FOR DATA COMPRESSION

Because of the relatively low complexity associated with the implementation of the predictors and interpolators, these algorithms are attractive alternatives for implementation in imaging systems. However, the problem associated with them in terms of being able to handle only one type of data activity must be overcome. The author considered combining several predictors and interpolators that could handle various types of data activity with an activity-measuring scheme that would select the best algorithm to compress the data at hand. In order to implement this system, an analysis of the ZOP, the FOP, and the first-order interpolator (FOI) was performed to determine which range of data activity is best handled by each algorithm. The test was performed on data that had a wide range of activity. To achieve that activity range, the data were passed through the filter of equation (23), and α was varied from 0.1 to 0.99. (The symbol α represents the filter parameter which corresponds to different values of σ^2, as shown in equation (28), by letting $n = 0$.) A gain function was defined as a criterion to determine at which value of α the switching should occur for each algorithm. This gain function is defined as

$$G = \frac{\mathrm{CR}}{\epsilon^2}$$

where CR is the compression ratio and ϵ^2 is the mean square error. Figures 8, 9, and 10 show plots of the gain as a function of the filter parameter α for values of $K = 2$, 4, and 6, respectively. It can be seen from these plots that the FOI has the highest gain for values of $\alpha \leq 0.91$. For $\alpha > 0.91$, the ZOP provides the highest gain.

According to the above analysis, the compression system would include two compression algorithms: the ZOP and the FOI. The FOI would be used to compress data for values of $\alpha \leq 0.91$, then the system would switch to the ZOP for values of $\alpha > 0.91$. A general block diagram of that system is shown in figure 11. Notice that in such a system, the image is divided into subimages or segments, and a decision is made for every segment. Overhead information is sent at the beginning of each block to inform the ground station of which compressor was used over that block. The algorithm-switching aproach offers a considerable improvement over the use of one simple compression algorithm for all types of image data.

CONCLUDING REMARKS

A survey of various compression algorithms and an evaluation of their performance and implementation complexity were presented. It was shown that the more complex algorithms are able to handle all types of data, whereas algorithms which are simple to implement are best suited for a specific type of data activity. An approach has been presented and described which employs a measure of scene activity as a criterion to switch between various simplistic algorithms. This approach offered a considerable improvement over the use of one simple compression algorithm for all types of image data.

Further evaluation of other algorithms described in the survey (e.g., zero-order predictor with an offset, first-order predictor with a slope correction, and zero-order interpolator) is necessary to determine their applicability to the system and to optimize performance. Furthermore, other scene activity switching mechanisms (e.g., entropy) warrant additional investigation.

NASA Langley Research Center
Hampton, VA 23665
April 26, 1985

REFERENCES

1. Benelli, G.; Cappellini, V.; and Lotti, F.: Data Compression Techniques and Applications. *Radio & Electron. Eng.*, vol. 50, no. 1–2, Jan. Feb. 1980, pp. 29–53.
2. Kortman, C. M.: Redundancy Reduction—A Practical Method of Data Compression. *Proc. IEEE,* vol. 55, no. 3, Mar. 1967, pp. 253–263.
3. Andrews, C. A.; Davies, J. M.; and Schwarz, G. R.: Adaptive Data Compression. *Proc. IEEE,* vol. 55, no. 3, Mar. 1967, pp. 267–277.
4. Soame, T. A.: Bandwidth Compression Using Transform Techniques for Image Transmission System. *Marconi Rev.*, vol. 43, no. 219, 1980, pp. 228–240.
5. Habibi, Ali: Comparison of *n*th-Order DPCM Encoder With Linear Transformations and Block Quantization Techniques. *IEEE Trans. Commun. Technol.*, vol. COM-19, no. 6, pt. 1, Dec. 1971, pp. 948–956.
6. Xydeas, C. S.; and Steele, R.: Dynamic Ratio Quantiser. *Proc. Inst. Electr. Eng.*, vol. 125, no. 1, Jan. 1978, pp. 25–29.
7. Jayant, Nuggehally S.: Digital Coding of Speech Waveforms: PCM, DPCM and DM Quantizers. *Proc. IEEE,* vol. 62, no. 5, May 1974, pp. 611-632.
8. Cummiskey, P.; Jayant, N. S.; and Flangan, J. L.: Adaptive Quantization in Differential PCM Coding of Speech. *Bell Syst. Tech. J.,* vol. 52, no. 7, Sept. 1973, pp. 1105–1118.

