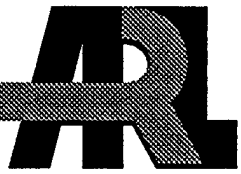
ARMY RESEARCH LABORATORY

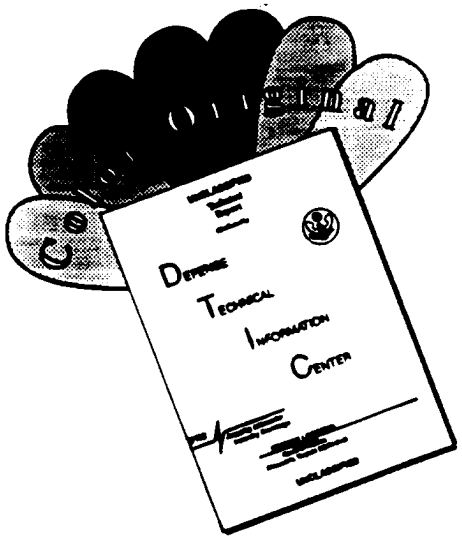


Robust Source Coding of Images With Predictive Trellis-Coded Quantization

Lisa M. Marvel

ARL-TR-1192 September 1996

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

NOTICES

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Final, Jul 95 - Jan 96

4. TITLE AND SUBTITLE

Robust Source Coding of Images With Predictive Trellis-Coded Quantization

5. FUNDING NUMBERS

PR: 611102.53A

6. AUTHOR(S)

Lisa M. Marvel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Laboratory
ATTN: AMSRL-IS-TP
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-1192

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The ability to transmit images over narrow bandwidth noisy channels has become desirable for many applications. Image integrity and timely transmission are imperative in many scenarios. The traditional method of image transmission requires a multistage process. The first stage is source coding, or the removal of redundancy, to compress the image for the narrow bandwidth channel. The second stage is channel coding, or the adding of redundant characters to protect the information from noise. This report pursues a method to perform robust source coding, providing both compression and noise mitigation. Specifically, Predictive Trellis Coding Quantization (PTCQ) incorporating various types of prediction filters is investigated. Trellis Coded Quanitization (TCQ) implies using an expanded set of quantization levels and the Viterbi algorithm to determine the minimal distortion path through a trellis, whose structure allows for low bit rate encoding. The prediction filter supplies correlation of the prediction differences and thereby provides the protection from noise at the decoder. PTCQ combines TCQ's encoding efficiency with predictive coding compression merits. Linear and nonlinear filter performance within the PTCQ scheme is shown under various channel conditions. Findings show that nonlinear filter implementation provides the highest noise immunity of those tested. The resulting algorithm is implementable in near real-time, allowing for the fast, efficient transmission of images over noisy channels.

14. SUBJECT TERMS	15. NUMBER OF PAGES
image coding, compression, noisy channel	67
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

Acknowledgments

This research would not be truly complete without recognizing those who were instrumental in its production. I have been extremely fortunate to have as my mentor Dr. A. Brinton Cooper of the U.S. Army Research Laboratory, whose motivation and encouragement has made the timely completion of this research possible. In addition, I would like to thank Richard Angelini of the Advance Simulation and High Performance Computing Directorate, U.S. Army Research Laboratory, for his assistance with providing the computational power necessary to perform simulations. I would also like to express my deep appreciation to my advisors, Dr. Ali Khayrallah of Ericsson, Inc., and Professor Charles G. Boncelet, Jr., of the University of Delaware, Department of Electrical Engineering, for their inspiration and guidance throughout the past years, and especially during the preparation of this report.

INTENTIONALLY LEFT BLANK.

1. Introduction

In this section the requirements for reliable image transmission over narrow bandwidth, noisy channels are presented. The compromise associated with source coding and channel coding as well as the traditional methods of image coding for transmission is then discussed. Finally, the subject research of this report is outlined.

1.1 Image Transmission

Many communication functions require the transmission of images over narrow bandwidth channels in a noisy environment. Radio, telephone, and satellite communications are all representative of narrow bandwidth channels. Noise can be generated by many sources, a few being electronic noise, cochannel interference, multipath interference, and intentional jamming. Applications for the capability of image transmission are limitless. They range from industrial to military applications including tele-medicine, digitization of the battlefield, weather services, and robotics. Of course one of the most difficult challenges is mobile communications, where restrictions are placed on the capabilities and size of the transmitter and the receiver. It is imperative that image integrity be preserved as much as possible during transmission. For instance, corruption of an image used for remote diagnosis could be a life-or-death situation. Of equal importance in wartime is a decision made from an image transmitted for identification of friend or foe to prevent an incident of friendly fire. The use of both compression and error protection is needed for image transmission over narrow bandwidth noisy channels.

The fact that images contain large amounts of data causes compression to be an important objective. In order to put into perspective the actual amount of data that an image contains, the following example is provided. An image obtained from a Charged Coupled-Device (CCD) camera and frame grabber contains 512 x 512 pixels. Each pixel has a dynamic range of 0-255, symbolizing gray scale values. This translates to 8 bits/pixel $\times$ 262,144 pixels for a total of 2,097,152 bits or approximately 2 megabits. For a narrow bandwidth channel of 16 kilobits/second, it would take a minimum of 2 minutes to transmit this image. In many applications this delay is neither practical nor acceptable. Therefore, it is necessary to compress the image data before transmission.

There are two categories of compression, lossless and lossy. Lossless compression implies that the reconstructed image will be an exact replica of the original. Two common types of lossless compression are Huffman coding and Ziv-Lempel coding. Lossy compression discards some nonredundant information to provide a greater reduction of data. This is accomplished at the cost of distortion in the reconstructed image. The Discrete Cosine Transform and Wavelet transform coding are lossy coding schemes. In many applications a certain amount of distortion is acceptable, in return for greater compression performance.

An image can become corrupted when transmitted over a noisy channel. Depending upon the form of the data during transmission, a small amount of noise can be catastrophic, causing the received image to be distorted such that it is rendered useless. If an image were

encoded utilizing a variable-length prefix code, a single bit error could cause the remaining data to be decoded incorrectly. Consequently, an effort must be made to prevent or minimize the effects of channel noise.

It is also desirable for the algorithm that processes the images to be of minimal computational complexity. In the application of mobile communications, this is particularly vital due to the constraints of the field transmitter/receiver. A less intensive algorithm allows real time or near real time processing and transmission of imagery. This aspect is very desirable in an image transmission system.

1.2 Joint Source Channel Coding

C. E. Shannon wrote his definitive paper on the mathematical theory of communication in 1948 [1]. In this paper, he defined the *capacity* of a channel to be the *maximum* rate, bits/symbol, at which information could be transmitted reliably through a channel. Shannon's source coding theorem states that if a channel is noiseless, it is possible to compact the information to the size of the source entropy, the minimum average word length in bits/symbol, by coding infinitely long extensions of the source. If the channel is noisy, the interest shifts from representing information as compactly as possible to encoding it so that reliable communication is possible. By sending information in a *redundant* form over a noisy channel, the probability that an error will occur can be reduced. Rate distortion theory states that the distortion incurred by the information is a decreasing function of the information rate. In addition, the rate is upper-bounded by the capacity of the channel. The capacity is a function of the channel bandwidth, noise properties of the channel and the intended signal. Therefore, one must compromise between rate and distortion over the noisy channel. This balance between source coding to minimize rate and channel coding to minimize distortion is dependent upon the application.

A source and channel coding system is depicted in Figure 1. In this example, $\mathbf{x}$ is the original image. It is compressed by the source encoder, which outputs $\mathbf{y}$, then protected from errors by the channel encoder, which produces $\mathbf{z}$. During transmission over a narrow bandwidth noisy channel, $\mathbf{z}$ is exposed to various forms of noise and enters the receiver as $\hat{\mathbf{z}}$. The channel decoder exploits the redundancy in $\hat{\mathbf{z}}$ to produce an estimate, $\hat{\mathbf{y}}$, of $\mathbf{y}$. The source decoder then constructs $\hat{\mathbf{x}}$, a possibly distorted version of $\mathbf{x}$.

It has been a long standing practice to treat source and channel coding separately. This practice was motivated by Shannon's joint source channel coding theorem [2]. This theory states that source coding followed by channel coding can be made to perform as well as any single- stage source channel coding procedure. In this scenario of separation, the source coding can be designed to suit the source, and the channel coding can be designed for good performance in a particular channel. The drawback to this method is high computational complexity and long delay. In contrast, joint source channel coding lessens the computational burden by performing both as a single process [3], as depicted in Figure 2.

In many ways source and channel coding form a duality. Source coding removes unstructured redundancy from the data — data compression — while channel coding introduces

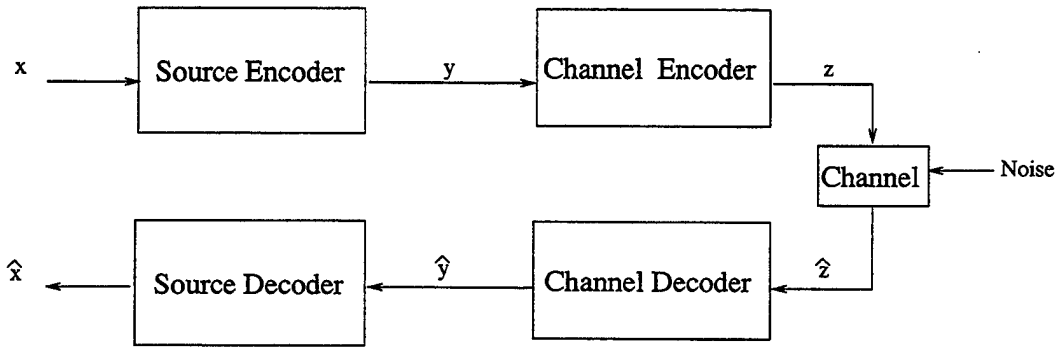


FIGURE 1.—*Traditional Coding System*

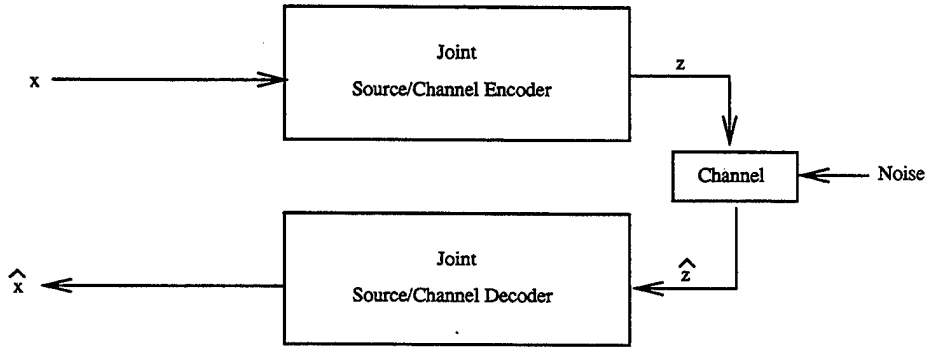


FIGURE 2.—*Joint Source/Channel Coding System*

structured redundancy to the data to combat errors caused by noise — error correction. If the source coding method is lossy, the cost for the compression is distortion. The expense for the protection provided by channel coding is more data and less distortion. The tradeoff one is willing to make between distortion and rate depends primarily on the application. As an example, little compression and high data protection may be desired for medical images transmitted on a wide bandwidth channel. Conversely, compression may be of the utmost concern for the transmission of satellite imagery over a narrow bandwidth channel. The ideal situation is efficient compression with low distortion, along with resilience to channel noise.

1.3 Subject Research

In this research, the *robust* source coding of images for transmission over noisy channels is presented. By providing source coding with good channel response, we address the source/channel coding problem using an algorithm of modest computational complexity. This coding is accomplished by utilizing Predictive Trellis-Coded Quantization (PTCQ) incorporating various prediction filters. The prediction filters are designed to mitigate the effect of errors caused by the the imperfect channel, thereby providing an image with low distortion. The structure of this report is as follows: essential knowledge is presented in sec-

tion 2, where background is provided for the Viterbi algorithm, Trellis-Coded Modulation, and predictive coding along with the design of optimal quantizers. Once these preliminaries have been addressed, section 3 delves into the details of PTCQ, and discusses modifications for the noisy channel and the incorporation of the linear and nonlinear filters. Section 4 presents the performance of PTCQ, incorporating the various prediction filters in both a noiseless and noisy environment. The PTCQ systems that provide good noise mitigation will be future evaluation for differing PTCQ configurations. Section 5 presents conclusions as well as recommendations for future work.

2. Preliminaries

Fundamental to the formulation of PTCQ are Viterbi decoding, TCM, predictive coding, and optimal quantizers. The PTCQ algorithm utilizes a trellis that is a redundant state machine mapped out in time. The use of a trellis, whose output is determined by a maximum likelihood algorithm called the Viterbi algorithm [4], results in low bit rate output with minimal distortion. The design of the trellis stems from the channel coding technique of TCM. PTCQ also incorporates the concept of lossy predictive coding to encode the image. Optimal quantizers designed using the Lloyd-Max algorithm are used to quantize the prediction error.

This section begins by explaining the trellis structure and its origin, then demonstrates how the Viterbi algorithm is used to find the path of minimum distortion through the trellis. TCM is then overviewed as the original channel coding dual from which the source coding method Trellis-Coded Quantization (TCQ) was derived. The theory and methods of predictive coding are then presented, followed by the design of optimal quantizers. The following section incorporates all of these techniques in order to formulate PTCQ.

2.1 Viterbi Algorithm

The Viterbi algorithm was developed in 1967 by Andrew Viterbi as a method for decoding convolutional codes [4]. A convolutional code is a discrete finite state Markov process that can be represented as a state transition diagram. The algorithm that optimally finds the shortest path through a directed graph is analogous to forward dynamic programming [5]. The recursive algorithm is a maximum likelihood decoding algorithm, which locates the most likely path through the encoder's state diagram and therefore minimizes the probability of error [6].

A convolutional code can be described by a state transition diagram, or state machine, as shown in Figure 3. In this figure the possible state progression is depicted by the directed graph. A more redundant description of the process can be constructed from this state machine by adding a time axis; the resulting structure is referred to as a *trellis*. Each node or state within the trellis represents a distinct state at a given time. Each branch of the trellis indicates a time-dependent state transition. The Viterbi algorithm takes advantage of the state nature of the code by tracing a *path* from state to state though the trellis diagram in an optimal fashion, thus decoding the output of the convolutional encoder. The trellis corresponding to the state transition diagram of Figure 3 is displayed in Figure 4.

A sequence $\mathbf{x} = (x_0, \ldots, x_n)$ is encoded using a convolutional encoder which outputs $\mathbf{z} = (z_0, \ldots, z_n)$. The sequence $\mathbf{z}$ is sent over a channel and received as $\hat{\mathbf{z}}$; the sequence $\mathbf{z}$ which may be corrupted by noise. The Viterbi algorithm is used to decode $\hat{\mathbf{z}}$, such that $\mathbf{P} = (z \textit{ was sent} | \hat{z} \textit{ was received})$ is maximized.

We now explain the Viterbi algorithm. Let ρ represent the cost function or metric that we are trying to minimize, and let k and j represent the states of the trellis.

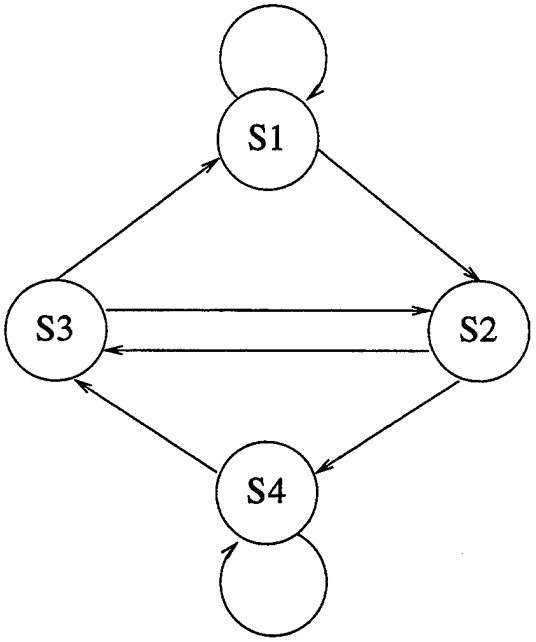


FIGURE 3.—*State Transition Diagram*

Initialize $\rho_k = 0$, state k=1, and time=0

1. Compute the current cost for all paths entering each state, j, by adding the cost metric, ρ_k, of the previous state to the cost incurred traversing the branch to this state j
2. For each state, j, save the path entering this state with the lowest cost metric (this becomes the *survivor path*),
   update metric $\rho_j = \rho_k + \rho$ (*traversing branch from k to j*), and
   eliminate all other branches entering this state
3. Increment time and repeat from step 1.

When the entire sequence has been processed, select the state at time n that has the minimum cost, ρ_k. Reconstruct the output sequence by following the path back to time 0 through this maximum likelihood path. An example is provided in Appendix A.

In the following sections the cost function is the Euclidean distance, defined in (1).

$$\rho(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{(x - \hat{x})^2}. \tag{1}$$

The Viterbi algorithm is an integral portion of many digital communication techniques; a few, namely TCQ and TCM, are covered in the following sections and subsections.

2.2 Trellis Coded Modulation (TCM)

Ungerboeck developed channel coding for multilevel/phase signals by combining channel coding and modulation [7]. This technique, known as TCM, can be described by a trellis

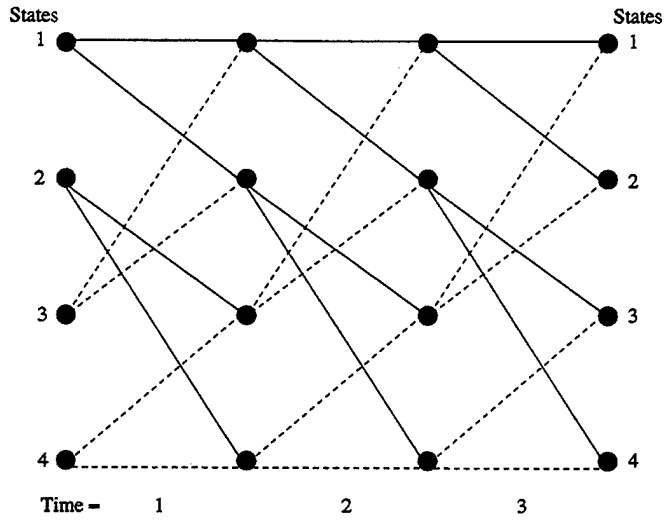


FIGURE 4.—*Corresponding Trellis*

diagram and uses maximum likelihood decoding, as in subsection 2.1. We begin with the simple case of a four-state trellis and confine our example to eight-level Phase-Shift Keying (PSK) to illustrate this concept. Eight PSK is a two-dimensional modulation scheme where symbols are mapped to discrete phases of a sinusoidal carrier. A block diagram of this TCM system is shown in Figure 5. Two input bits, $\mathbf{x} = (x_1, x_2)$, determine the phase modulated waveform at the output of the encoder. The first bit, x_1, is used as input to a binary convolutional encoder with rate 1/2. The output of this encoder along with the second input bit, x_2, is used to select one of the eight phase signals. The mapping and selection process of these phases are the unique design property of TCM.

