MOMENT PRESERVING QUANTIZATION AND ITS APPLICATION IN BLOCK TRUNCATION CODING

A Thesis

Submitted to the Faculty

of

Purdue University

by

Edward John Delp III

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 1979

Grad. School
Form No. 9
Revised

PURDUE UNIVERSITY

Graduate School

This is to certify that the thesis prepared

By Edward John Delp III

Entitled MOMENT PRESERVING QUANTIZATION AND ITS APPLICATION IN BLOCK TRUNCATION CODING

Complies with the University regulations and that it meets the accepted standards of the Graduate School with respect to originality and quality

For the degree of:

Doctor of Philosophy

Signed by the final examining committee:

O.R. Mitchell, chairman

George R. Cooper

Raymond De Carlo

T.W. Mullikin

Approved by the head of school or department:

May 10 1979

To the librarian:

~~is~~

This thesis is not to be regarded as confidential

O.R. Mitchell
Professor in charge of the thesis

This thesis is dedicated to the memory of my late father,

Edward J. Delp Jr.,

who in many ways was a better engineer than I will ever be.

ACKNOWLEDGEMENTS

I would like to thank my advisory committee consisting of Professor Mitchell, Professor Cooper, Professor Saridis, Professor Mullikin, and Professor DeCarlo for their help and advice in preparing this thesis. In particular I appreciate the committee attending all the meetings and signing all the forms. I will forever be indebted to Professor Mitchell for his brainstorm of March 17, 1977 when Block Truncation Coding was born. I sincerely appreciate the discussions I had with Professor Studdin of the Statistics Department concerning orthogonal polynomials.

I would also like to thank Dr. John Gamble of the Rome Air Development Center for his advice and criticism of Chapter 3, and Richard R. Petroski and his colleagues of the Rome Research Corporation for providing the expert photo reconnaissance evaluations of Chapter 3. I appreciate the transform and Hybrid coding results provided by Ali Tabatabai and Paul Stiling for the comparison study of Chapter 3.

My typists, Mellanie Boes and Chris Ramsey, always rose to the occasion and met my ridiculous deadlines with professional results. I thank them sincerely.

Finally, to my wife Maureen who put up with all of the lonely nights and weekends while I was out chasing new results, I appreciate the understanding and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Delp III, Edward John. Ph. D., Purdue University, August 1979. Moment Preserving Quantization and its Application in Block Truncation Coding. Major Professor: Owen Robert Mitchell, Jr.

A new criterion for quantization design is presented whereby a quantizer is obtained such that a finite set of moments of the output of the quantizer is identical to that of the input to the quantizer. The general moment preserving quantizer (MP) is shown to be related to the Gauss-Jacobi mechanical quadrature problem used in numerical analysis. The output levels of an N level MP quantizer are shown to be the N zeroes of an Nth degree orthogonal polynomial associated with the input distribution. The N-1 thresholds of the MP quantizer are shown to be related to the Christoffel numbers through the Separation Theorem of Chebyshev-Markov-Stieltjes. The statistical convergence of the MP quantizer is presented whereby convergence in distribution is guaranteed if the input probability distribution is characterized by its moments (i.e., a solution to the Hamburger, Stieltjes, Hausdorff moment problem). Mean square convergence is also investigated. MP quantizer tables are presented for the uniform, normal and Laplacian density functions. The MP quantizer is compared with the minimum mean square error quantizer of Max.

An image coding scheme is presented using a non-parametric formulation of the MP quantizer. This coding technique is known as Block Truncation Coding (BTC). Reconstructed images using BTC are shown to compare very favorably to transform coding at bit rates of 1.63 bits/pixel. BTC is compared with both the non-parametric minimum mean square error and minimum mean absolute error quantizers. The performance of BTC in the presence of channel errors is discussed along with a hybrid formulation. A differential form of BTC is presented at bit rates of 1.18 bits/pixel.

Image modeling is discussed in the context of quarter plane Gaussian-Markov fields. Through appropriate use of initial conditions these models are shown to be related to a seasonal autoregressive time series. These models are fitted to test images; the images are then regenerated using the model and a random number generator. These models are shown to have promise in texture synthesis at low bit rates (0.33 bits/pixel). An application is presented where the model is used to generate crude background scenes with geometric features (edges) displayed independently.

CHAPTER 1

OVERVIEW OF IMAGE COMPRESSION AND STATEMENT OF PROBLEM

1.1. Introduction

Since the beginning of the use of digital techniques in the area of image processing, there has been a desire to find ways of coding images for efficient transmission and/or archival storage. We shall investigate in this thesis a new method of image coding developed at Purdue University over the past two years. This new technique has been given the name Block Truncation Coding (BTC).

It shall be assumed throughout this work that the desired image has already been sampled and quantized to obtain an acceptable discrete representation of the image. It is assumed that this discretized image is to be coded for bandwidth compression. The compression shall be expressed as a bit rate in bits per pixel (picture element). This is obtained by dividing the total number of bits in the picture representation by the total number of pixels. The original discrete images will have a nominal gray level resolution of either 6 or 8 bits.

1.2. Overview of Image Compression

Image coding for bandwidth compression can usually be grouped into two general methods, information and non-information preserving coding [35], [32], [69], [63], [30], [37]. Information preserving coding takes the form of entropy coding. This is accomplished by first obtaining a histogram of the image pixel gray levels (or perhaps differences of pixels) and using optimum code words for the distribution of the gray levels [59]. One may implement an optimum (variable-length) code by using a Huffman or Shannon-Fano code [1], [29]. These codes are cumbersome to implement and one usually uses a suboptimum code [36]. A good review of information preserving coding is presented in [30]. Two distinct disadvantages of entropy coding are the poor performance of these codes in the presence of errors and the fact that reduction in bit rate is not that large (for image data bit rates of 3 bits/pixel are typical).

The area of non-information preserving coding is usually further subdivided into transform and non-transform techniques. In transform coding one first transforms the image usually using a linear orthogonal discrete transform such a Karhunen-Loeve, Fourier, Cosine, Slant, Walsh, or Haar transform [80], [7], [4], [64], [65], [20], [5], [15]. Once the desired transform is obtained, some of the resulting coefficients are discarded and the remaining coefficients are then quantized, coded, and transmitted. The receiver reconstructs the image by inverse transformation. The method of discarding the coefficients is usually one of either thresholding the coefficients or zonal filtering. In the thresholding method one discards coefficients whose amplitudes are below a fixed threshold. The threshold is usually set using a percentage of the total energy contained in the picture. Besides quantizing the magnitude of the coefficients retained, it is also necessary to code the coefficient location. In the zonal filtering method only those coefficients in a fixed zone in the transform domain are retained. This method is somewhat easier and more noise immune since one can avoid the overhead of coding the coefficient locations.