9. Jayant, N. S.: Adaptive Quantization With a One-Word Memory. *Bell Syst. Tech. J.*, vol. 52, no. 7, Sept. 1973, pp. 1119-1144.

10. Chen, Wen-Hsiung; Smith, C. Harrison; and Fralick, S. C.: A Fast Computational Algorithm for the Discrete Cosine Transform. *IEEE Trans. Commun.*, vol. COM-25, no. 9, Sept. 1977, pp. 1004–1009.

TABLE I. COMPARISON OF DIFFERENT DATA COMPRESSION ALGORITHMS

Algorithm	Scene activity giving best results	Compression efficiency	Implementation complexity	Immunity to channel noise	Energy compaction property (a)	Additional information required (a)
Zero-order predictor	Nearly constant and slowly varying data	Very good	Low	Poor	N/A	N/A
Zero-order predictor with offset	Constant and slowly varying data	Good	Low	Poor	N/A	N/A
First-order predictor	Data varying at medium rate	Good	Low	Poor	N/A	N/A
First-order predictor with slope correction	Moderately active data	Good	Medium	Poor	N/A	N/A
Optimum linear predictor	Any nonburst data	Good	High	Poor	N/A	Auto-correlation
Zero-order interpolator	Nearly constant and slowly varying data	Very good	Medium	Poor	N/A	N/A
First-order interpolator	Any nonburst data	Very good	Medium	Poor	N/A	N/A
Basic DPCM	Constant and slowly varying data	Good	Low	Poor	N/A	N/A
Adaptive DPCM	Any nonburst data	Fair	Medium	Poor	N/A	N/A
Fast K-LT	Any type	Very good	Very high	Good	Optimum mean square error	Boundary values
DCT	Any type	Very good	High	Good	Approaches that of K-LT as block size increases	None
Hadamard	Any type	Fair	Medium	Good	Poor	None

[a] N/A indicates not applicable.