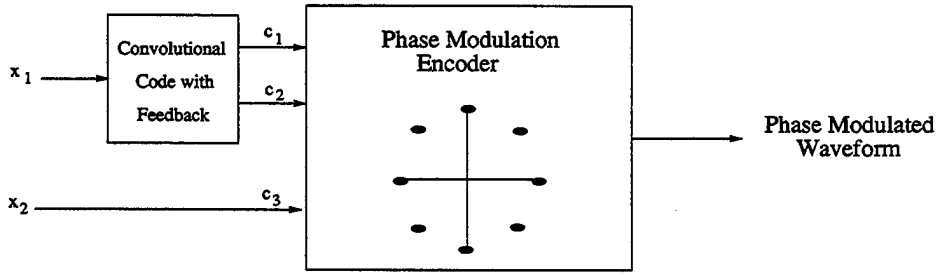


FIGURE 5.—*Block Diagram of the Eight-PSK TCM System*

Since the unquantized demodulator output is input directly into the maximum likelihood algorithm, it is implied that to provide low probability of error at the decoder, the signals must have a large *Euclidean distance.* The minimum Euclidean distance of a code is the free distance, d_{free}, defined in (2), where d is the Euclidean distance between signal x_n and $\hat{x}_n$. Coding and signal mapping are designed together to maximize d_{free} between all code sequences, Ungerboeck defines this process as *mapping by set partition.*

$$d_{free} = \min_{\{x_n\} \neq \{\hat{x}_n\}} \left[\sum_n d^2(x_n, \hat{x}_n) \right]^{1/2}. \tag{2}$$

One of the ways that TCM maximizes d_{free} is by expanding the signal set; when transmitting m bits/symbol, a signal set, or alphabet, of size 2^{m+1} is used. This technique provides redundancy for coding with little to no bandwidth expansion. Set partitioning divides the signal constellation into subsets that maximize d_{free}. The guidelines for set partitioning are as follows:

1. Maximize distance between parallel transitions,
2. Maximize distance between transitions originating *or* ending at the same state,
3. Use all symbols with equal frequency.

In Figure 5, the input vector is $\mathbf{x} = (x_1, x_2)$, where x_1 is the input to a convolutional encoder whose output selects the one of four possible signal subsets and x_2, which selects the signal within that subset. TCM's typical convolutional encoder is designed with feedback and rate equal to 1/2.

For the eight-level phase modulation system in this example, there are four possible signal subsets that maximize d_{free}, each containing two symbols with maximum Euclidean distance. These subsets are shown in Figure 6.

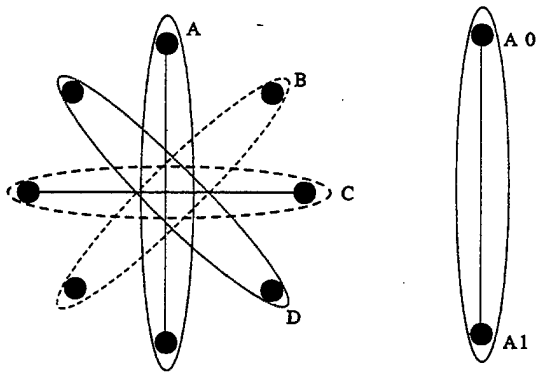


FIGURE 6.—*Example of Set Partitioning*

The two symbols within each subset represent the parallel transition in the trellis diagram. Corresponding to the example in Figure 6, x_1 selects subset A, B, C, or D based on the output of the convolutional encoder, and x_2 selects the 0 or 1 signal within the selected subset.

The trellis diagram for the described system is shown in Figure 7. The path from state 1 to state 1, is the parallel path representing signals A0 and A1, respectively. Adhering to the first set-partitioning guideline, parallel path symbols are selected from the same signal set, as shown in Figure 8. By mapping A0 and A1 as parallel paths the signals have the maximum distance possible, 180° apart. As specified by the second set-partitioning guideline, transitions originating *or* ending at the same state are assigned in order to maximum d_{free}. Subset A and subset C originate at the same state as do subsets B and D; these signal subsets are 90° degrees apart, as is evident in Figure 8.

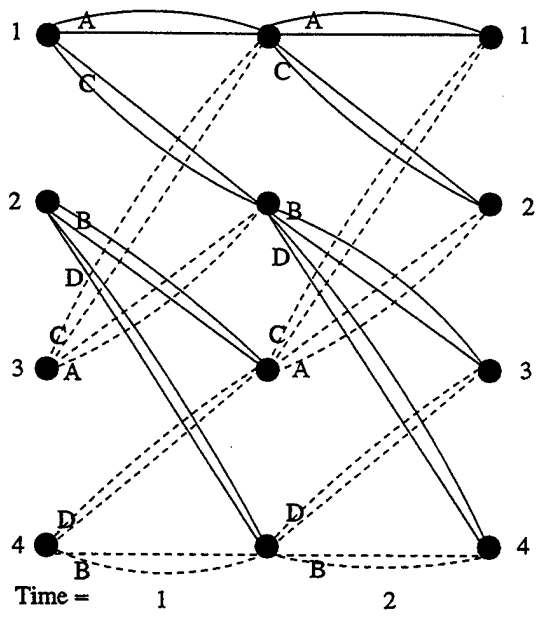


FIGURE 7.—*Example of Trellis States*

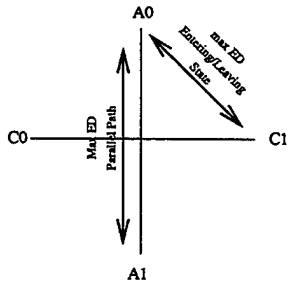


FIGURE 8.—*Adherence to Set-Partitioning Guidelines*

The fundamental concept of TCM, namely mapping by set partition, has been demonstrated using eight-PSK TCM. TCM offers high-spectral efficiency to operate within a few decibels of channel capacity with only moderate complexity. For a more detailed explanation of the TCM algorithm, we refer the reader to the subsequent paper by Ungerboeck [8].

2.3 Predictive Coding

Predictive coding is one of the many methods of compression. As with most types of compression, it can be lossy or lossless. The principle of predictive coding is based on eliminating the interpixel redundancy that exists between adjacent pixels. The redundancy is removed by taking the difference between the predicted value of the pixel and its actual value. This difference is the prediction error, or the *residual*, which constitutes the new information in the pixel. Only this new information is encoded in the compressed image. In this section, we begin by describing the lossless method of predictive coding, and then proceed to lossy predictive coding, which is the method used for PTCQ.

An image has high spatial correlation due to the physical system used to acquire it. For example, a CCD camera, which is used to obtain the image, consists of an array of photocell elements that leak energy into one another. This physical phenomenon is evident in the gradual change of pixel values throughout the image. To illustrate this fact, an example is provided by extracting a row of pixels from an image (shown in Figure 9). Here it is evident that the change from one pixel to the next is subtle.

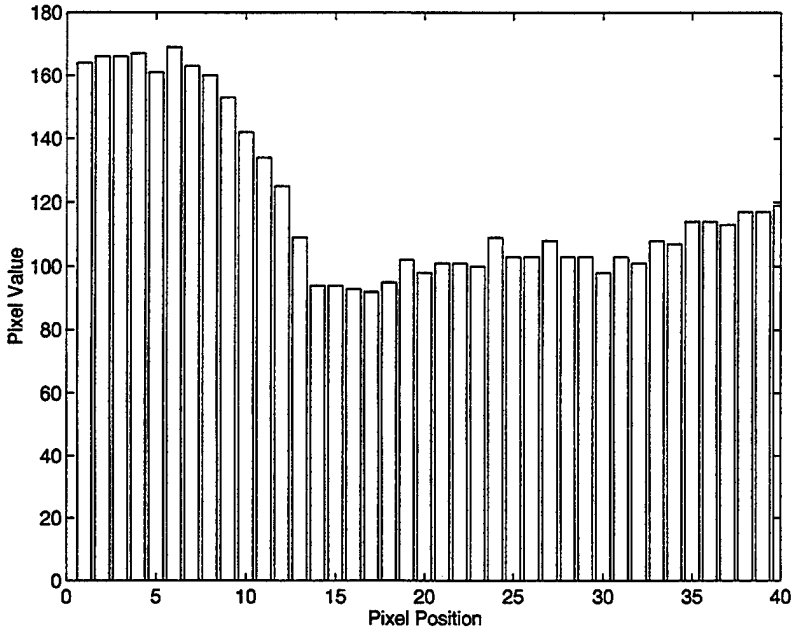


FIGURE 9.—*A Row of Pixels Extracted From an Image*

2.3.1 Lossless Predictive Coding

The basic components of a lossless predictive coding encoder are presented in Figure 10 [9]. For each pixel, u_n, of the input image, the predictor utilizes the weighted sum of neighboring pixel intensities to generate an estimate, $\hat{u}_n$. A prediction error, v_n, is then computed by subtracting the estimate from the corresponding pixel intensity. The residual is then encoded in some fashion using the symbol encoder.

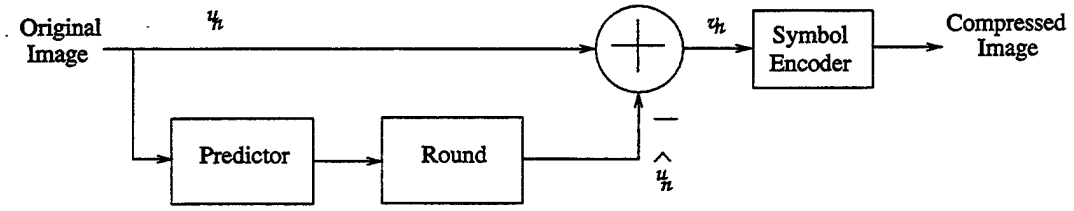


FIGURE 10.—*Predictive Encoder*

The decoder of Figure 11 performs the inverse operation utilizing the same prediction component of the encoder. The encoded residual, v_n, is received. It is decoded and added to the result of the predictor. The result is the reconstructed value, $u_n = v_n + \hat{u}_n$.

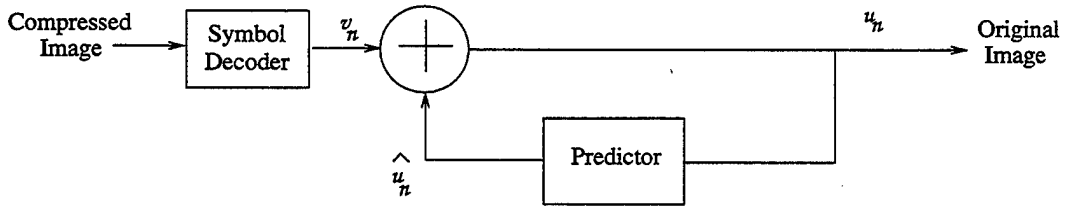


FIGURE 11.—*Predictive Decoder*

There are many types of predictors including local, global, linear, and nonlinear. The focus of this section is linear prediction, which is accomplished via a linear combination of *past* pixel values, as specified in (3). The raster scan method defines the concept of the past, and image can be thought of as a sequence in time, where pixels are encoded row by row, left to right, and top to bottom.

$$\hat{u}_n = Round[\sum_{i=1}^{m} \alpha_i u_{n-i}]. \tag{3}$$

The order of the predictor is m, and α_i, where $i = \{1, 2, \ldots, m\}$, represents the prediction coefficients. For lossless coding the first m pixels must be encoded using other methods since all subsequent values depend on the value of the first m pixels.

As an example of the compression obtainable through lossless predictive coding, the most basic linear predictor, the difference filter, is used to demonstrate. This is a first-order prediction filter, $m = 1$ and $\alpha = 1$. Therefore, the difference between pixel n and pixel $n-1$ becomes the residual. In Figure 12a, the histogram represents the distribution of the Lena (256×256) image, with mean, $\mu = 123$, standard deviation, $\sigma = 47.92$, and an entropy rate of 7.45 bits/pixel. After applying the linear prediction utilizing the difference filter, the distribution becomes more compact with a smaller dynamic range, with $\mu = 0.002$, $\sigma = 19.12$, and an entropy rate of 5.66 bits/pixel, as shown in Figure 12b.

Using the lossless encoder/decoder, the original image can be reconstructed from this compressed image without incurring any distortion. The prediction error is usually well modeled by a zero mean, variance σ^2, uncorrelated Laplacian probability distribution function, as given by (4).

$$p_u(u) = \frac{\alpha}{2} \exp(-\alpha \mid u - \mu \mid), \quad where \quad \sigma^2 = \frac{2}{\alpha}. \tag{4}$$

This function is very similar in shape to the residual histogram of Figure 12b and is in fact typical of prediction errors for all images. It is for this reason that the residual of predictive coding can be modeled as a Laplacian random process.

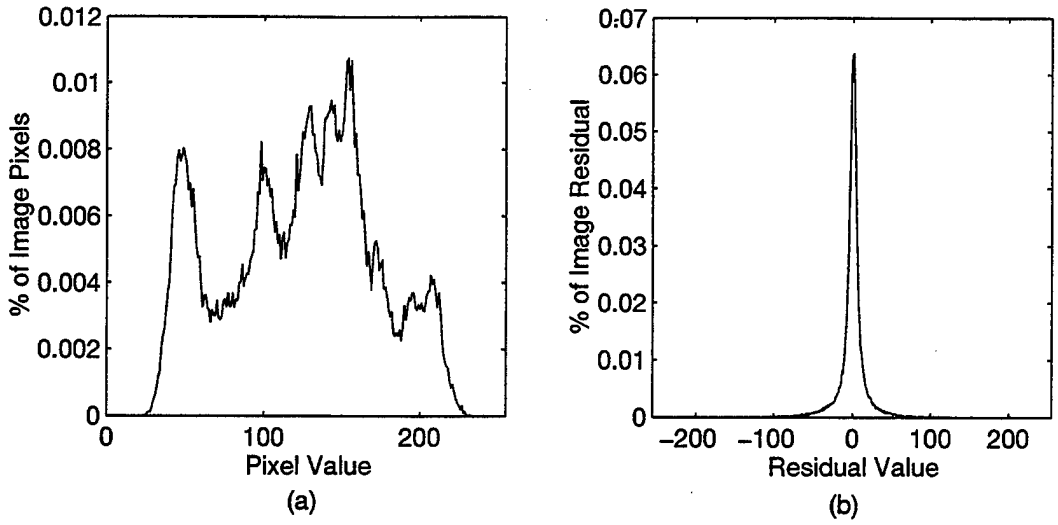


FIGURE 12.—*Histogram of (a) Lena Image and (b) Residual*

2.3.2 Lossy Predictive Coding

The model for lossy predictive coding is similar to that of the lossless model with the addition of a *Quantization* step and feedback as shown in Figure 13.

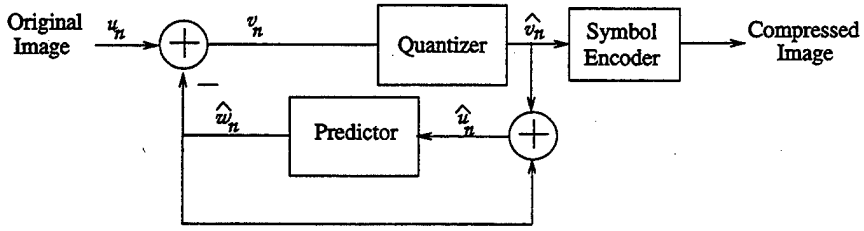


FIGURE 13.—*Lossy Predictive Encoder*

As for all lossy compression schemes, there is a compromise between distortion and compression. The quantization in the lossy scheme maps the prediction error to a limited number of outputs, denoted $\hat{v}$. Along with the predictor, the quantizer defines the amount of compression and distortion associated with this lossy system.

In this system, u_n is the input pixel and $\hat{w}_n$ is the prediction of u_n, as determined by the prediction filter. The prediction error, v_n, is the difference between the actual pixel value u_n and the predicted value $\hat{w}_n$. The prediction error is then quantized to $\hat{v}_n$ and is added with the prediction $\hat{w}_n$ to form $\hat{u}_n$, the estimated value of u_n. Then $\hat{u}_n$ is used as the input to the prediction filter, which produces $\hat{w}_{n+1}$, the predicted value of u_{n+1}. The feedback loop for the predictor is added so the inverse operation can be performed at the decoder stage and to compensate for the distortion induced by the quantizer. This closed loop configuration also prevents error buildup at the decoder's output [9].

The decoder for lossy predictive coding is identical to that of the lossless scheme of Section 2.3.1, shown in Figure 11. Assuming a perfect noiseless channel, $\hat{v}_n$ is received and

$\hat{w}_n$ is added, producing $\hat{u}_n$, the reconstructed pixel. The only distortion in this noiseless system is that caused by the quantizer.

Optimal predictors are used to minimize the distortion produced by lossy predictive coding. The distortion measure is the mean squared error, which is given by (5).

$$E\{v_n^2\} = E\{[u_n - \hat{u}_n]^2\}. \tag{5}$$

The optimizing criterion is chosen to minimize the mean squared prediction error. The quantizer error is assumed negligible ($v \approx \hat{v}$). The prediction is constrained to a linear combination of m previous pixels. These constraints allow simplification of the analysis and decrease the computational complexity of the predictor. The optimal predictor design is reduced to the process of selecting the α that minimizes (6).

$$E\{v_n^2\} = E\{[u_n - \sum_{i=1}^{m} \alpha u_{n-i}]^2\}. \tag{6}$$

After differentiating with respect to each coefficient α_i, equating the derivatives to zero, and solving the set of simultaneous equations under the assumption that u has zero mean and variance, σ^2, the resulting equation is (7).

$$\alpha = \mathbf{R}^{-1}\mathbf{r}, \tag{7}$$

where $\mathbf{R}^{-1}$ is the inverse of the $m \times m$ autocorrelation matrix $\mathbf{R}$, as indicated below.

$$R = \begin{bmatrix} E\{u_{n-1}u_{n-1}\} & E\{u_{n-1}u_{n-2}\} & \cdots & E\{u_{n-1}u_{n-m}\} \\ E\{u_{n-2}u_n - 1\} & \cdot & \cdots & \cdot \\ \vdots & \cdot & \cdots & \cdot \\ E\{u_{n-m}u_{n-1}\} & E\{u_{n-m}u_{n-2}\} & \cdots & E\{u_{n-m}u_{n-m}\} \end{bmatrix} \tag{8}$$

and $\mathbf{r}$ and α are the m-element vectors:

$$r = \begin{bmatrix} E\{u_n u_{n-1}\} \\ E\{u_n u_{n-2}\} \\ \vdots \\ E\{u_n u_{n-m}\} \end{bmatrix} \tag{9}$$

and

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}. \tag{10}$$

For any input image, the coefficients of α that minimize (6) can be determined using (7). The coefficients depend only on the autocorrelation of the pixels in the original image. As

in any system with feedback, the sum of the coefficients is typically restricted to less than or equal to one, as exhibited in (11).

$$\sum_{i=1}^{m} \alpha_i \leq 1. \tag{11}$$

This constraint ensures that the propagation of error caused by transmission errors will be reduced and that the prediction output falls within the allowable range for gray scale values. This restriction also prevents the decoder's output from becoming unstable. Depending on the configuration of the filter, transmission errors will appear as streaks propagating and diminishing in the direction of the filter as it slides across the image.