The Cosine transform seems to be a rather robust orthogonal transform for a large class of images and in some cases the performance of this transform resembles that of the Karhunen-Loeve expansion [15]. In most cases for high resolution images, transform coding has a tendency to blur the image. One of the distinct advantages of transform coding is that it is quite easy to obtain data rates below 1 bit/pixel. While the coding artifacts are quite noticeable at data rates below 1 bit/pixel the ease of obtaining these data rates should not be overlooked. An obvious disadvantage of transform coding is the large computational load necessary to first obtain the transform and then operate on the coefficients. Some transform coding techniques such as the Chen and Smith method [16] require multiple passes through the transform coefficients to collect statistics. Transform coding also usually requires sophisticated error protection for the coding of overhead information (i.e., coefficient assignment tables and bit maps) when the image is transmitted over a noisy channel.

The second method of non-information preserving image compression, non-transform techniques, exploit some local properties of the image. Predictive coding (DPCM) [58], [17] is a technique where the local correlation properties of the image are used to obtain bandwidth compression. In DPCM the difference between the actual pixel gray level and a predicted pixel value is quantized, coded and transmitted. At the receiver (decoder) the image is reconstructed by using the quantized difference signal and the predictor model. Using this predictive method it is usually not possible to obtain results less than 1 bit/pixel unless the difference signal is modified in some sense. It is possible to formulate some predictive coding schemes as really being a variation of transform coding [8], however these types of predictive coding do not exploit all the properties of the image. These predictor models belong to a very small class of predictors. An advantage of predictive coding is the fact that a transform is not required. A disadvantage is that of obtaining adequate predictor and quantizer models. Block Truncation Coding belongs to this class of non-transform techniques.

Hybrid coding [33] tries to exploit the benefits of both the transform and predictive methods. In this method transform coding is performed along the rows (or columns) of the picture and predictive coding along the columns (or rows) of the image. This method can be used to obtain data rates less than 1 bit/pixel.

It should be mentioned that the coding artifacts of all the non-information preserving techniques are indeed different and in some cases this prevents accurate comparisons between the methods.

1.3. Statement of Problem

Block Truncation Coding (BTC) will be formulated as a problem of obtaining an adaptive two-level (one bit) quantizer such that a set of sample moments of the image are preserved. If one could obtain a fidelity criterion for an image that would somehow represent the desired properties of an image (i.e. edge preservation, no false contouring, texture preservation) then rate-distortion theory could be used to obtain a one bit quantizer to match this fidelity criterion [11], [14], [75], [54], [70], [53]. This is indeed a Herculean effort and will not be pursued in this thesis.

In this thesis we will examine quantizers that preserve moments of the input signal. We will find that these types of quantizers can be related to the Gauss-Jacobi mechanical quadrature problem where the output levels are the zeroes of the orthogonal polynomials associated with the input probability distribution. The quantizer threshold levels will be obtained by the so-called Separation Theorem. The uniqueness of the quantizer will be guaranteed by the above theorems. The statistical convergence of these quantizers will be investigated and the quantizer compared to the mean square error quantizer of Max [46].

BTC will be examined in this context except the quantizer will have a non-parametric formulation where the one-bit quantizer can be written in closed form. We will examine BTC in the presence of channel errors and a hybrid method of BTC will be discussed. This will be accomplished by the use of supplemental information such as a highly compressed transform coded image as a means of supplying a better low-frequency response to BTC. BTC will be compared to non-parametric mean square error and mean absolute error quantizers. A differential form of BTC will be discussed and image modeling will be examined. This quantizer scheme produces images that appear to be enhanced and the edge locations are preserved. BTC appears to be superior to transform coding in the sense that the compressed image is not blurred (i.e. edge preservation) and BTC requires no great computational effort as does transform coding. This technique applies to most situations when images must be either transmitted or stored.

CHAPTER 2

MOMENT PRESERVING QUANTIZATION

2.1 Introduction

Since the advent of the use of pulse code modulation (PCM) systems there has been great interest in the design of quantizers. It became obvious very early that non-uniform quantizers possessed properties that could be used to achieve results such as lower mean square error or enhanced subjective performance in areas such as speech and image processing [61],[41],[71]. These types of quantizers are designed for a particular input probability density function.

Optimal quantizers are designed relative to a particular performance index or fidelity criterion. The most popular fidelity criterion used is that of the mean square error (MSE) between the input and output with the quantizer found to minimize this mean square error [46]. Recently, some interest has also been shown using the mean absolute error criterion [43]. Studies have shown that both of these fidelity criteria cannot be used reliably in areas such as speech and image processing [66],[50].

In this chapter we will examine quantizers that preserve moments of the input signal. We will compare this technique to standard quantization design such as minimum mean square error quantizers. We will show that quantizers which preserve moments are easy to derive in closed form when the input density is symmetric and the number of levels is relatively small. We will further show that the moment preserving quantization problem can be formulated as the classical Gauss-Jacobi mechanical quadrature problem where the output levels of the quantizer are the zeroes of orthogonal polynomials associated with the input probability distribution. The thresholds of the quantizer are then related to the so-called Christoffel numbers.

2.2 The General Moment Preserving Quantizer

We will approach the problem of using the moment reserving (MP) fidelity criterion by first examining the problem of a two level MP quantizer and then generalize the result to N levels. The notation used here is that of Max [46].

Let X denote the input to the quantizer whose probability distribution function is $F(x)$, $x \varepsilon [a,b]$. The interval $[a,b]$ can be finite, infinite, or semi-infinite. Let Y denote the output of the quantizer. For a two level quantizer, the random variable Y is discrete and takes on the values $\{y_1,y_2\}$ with probabilities $P_1 = \text{Prob}(y=y_1)$ and $P_2 = \text{Prob}(y=y_2)$. The output Y takes on the value y_1 whenever the input x is below some threshold x_1 otherwise the output is y_2. This is shown in Figure 2.1. Therefore in general to design any two-level quantizer one must choose the two output levels y_1 and y_2 and the input threshold x_1. When using a design criterion of having the two-level quantizer preserve moments of the input it is necessary that the quantizer preserve the first three moments of the input, otherwise one of the three parameters would have to be known (or guessed) initially. To specify the quantizer one must solve the following equations for y_1, y_2,