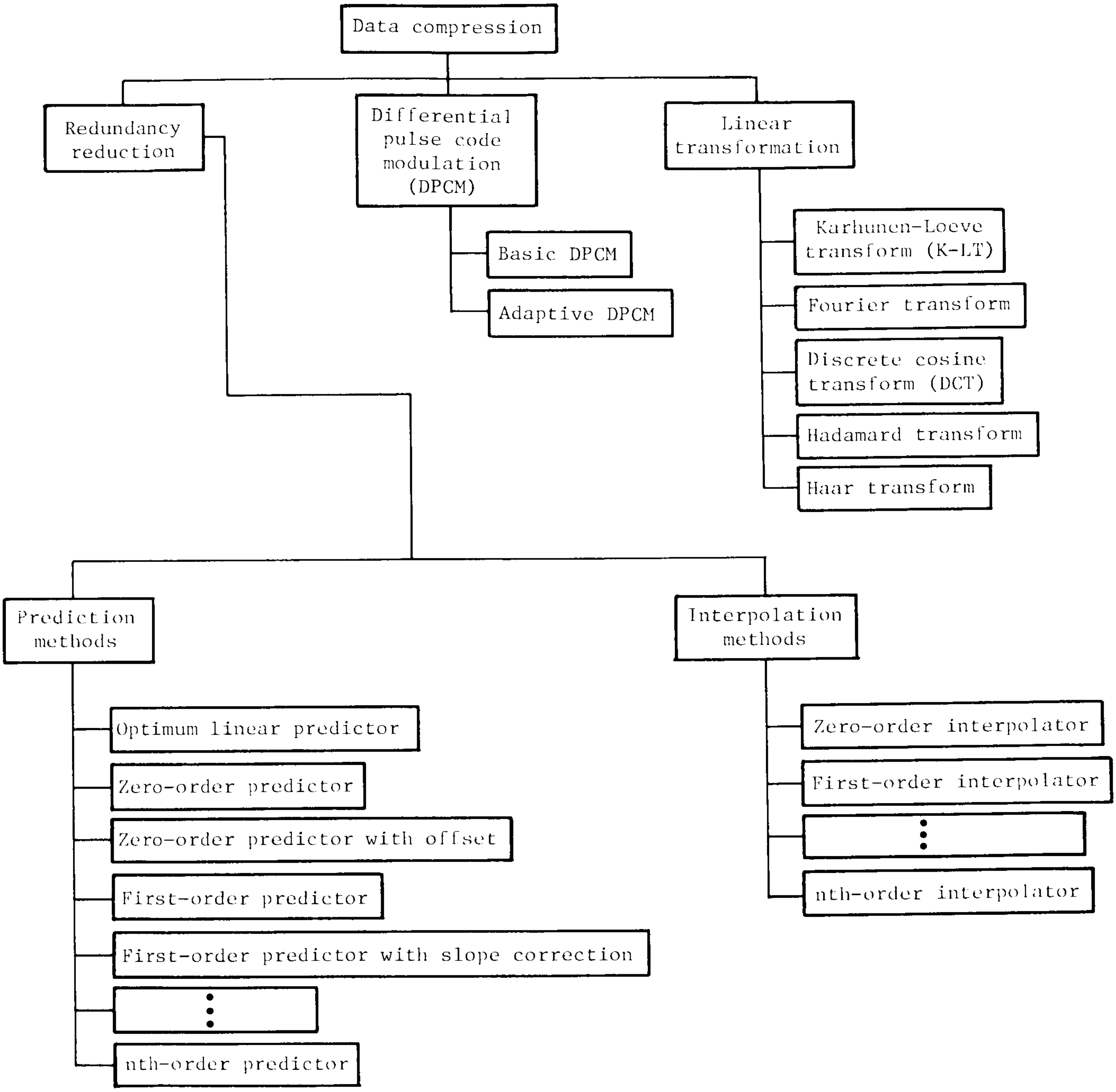

Figure 1. Reversible data compression techniques.

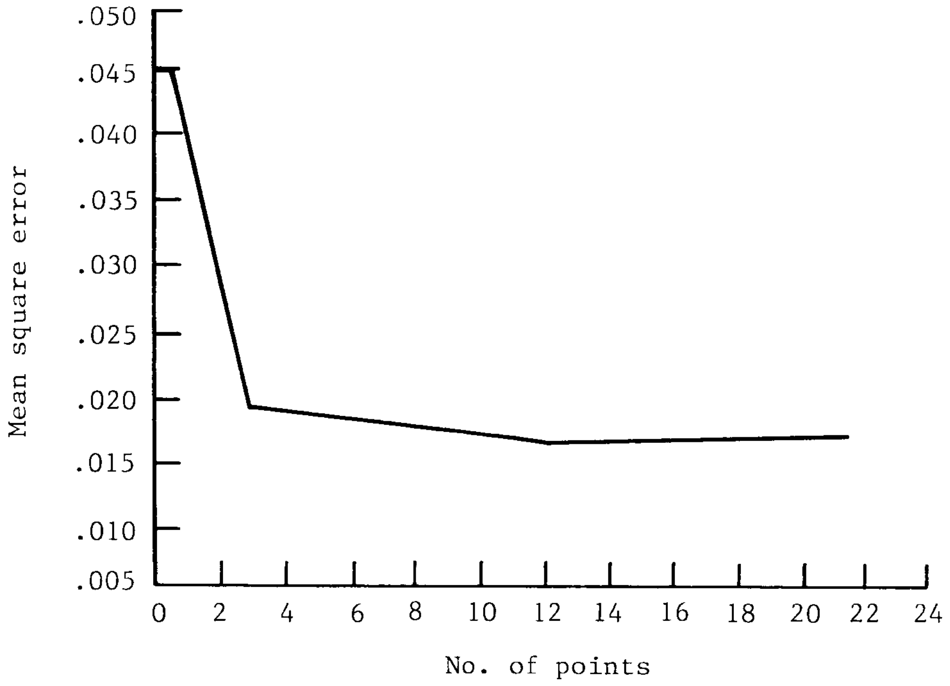

Figure 2. Mean square error versus number of points employed in predictor.

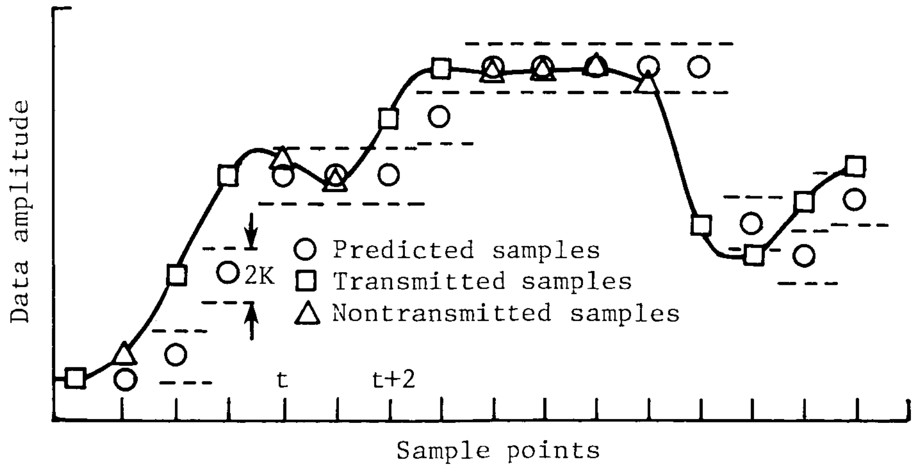

Figure 3. Data sampling and selection: zero-order predictor.