The following prediction filters contain a relatively robust set of coefficients that provide satisfactory performance over a wide range of images [9]. The filters are specified in (12) - (14).

$$\hat{u}_{i,j} = 0.97u_{i,j-1}. \tag{12}$$

$$\hat{u}_{i,j} = 0.5u_{i,j-1} + 0.5u_{i-1,j}. \tag{13}$$

$$\hat{u}_{i,j} = 0.75u_{i,j-1} + 0.75u_{i-1,j} - 0.5u_{i-1,j-1}. \tag{14}$$

Predictive coding utilizing the optimal filter method, as described previously, is commonly known as Differential Pulse Code Modulation (DPCM). In this section we have covered the major concepts of predictive coding. The optimal predictor is revisited in the following sections.

2.4 Lloyd-Max Quantizers

Quantizers map a range of values within the input alphabet to a limited set of values in the output alphabet. Analog-to-digital converters are quantizers, as are math functions such as the rounding function. To describe a quantizer, let u be a real scalar random variable with continuous probability density function $p_u(u)$. An L-level quantizer is desired whose thresholds are t_k, where $(k = 1, \cdots, L)$ and the quantization or reconstruction levels are r_k, where $(k = 1, \cdots, L)$. Let $g(u)$ be the mapping function, input to output, of the quantizer as defined in (15). The quantizer error, the distortion introduced by quantization, is computed in (16). The diagram describing a uniform quantizer is shown in Figure 14.

$$g(u) = r_i \quad if\ t_i \leq u < t_i + \Delta. \tag{15}$$

$$D = E[(u-r)^2] = \int_{t_1}^{t_{L+1}} (u-r)^2 p_u(u) du. \tag{16}$$

This is the simplest form of a quantizer that *uniformly* maps the input to the output. The distortion, as defined in (16), introduced during uniform quantization of a source with a uniform probability density function is (17), where Δ is the distance between the thresholds, $1/L$.

$$D = L\frac{\Delta^2}{12}. \tag{17}$$

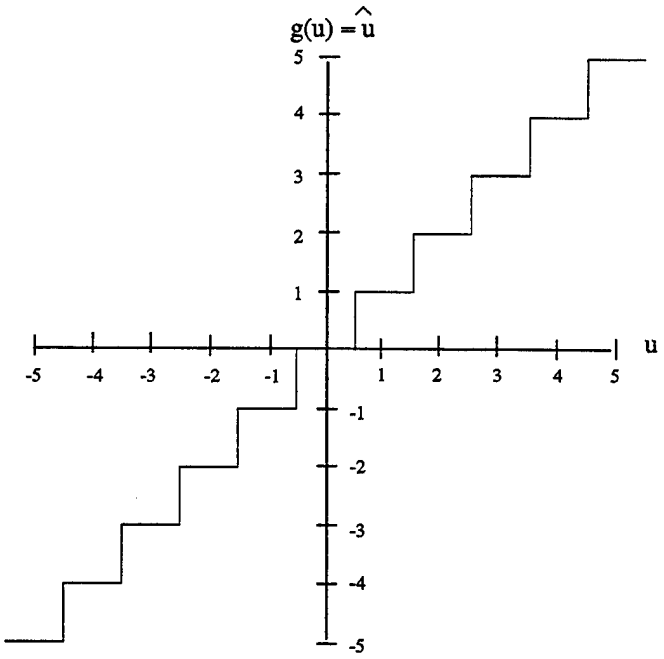


FIGURE 14.—*Uniform Quantizer*

When a uniform quantizer is used to quantize a source whose distribution is not uniform, the quantization distortion increases. A method of quantizing a nonuniform source with a uniform quantizer and minimizing this increase in distortion is *companding*. Any random variable can be converted to a uniform random variable using a nonlinear transformation, $f(\cdot)$. Companding, depicted in Figure 15, is the process of transforming the nonuniform source into a uniform source, quantizing using a uniform quantizer and inversing the transformation to result in a variable with the initial probability density function. However, this transformation is nonlinear in nature, the original sequence may not be recoverable, and the transformation introduces its own distortion. Consequently, quantizers are designed optimally to minimize the distortion when processing a nonuniform source.

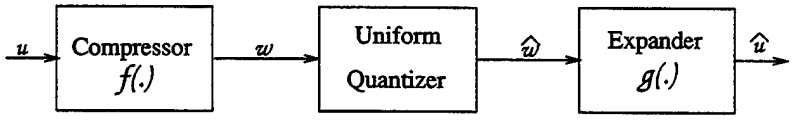


FIGURE 15.—*Companding*

To minimize quantizer distortion, Lloyd [10] and Max [11] designed an algorithm to iteratively find the thresholds and reconstruction levels that minimize the mean squared quantization distortion [12]. The following describes the Lloyd-Max algorithm. Again, let u be a real scalar random variable with continuous probability density function, $p_u(u)$. If an L-level quantizer is desired, the thresholds, t_k, $(k = 1, \cdots, L)$, and the reconstruction levels,

r_k, $(k = 1, \cdots, L)$, which minimize the mean squared error, must be determined. The quantizer error, as defined in (16), is minimized over all reconstruction levels:

$$D = \sum_{i=1}^{L} \int_{t_1}^{t_{i+1}} (u - r_i)^2 p_u(u) du. \tag{18}$$

Equations (19) and (20) display the results, after differentiating with respect to t_k and r_k, equating to zero and simplifying.

$$t_k = \frac{(r_k + r_{k+1})}{2}, \tag{19}$$

and

$$r_k = \frac{\int_{t_1}^{t_{k+1}} u p_u(u) du}{\int_{t_1}^{t_{k+1}} p_u(u) du}. \tag{20}$$

Based on this result, the optimal thresholds lie half way between the optimal reconstruction levels, which lie at the center of the mass of the probability density function, $p_u(u)$. Both (19) and (20) are nonlinear equations and must be solved simultaneously, given the boundary values, t_1 and t_{L+1}, which specify the dynamic range of the quantizer. The equations can be solved by an iterative scheme such as Newton's method.

It is sufficient to design a optimal quantizer with $\mu = 0$ and $\sigma = 1$ for all sources with the same probability density function [12]. If r_k and t_k are the reconstruction and threshold levels for a zero mean and unity variance random variable u with probability density function $p_u(u)$, a source with mean μ, variance σ^2, and the same probability density function can be transformed to zero mean and unity variance using the linear transformations specified in (21).

$$r_k = \mu + \sigma r_k \ \ and \ \ t_k = \mu + \sigma t_k. \tag{21}$$

A diagram of this optimal quantizer system can be viewed in Figure 16.

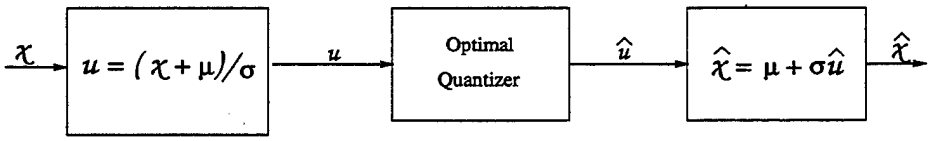


FIGURE 16.—*Optimal Quantization*

For the purpose of quantizing the prediction error for predictive coding, as in section 2.3.2, the random variable $p_u(u)$ is modeled by a Laplacian random process, and we assume u is a Laplacian random variable, as specified in (4).

In this section, necessary background for PTCQ — specifically, the Viterbi algorithm, set partitioning for TCM, lossy predictive coding, and the design of Lloyd-Max Optimal quantizers — has been provided. All of which are relevant to the formulation of PTCQ.

3. Predictive Trellis-Coded Quantization

The fundamental concepts presented in section 2 will now be integrated to formulate PTCQ as a robust source coding scheme. The source coding dual of TCM — namely, TCQ — as well as the modifications necessary for good performance over noisy channels is described. Lossy predictive coding is then added to the TCQ structure. Finally, the prediction filters that will be incorporated into PTCQ are detailed.

3.1 Trellis Coded Quantization (TCQ)

TCQ combines the optimal quantizers designed using the Lloyd-Max Algorithm with TCM. For a memoryless source, this method of quantization outperforms the optimal scalar quantizer and comes within 0.21 dB of the distortion rate lower bound. TCQ exploits the duality between modulation for digital communication systems and source coding while following the general principles of TCM signal expansion described in subsection 2.2 [13].

To encode a memoryless source at the rate of R bits/sample, the output alphabet is chosen at the rate $R+1$, similar to TCM's signal expansion of $m+1$. The Lloyd-Max Quantizer presented in subsection 2.4, is used to generate a quantizer with 2^{R+1} levels. These scalar quantizer points are partitioned according to the set-partitioning guidelines of subsection 2.2, and the number of subsets is specified by the number of states in the trellis. As an example, we use the case of a four-state trellis with rate, $R=3$. The optimal quantizer with $2^{3+1}=16$ levels is partitioned into four subsets. Beginning at the left-most point, the quantization points are sequentially labeled with subsets as demonstrated in Figure 17.

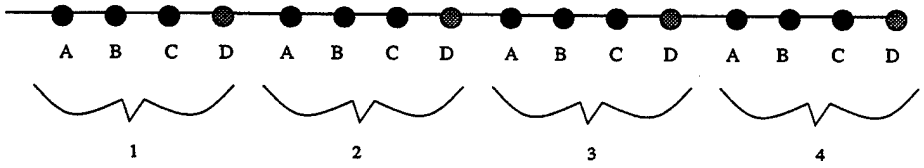


FIGURE 17.—*Partitioned Scalar Quantizer*

The trellis diagram is generated from the output of a convolutional encoder with feedback, as shown in Figure 18. The resulting trellis is also depicted in Figure 18. The parallel paths that represent the quantization levels within the subset are drawn as single lines for readability. The branches are labeled with the appropriate input bit, x_1, which symbolizes the branch taken from a state. This is termed the *branch bit.* In addition the trellis branches are labeled with subsets following the set-partition guidelines; therefore, quantization points along parallel paths and subsets entering *or* leaving a state should have the maximum Euclidean distance possible. Figure 19 demonstrates the adherence of the trellis to the set-partitioning guidelines of subsection 2.2.

To encode a source using TCQ, the Viterbi algorithm is used, as described in subsection 2.1. The algorithm determines the sequence of allowable output symbols that minimizes

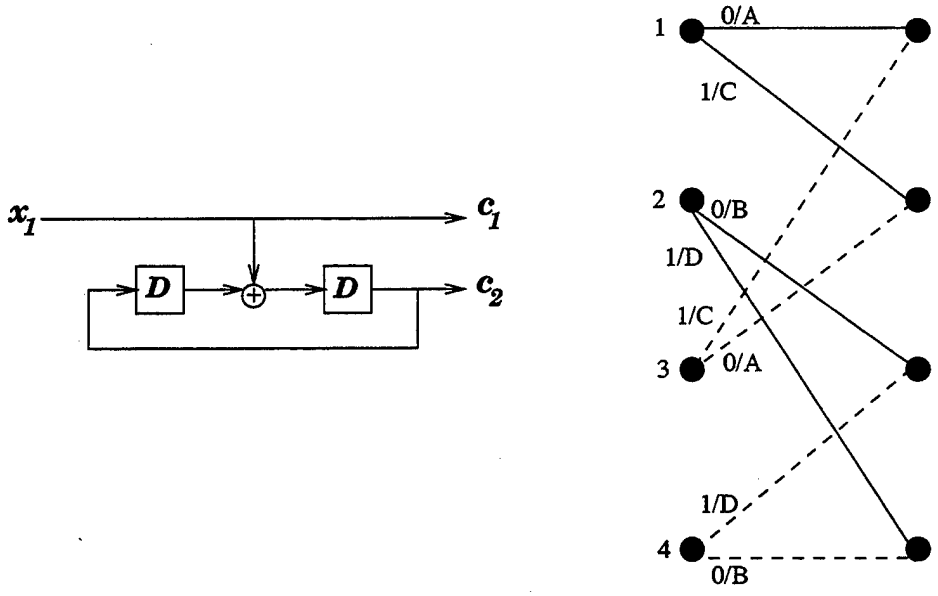


FIGURE 18.—*Convolutional Encoder With Feedback, TCQ Trellis*

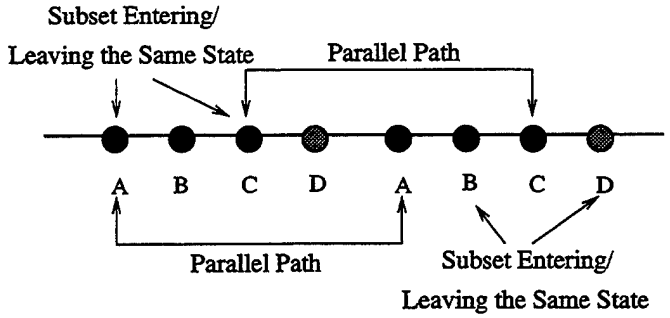


FIGURE 19.—*TCQ Partitions*

the distortion, $\rho(x, \hat{x})$, as defined in (22).

$$\rho(x, \hat{x}) = \sqrt{(x - \hat{x})^2}. \tag{22}$$

This output sequence can be represented by a bit sequence specifying the path through the trellis. This path is indicated by using the branch bit to stipulate the the branch taken from a state, thereby determining the subset, and the remaining $R-1$ bits are used to designate the quantization level selected within this chosen subset.

3.2 Trellis Modification for the Noisy Channel

When Ungerboeck designed the trellis for TCM, he utilized convolutional encoders with feedback. For TCQ on the other hand, using a feedback encoder causes difficulties. If the TCQ encoder output sequence is sent over a noisy channel, a single bit error can result in the TCQ decoder diverging indefinitely from the intended trellis path. This is a form

of catastrophic failure and should be avoided. A demonstration of this event is shown in Figure 20. The four-state, four-stage trellis is produced by the convolutional encoder with feedback given previously in Figure 18. For this example only, the branch bits which specify the path through the trellis will be provided. Suppose the sequence of branch bits sent is $x = (0,0,0,0)$. A bit error occurs as a result of the noisy channel, and the sequence is received as $x' = (1,0,0,0)$. In Figure 20, the correct path, x, is symbolized by the solid line and the path resulting from receiving x' rather than x is indicated by the dashed line.

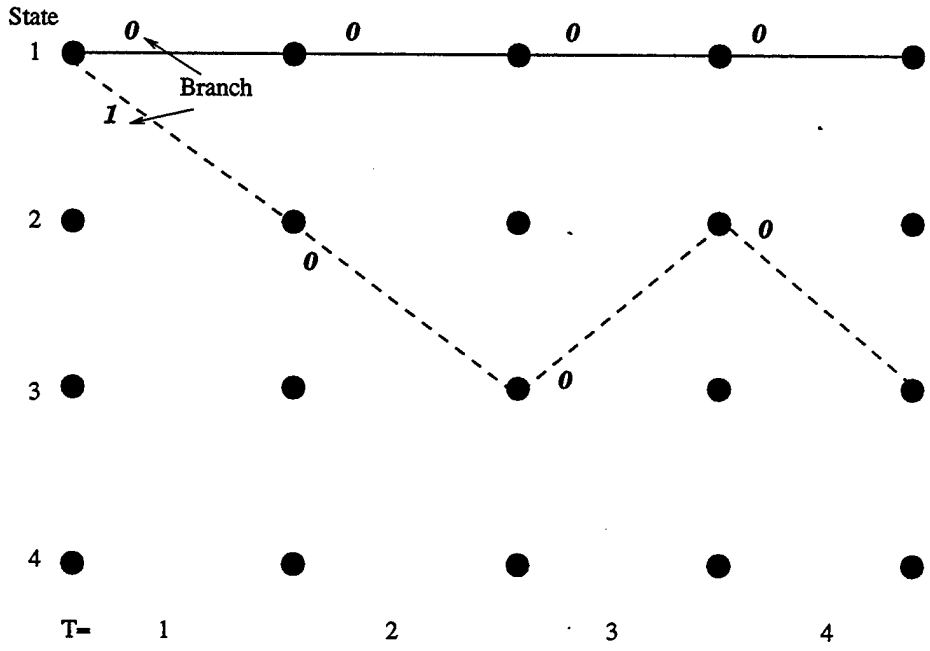


FIGURE 20.—*TCQ Trellis Error*

As shown in this example, a single bit error, which appears in the received branch bit sequence, causes a diversion from the intended maximum likelihood path that is not recovered. The error could propagate throughout the remaining sequence, causing increased distortion due to a single channel error.

To remedy this catastrophic failure, a convolutional encoder without feedback can be used to specify the branch selection. Fortunately, for every convolutional encoder with feedback, there exists a *feedback-free* encoder, for which any given input bit can affect no more that $1 + log_2(N)$ outputs, where N is defined as the number of trellis states [13]. One such convolutional encoder is shown in Figure 21 [14].

This feedback-free convolution encoder produces a similar trellis but with the branch bits labeled differently, as illustrated in Figure 21. Using the new trellis produced from this feedback-free encoder, let us recreate the previous example where $x = \{0,0,0,0\}$. This intended branch bit sequence is corrupted by noise, and the received sequence is again $x' = \{1,0,0,0\}$. The ensuing path through the trellis is displayed in Figure 22. This example demonstrates that the intended path through the trellis is recoverable after a bounded delay when transmission errors occur.