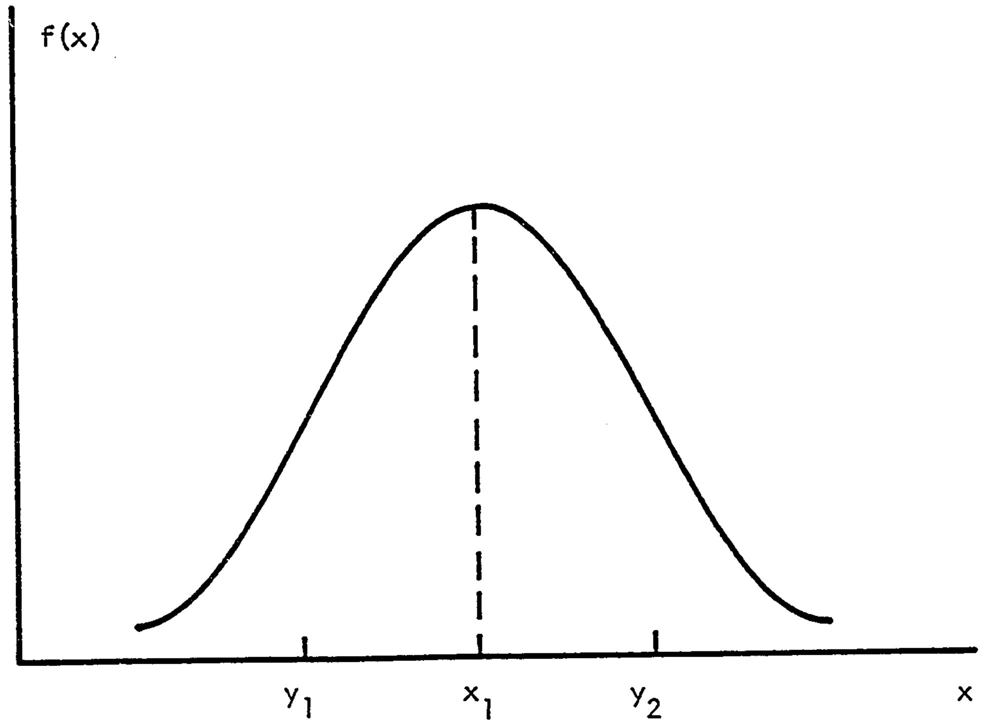

Figure 2.1 In designing any one bit quantizer one must find a threshold x_1 and two output levels y_1 and y_2. f(x) is the probability density function of X.

and x_1:

$$\begin{aligned} E[Y] &= E[X] = y_1P_1 + y_2P_2 \\ E[Y^2] &= E[X^2] = y_1^2P_1 + y_2^2P_2 \\ E[Y^3] &= E[X^3] = y_1^3P_1 + y_2^3P_2 \\ P_1 &+ P_2 = 1 \end{aligned} \tag{2.1}$$

where the expectation operator is defined by the Lebesgue-Stieltjes integral $E[X^i] = \int_a^b x^i dF(x)$ and $P_i = \text{Prob}(Y=y_i)$, $y_1 \leq x_1 \leq y_2$.

We shall assume throughout this presentation that the moments exist and are finite. The total variation of F(x) is of course identically equal to one. We will ignore the "defective" probability measure discussed by Feller [26]. For the case where F(x) is absolutely continuous a probably density function f(x) is admitted. The function f(x) is a non-negative function measurable in the Lebesgue sense. Equation 2.1 can be rewritten as:

$$\begin{aligned} m_1 &= y_1F(x_1) + y_2(1-F(x_1)) \\ m_2 &= y_1^2F(x_1) + y_2^2(1-F(x_1)) \\ m_3 &= y_1^3F(x_1) + y_2^3(1-F(x_1)) \end{aligned} \tag{2.2}$$

where $m_i = E[X^i]$

$P_1 = P(X < x_1) = F(x_1)$

$P_2 = P(X \geq x_1) = 1-F(x_1)$

By solving Equation 2.2 for y_1, y_2, and x_1 the quantizer obtained is such that the first three moments of X and Y are identical. Equation 2.2 is a set of nonlinear algebraic equations in y_1, y_2 and $F(x_1)$. To find x_1 we shall assume $F^{-1}(x_1)$ exists.

Without loss of generality we shall further assume that $m_1 = 0$ and $m_2 = 1$, this amounts to assuming X is zero mean and unit variance; Equation 2.2 becomes

$$\begin{aligned} 0 &= y_1 F(x_1) + y_2(1-F(x_1)) \\ 1 &= y_1^2 F(x_1) + y_2^2(1-F(x_1)) \qquad (2.3) \\ m_3 &= y_1^3 F(x_1) + y_2^3(1-F(x_1)) \end{aligned}$$

By solving the first two equations for y_1 and y_2 in terms of $F(x_1)$ and using these solutions in the last equation we arrive at the desired results:

$$\begin{aligned} y_1 &= -\sqrt{\frac{1-F(x_1)}{F(x_1)}} = -\sqrt{\frac{P_2}{P_1}} \\ y_2 &= \sqrt{\frac{F(x_1)}{1-F(x_1)}} = \sqrt{\frac{P_1}{P_2}} \qquad (2.4) \\ \\ F(x_1) &= \frac{1}{2} + \frac{m_3}{2}\sqrt{\frac{1}{4+m_3^2}} \end{aligned}$$

This result is interesting in that the quantizer can be written in closed form. When using other fidelity criterion such as mean square error it is usually impossible to arrive at some sort of closed form expression for the quantizer with a general density function. The above result in Equation 2.4 also indicates that the threshold, x_1, is nominally the median of X and not the mean as one would expect. The third moment, m_3, is in general a signed number and can be thought of as a measure of skewness in the density function. The above result indicates that the threshold is biased above or below the median according to the sign and magnitude of this skewness. It should be noted that at this point we have no guarantee that $y_1 \leq x_1 \leq y_2$. Particularly we can not yet state that x_1 lies between y_1 and y_2. The problem will be addressed in Section 2.4 by the Separation Theorem of Chebyshev Markov-Stieltjes.

The MP quantizer can be generalized to the N-level quantizer. One needs to recognize that for the N-level quantizer there are N output levels and N-1 thresholds. So if we desire an N-level MP quantizer we need to know the first 2N-1 moments, i.e., the N-level MP quantizer preserves 2N-1 moments. This statement will be shown in Section 2.4 to guarantee uniqueness of the quantizer by the Gauss-Jacobi mechanical quadrature theorem. For large N this does lead to the problem of knowing a large set of moments for a given distribution. However for most distribution functions we are interested in, one can exploit recursion relationships among the moments. This will be discussed later.