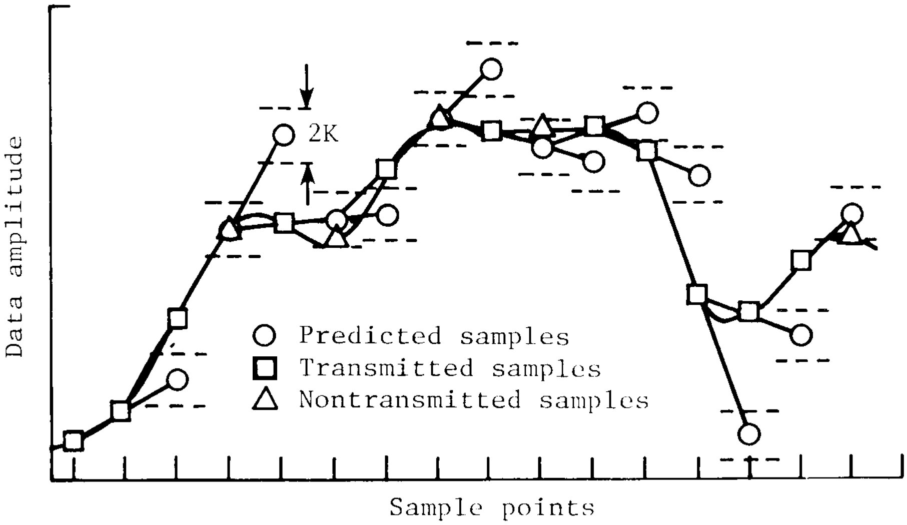

Figure 4. Data sampling and selection: first-order predictor.

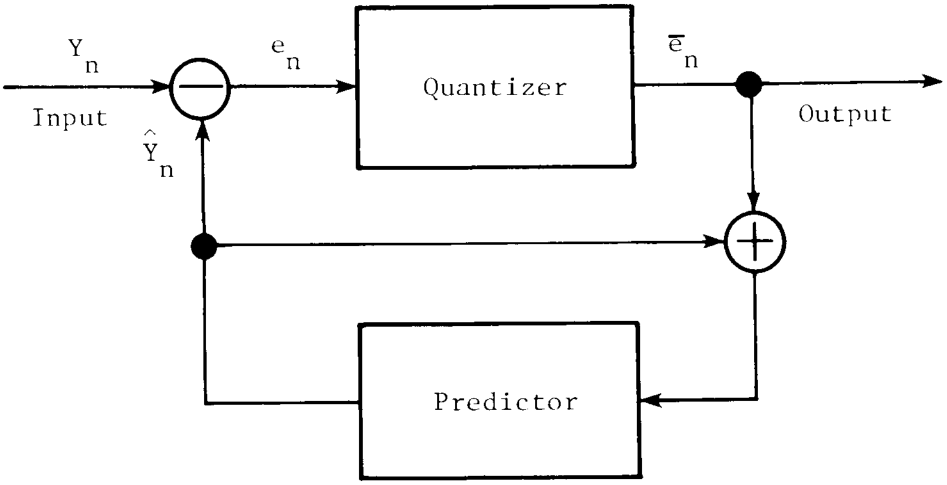

Figure 5. Block diagram of differential pulse code modulation system.