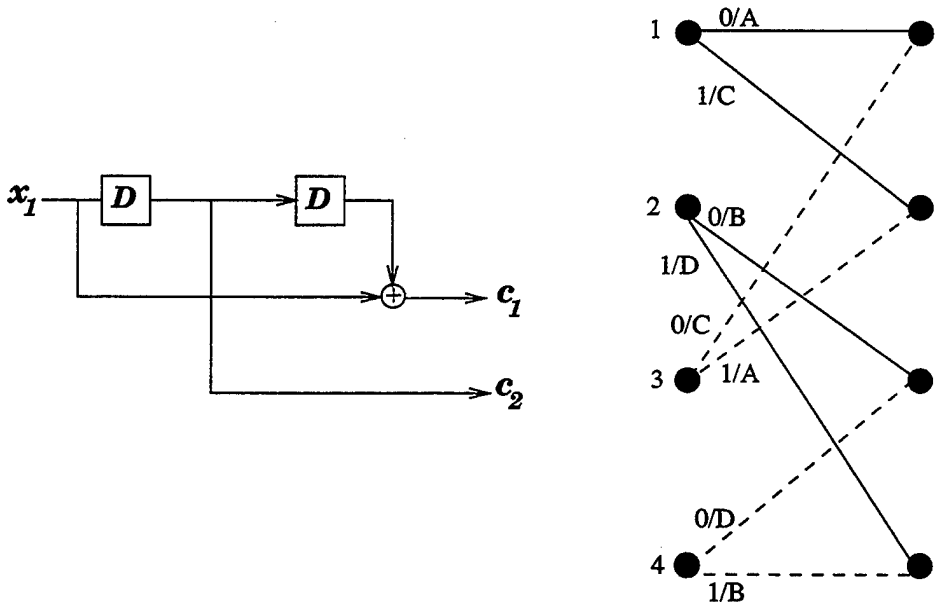


FIGURE 21.—*Convolutional Encoder Without Feedback, Modified TCQ Trellis*

3.3 Combining TCQ and Predictive Coding

Lossy predictive coding utilizing a scalar quantizer performs well at compressing highly correlated data. As stated in subsection 3.1, TCQ outperforms scalar quantization and comes closer to the distortion rate lower bound. Given this fact, a logical extension is to incorporate the quantization method of least distortion into the predictive coding configuration. This will achieve lower distortion than traditional lossy predictive coding of subsection 2.3.2.

Recall the Viterbi algorithm described in subsection 2.1. At each stage in time, each survivor path at a state indicates a sequence of output symbols. At time of transmission, the survivor path of minimum distortion is chosen as the output sequence. Also recall the lossy predictive coding of subsection 2.3.2, where previous values of the data are used as input to the predictor. For PTCQ, the survivor path at each state specifies these previous values of the data. The prediction residual is formed at each state as the difference between the predicted value, calculated at each state from its own survivor path, and the current data sample. For each branch emanating from a state, a scalar quantization is performed to determine the best quantization point associated with the prediction residual within the subset of this branch. Cumulative distortion is computed at each state by adding the distortion for the survivor path to the distortion incurred by the scalar quantization. A single path entering the next state is selected based on the cumulative distortion, while the other path entering this state is discarded. This now becomes the survivor path for this state at this stage in time. The new survivor path specifies the previous values to be input into the prediction, and the entire procedure is repeated.

To explain the algorithm further, consider the following example from a paper by Marcellin and Fischer [15]. Given a sequence to be encoded, $x = \{x_1, x_2, \ldots, x_n\}$, the ith step in the encoding process is as such. Let the survivor path at state k at stage $i-1$ be called

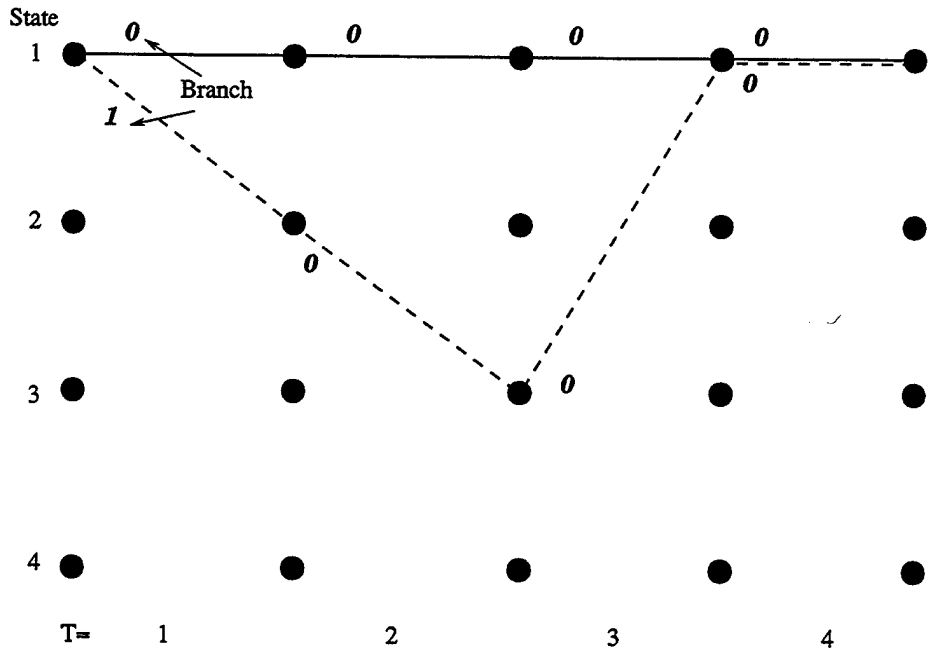


FIGURE 22.—*Modified TCQ Trellis Error*

survivor-k, and let $\hat{x}^k_{i-j}$, $k = 1, 2, \ldots, N$, be the encoded value of x_{i-j} $j = 1, 2, 3, \ldots$, associated with survivor-k, where N is the number of trellis states. Let $\hat{x}^k_{i|i-1}$ denote the predicted value of the current data sample x_i, given the survivor path, and let $d^k_i = (x_i - \hat{x}^k_{i|i-1})$ be the prediction residual associated with survivor-k. Furthermore, let $\rho_{i-1}(x, \hat{x}^k)$ be the distortion associated with survivor-k. For the PTCQ trellis, there are two branches entering and leaving each state. Denote the subset associated with the branch leaving state k and entering state l as D^k_l. For each subset, a scalar quantization operation is performed to determine the subset element closest to d^k_i. This element is denoted $\bar{D}^k_l$. All elements of each subset are discarded except for the one selected by the scalar quantization. This process is performed for all N states. At the next stage i, there are two branches entering each state, these branches are labeled $\bar{D}^{k_1}_l$ and $\bar{D}^{k_2}_l$, corresponding to the *from states* of which these branches emanate. At this time, the cumulative distortion is computed for each branch as specified in (23).

$$\rho_i(x, \hat{x}^l) = \min_{k \in \{k_1, k_2\}} (\rho_{i-1}(x, \hat{x}^k) + (d^k_i - \bar{D}^k_l)^2). \tag{23}$$

The survivor path at state l now becomes

$$\hat{x}^l_i = \hat{x}^{k'}_{i|i-1} + \bar{D}^{k'}_l, \tag{24}$$

where k' is the value of k that achieves the minimum in (23). This recursion is carried out until the end of the data sequence ($i = n$).

An abstract diagram of the PTCQ algorithm is shown in Figure 23; a two-state trellis is used for simplicity.

It should be noted that the PTCQ algorithm is suboptimal [15]. This suboptimality arises from the prediction. Future prediction values depend greatly on the quantization of

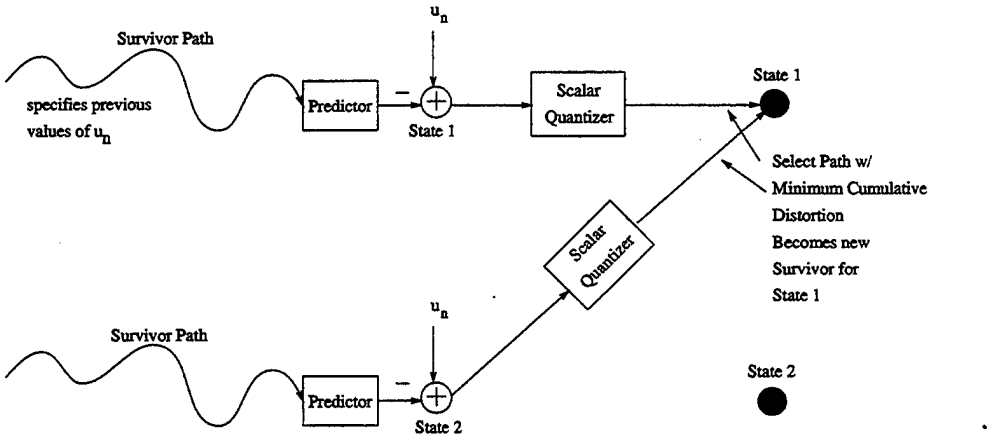


FIGURE 23.—*Abstract Diagram of PTCQ*

the past prediction values. Since a path is eliminated in favor of the survivor path at each stage in time, it is reasonable to assume that the eliminated path may have produced a lower *overall* distortion than the survivor path. However, results will show that the effect of the suboptimality is negligible.

3.4 Prediction Filters

There are various types of prediction filters available for predictive coding. The filters that will be incorporated into the PTCQ structure are the difference, flat, fixed, optimal linear, and nonlinear prediction filters. To describe these filters, the filter window is provided, as shown in Figure 24, along with the filter coefficients α_n where $n = (1, \ldots, m)$ where m is the order of the filter. The variable $\hat{w}_0$, represents the *predicted* value of the next pixel indicated by the *zero*th element within the filter window. All of the filters used for PTCQ are causal; image reconstruction incurs no delay because reconstructed pixels depend only on the past pixel values.

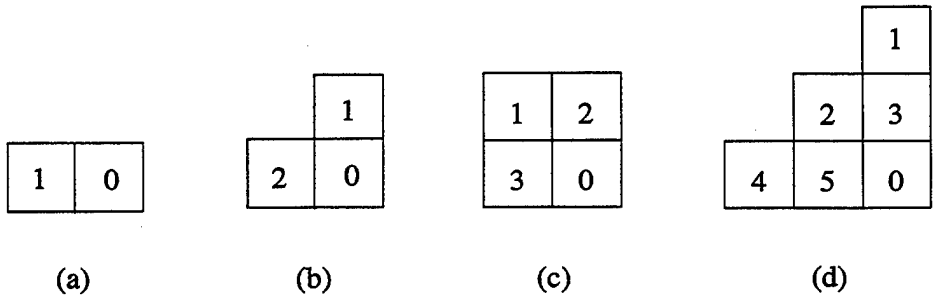


FIGURE 24.—*Filter Windows*

The difference filter, $m = 1$, Figure 24a, as mentioned in subsection 2.3.1, is represented by the following equation:

$$\hat{w}_0 = .97\hat{u}_1, \quad where \ \hat{u}_1 = \hat{u}_{i,j-1}. \tag{25}$$

The flat filter, $m = 2$, Figure 24b, applies equal weights to the surrounding pixel values to formulate the prediction value.

$$\hat{w}_0 = 0.5\hat{u}_1 + 0.5\hat{u}_2, \quad where \ \hat{u}_1 = \hat{u}_{i-1,j} \ and \ \hat{u}_2 = \hat{u}_{i,j-1}. \tag{26}$$

Two three-point filters will be used, namely the fixed and optimal linear filters. In theses cases $m = 3$ and the filter window is shown in Figure 24c, where the one-dimensional elements represent:

$$\hat{u}_1 = \hat{u}_{i-1,j-1}, \quad \hat{u}_2 = \hat{u}_{i-1,j}, \quad and \ \hat{u}_3 = \hat{u}_{i,j-1}. \tag{27}$$

The three-point fixed filter of subsection 2.3.2 applies the following coefficient to produce the prediction value:

$$\hat{w}_0 = -0.5\hat{u}_1 + .75\hat{u}_2 + .75\hat{u}_3. \tag{28}$$

The optimal linear filter is computed based on the autocorrelation of the three surrounding pixel values as specified in subsection 2.3.2, which details optimal prediction. Using the following matrix operation to compute α.

$$\alpha = \mathbf{R}^{-1}\mathbf{r}, \tag{29}$$

where

$$R = \begin{bmatrix} E\{u_1u_1\} & E\{u_1u_2\} & E\{u_1u_3\} \\ E\{u_2u_1\} & E\{u_2u_2\} & E\{u_2u_3\} \\ E\{u_3u_1\} & E\{u_3u_2\} & E\{u_3u_3\} \end{bmatrix}, \tag{30}$$

$$r = \begin{bmatrix} E\{u_0u_1\} \\ E\{u_0u_2\} \\ E\{u_0u_3\} \end{bmatrix}, \tag{31}$$

and

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}. \tag{32}$$

In these equations, $E\{\cdot\}$ is the empirical average obtain from the observed pixels values in each image. By computing $\mathbf{R}$ and $\mathbf{r}$ for the entire image, a constant α is determined using (29).

The nonlinear filter utilized in the PTCQ scheme is a simple case of the Ll filter [16]. The nonlinear filter coefficients, α_j^k, are selected based on the ranking of the elements, $\hat{u}_j$. The elements of the five-point window are shown in Figure 24d.

$$\begin{aligned} \hat{w}_0 &= \alpha_1^k\hat{u}_1 + \alpha_2^k\hat{u}_2 + \alpha_3^k\hat{u}_3 + \alpha_4^k\hat{u}_4 + \alpha_5^k\hat{u}_5, \\ &where \ \hat{u}_1 = \hat{u}_{i-2,j}, \quad \hat{u}_2 = \hat{u}_{i-1,j-1}, \quad \hat{u}_3 = \hat{u}_{i-1,j}, \quad \hat{u}_4 = \hat{u}_{i,j-2}, \quad and \ \hat{u}_5 = \hat{u}_{1,j-1}. \end{aligned} \tag{33}$$

The filter coefficients for the minimum and maximum values within the filter window are set to zero, while a linear filter is invoked for the three remaining elements in the window. There

are $\binom{5}{2} = 10$ filters, indicated by α^k, $(k = 1, \ldots, 10)$, since the minimum and maximum of the window elements are discarded. The coefficients of the 10 filters are calculated by taking the autocorrelation of the three remaining pixels. The linear filter is selected using the coefficients of the filter that minimizes mean squared error, namely the filter that solves (29).

As an example, the following window of data is provided in Figure 25. In the example, pixels $j = 1$ and $j = 5$ form the set $J = \{1, 5\}$, which indicates the indices of the minimum and maximum pixel values. The remaining elements result in the configuration shown in Figure 26.

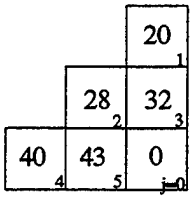


FIGURE 25.—*Example of Ll-Filter Implementation*

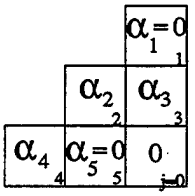


FIGURE 26.—*Resulting Ll-Filter Window*

In this particular case where $J = \{1, 5\}$, the 3×3 autocorrelation matrix $\mathbf{R}$ along with $\mathbf{r}$ are the empirical average of the remaining surrounding pixels with the filter window as specified in (34) and (35), respectively.

$$R = \begin{bmatrix} E\{u_2u_2\} & E\{u_2u_3\} & E\{u_2u_4\} \\ E\{u_3u_2\} & E\{u_3u_3\} & E\{u_3u_4\} \\ E\{u_4u_2\} & E\{u_4u_3\} & E\{u_3u_4\} \end{bmatrix}, \tag{34}$$

and

$$r = \begin{bmatrix} E\{u_0u_2\} \\ E\{u_0u_3\} \\ E\{u_0u_4\} \end{bmatrix}. \tag{35}$$

For the instance where the minimum and maximum values are located at $J = \{1, 4\}$, the resulting filter coefficients are identical to those used for optimal linear prediction filter, (29)-(32).

All of the prediction filters described in this subsection will be utilized within the PTCQ structure. In the following section, a comparison of the compression performance of these filters is presented in addition to their ability to mitigate the effects of channel errors.

4. Performance of PTCQ

In this chapter PTCQ performance is demonstrated through the evaluation of various configurations. The configuration variables are the number of states in the trellis, bit rate in bits/pixel (bpp), and prediction filter selection. To begin, the PTCQ quantizer error, the distortion introduced by the algorithm, is computed for each prediction filter. To evaluate the various prediction filters in a noisy environment, the channel is represented as a binary symmetric channel while the rate and number of states are held constant. The prediction filters which display the best noise immunity are further evaluated by varying the rate and the number of states. Finally, the encoded data is exposed to multiple levels of noise to establish the performance achievable over noisy channels.

The fidelity criteria for comparison are the mean squared error (MSE), and the power to signal noise ratio (PSNR), as indicated in (36) and (37), respectively. The variable x_i represents the pixels from the input image and $\hat{x}_i$ represents the corresponding pixel values after a process. Accompanying these error measurements, a qualitative comparison is also used to assess performance.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2, \tag{36}$$

and

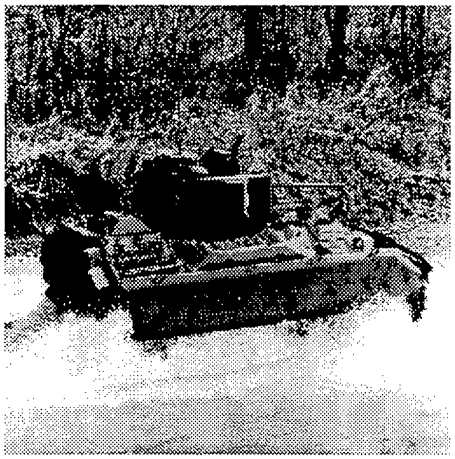
$$PSNR = 10\log_{10}\frac{255^2}{MSE}. \tag{37}$$

Three different images are used throughout this chapter to evaluate the PTCQ systems. For future reference, the original images are displayed in Figure 27. The MSE and PSNR are used to assess quantizer error, channel error, and total system error. The quantizer error is the error incurred by the utilization of the PTCQ system alone, the error between the original image, and the quantized image that has been encoded and then decoded by the PTCQ algorithm. The channel error is the error introduced by the channel, the error of the quantized image to the image that has been encoded, exposed to a noisy channel and decoded. The total system error is the error of the original image to the PTCQ resultant image, the image encoded, exposed to a noisy channel and then decoded.