To arrive at the desired quantizer we need to know N output levels $\{y_1,y_2,\ldots,y_N\}$ and N-1 thresholds $\{x_1,\ldots,x_{N-1}\}$; with $y_1 \leq x_1 \leq y_2 \ldots \leq x_{N-1} \leq y_N$. We again assume $m_1 = 0$ and $m_2 = 1$. We must solve:

$$m_n = \int_a^b x^n dF(x) = \sum_{i=1}^{N} y_i^n P_i$$

$$n = 0, 1, 2, \ldots, 2N-1 \tag{2.5}$$

where $x_0 = a$

$x_N = b$

$m_n = E[X^n]; \; m_1 = 0, \; m_2 = 1$

$P_i = F(x_i) - F(x_{i-1})$

$F(x_i) = \sum_{j=1}^{i} P_j$

For a large class of practical problems where F(x) admits a density f(x), one can assume that f(x) is even, i.e., f(x) = f(-x). For this assumption the complexity of Equation 2.5 is cut in half since $m_n \equiv 0$ for n odd and the quantizer itself is symmetric. The symmetry of the quantizer manifests itself as:

N even

$$y_i = -y_{N+1-i} \qquad i = 1, 2, \ldots, \frac{N}{2}$$

$$x_{\frac{N}{2}} \equiv 0 \rightarrow F(x_{\frac{N}{2}}) = \frac{1}{2}$$

$$x_k = -x_{N-k} \qquad k = 1, 2, \ldots, (\frac{N-2}{2})$$

N odd

$$y_{\frac{N+1}{2}} \equiv 0$$

$$y_i = -y_{N+1-i} \qquad i = 1, 2, \ldots, (\frac{N-1}{2})$$

$$x_k = -x_{N-k} \qquad k = 1, 2, \ldots, (\frac{N-1}{2})$$

Using the above results and the fact that the odd moments are zero it is obvious that only the even moment equations must be solved in Equation 2.5. Hence Equation 2.5 becomes:

$$\begin{aligned} m_n &= \sum_{i=1}^{M} 2y_i^n \,(F(x_i) - F(x_{i-1})) \\ &\quad n = 2, 4, 6, \ldots, 2N-2 \\ &\quad m_n = E[X^n] \;;\; m_2 = 1 \end{aligned} \tag{2.6}$$

$$\text{where}\quad M = \frac{N-1}{2} \text{ if N odd}$$
$$M = \frac{N}{2} \quad \text{if N even}$$

The case where N = 2 is immediately specified:

$$\begin{aligned} x_1 &= 0 \\ y_1 &= -y_2 = -1 \end{aligned} \tag{2.7}$$

and the N = 3 case is obvious:

$$\begin{aligned} y_1 &= -y_3 = -\sqrt{m_4} \\ y_2 &= 0 \end{aligned} \tag{2.8}$$

$$y_1 = -x_2 \qquad \text{where } F(x_1) = \frac{1}{2m_4}$$

In Section 2.5 we will present the general solution. For a symmetrical density function a closed form solution has been obtained for N = 2,3,4. These results are summarized in Table 2.1. The general solution for N = 2 is presented above by Equation 2.4.

In the next two section we digress for awhile to introduce orthogonal polynomials.

2.3 Some Preliminaries

In the next two sections a brief discussion of orthogonal polynomials will be presented. This discussion is by no means meant to be complete. The literature on orthogonal polynomials is almost unbounded. We will present only material needed to examine the moment preserving quantizer. Only pertinent theorems will be proved. For a complete discussion of orthogonal polynomials the reader is referred to the literature particularly the classical work by Szego [76], the somewhat dated but still relevant bibliography by Shohat [72], and the work by Askey [9], Jackson [39], Davis [18] and Krylov [44]. The notation used in this section is a slight modification of Szego's notation.

Let F(x) be a non-decreasing real-valued function on [a,b] which is not constant. If $a = -\infty$ and/or $b = +\infty$ we will require that $F(-\infty) = \lim_{x \to -\infty} F(x) = 0$ and/or $F(\infty) = \lim_{x \to \infty} F(x) = 1$. As previously stated we will assume the total variation of F(x) is identically equal to one. The class of functions g(x) which are measurable with respect to F(x) and for which the Lebesgue-Stieltjes integral:

$$\int_a^b |g(x)|^2 \, dF(x) \tag{2.9}$$

is finite is called $L_F^2(a,b)$. For a complete discussion of L_F^2 spaces the

Table 2.1 Summary of closed formed relationships of a moment preserving quantizer when the input density is symmetric. Where $m_i = E[X^i]$.

N = 2

$$y_1 = -y_2 = -1$$

$$X_1 = 0$$

N = 3

$$y_1 = -y_3 = -\sqrt{m_4}$$

$$y_2 = 0$$

$$x_1 = -x_2$$

$$\text{where } F(x_1) = \frac{1}{2m_4}$$

N = 4

$$y_1 = -y_4 = -\left(1 + \left[\frac{1-2F(x_1)}{2F(x_1)}\,(m_4-1)\right]^{1/2}\right)^{1/2}$$

$$y_2 = -y_3 = -\left(1 - \left[\frac{2F(x_1)}{1-2F(x_1)}\,(m_4-1)\right]^{1/2}\right)^{1/2}$$

$$x_1 = -x_3$$

$$x_2 = 0$$

$$\text{where } F(x_1) = \frac{1}{4} - \frac{R}{4}\sqrt{\frac{1}{4+R^2}}$$

$$R = \frac{(m_6-1) - 3(m_4-1)}{(m_4-1)^{3/2}}$$

reader is referred to Natanson [57].

We shall define the inner product (metric) of the functions $g(x)$, $h(x) \varepsilon L_F^2(a,b)$ to be

$$(g,h) = \int_a^b g(x)h(x)dF(x) \tag{2.10}$$

The norm shall be defined as

$$||g|| = (g,g)^{1/2} \tag{2.11}$$

Definition 2.1. Let the set of functions

$$g_0(x),\ g_1(x),\ \ldots,\ g_n(x),\ \ldots \tag{2.12}$$

be of the class $L_F^2(a,b)$; i.e. $g_i(x) \varepsilon L_F^2(a,b)$ for all i. The set of Equation 2.12 is said to be closed in $L_F^2(a,b)$ if for every $g(x) \varepsilon L_F^2(a,b)$ and for each $\varepsilon > 0$ there exists an integer n such that a function of the form

$$h(x) = \sum_{i=0}^{n} c_i g_i(x)$$

with

$$\int_a^b |g(x) - h(x)|^2 \, dF(x) < \varepsilon.$$

Definition 2.2. A finite set of functions

$$g_0(x), \dots, g_n(x)$$

is said to be linearly independent if the equation

$$|| \sum_{i=0}^{n} \lambda_i g_i(x) || = 0$$

can be true only for

$$\lambda_0 = \lambda_1 = \lambda_2 \dots = \lambda_n \equiv 0.$$

The system $g_i(x)$ cannot contain the zero function i.e. $||g_i(x)|| \neq 0$ for all i. A countably infinite set of functions is linearly independent if the above conditions hold for every finite subset.

This leads to the following theorem stated without proof.

Theorem 2.1. Let F(x) have the same meaning as in Definition 2.1 and a and b be finite. The system of functions $\{x^n\}$, $n = 0,1,2,\dots$ is linearly independent and closed. For proof see [76].

This theorem will ensure closure of orthogonal sets of polynomials on a finite interval. The system $\{x^n\}$ is also linearly independent on an infinite or semi-infinite interval.