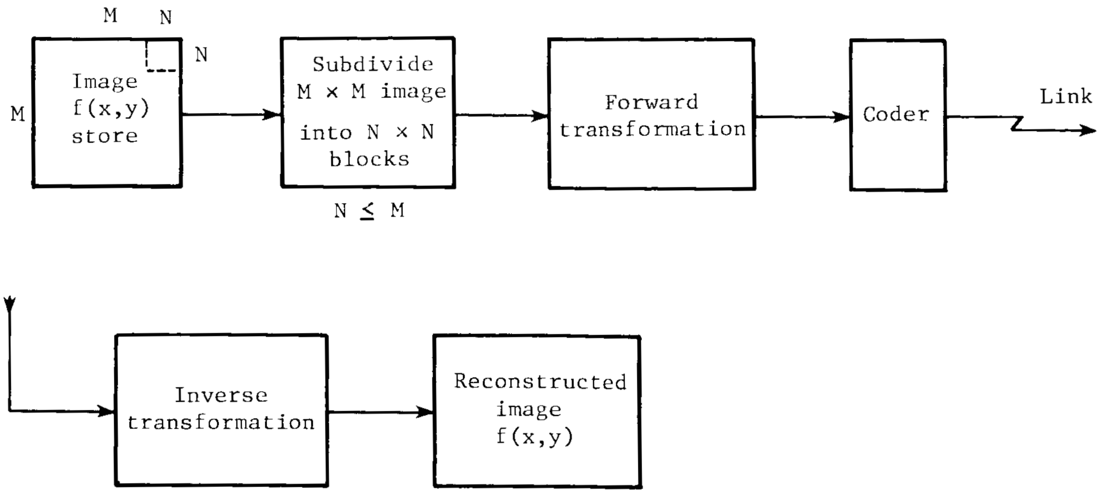

Figure 6. Block diagram of transform coding system.

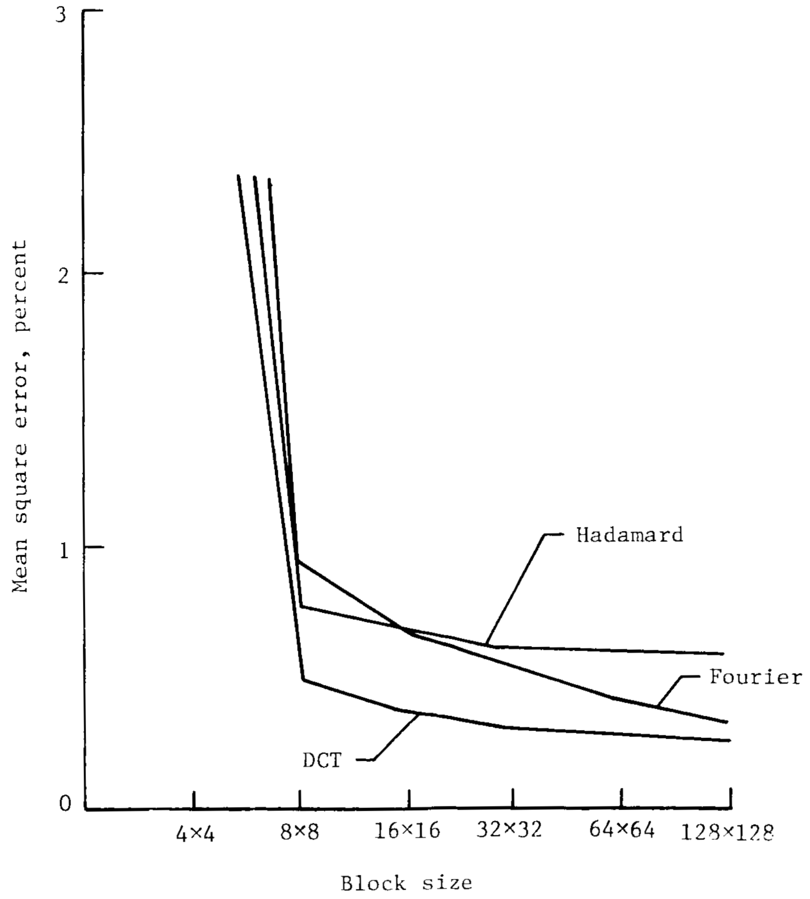

Figure 7. Mean-square-error performance of different transforms for two-dimensional Markov image source.