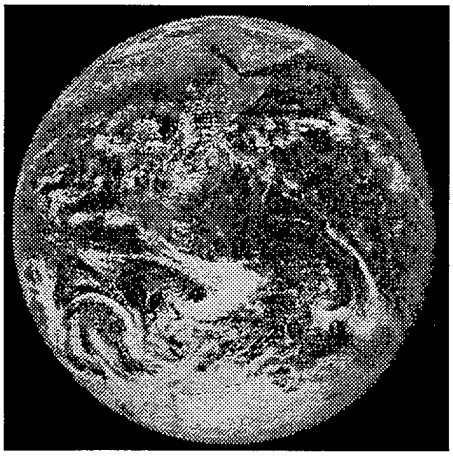
4.1 Quantizer Error

To determine the quantization error, the image is encoded and then decoded utilizing the PTCQ algorithm, as if it were passed through a noiseless channel. For this example, the image data are processed using a four-state PTCQ scheme with a bit rate of 3 bpp for each prediction filter. The resultant image is used to compute the distortion caused by the utilization of PTCQ. Appendix B contains tables of optimal linear and nonlinear filter coefficients for the subject images. Table 1 reflects the MSE and the PSNR for each prediction filter. All resultant images compare reasonably well with the original images; as

Lena (256×256)



M2 Bradley (256×256)



Earth (256×256)

FIGURE 27.—*Original Images*

a matter of fact, is is difficult to tell the result of one prediction filter implementation from another. A sampling of the decoded images using the various prediction filters is displayed in Figure 28. The distortion incurred by the PTCQ algorithm is negligible. It can therefore be concluded that all of the prediction filters perform well over the noiseless channel.

TABLE 1.—*Quantizer Error: Four-State PTCQ, Rate = 3 bpp*

Filter	Lena		M2 Bradley		Earth	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
Difference	19.12	35.31	83.63	28.90	46.78	33.39
Flat	12.74	37.07	57.08	30.56	33.91	32.82
Fixed	11.83	37.39	84.13	28.88	51.72	30.99
Linear (l)	11.43	37.55	54.74	30.74	31.78	33.11
Nonlinear (Ll)	14.69	36.47	55.82	30.66	31.58	33.14

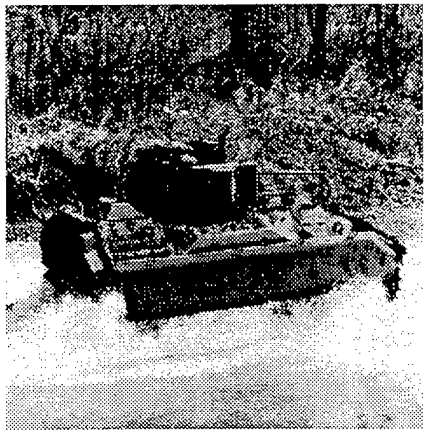
4.2 Performance Over the Noisy Channel

The encoded images of subsection 4.1 are used to demonstrate each filter's response to a noisy channel. A binary symmetric channel with crossover error probability, $\epsilon = .00316$, is simulated. The channel error, the distortion introduced by the channel, is exhibited in Table 2, and the total system error is indicated in Table 3. Figures 29 through 31 display the PTCQ output for each image. It is apparent from the error measurement and the resulting images that the difference, flat and fixed prediction filters have very low noise immunity properties. The difference filter provides the worst results in a noisy environment. Channel errors appear as horizontal streaks because only the previous pixel is used in the prediction, and the error propagates throughout the row of pixels in which the error occurred. Both the flat and fixed linear filters perform in a mediocre fashion; channel errors are very evident and appear as large blocky diagonal streaks on the resulting images. The performance of the linear and nonlinear filters is much better than the other three filters. The increase in performance is attributed to the *optimal* design technique by which the filter coefficients are defined to suit each image. The nonlinear filter performs visually better than the linear filter. This filter reduces the propagation of error because outliers, which may be caused by channel errors, are removed from the prediction when the coefficients of the maximum and minimum pixel values within the filter window are set to zero. In the succeeding subsections, the linear and nonlinear filters are further evaluated by varying the number of states in the trellis, the bit rate of the encoded image, and the channel noise.

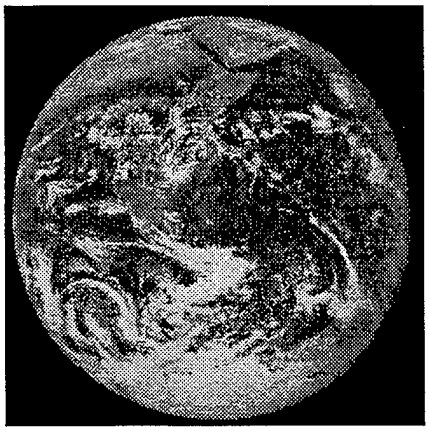
4.3 Variations in PTCQ Configuration

This subsection begins by altering the number of trellis states in the PTCQ scheme, incorporating the better performing prediction filters, linear (l-PTCQ) and nonlinear (Ll-PTCQ), as determined in subsection 4.2. Figures 32 through 34 display the resultant images

Difference Prediction Filter

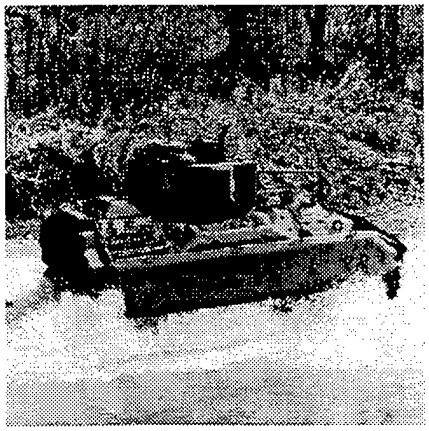


Flat Prediction Filter



Fixed Prediction Filter

Linear Prediction Filter



Nonlinear L*l* Prediction Filter

FIGURE 28.—*Sampling of PTCQ Systems, Rate = 3 bpp*

Difference Prediction Filter

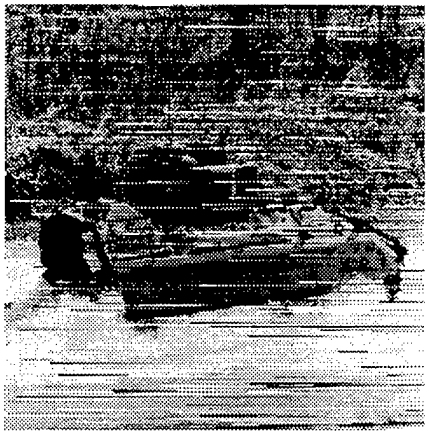
Flat Prediction Filter

Fixed Prediction Filter

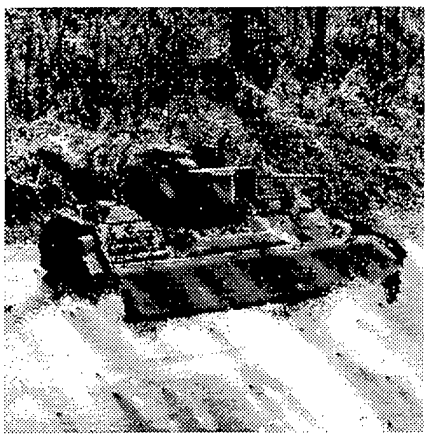
Linear Prediction Filter

Nonlinear L*l* Prediction Filter

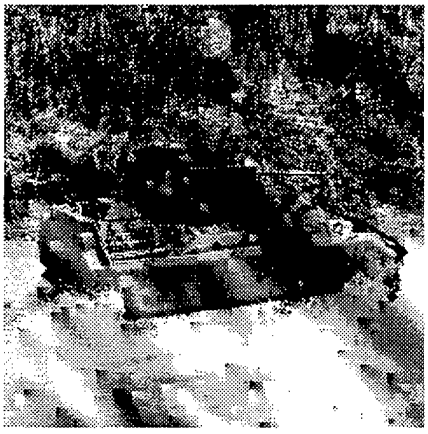
FIGURE 29.—*PTCQ Systems With* $\epsilon = 0.00316$, *Lena Image*



Difference Prediction Filter

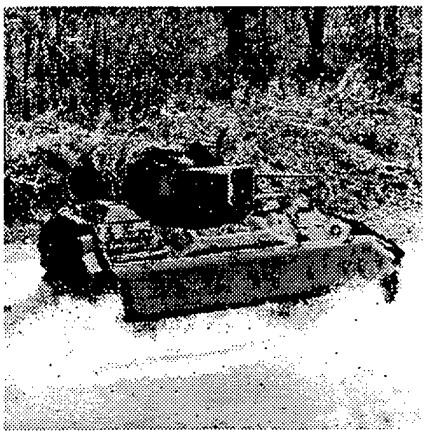


Flat Prediction Filter



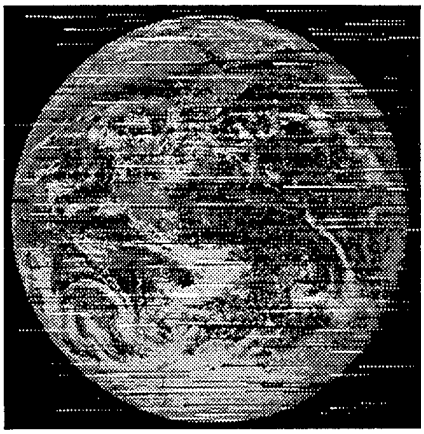
Fixed Prediction Filter

Linear Prediction Filter

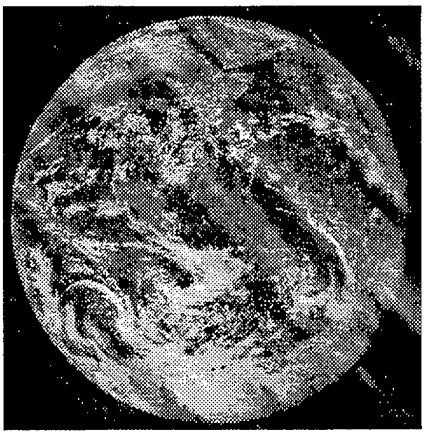


Nonlinear L*l* Prediction Filter

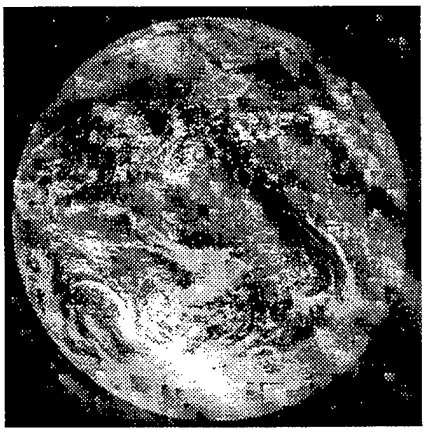
FIGURE 30.—*PTCQ Systems With* $\epsilon = 0.00316$, *M2 Bradley Image*



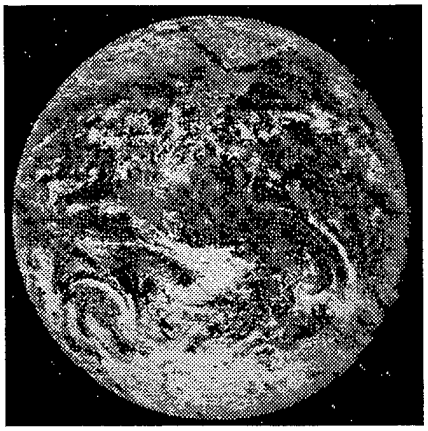
Difference Prediction Filter



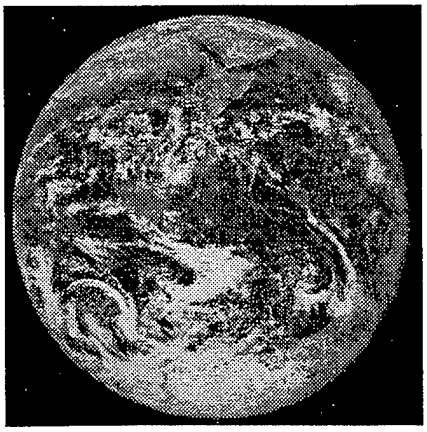
Flat Prediction Filter



Fixed Prediction Filter



Linear Prediction Filter



Nonlinear L*l* Prediction Filter

FIGURE 31.—*PTCQ Systems With* $\epsilon = 0.00316$, *Earth Image*

TABLE 2.—*Channel Error: Four-State PTCQ, Rate = 3 bpp, $\epsilon = 0.00316$*

Filter	Lena		M2 Bradley		Earth	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
Difference	302.67	23.32	2082.70	.0149	989.55	18.18
Flat	95.39	28.34	1190.80	.0174	700.28	19.68
Fixed	240.07	24.33	4736.50	.0114	1785.70	.0156
Linear (l)	51.95	30.97	261.51	23.96	128.27	27.05
Nonlinear (Ll)	55.99	30.65	587.07	20.44	69.21	29.72

TABLE 3.—*Total Error: Four-State PTCQ, Rate = 3 bpp, $\epsilon = 0.00316$*

Filter	Lena		M2 Bradley		Earth	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
Difference	320.84	23.07	2164.70	.0148	1035.20	.018
Flat	108.58	27.77	1246.40	.0172	735.22	19.46
Fixed	253.51	24.09	4818.60	.0113	1837.70	.0155
Linear (l)	63.21	30.12	313.63	23.17	160.59	26.07
Nonlinear (Ll)	70.89	29.62	638.38	20.08	100.49	28.11

of the two-, four- and eight-state PTCQ system at rate 3 bpp for both prediction filters with noise level $\epsilon = 0.00316$. Tables 4 and 5 reflect the channel error and total system error, respectively. It is apparent from the channel error and the images that the two-state system performs better in a noisy environment than the eight-state system. This is due to the use of a feedback free convolutional encoder. This guarantees that an error in the branch bit is limited to $1 + \log_2 N$ outputs, where N is the number of trellis states. Therefore, a diversion from the correct path would affect more output symbols as the number of trellis states increase.

TABLE 4.—*Channel Error: Two-, Four-, and Eight-State PTCQ, Rate = 3 bpp*

Configuration	Lena		M2 Bradley		Earth	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
2-l-PTCQ	34.39	32.76	177.50	25.63	95.52	28.33
4-l-PTCQ	51.95	30.97	261.51	23.95	128.27	27.05
8-l-PTCQ	70.60	29.64	340.57	22.81	173.43	25.74
2-Ll-PTCQ	52.75	30.91	99.75	28.14	56.46	30.61
4-Ll-PTCQ	55.99	30.65	128.19	27.05	69.21	29.73
8-Ll-PTCQ	66.33	29.91	155.31	26.22	80.54	29.07

The bit rate is now be altered within the PTCQ scheme. For simplicity, the number of states is fixed at two trellis states. Figures 35 through 37 display the resulting images encoded at rates 2, 3, 4, and 5 bpp, alternating between l-PTCQ and Ll-PTCQ. Tables 6

Two-State l-PTCQ

Two-State Ll-PTCQ

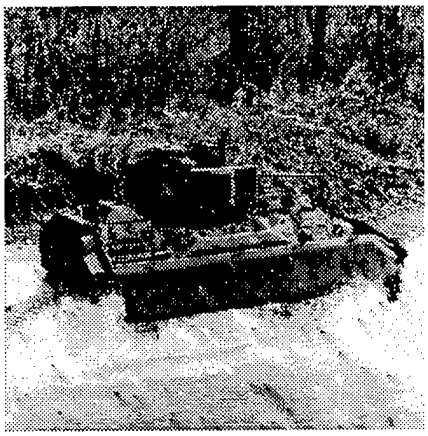
Four-State l-PTCQ

Four-State Ll-PTCQ

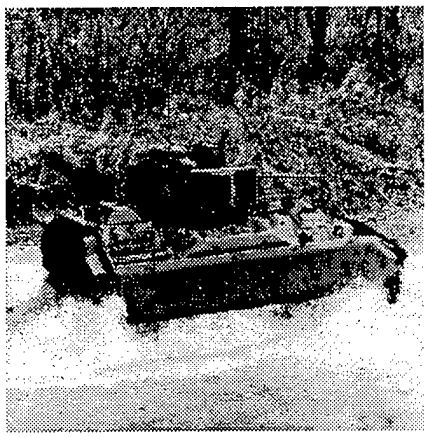
Eight-State l-PTCQ

Eight-State Ll-PTCQ

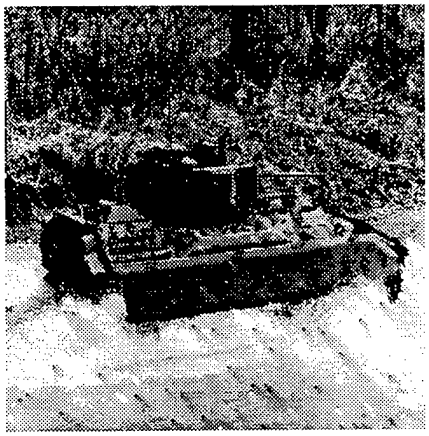
FIGURE 32.—*PTCQ System: Two-, Four-, and Eight-States, Lena Image*



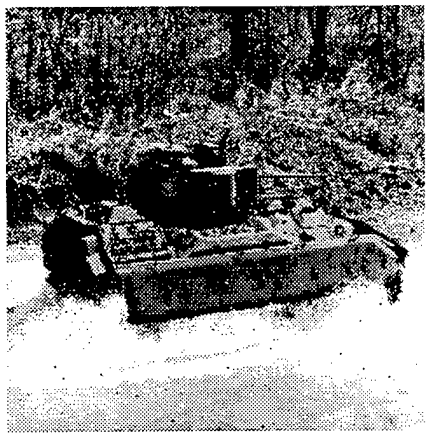
Two-State l-PTCQ



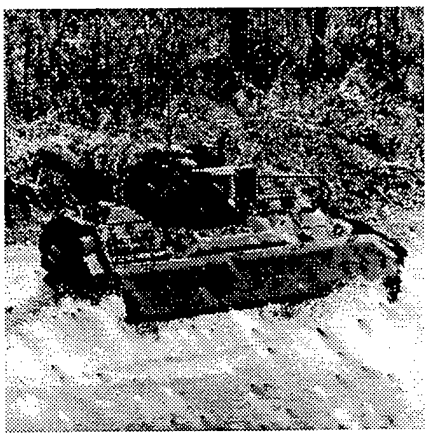
Two-State Ll-PTCQ



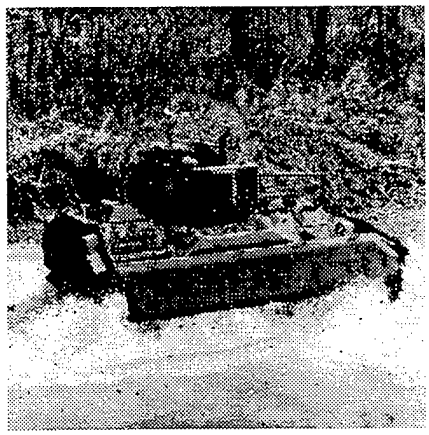
Four-State l-PTCQ



Four-State Ll-PTCQ

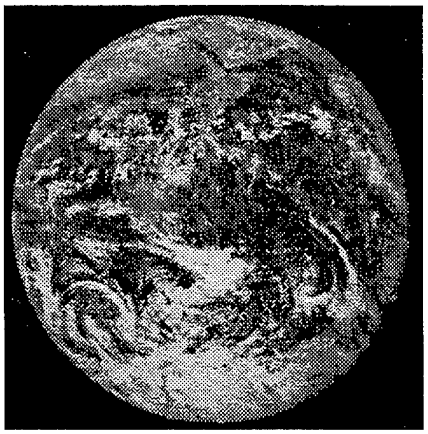


Eight-State l-PTCQ

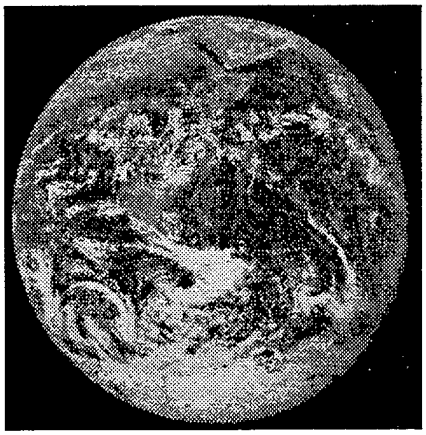


Eight-State Ll-PTCQ

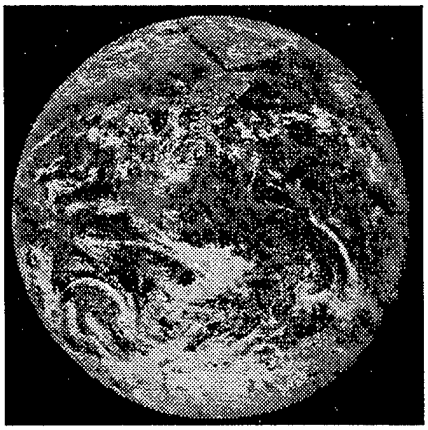
FIGURE 33.—*PTCQ System: Two-, Four-, and Eight-States, M2 Image*