Definition 2.3. The set of functions

$$g_0(x),\ g_1(x),\ldots,\ g_\ell(x)$$

ℓ finite or infinite is said to be orthogonal with respect to F(x) if

$$(g_n(x), g_m(x)) = k_n \delta_{nm} \qquad n,m = 0,1,\ldots,\ell$$

where

$$\delta_{nm} = \begin{matrix} 1 & \text{if} & m = n, \\ 0 & \text{if} & m \neq n \end{matrix}$$

and

$$k_n = (g_n(x), g_n(x))$$

Here $g_i(x) \varepsilon L_F^2(a,b)$ for all i and $g_i(x)$ is assumed to be real-valued. If $k_n = 1$ for all n then the set of function is said to be orthonormal.

Orthogonal functions are necessarily linearly independent. Any set of a linearly independent functions $\{h_i(x);\ h_i(x) \varepsilon L_F^2(a,b)\}$ can be orthogonalized using the Gram-Schmidt procedure stated in the following theorem.

Theorem 2.2. Let the real-valued functions

$$h_0(x),\ h_1(x),\ \ldots,\ h_\ell(x)\ ;\ \ell \text{ finite or infinite}$$

where $h_i(x) \varepsilon L_F^2(a,b)$ be linearly dependent. Then an orthonormal set

$$\Phi_0(x), \Phi_1(x), \ldots, \Phi_\ell(x)$$

exists such that, for n = 0,1,2,...,ℓ.

$$\Phi_n(x) = \sum_{i=0}^{n} \lambda_{ni}\, h_i(x) \quad . \tag{2.13}$$

The set $\{\Phi_i(x)\}$ is uniquely determined. Proof see [76].

By using Definition 2.3 and Equation 2.13 we can show (Natanson [57]) the orthonormal functions of the above theorem can be constructed as:

$$\Phi_n(x) = (\Delta_{n-1}\Delta_n)^{-1/2}\Delta_n(x) \qquad n \geq 0$$

where

$$\Delta_n(x) = \begin{bmatrix} (h_0,h_0) & (h_0,h_1) & \cdots & (h_0,h_n) \\ (h_1,h_0) & (h_1,h_1) & \cdots & (h_1,h_n) \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ (h_{n-1},h_0) & & & (h_{n-1},h_n) \\ h_0(x) & & & h_n(x) \end{bmatrix} \tag{2.14}$$

and $$\Delta_n = \begin{bmatrix} (h_0,h_0) & (h_0,h_1) & \cdots & (h_0,h_n) \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ (h_n,h_0) & & & (h_n,h_n) \end{bmatrix}$$

Δ_n is known as the Gram determinant

$$\Delta_0(x) = h_0(x)$$

$$\Delta_{-1} = 1$$

Equation 2.14 is really no more than a restatement of Cramers Rules.

While many more statements can be made relative to orthogonal functions we have developed most of the basic concepts needed at this point to introduce orthogonal polynomials. In the next section we will develop some properties of orthogonal polynomials.

2.4 Orthogonal Polynomials

Let π_n denote the set of all polynomials $\{\rho(x)\} \varepsilon L_F^2(a,b)$ with degree less than or equal to n. Hence π_n is a subspace of $L_F^2(a,b)$.

Definition 2.4. Let F(x) be a fixed non-decreasing function with infinitely many points of increase in the finite or infinite interval [a,b], and let the moments

$$m_n = \int_a^b x^n dF(x) \qquad n = 0,1,2,...$$

exist and be finite. If we orthogonalize the set of functions $\{x^n\}$, n = 0,1,2,... in the sense explained in Theorem 2.2 and Equation 2.14 we obtain a set of polynomials

$$p_0(x),\ p_1(x),\ ...,\ p_n(x),\ ...$$

uniquely determined by the following:

a) $p_n(x)$ is a polynomial of degree n where the coefficient of x^n positive

b) the system $\{p_n(x)\}$ is orthogonal:

$$(p_n(x), p_m(x)) = k_n \delta_{nm}$$

A similar definition holds if F(x) admits a density function f(x). The set $\{p_n(x)\}$ of orthogonal polynomials is said to be associated with F(x) (or f(x)). We shall use the notation $\{\Psi_n(x)\}$ to denote the normalized set of $\{p_n(x)\}$. If F(x) has only a finite number of points of increase then we obtain a finite system of polynomials. Using Equation 2.14 we arrive at

$$p_n(x) = \begin{bmatrix} m_0 & m_1 & \cdots & m_n \\ m_1 & m_2 & \cdots & m_{n+1} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ m_{n-1} & m_n & \cdots & m_{2n-1} \\ 1 & x & \cdots & x^n \end{bmatrix} \qquad (2.15)$$

$$\Psi_n(x) = (\Delta_{n-1}\Delta_n)^{-1/2} \; p_n(x)$$

where Δ_n is the Gram determinant.

Every polynomial $\rho(x) \in \pi_n$ can of course be expressed as a linear combination of $\Psi_0(x)$, $\Psi_1(x)$, ..., $\Psi_n(x)$. Each $\Psi_n(x)$ is orthogonal to every polynomial of lower degree. This is obvious from the orthogonal property, i.e., if $q(x) \in \pi_{n-1}$ then

$$(q(x), \Psi_n(x)) = 0$$

in particular

$$(x^i, \Psi_n(x)) = 0 \quad i = 0,1,2,...,n-1. \qquad (2.16)$$

This condition determines $\Psi_n(x)$ except for a constant factor. Equation 2.16 can be thought of as a wider orthogonality condition.

If [a,b] is symmetric with respect to the origin (i.e., b = -a) and F(x) admits a density function that is itself an even function (f(x) = f(-x)), then if q(x) is a polynomial, q(-x) is a polynomial of the same degree. Let $q(x) \varepsilon \pi_{n-1}$ then the integral

$$\int_{-b}^{b} \Psi_n(-x)q(x)f(x)dx = \int_{-b}^{b} \Psi_n(x)q(-x)f(x)dx = 0$$

since $\Psi_n(t)$ is orthogonal to every polynomial of lower degree by Equation 2.16. Since $\Psi_n(-x)$ is normalized, and $(-1)^n\Psi_n(-x)$ has a positive coefficient for x^n. Then by the Definition 2.4:

$$\Psi_n(x) = (-1)^n\Psi_n(-x) \qquad (2.17)$$

That says that $\Psi_n(x)$ is an odd or even function depending on n being odd or even. Hence $\Psi_n(x)$ contains only even or odd powers depending on whether n is odd or even.

<u>Definition 2.5</u>. Let $\{\Psi_n(x)\}$, n = 0,1,.. be a set of real orthonormal polynomials. The symmetric function

$$K_n(x,y) = K_n(y,x) = \sum_{k=0}^{n} \Psi_k(x)\Psi_k(y)$$

is called the <u>kernel polynomial</u> of order n.