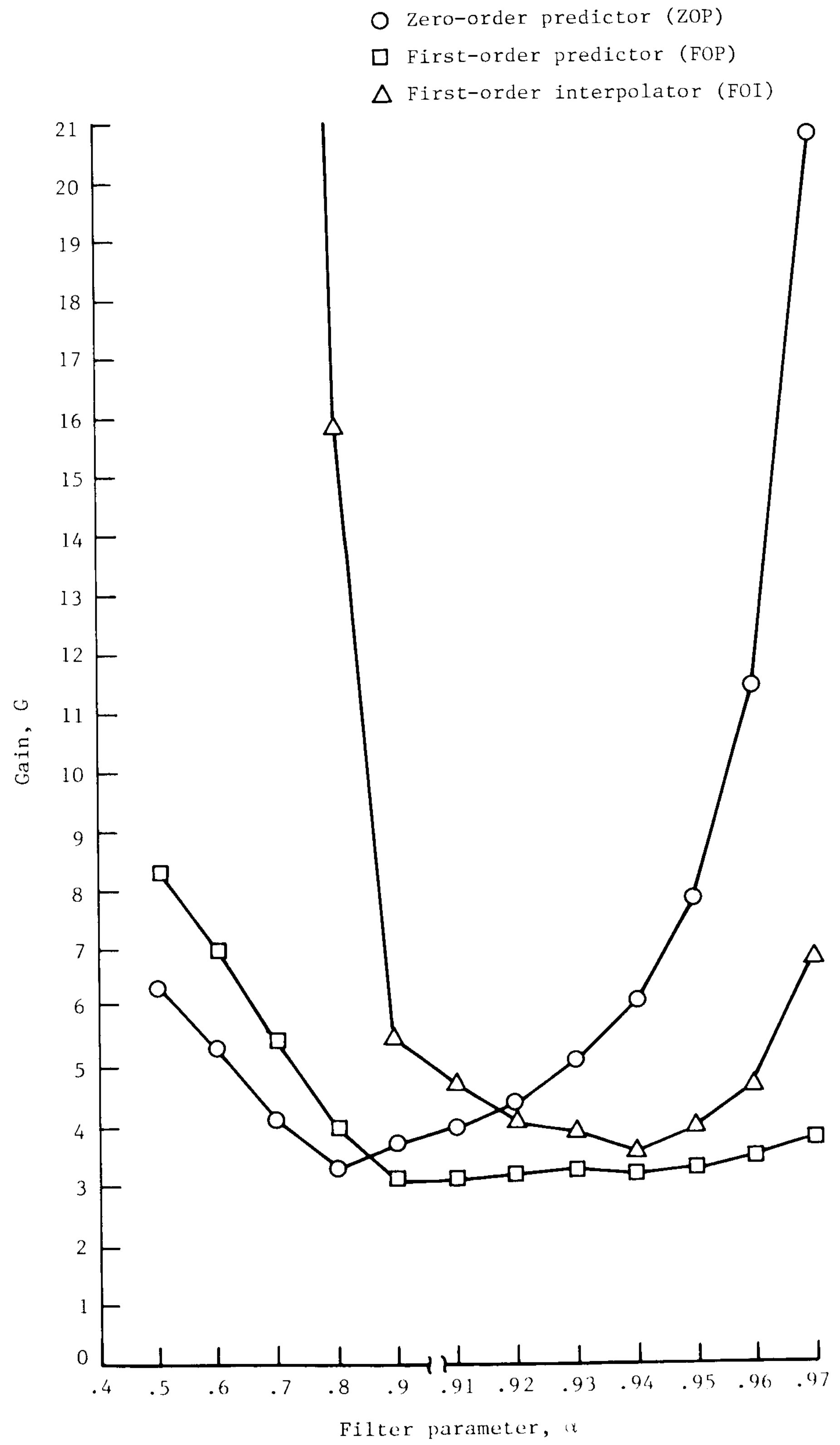

Figure 8. Gain G versus filter parameter α for $K = 2$.