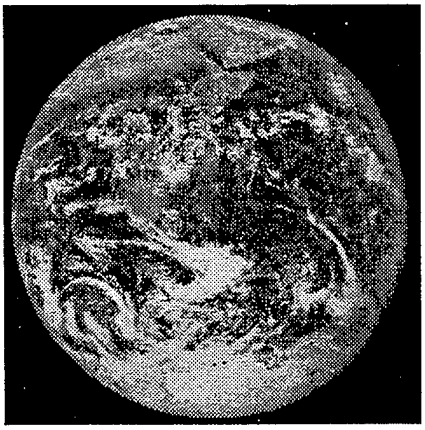
Two-State l-PTCQ



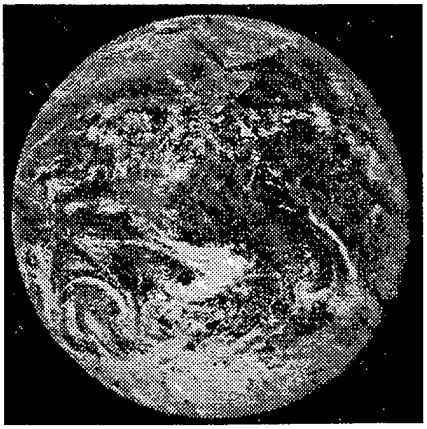
Two-State Ll-PTCQ



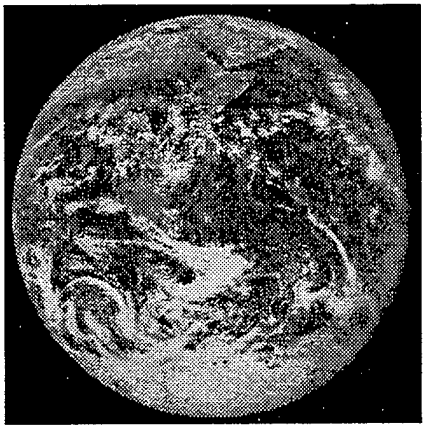
Four-State l-PTCQ



Four-State Ll-PTCQ



Eight-State l-PTCQ



Eight-State Ll-PTCQ

FIGURE 34.—*PTCQ System: Two-, Four-, and Eight-States, Earth Image*

TABLE 5.—*Total Error: Two-, Four-, and Eight-State PTCQ, Rate = 3 bpp*

Configuration	Lena		M2 Bradley		Earth	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
2-*l*-PTCQ	59.85	30.36	246.75	24.21	143.08	26.57
4-*l*-PTCQ	63.21	30.12	313.63	23.17	160.59	26.07
8-*l*-PTCQ	78.01	29.21	395.43	22.16	203.86	25.04
2-L*l*-PTCQ	82.36	28.87	172.70	25.76	104.84	27.93
4-L*l*-PTCQ	70.89	29.62	183.00	25.50	100.49	28.11
8-L*l*-PTCQ	76.37	29.30	209.56	24.92	110.66	27.69

and 7 indicate the channel error and total system error. From the images in Figure 35 through 37 along the channel and total error, it is evident that a lower bit rate leads to better performance over the noisy channel. This is attributed to the simple fact that there are fewer bits subject to corruption at the lower bit rate.

TABLE 6.—*Channel Error: Two-State PTCQ, Rate = 2, 3, 4, and 5 bpp*

Configuration	Lena		M2 Bradley		Earth	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
l-PTCQ, rate = 2	24.59	34.22	101.27	28.07	48.69	31.25
l-PTCQ, rate = 3	34.39	32.76	177.50	25.63	95.51	28.32
l-PTCQ, rate = 4	51.01	31.05	251.19	24.13	130.28	26.98
l-PTCQ, rate = 5	69.38	29.71	326.75	22.98	165.67	25.93
L*l*-PTCQ, rate = 2	36.96	32.45	69.54	29.70	33.18	32.92
L*l*-PTCQ, rate = 3	52.74	30.90	99.75	28.14	56.46	30.61
L*l*-PTCQ, rate = 4	68.51	29.77	124.54	27.17	68.94	29.74
L*l*-PTCQ, rate = 5	68.28	29.78	157.96	26.14	87.97	28.68

TABLE 7.—*Total Error: Two-State PTCQ, Rate = 2, 3, 4, and 5 bpp*

Configuration	Lena		M2 Bradley		Earth	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
l-PTCQ, rate = 2	104.10	27.95	347.55	22.72	208.93	24.93
l-PTCQ, rate = 3	59.84	30.36	246.75	24.20	143.08	26.57
l-PTCQ, rate = 4	60.03	30.34	272.77	23.77	143.31	26.56
l-PTCQ, rate = 5	71.82	29.56	332.27	22.91	169.19	25.84
L*l*-PTCQ, rate = 2	135.04	26.82	320.16	23.07	192.07	25.29
L*l*-PTCQ, rate = 3	82.35	28.97	172.70	25.75	104.84	27.92
L*l*-PTCQ, rate = 4	78.24	29.19	144.85	26.52	82.63	28.95
L*l*-PTCQ, rate = 5	71.23	29.60	163.38	25.99	90.99	28.54

The PTCQ performance for variations in the number of states and bit rate has been

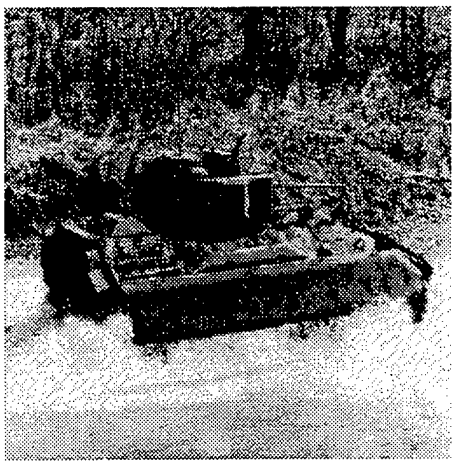
Two-State l-PTCQ, rate = 2 bpp

Two-State Ll-PTCQ, rate = 3 bpp

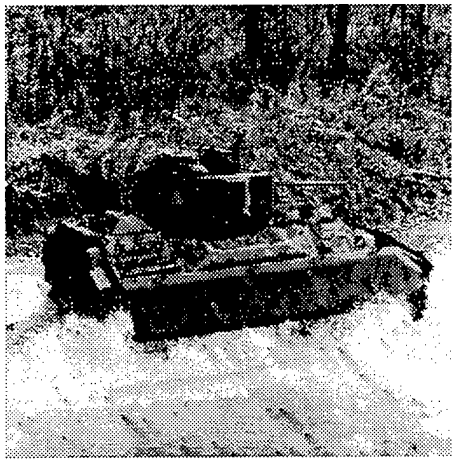
Two-State l-PTCQ, rate = 4 bpp

Two-State Ll-PTCQ, rate = 5 bpp

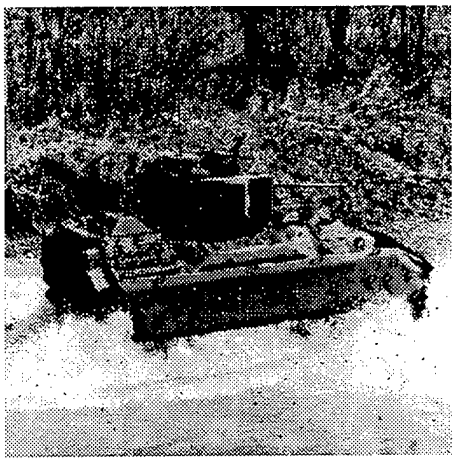
FIGURE 35.—*l-PTCQ and Ll-PTCQ System, Two-State, Varying Rates, Lena Image*



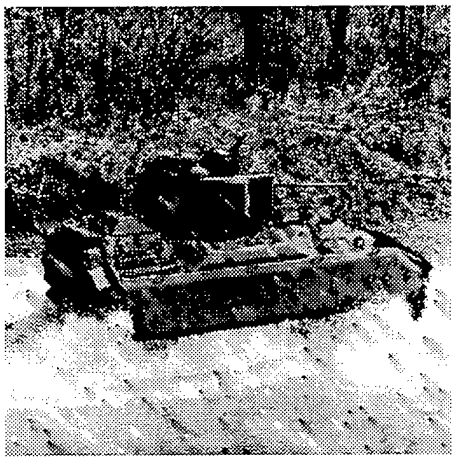
Two-State Ll-PTCQ, rate = 2 bpp



Two-State l-PTCQ, rate = 3 bpp

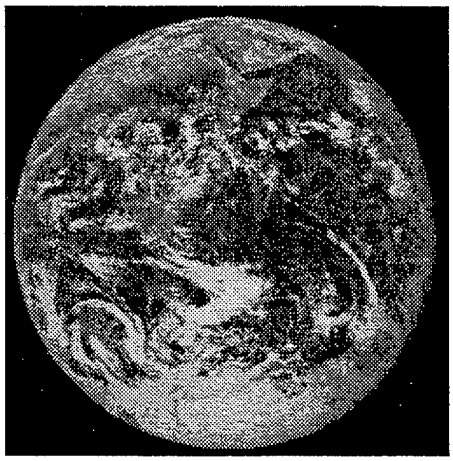


Two-State Ll-PTCQ, rate = 4 bpp

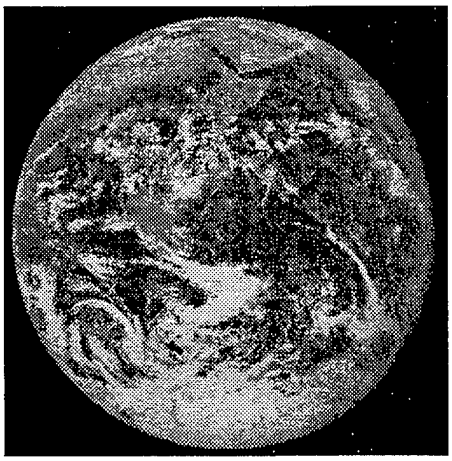


Two-State l-PTCQ, rate = 5 bpp

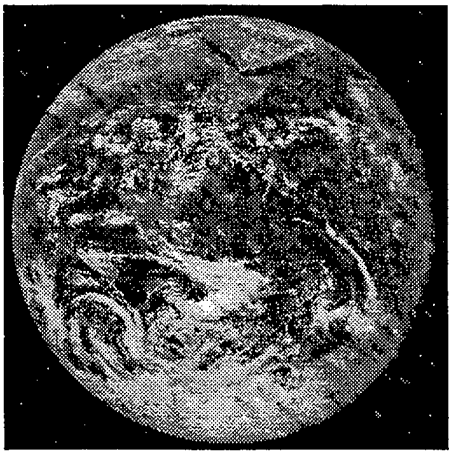
FIGURE 36.—*l-PTCQ and Ll-PTCQ System, Two-State, Varying Rates, M2 Image*



Two-State l-PTCQ, rate = 2 bpp



Two-State Ll-PTCQ, rate = 3 bpp



Two-State l-PTCQ, rate = 4 bpp

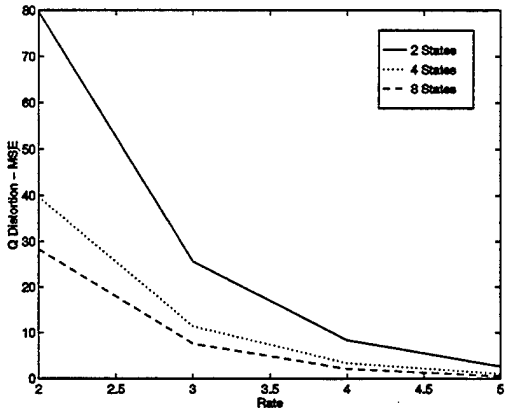
Two-State Ll-PTCQ, rate = 5 bpp

FIGURE 37.—*l-PTCQ and Ll-PTCQ System, Two-State, Varying Rates, Earth Image*

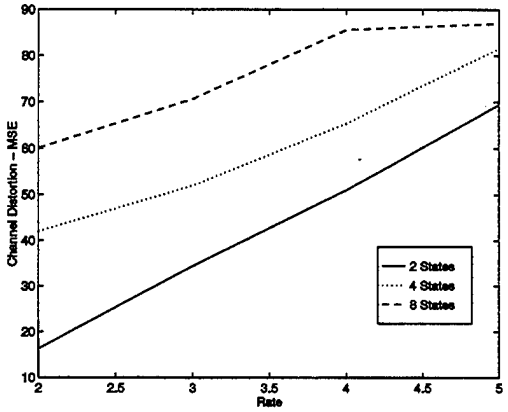
presented utilizing the output images and error tables. However, a more composite evaluation can be performed utilizing graphical representation of the error measurements for each test image. Figures 38 through 40 display distortion vs. rate for the two-, four-, and eight-state l-PTCQ system for each image. Figures 41 through 43 display distortion versus rate for the two-, four-, and eight-state Ll-PTCQ system. As expected, the quantizer distortion is inversely proportional to the bit rate as indicated in each Quantizer Distortion graph. The relationship between the channel distortion and the bit rate demonstrates that the distortion increases as the bit rate increases. This phenomenon is due to the fact that as the bit rate increases, there are more bits to be subjected to channel errors. The total system error curves reflect an operating point for each image where the total system error is minimized. These graphs indicate that the best performing system for the subject images is the two-state PTCQ system operated at a bit rate of about 3 or 4 bpp. This particular system demonstrates the necessary tradeoff between rate and the two cases of quantizer and channel distortion. The two-state system generally outperforms the other systems because of the branch bit error limitation of $1 + \log_2 N$, where N is the number of trellis states. The performance at 3 and 4 bpp demonstrates the balance between increasing the bit rate for lower quantization error and decreasing the bit rate for decreased channel distortion. When viewing the system as a whole by comparing the resultant images to the originals, the product of the best-performing system is obvious.

4.4 Performance Over Very Noisy Channels

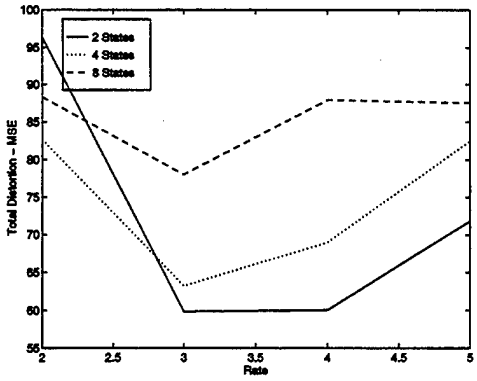
In the previous subsections, it has been shown that l-PTCQ and Ll-PTCQ provide good performance in noisy channels. In addition, results indicate that the two-state systems operating at rate of 3 or 4 bpp provide the best overall performance of all configurations in the noisy channel. This subsection explores the noise mitigation capabilities of these systems at high noise levels, namely $\epsilon = 0.01$ and $\epsilon = 0.05$. Figure 44 and 45 represent the channel distortion and total distortion vs. rate when the channel noise level is $\epsilon = 0.01$, for both l-PTCQ and Ll-PTCQ. From these graphs, the best operating point remains the two-state PTCQ system at rate 3 bpp. However, when the noise level is increased to $\epsilon = 0.05$, the best performing system is the two-state PTCQ system with a bit rate of 2 bpp as demonstrated in Figures 46 and 47. Figure 48 displays the best resultant images in the very noisy environments. It can be further concluded from the images in subsection 4.3 and Figure 48 that the nonlinear Ll-PTCQ is the most effective of the two systems in combating the effects of a noisy environment. This is due largely to the Ll filter's nonlinear properties, which eliminate outliers caused by channel errors, thereby minimizing error propagation.