Theorem 2.3. The following recurrence relationship holds for any three consecutive orthogonal polynomials:

$$x\Psi_n(x) = \frac{a_n}{a_{n+1}}\Psi_{n+1}(x) + \left(\frac{b_n}{a_n} - \frac{b_{n+1}}{a_{n+1}}\right)\Psi_n(x) + \frac{a_{n-1}}{a_n}\Psi_{n-1}(x). \tag{2.18}$$

where a_n is the coefficient of x^n in $\Psi_n(x)$ and b_n is the coefficient of x^{n-1} in $\Psi_n(x)$. The proof is omitted. See Jackson [39].

Equation 2.18 can be used to generate sets of polynomials by machine computation. A variation of Equation 2.18 due to Davis [18] is particularly convenient for machine computation:

$$p_{n+1}(x) = x\Psi_n(x) - (x\Psi_n(x),\Psi_n(x))\Psi_n(x) - (p_n,p_n)^{1/2}\,\Psi_{n-1}(x) \tag{2.19}$$

$$\Psi_{n+1}(x) = p_{n+1}(x)/(p_{n+1}(x),p_{n+1}(x))^{1/2}$$

Another useful recurrence relationship involving the kernel polynomial is the Christoffel-Darboux identity:

Theorem 2.4. The following recurrence relationship holds:

$$K_n(x,t) = \frac{a_n}{a_{n+1}}\,\frac{\Psi_{n+1}(t)\Psi_n(x) - \Psi_n(t)\Psi_{n+1}(x)}{t-x} \tag{2.20}$$

The proof is omitted. See Jackson [39].

We now turn our attention to some elementary properties of the zeroes of orthogonal polynomials.

Theorem 2.5. The zeroes of real orthogonal polynomials are real, simple and if [a,b] is a finite interval the zeroes are located in the interior of [a,b].

The proof if omitted. See Davis [18, p. 236] or Szego [76, Section 3.3].

Theorem 2.6. Let $z_1 < z_2 < \ldots < z_n$ be the zeroes of $\Psi_n(x)$; $z_0 = a$ and $z_{n+1} = b$. Then each interval $[z_i, z_{i+1}]$, $i = 0,1,2,\ldots,n$, contains exactly one zero of $\Psi_{n+1}(x)$.

The proof is omitted. See Szego [76, Section 3.3].

Theorem 2.7. Between two zeroes of $\Psi_n(x)$ there is at least one zero of $\Psi_m(x)$, $m > n$.

The proof is omitted. See Szego [76, Section 3.3].

These three theorems will be used when we discuss the convergence of the MP quantizer.

We now state the three theorems that totally specify the MP quantizer problem.

Theorem 2.8. (Gauss-Jacobi Mechanical Quadrature)

If $z_1 < z_2 < \ldots < z_n$ denote the zeroes of $\Psi_n(x)$, there exist real numbers $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that

$$\int_a^b \rho(x) dF(x) = \sum_{i=1}^{n} \lambda_i \rho(z_i) \tag{2.21}$$

whenever $\rho(x) \varepsilon \pi_{2n-1}$. The distribution F(x) and the integer n uniquely determine the numbers λ_i. The λ_i are known as the Christoffel numbers.

Proof: (after Szego [76, p. 47])

We construct the Lagrange interpolation polynomial [76, ch. 14].

$$Q(x) = \sum_{k=1}^{n} \frac{\rho(z_k)\Psi_n(x)}{\Psi_n'(z_k)(x-z_k)}$$

or

$$Q(x) = \sum_{k=1}^{n} \rho(z_k)\ell_k(x) \tag{2.22}$$

where $$\ell_k(x) = \frac{\Psi_n(x)}{\Psi_n'(z_k)(x-z_k)}$$

Since $\rho(z_k)$ and $\Psi_n'(z_k)$ are constants, Q(x) is of degree n-1. Also by using L'Hospital's rule it is obvious that $Q(x_k) = \rho(z_k)$. $(\ell_k(z_k) = 1)$. Hence the polynomial $\rho(x)-Q(x)$ has zeroes at $z_1 < z_2 < \ldots < z_n$. Therefore $\rho(x)-Q(x)$ is divisible by $\Psi_n(x)$ or alternatively

$$\rho(x)-Q(x) = \Psi_n(x)r(x)$$

where $r(x) \varepsilon \pi_{n-1}$. Hence

$$\rho(x) = Q(x) + \Psi_n(x)r(x)$$

$$\int_a^b \rho(x)dF(x) = \int_a^b Q(x)dF(x) + \int_a^b \Psi_n(x)r(x)dF(x)$$

But $\Psi_n(x)$ is orthogonal to any polynomial of degree less than n. Therefore

$$\int_a^b \Psi_n(x)r(x)dF(x) = 0$$

So

$$\int_a^b \rho(x)dF(x) = \int_a^b Q(x)dF(x)$$

$$= \sum_{k=1}^{n} \rho(z_k) \int_a^b \ell_k(x)dF(x)$$

Therefore Equation 2.21 is immediately given with

$$\lambda_k = \int_a^b \ell_k(x)dF(x)$$

$$= \int_a^b \frac{\Psi_n(x)}{\Psi_n'(z_k)(x-z_k)} dF(x) \qquad k = 1,2,...,n. \tag{2.23}$$

QED

Note that the λ_k's are independent of $\rho(x)$.

The result above is often used in numerical integration where $\rho(x)$ is replaced by a general function $g(x) \in L_F^2(a,b)$. The error can be predicted and n can be chosen to find the degree of accuracy needed [18, Ch. 14]. In the next two theorems we state some important properties of the Christoffel numbers.

Theorem 2.9. The Christoffel numbers, λ_k are positive, and

$$\sum_{k=1}^{n} \lambda_k = \int_a^b dF(x) = F(b)-F(a) = 1 \qquad (2.24)$$

That is the sum of the Christoffel numbers is equal to the total variation of F(x).

Proof:

Using the conventions of Theorem 2.8 we have

$$\ell_k(x) = \frac{\Psi_n(x)}{\Psi_n'(z_k)(x-z_k)}$$

with $\ell_k(z_k) = 1$; $\ell_k(z_m) = 0$; $m \neq k$

also $\ell_k(x) \in \pi_{n-1}$ so $(\ell_k(x))^2 \in \pi_{2n-2}$

So letting $\rho(x) = \ell_k^2(x)$ and applying Theorem 2.8 we have

$$\int_a^b \ell_k^2(x)dF(x) = \sum_{m=1}^{n} \lambda_m \ell_m^2(z_m)$$

therefore

$$\lambda_k = \int_a^b \left[\frac{\Psi_n(x)}{\Psi_n'(z_k)(x-z_k)} \right]^2 dF(x) \qquad (2.25)$$

This guarantees that the λ_k's are positive. We also have obtained another method of calculating the λ_k's. Also by letting $\rho(x) = 1$ and using Theorem 2.8 we have the desired result.

QED

An additional way of finding the Christoffel numbers is:

$$\lambda_k^{-1} = \sum_{m=0}^{n} \Psi_m^2(z_k) \qquad k = 1,2,\ldots,n. \tag{2.26}$$

This can be shown quite easily by using Equation 2.22 and Theorem 2.4.