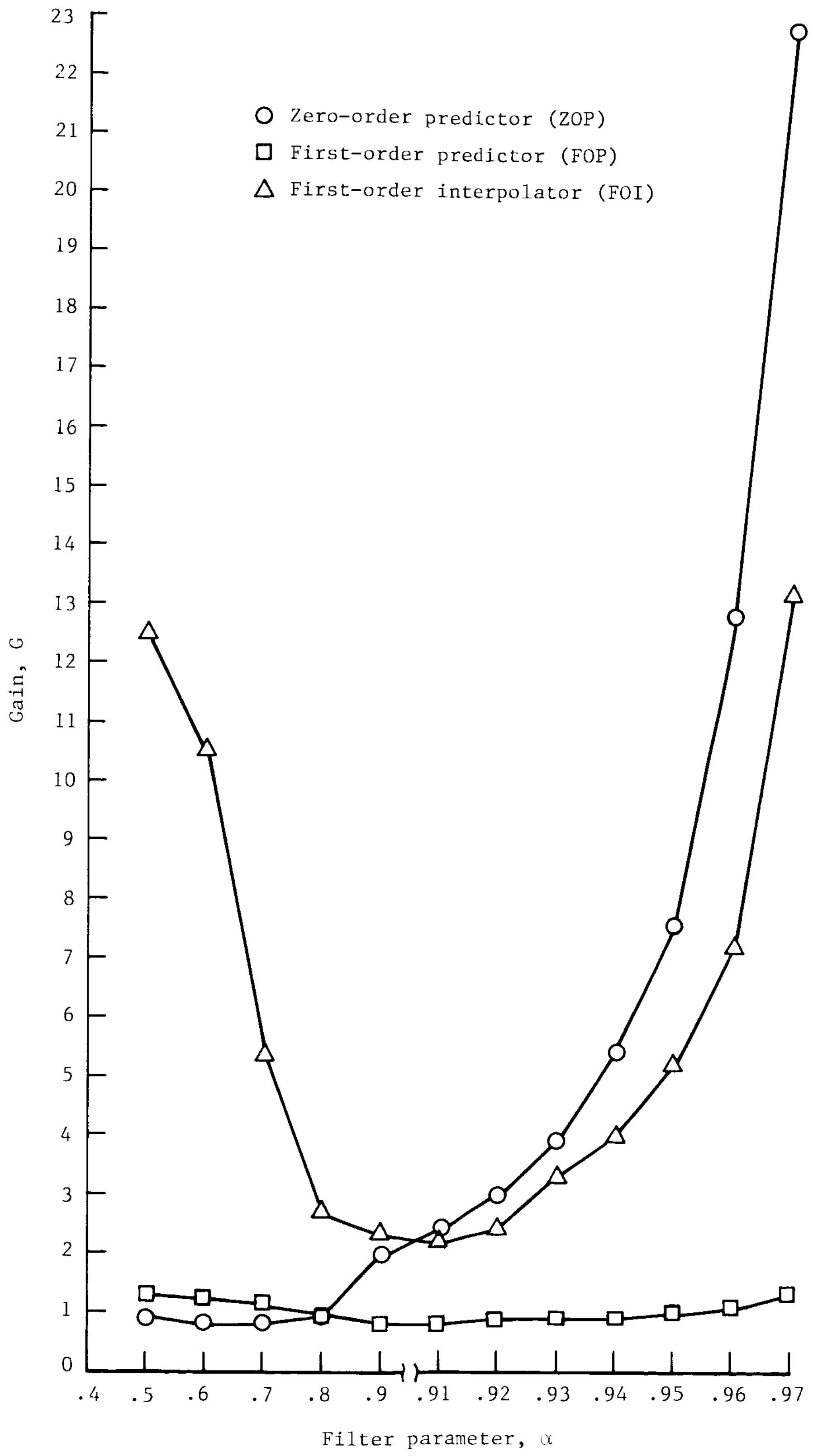

Figure 9. Gain G versus filter parameter α for $K = 4$.