Quantizer Error

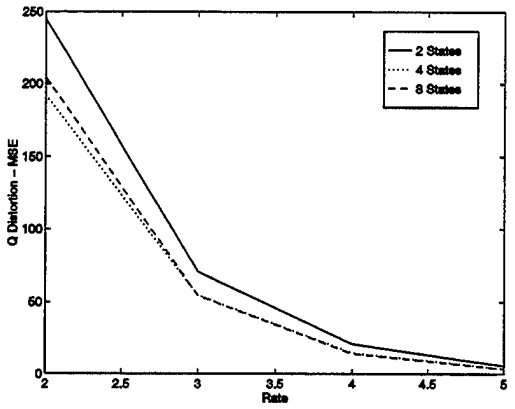


Channel Error

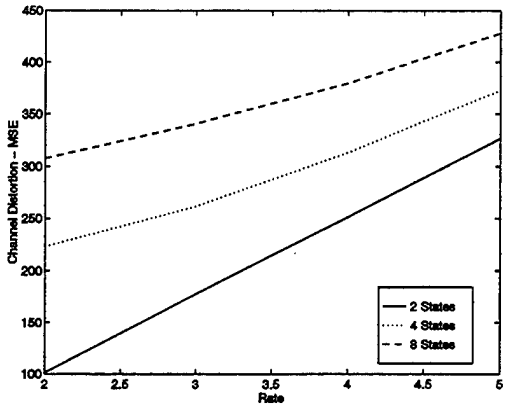


Total Error

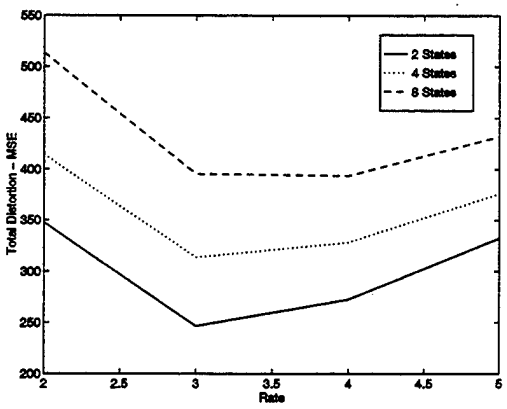
FIGURE 38.—*l-PTCQ, Lena Image*



Quantizer Error

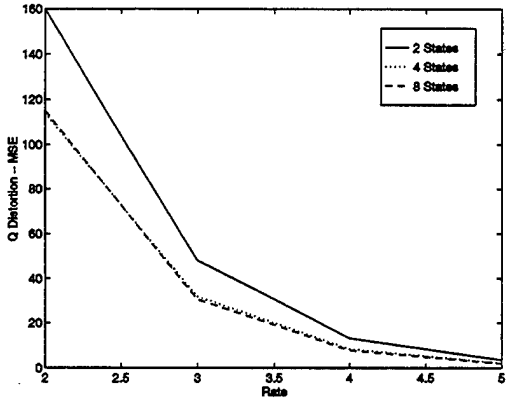


Channel Error

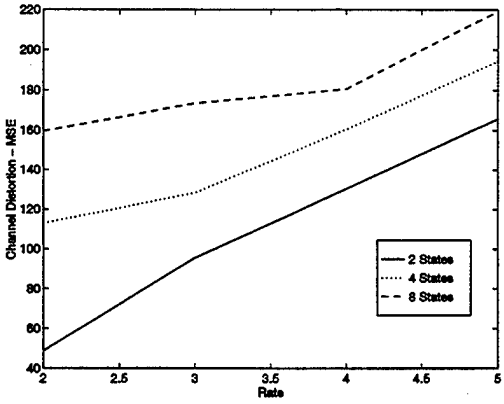


Total Error

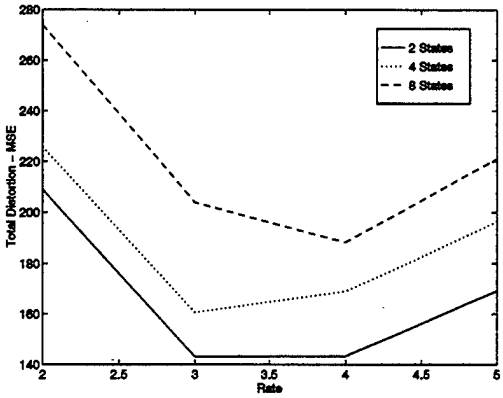
FIGURE 39.—*l-PTCQ, M2 Image*



Quantizer Error -

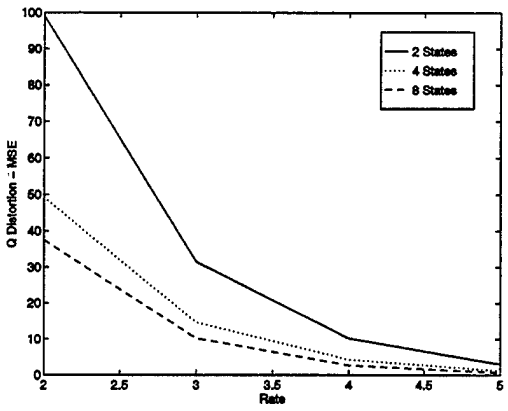


Channel Error

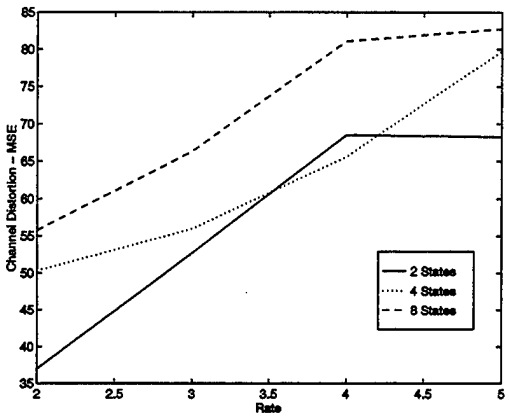


Total Error

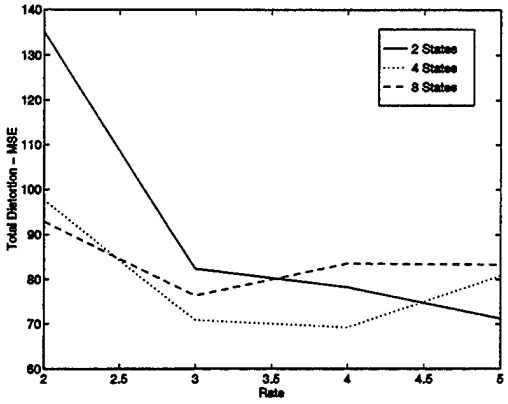
FIGURE 40.—*l-PTCQ, Earth Image*



Quantizer Error

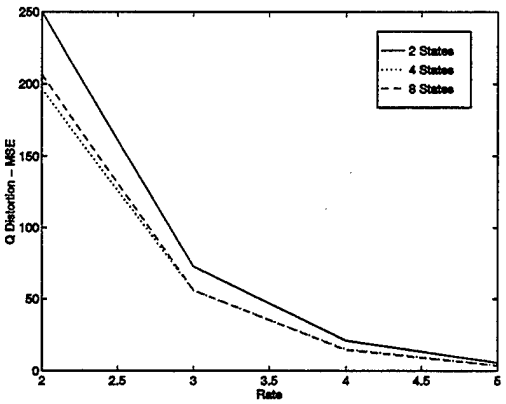


Channel Error

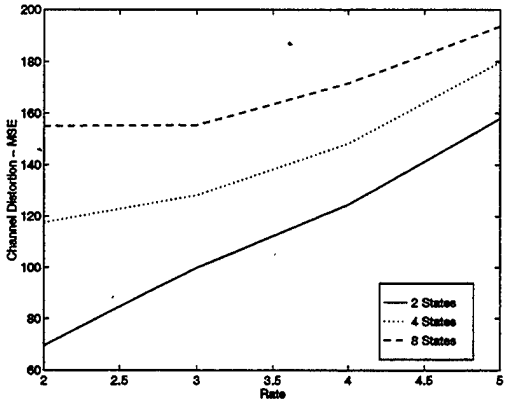


Total Error

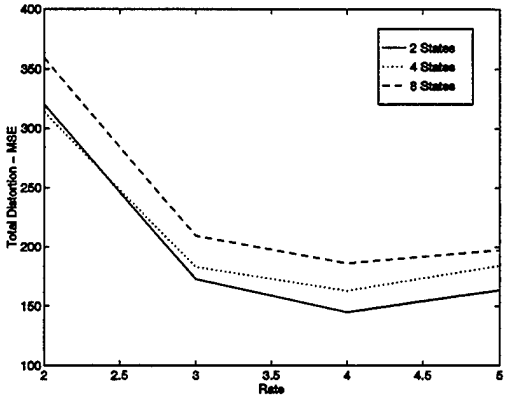
FIGURE 41.—*Ll-PTCQ, Lena Image*



Quantizer Error

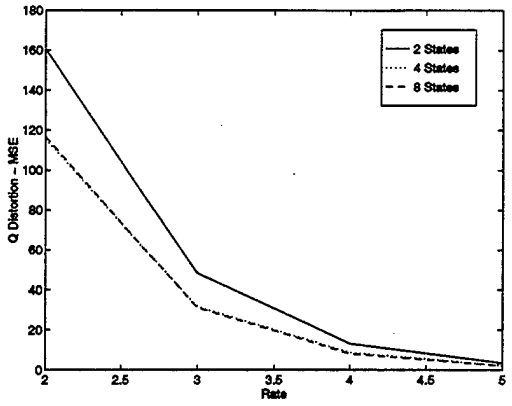


Channel Error

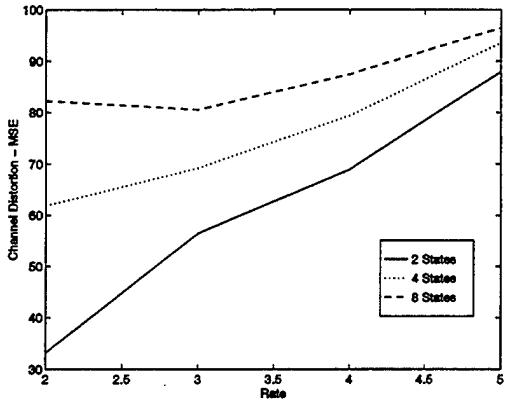


Total Error

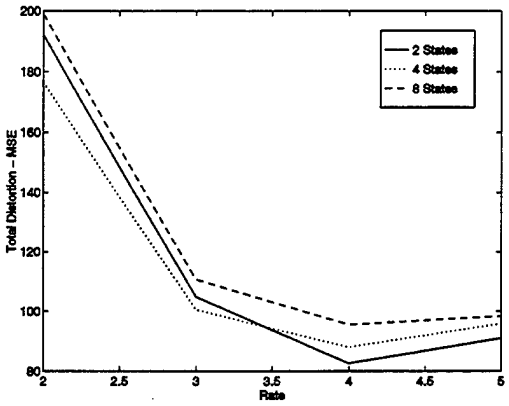
FIGURE 42.—*Ll-PTCQ, M2 Image*



Quantizer Error

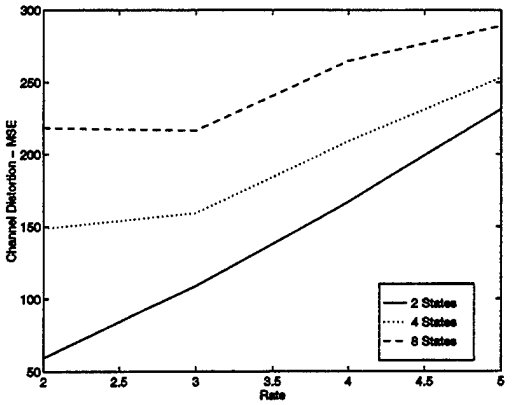


Channel Error

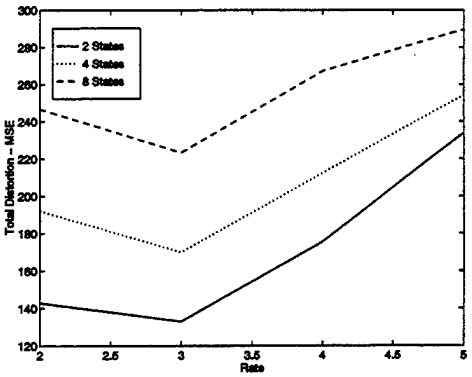


Total Error

FIGURE 43.—*Ll-PTCQ, Earth Image*

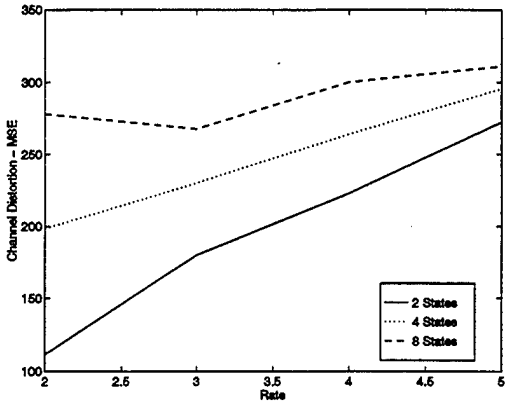


Channel Error

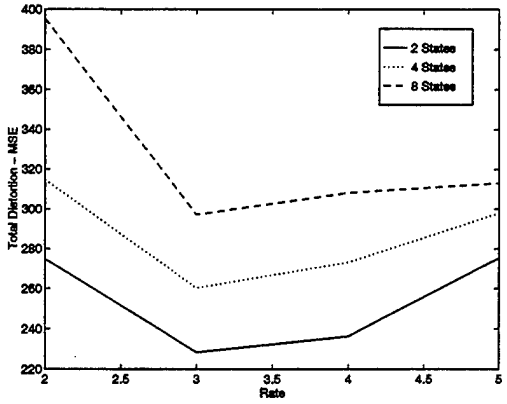


Total Error

FIGURE 44.—*l-PTCQ Performance Lena Image* $\epsilon = 0.01$

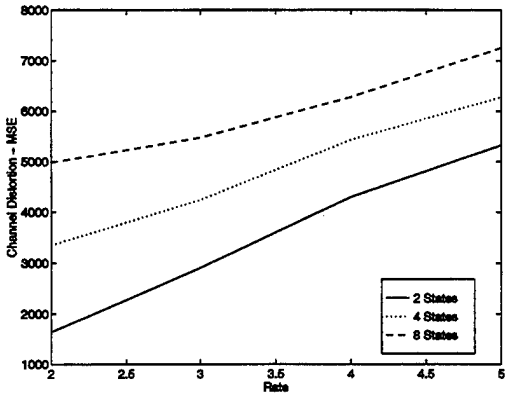


Channel Error

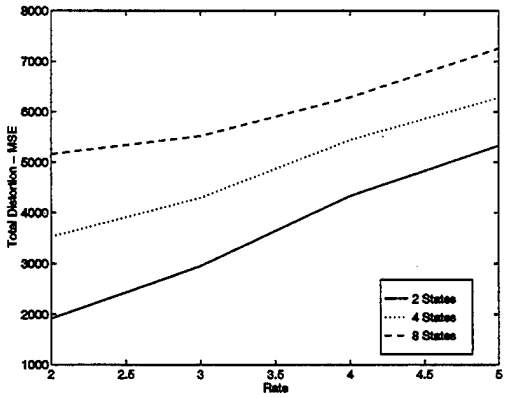


Total Error

FIGURE 45.—*Ll-PTCQ Performance Earth Image* $\epsilon = 0.01$

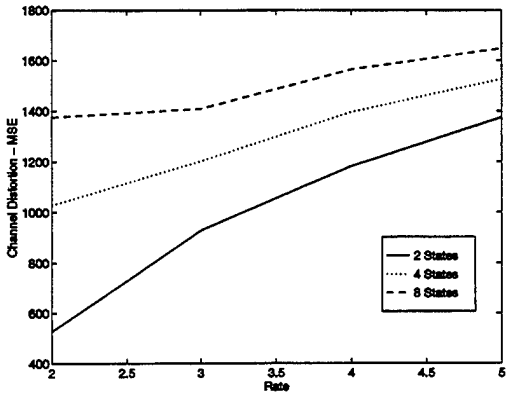


Channel Error

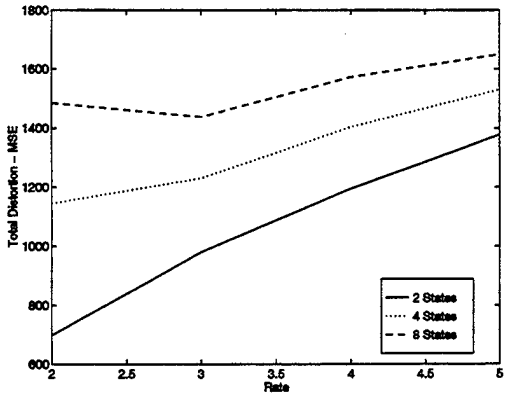


Total Error

FIGURE 46.—*l-PTCQ Performance M2 Image* $\epsilon = 0.05$



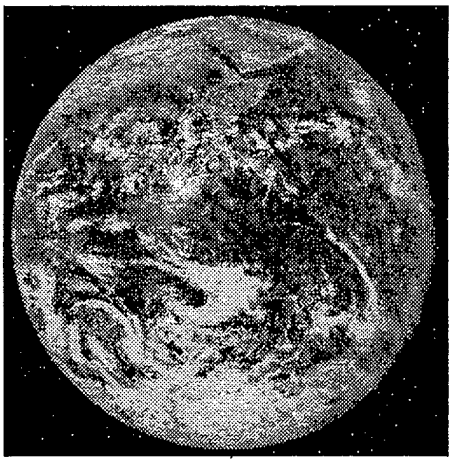
Channel Error



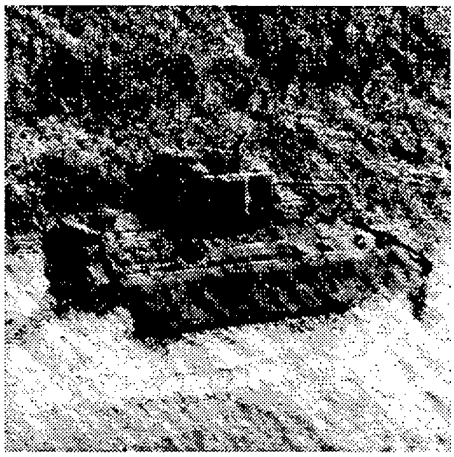
Total Error

FIGURE 47.—*Ll-PTCQ Performance Earth Image* $\epsilon = 0.05$

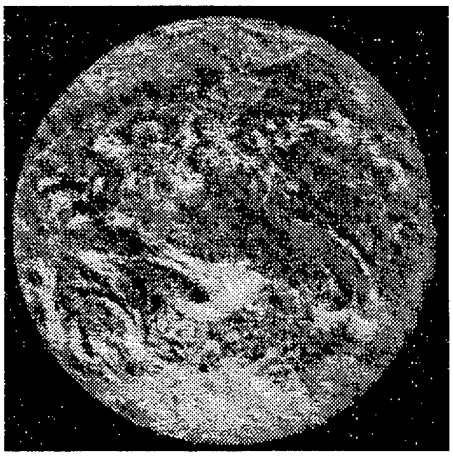
Two-State l-PTCQ 3 bpp,
$\epsilon = 0.01$, Lena



Two-State Ll-PTCQ 3 bpp,
$\epsilon = 0.01$, Earth



Two-State l-PTCQ 2 bpp,
$\epsilon = 0.05$, M2



Two-State Ll-PTCQ 3 bpp,
$\epsilon = 0.05$, Earth

FIGURE 48.—*Very Noisy PTCQ Performance*

5. Conclusions

This technical report has presented PTCQ with linear and nonlinear prediction filters as a method for *robust* source coding. This technique provides both the compression and noise mitigation necessary for image transmission over noisy channels. TCQ has been reviewed, and it has been demonstrated that when feedback-free convolutional encoders are employed, error propagation is limited. By evaluating PTCQ systems of various configurations, it has been determined that all prediction filters perform well in a noiseless environment. However, when the image is transmitted over a noisy channel, the optimal-linear and nonlinear PTCQ systems demonstrate the most noise immunity. Variations in the PTCQ configuration, namely changing bit rate and the number of trellis states, further exhibit that the best configuration for the noisy channel is the two-state PTCQ with rates of 3-4 bpp. When the noise was increased to a high level, it was shown that the better performing configuration is the two-state PTCQ algorithm operating at a bit rate of 2 bpp. Judging from the quality of the images, the Ll-PTCQ system provides better noise mitigation over the l-PTCQ system with identical configuration variables. The Ll filter provides the nonlinearity necessary to conceal the effects of noise and minimize the propagation of error.