From the above result of the positiveness of the λ_k's, and noting the properties of F(x) previously stated, there exists numbers $q_1 < q_2 < \ldots q_{n-1}$, $a < q_1$, $q_{n-1} < b$ such that

$$\begin{gathered} \lambda_k = F(q_k) - F(q_{k-1}) \\ k = 1,2,\ldots,n \\ q_0 = a \\ q_n = b \end{gathered} \tag{2.27}$$

We should of course worry about points of discontinuity of F(x) but this does not effect the results of Theorem 2.8. Also the q_k's are <u>not</u> in general uniquely determined. However for most cases of practical interest $F^{-1}(x)$ will exist; this of course guarantees the uniqueness of the q's. We shall now present the Separation Theorem of Chebyshev-Markov-Stieltjes which along with Theorems 2.8 and 2.9 will specify the MP quantizing.

<u>Theorem</u> <u>2.10</u>. (Separation Theorm)

The zeroes $z_1 < z_2 < \ldots < z_n$ alternate with the numbers $q_1 < q_2 \ldots q_n$ that is

$$q_k < z_k < q_{k+1} \, .$$

Hence Theorem 2.8 could be written as:

$$\int_a^b \rho(x)dF(x) = \sum_{k=1}^{n} \rho(z_n)(F(x_k) - F(x_{k-1})) \qquad (2.28)$$

The proof is omitted. Szego [76, p. 50] presents three proofs of this remarkable theorem.

Before leaving this section we present in Table 2.2 a brief list of classical orthogonal polynomials along with their distributions. Note that some distributions are of the "discrete" type.

2.5 The MP Quantizer Reconsidered

In Section 2.2 we stated that the MP quantizer was obtained by solving Equation 2.5

$$\begin{aligned} m_n = \int_a^b x^n dF(x) &= \sum_{i=1}^{n} y_i^n P_i \\ P_i &= F(x_i) - F(x_{i-1}) \qquad (2.29) \\ n &= 0,1,2,\ldots,2N-1. \end{aligned}$$

By inspection Equation 2.29 is just a special case of Theorems 2.8-2.10 with $\rho(x) = x^n$, $y_i = z_i$, $\lambda_i = P_i$, and $q_i = x_i$.

Hence we can state:

> The output levels, y_i, of a N level MP quantizer are the zeroes of the Nth degree orthogonal poloynomial associated with F(x). The P_i are the Christoffel numbers and the x_i and y_i alternate by the Separation Theorem. The formulation of quantizer is unique as guaranteed by Theorems 2.8-2.10.

Table 2.2 A partial list of orthogonal polynomials and their associated probability distributions [76].

Distribution	Polynomials
Uniform	Legerdre
Normal	Hermite
Gamma	Generalized Laguerre
Beta	Jacobi
Poisson	Charlier
Binomial	Krawtchouk
Negative Binomial	Meixner
Discrete Uniform	(Discrete) Chebyshev

This result is indeed interesting. The closed form results presented in Table 2.1 can be shown to conform to this statement as special cases where the zeroes are written in closed form. In general to find the quantizer for any N requires the computation of the zeroes of an Nth order polynomial, which for large N can present problems. In the next section we will compare the MP quantizer with other quantizers. The polynomials will be generated by using Equation 2.19. The zeroes will be obtained by numerical methods and P_i's will be obtained by Equation 2.26 the x_i's will then be obtained relative to Theorem 2.10. In Section 2.7 we shall discuss the statistical convergence of the MP quantizer and the quantization noise relationship.

2.6 Examples of MP Quantizers

In this section we will present some tables of the MP quantizer thresholds and output levels for the uniform, normal and Laplacian probability distribution functions. We continue to make the zero mean, unit variance assumption. We will compare these quantizers to the minimum mean square error quantizer of Max. For these examples we used Equation 2.19 to generate a set of orthogonal polynomials then the zeroes were obtained by using numerical techniques. The Christoffel numbers were obtained by Equation 2.26. The thresholds were obtained by using the fact that

$$F(x_i) = \sum_{j=1}^{i} P_j$$

and

$$x_i = F^{-1}(x_i)$$

As mentioned previously there can be problems if F(x) has points of discontinuity; however for these examples the $F^{-1}(x)$ exist. We also calculated the mean square error of the quantizer and the entropy of the output:

$$E[(Y-X)^2] = E[Y^2 - 2XY + X^2]$$

but

$$E[Y^2] = E[X^2] = 1$$

$$E[(Y-X)^2] = 2(1 - E[XY])$$

$$E[(Y-X)^2] = 2(1 - \int_a^b xy \, dF(x))$$

$$E[(Y-X)^2] = 2(1 - \sum_{i=1}^{N} y^i \int_{x_{i-1}}^{x_i} x \, dF(x)) \qquad (2.30)$$

$$\text{entropy} = - \sum_{i=1}^{N} P_i \log_2 P_i$$

The probability density functions are:

a) uniform:

$$f(x) = \frac{1}{2q}, \quad x \varepsilon [-q,q]$$

$$m_i \begin{array}{ll} = 0 & , \; i \text{ odd} \\ = \frac{(q)^i}{i+1} & , \; i \text{ even} \end{array}$$

for $m_2 = 1 \Longrightarrow q = \sqrt{3}$

b) normal:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} , \quad x \in (-\infty,\infty)$$

$$m_i = \begin{array}{ll} 0 & , \; i \text{ odd} \\ 1 \cdot 3 \cdot 5 \cdot \quad (i-1) & , \; i \text{ even} \end{array}$$

c) Laplacian:

$$f(x) = \frac{1}{\sqrt{2}} e^{-\sqrt{2}\,|x|} , \quad x \in (-\infty,\infty)$$

$$m_i = \begin{array}{ll} 0 & , \; i \text{ odd} \\ 2^{-\frac{i}{2}} i! & , \; i \text{ even} \end{array}$$

The results for the MP quantizer (N = 2-16) are shown in Tables 2.3, 2.5, and 2.7 respectively. For the uniform density the polynomials are the Legendre polynomials; for the normal density the polynomials are the Hermite polynomials. The polynomials for the Laplacian density are not members of the classical polynomials (we have strongly resisted the temptation to call them the Delp-Mitchell polynomials).