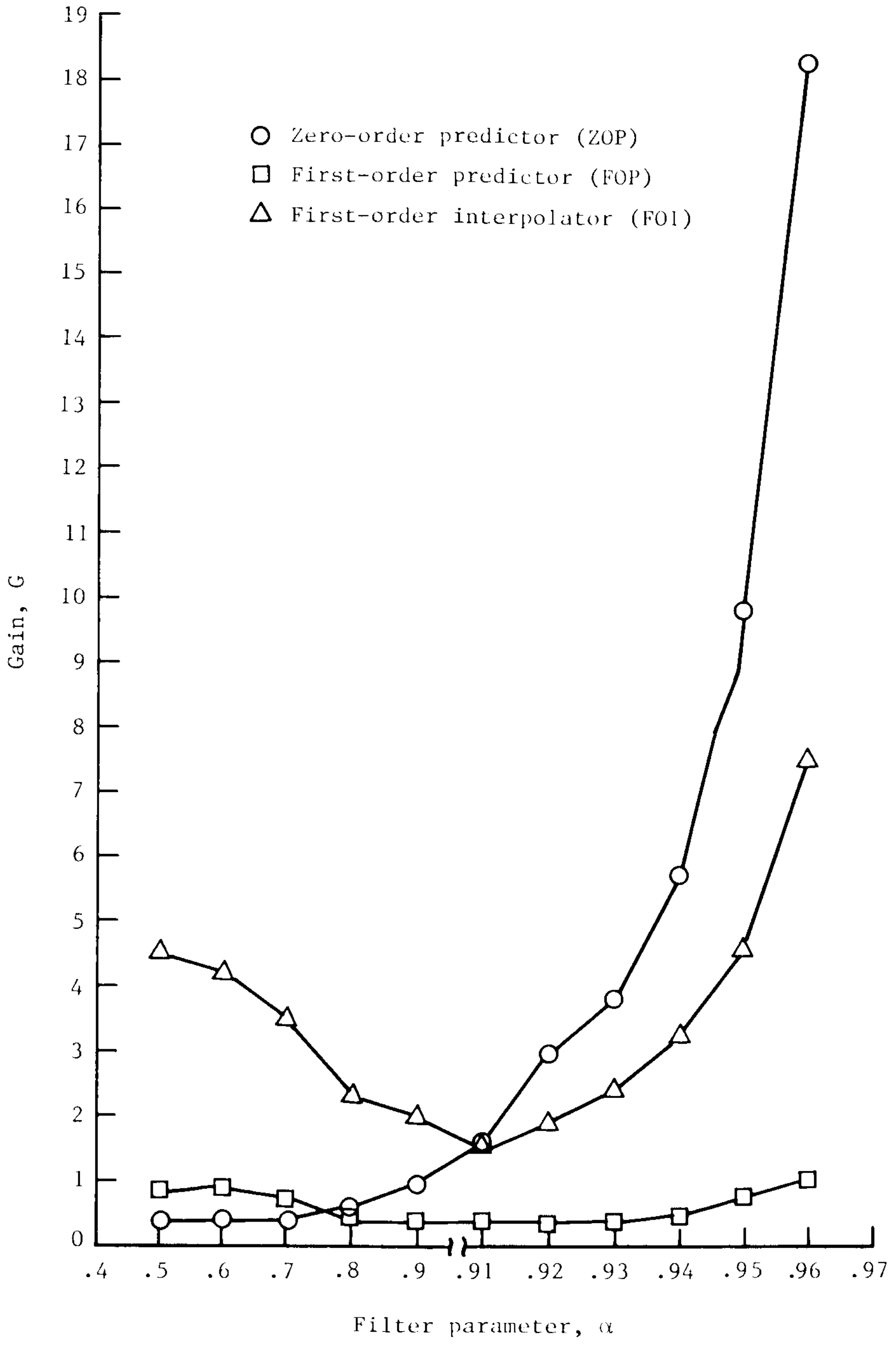

Figure 10. Gain G versus filter parameter α for $K = 6$.

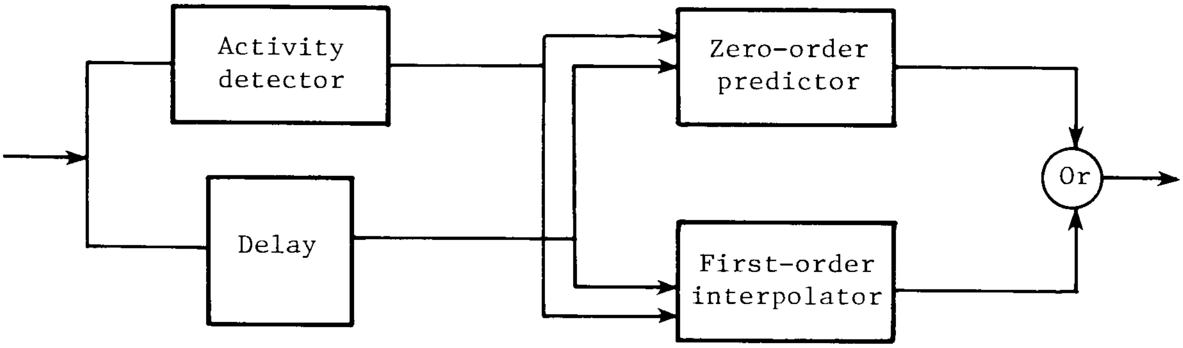

Figure 11. Block diagram of algorithm-switching system.

1. Report No.
NASA TP-2458

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
An Image Compression Survey and Algorithm Switching Based on Scene Activity

5. Report Date
August 1985

6. Performing Organization Code
506-58-13-02

7. Author(s)
Michael M. Hart

8. Performing Organization Report No.
L-15957

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, VA 23665

10. Work Unit No.

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546

13. Type of Report and Period Covered
Technical Paper

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A comprehensive study of data compression techniques is presented in this paper. A description of these techniques is provided along with a performance evaluation. The complexity of the hardware resulting from their implementation is also addressed. The compression effect on channel distortion and the applicability of these algorithms to real-time processing are presented. Also included is a proposed new direction for an adaptive compression technique for real-time processing.

17. Key Words (Suggested by Authors(s))
Reversible compression algorithm
Scene-activity-measuring algorithm
Switching compression system

18. Distribution Statement
Unclassified - Unlimited

Subject Category 32

19. Security Classif.(of this report)
Unclassified

20. Security Classif.(of this page)
Unclassified

21. No. of Pages
18

22. Price
A02

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 19