As recommendations for future work to provide even more robust source coding, the following avenues could be pursued. An error-correcting code could be employed to protect the encoded bit stream from corruption by noise. Since errors in the branch bit result in more serious repercussions than the scalar quantizer bits, an error correcting code that protects the branch bit more than the other bits, such as the (3,5;2) code could be used with a modest increase in bit rate [17]. A further enhancement of the algorithm would be the mapping of scalar quantization points so that the Euclidean distance *and* Hamming distance were proportional. This would cause quantization points that are close to one another to have a small Hamming distance. Conversely, quantization points that are far apart would have a large Hamming distance. A single bit error would cause a quantization point to be mapped to a close neighbor quantization point and therefore minimize the magnitude of error [18]. Ll-PTCQ incorporating these design criteria would possess an improved noise immunity.

6. References

[1] Shannon, C. E. "A Mathematical Theory of Communication." The Bell Systems Technical Journal, vol. XXVII, no. 3, July 1948.

[2] Cover, T. M., and J. A. Thomas. Elements of Information Theory. New York: John Wiley and Sons, Inc., 1991.

[3] Ayanoğlu, E., and R. M. Gray. "The Design of Joint Source and Channel Trellis Waveform Coders." IEEE Transactions on Information Theory, vol. 33, no. 6, pp. 855–865, November 1987.

[4] Viterbi, A. J. "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm." IEEE Transactions on Information Theory, vol. IT-13, pp. 260–269, April 1967.

[5] Forney, G. D., Jr. "The Viterbi Algorithm." Proceedings IEEE, vol. 61, no. 3, pp. 268–278, March 1973.

[6] Lin, S., and C. J. Costello, Jr. Error Control Coding: Fundamentals and Applications. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

[7] Ungerboeck, G. "Channel Coding with Multilevel/Phase Signals." IEEE Transactions on Information Theory, vol. 28, no. 1, pp. 55–67, January 1982.

[8] Ungerboeck, G. "Trellis-Coded Modulation with Redundant Signal Sets, Parts 1 and 2." IEEE Communications Magazine, vol. 25, no. 2, pp. 5–21, February 1987.

[9] Gonzalez, R. C., and R. E. Woods. Digital Image Processing. New York: Addison-Wesley Publishing, 1992.

[10] Lloyd, S. P. "Least Squares Quantization in PCM." Unpublished memorandum, Bell Laboratories, 1957.

[11] Max, J. "Quantizing for Minimum Distortion." IRE Transaction on Information Theory, vol. 6, no. 7–12, 1960.

[12] Jain, A. K. Fundamentals of Digital Image Processing. Englewood Cliffs, NJ: Prentice Hall, 1989.

[13] Marcellin, M. W., and T. R. Fischer. "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources." IEEE Transactions on Communications, vol. 38, no. 1, pp. 82–93, January 1990.

[14] Wang, M., and T. R. Fischer. "Trellis-Coded Quantization Designed for Noisy Channels." IEEE Transactions on Information Theory, vol. 40, no. 6, pp. 1792–1802, November 1994.

[15] Marcellin, M. W., and T. R. Fischer. "Predictive Trellis Coded Quantization of Speech." IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 1, pp. 46–55, January 1990.

[16] Palmieri, F., and C. G. Boncelet, Jr. "L*l*-filters - A New Class of Order Statistic Filters." IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 5, pp. 691–701, May 1989.

[17] Khayrallah, A. S. "On the Minimum Bit-Error Rate of Linear Codes." IEEE Transactions on Information Theory, vol. 41, no. 5, pp. 1457–1466, September 1995.

[18] Girod, B., B. Belzer, and J. D. Villasenor. "Joint Source Channel Coding of Images with Trellis Coded Quantization and Convolutional Codes." Presented at the IEEE International Conference on Image Processing, Washington, DC, October 1995.

Appendix A:

Example of the Viterbi Algorithm

As a description of the Viterbi algorithm, the following example is provided. By assuming the code is linear, we can begin in state 1 and end in state 1. The cost incurred traversing a branch is indicated in the parentheses, $(\cdot)$, and the cumulative costs is labeled at each state. The cost, ρ, is defined as the Euclidean distance (38), where $\mathbf{x}$ is the output bits generated by the convolutional code and $\hat{\mathbf{x}}$ is the received sequence from a noisy channel.

$$\rho(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{(x - \hat{x})^2}. \tag{38}$$

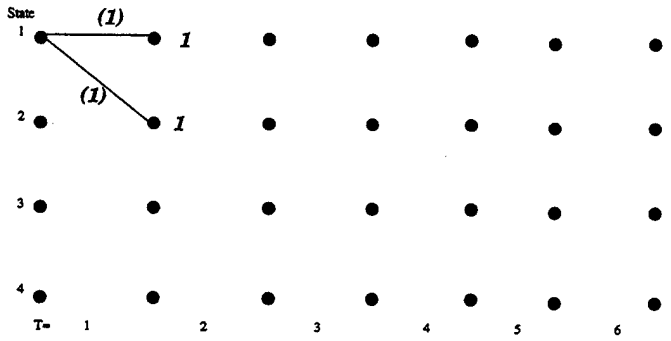


FIGURE A-1.—*Viterbi Algorithm - First Stage. (Compute the cost, ρ, incurred traversing the branches that emanate from state 1. This cost becomes the cumulative cost at Stage two for states 1 and 2, respectively.)*

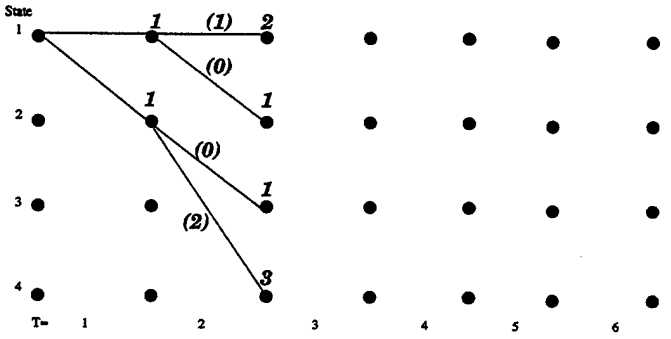


FIGURE A-2.—*Viterbi Algorithm - Second Stage. (Compute ρ for each branch and add to the cumulative cost. This is now the cumulative cost at Stage three.)*

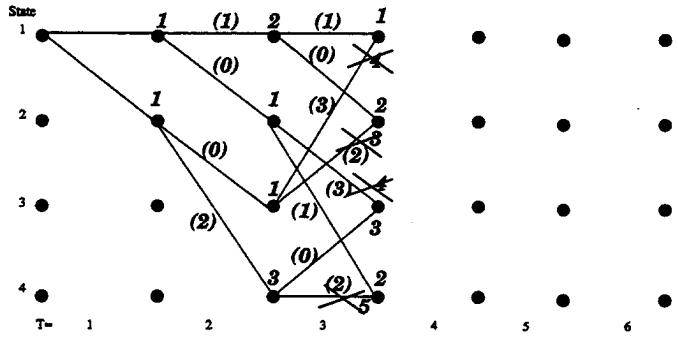


FIGURE A-3.—*Viterbi Algorithm - Third Stage. (Repeat process performed at state two. Select branch entering a state with the least cumulative distortion as the survivor and discard all other branches. If the entering branches have equivalent cumulative distortion, the survivor can be selected arbitrarily.)*

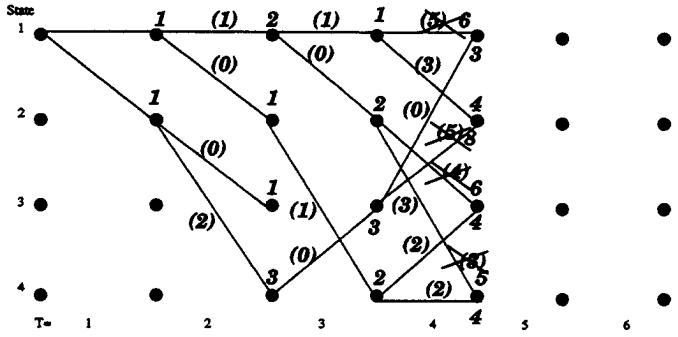


FIGURE A-4.—*Viterbi Algorithm - Fourth Stage. (Repeat as in Stage three.)*

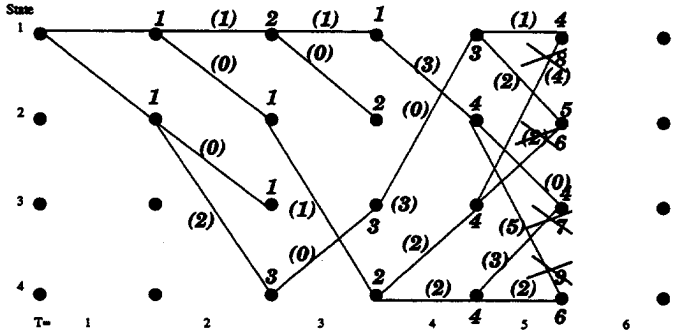


FIGURE A-5.—*Viterbi Algorithm - Fifth Stage. (Repeat as in Stage three.)*

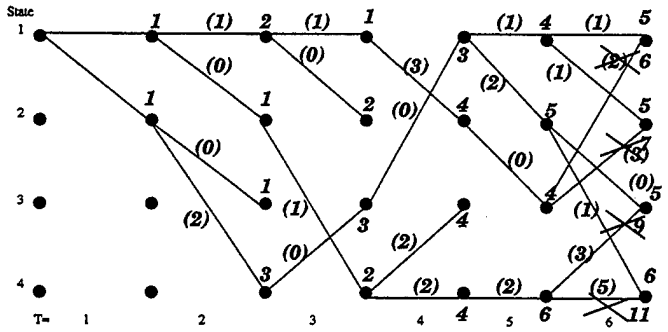


FIGURE A-6.—*Viterbi Algorithm - Sixth Stage. (Repeat as in Stage three.)*

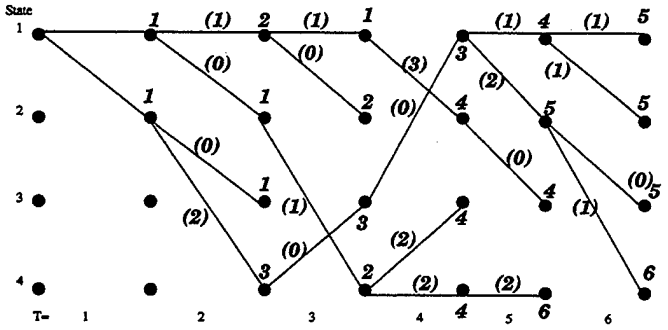


FIGURE A-7.—*Viterbi Algorithm - Final Stage. (At this final stage of the Viterbi Algorithm, one survivor path is displayed for each state.)*

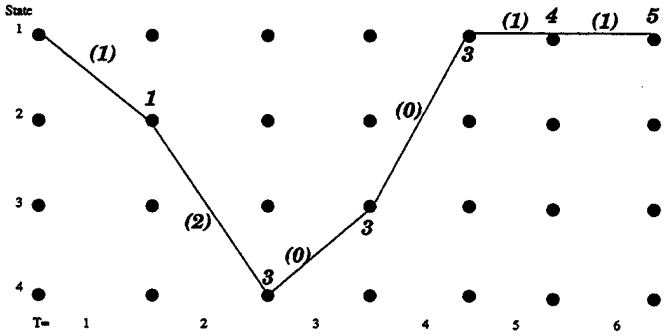


FIGURE A-8.—*Viterbi Algorithm - Maximum Likelihood Path. (At this final step, select the branch with minimum cumulative cost. Trace path backward to state one. The resultant path is the maximum likelihood path.)*

Appendix B:

Linear and Nonlinear Filter Coefficients

TABLE B-1.—*Linear Filter Coefficients - All Images*

Image	α_1	α_2	α_3
Lena	-.1713	.7632	.3850
M2 Bradley	.1365	.4020	.4127
Earth	.1871	.3441	.4202

TABLE B-2.—*Nonlinear Ll-filter Coefficients - Lena Image*

i	J	α_1^i	α_2^i	α_3^i	α_4^i	α_5^i
1	{1,2}	0	0	.7158	-.0828	.3411
2	{1,3}	0	.2767	0	-.1751	.8240
3	{1,4}	0	-.1713	.7632	0	.3850
4	{1,5}	0	.0432	.8643	.0668	0
5	{2,3}	.4290	0	0	-.1607	.6901
6	{2,4}	-.1610	0	.8620	0	.2738
7	{2,5}	-.1702	0	1.0318	.1083	0
8	{3,4}	.4750	-.1866	0	0	.6799
9	{3,5}	.5254	.3625	0	.0621	0
10	{4,5}	-.1996	.1581	1.0059	0	0

TABLE B-3.—*Nonlinear Ll-filter Coefficients - M2 Image*

i	J	α_1^i	α_2^i	α_3^i	α_4^i	α_5^i
1	{1,2}	0	0	.4154	.1769	.3666
2	{1,3}	0	.2866	0	.2215	.4285
3	{1,4}	0	.1365	.4020	0	.4127
4	{1,5}	0	.2100	.4443	.2956	0
5	{2,3}	.2957	0	0	.2147	.4429
6	{2,4}	.1493	0	.3733	0	.4340
7	{2,5}	.1788	0	.4425	.3312	0
8	{3,4}	.2649	.2191	0	0	.4640
9	{3,5}	.2948	.3148	0	.3361	0
10	{4,5}	.1906	.3002	.4407	0	0

TABLE B-4.—*Nonlinear Ll-filter Coefficients - Earth Image*

i	J	α_1^i	α_2^i	α_3^i	α_4^i	α_5^i
1	{1,2}	0	0	.3963	.1567	.4010
2	{1,3}	0	.3245	0	.1605	.4515
3	{1,4}	0	.1871	.3441	0	.4202
4	{1,5}	0	.2544	.4152	.2770	0
5	{2,3}	.2833	0	0	.1868	.4814
6	{2,4}	.1473	0	.3449	0	.4624
7	{2,5}	.1788	0	.4374	.3332	0
8	{3,4}	.2193	.2551	0	0	.4758
9	{3,5}	.2663	.3567	0	.3198	0
10	{4,5}	.1647	.3462	.4146	0	0

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFO CTR ATTN DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL OP SD TA 2800 POWDER MILL RD ADELPHI MD 20783-1145
3	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL OP SD TL 2800 POWDER MILL RD ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LAB ATTN AMSRL OP SD TP 2800 POWDER MILL RD ADELPHI MD 20783-1145
	ABERDEEN PROVING GROUND
2	DIR USARL ATTN AMSRL OP AP L (305)

NO. OF COPIES	ORGANIZATION
4	COMMANDER US ARMY TACOM ATTN AMSTA TA D P HANIACK MS 201B WARREN MI 48397
1	COMMANDER US ARMY TACOM ATTN AMSTA TR R K ADAMS MS 264 WARREN MI 48397
4	COMMANDER US ARMY ATCOM ATTN AMSAC R TD CC R WALL FORT EUSTIS VA 23604-1104
2	COMMANDER US ARMY CECOM ATTN AMSEL RD C3 AC P SASS FORT MONMOUTH NJ 07703
2	COMMANDER US ARMY CECOM ATTN NVESD DR GARD D II FORT BELVOIR VA 22060-5806
1	DIRECTOR US ARMY RSRCH OFC ATTN AMXRO EL DR W SANDER PO BOX 12211 RESEARCH TRIANGLE PARK NC 22709-2211
1	COMMANDANT US MILITARY ACADEMY WEST POINT NY 10996
1	COMMANDANT US NAVAL ACADEMY ANNAPOLIS MD 21404
1	COMMANDER NAVAL AIR WARFARE CENTER ATTN S GATTIS 1 ADMIN CIRCLE CHINA LAKE CA 93555-6001

NO. OF COPIES	ORGANIZATION
1	COMMANDANT USAF ACADEMY COLORADO SPRINGS CO 80840
1	DIRECTOR NAVAL RSRCH LAB WASHINGTON DC 20375-5000
1	ERICSSON INC ADV DEV & RSRCH GRP ATTN DR A KHAYRALLAH RESEARCH TRIANGLE PARK NC 27709
1	UNIVERSITY OF DELAWARE DEPT OF ELECTRICAL ENGRNG ATTN PROF C BONCELET JR NEWARK DE 19716
	ABERDEEN PROVING GROUND, MD
1	CDR, USATECOM ATTN: AMSTE
37	DIR, USARL ATTN: AMSRL-CI-AC, MR. ANGELINI AMSRL-IS, MR. DEMONTE MR. SLIFE DR. EMMERMAN AMSRL-IS-TP, DR. GOWENS DR. COOPER DR. RETTER DR. CHAMBERLAIN DR. TORRIERI DR. SADLER MR. CIRINCIONE MR. HARTWIG MR. CATON MS. LOPEZ MR. BRUNDICK MR. MARKOWSKI MS. SARAFIDIS MR. HAYES MS. MARK AMSRL-IS-ST, MS. BROOME DR. MILLS (3 CPS) MR. KASTE MR. DOWNS MS. BRODEEN MR GWYN MS. WRENCHER

NO. OF
COPIES ORGANIZATION

ABERDEEN PROVING GROUND, MD (Continued)

AMSRL-SC-I,
A. RAGLIN
LT HANRATTY
AMSRL-SC-SA, LTC WALL
AMSRL-SE-RI, S. WELBY
AMSRL-SE-RL, A. MARINELLI
AMSRL-SE-RT, DR. HAMILTON
AMSRL-SL-BV, M. MUSS
AMSRL-WT-WB, DR. D'AMICO (2 CPS)

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. ARL Report Number/Author ARL-TR-1192 (Marvel) Date of Report September 1996

2. Date Report Received _____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. Specifically, how is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

7. If indicating a Change of Address or Address Correction, please provide the Current or Correct address above and the Old or Incorrect address below.

OLD ADDRESS

_____
Organization

_____
Name

_____
Street or P.O. Box No.

_____
City, State, Zip Code

(Remove this sheet, fold as indicated, tape closed, and mail.)
(DO NOT STAPLE)