The minimum mean square error quantizer (denoted MSE) of Max is found by:

$$\min_{x_i\text{'s},y_i\text{'s}} E[(Y-X)^2] \quad \text{for a fixed N.}$$

Tables of the MSE quantizer (N = 2,4,8,16) are shown in Table 2.4, 2.6, and 2.8. Plots of the mean square error are shown in Figures 2.2-2.4. The uniform density MSE quantizer is one of the few density functions for which closed form relationships are available for the y_i's and x_i's. The results are that the MSE quantizer for a uniform density is a uniform quantizer. It can be shown quite easily that:

$$y_i = \frac{(2i - (N+1))q}{N} \qquad i = 1,2,\dots,N$$

$$P_i = \frac{1}{N}$$

$$x_j = \frac{(2j - N)q}{N} \qquad j = 1,2,\dots,N-1$$

$$F(x_j) = \frac{j}{N}$$

$$\text{mse} = \frac{q^2}{3N^2}$$

$$\text{entropy} = \log_2 N$$

The results for MP quantizer for a uniform density are interesting in that as N increases the output levels tend to group closely to ±q. In fact it can be shown that on a finite interval the zeroes of any

Table 2.3 Positive thresholds and positive output levels for a MP quantizer (N = 2-16) for zero mean, unit variance uniform probability density function. (mse = mean square error).

		output levels	thresholds
N=2		1.00	0.00
entropy	1.00		
mse	0.2679		
N=3		0.00	0.7698
		1.3416	
entropy	1.547		
mse	0.1352		
N=4		0.5889	0.0
		1.4915	1.1295
entropy	1.9321		
mse	0.0815		
N=5		0.00	0.4927
		0.9327	1.3217
		1.5695	
entropy	2.2325		
mse	0.0545		
N=6		0.4133	0.0
		1.1452	0.8105
		1.6151	1.4353
entropy	2.4794		
mse	0.039		
N=7		0.00	0.3620
		0.7029	1.0233
		1.2894	1.5078
		1.6439	
entropy	2.6893		
mse	0.02927		

Table 2.3, cont.

		output levels	thresholds
N=8		0.3177	0.00
		0.9102	0.6282
		1.3799	1.1715
		1.6633	1.5567
entropy	2.8722		
mse	0.0228		
N=9		0.00	0.2860
		0.5616	0.8270
		1.0624	1.2784
		1.4480	1.5913
		1.6769	
entropy	3.0342		
mse	0.0182		
N=10		0.2579	0.00
		0.7507	0.5119
		1.1768	0.9782
		1.4983	1.3577
		1.6869	1.6166
entropy	3.1796		
mse	0.0149		
N=11		0.00	0.2364
		0.4669	0.6915
		0.8991	1.0955
		1.2647	1.4181
		1.5364	1.6356
		1.6943	
entropy	3.3116		
mse	0.0124		
N=12		0.2169	0.00
		0.6371	0.4315
		1.0172	0.8359
		1.3335	1.1878
		1.5660	1.4651
		1.7001	1.6503
entropy	3.4325		
mse	0.0105		

Table 2.3, cont.

		output levels	thresholds
N=13		0.00	0.2014
		0.3991	0.5933
		0.7768	0.9533
		1.1126	1.2618
		1.3883	1.5023
		1.5893	1.6619
		1.7046	
entropy	3.5439		
mse	0.0090		
N=14		0.1872	0.00
		0.5527	0.3728
		0.8924	0.7283
		1.1904	1.0496
		1.4327	1.3219
		1.6081	1.5324
		1.7083	1.67122
entropy	3.6474		
mse	0.0078		
N=15		0.00	0.1754
		0.3485	0.5191
		0.6827	0.8416
		0.9889	1.1296
		1.2547	1.3713
		1.4691	1.5569
		1.6234	1.6788
		1.7113	
entropy	3.7439		
mse	0.0068		
N=16		0.1646	0.0
		0.4878	0.3281
		0.7933	0.6444
		1.0702	0.9374
		1.3084	1.1965
		1.4993	1.4124
		1.6361	1.5772
		1.7137	1.6850
entropy	3.8343		
mse	0.0061		

Table 2.4 Positive thresholds and positive output levels for an MSE quantizer (N=2,4,8,16) for a zero mean, unit variance uniform probability density function (mse = mean square error)

		output levels	thresholds
N=2		0.8660	0.00
entropy	1.00		
mse	0.25		
N=4		0.4330	0.00
		1.2990	0.8660
entropy	2.00		
mse	0.0625		
N=8		0.2165	0.00
		0.6495	0.433
		1.0825	0.8660
		1.5155	1.2990
entropy	3.00		
mse	0.0156		
N=16		0.1083	0.00
		0.3248	0.2165
		0.5413	0.4330
		0.7578	0.6495
		0.9743	0.8660
		1.1908	1.0825
		1.4073	1.2990
		1.6238	1.5155
entropy	4.00		
mse	0.0039		

Table 2.5 Positive thresholds and positive output levels for a MP quantizer (N=2-16) for a zero mean, unit variance normal probability density function (mse = mean square error)

	output levels	thresholds
N=2	1.00	0.00
entropy 1.00		
mse 0.4042		
N=3	0.00	0.9673
	1.7321	
entropy 1.2516		
mse 0.2689		
N=4	0.7419	0.00
	2.3344	1.6866
entropy 1.4423		
mse 0.2032		
N=5	0.00	0.7277
	1.3557	2.2820
	2.8570	
entropy 1.5936		
mse 0.1626		
N=6	6.6167	0.00
	1.8892	1.3338
	3.3242	2.8003
entropy 1.7188		
mse 0.1362		
N=7	0.00	0.6081
	1.1544	1.8624
	2.3667	3.2648
	3.7504	
entropy 1.8255		
mse 0.1166		

Table 2.5, cont.

		output levels	thresholds
N=8		0.5391	0.00
		1.6365	1.1408
		2.8025	2.3364
		4.1445	3.6890
entropy	1.9185		
mse	0.1024		
N=9		0.00	0.5332
		1.0233	1.6193
		2.0768	2.7694
		3.2054	4.0818
		4.5127	
entropy	2.0008		
mse	0.0909		
N=10		0.4849	0.00
		1.465	1.0137
		2.4843	2.0568
		3.5818	3.1702
		4.8595	4.4491
entropy	2.0748		
mse	0.0820		
N=11		0.00	0.4805
		0.9288	1.4537
		1.8760	2.4620
		2.8651	3.5449
		3.9361	4.7951
		5.1880	
entropy	2.1419		
mse	0.0745		
N=12		0.4444	0.00
		1.3404	0.9216
		2.2595	1.8615
		3.2237	2.8409
		4.2718	3.8979
		5.5009	5.1232
entropy	2.2032		
mse	0.06841		

Table 2.5, cont.

		output levels	thresholds
N=13		0.00	0.4409
		0.8567	1.3309
		1.7254	2.2429
		2.6207	3.1978
		3.5634	4.2324
		4.5914	5.4358
		5.8002	
entropy	2.2598		
mse	0.0631		
N=14		0.4126	0.00
		1.2427	0.8509
		2.0883	1.7142
		2.9630	2.6026
		3.8869	3.5363
		4.8969	4.5512
		6.0874	5.7349
entropy	2.3123		
mse	0.0587		
N=15		0.00	0.4096
		0.7991	1.2352
		1.6067	2.0755
		2.4324	2.4435
		3.2891	3.8586
		4.1962	4.8560
		5.1901	6.0221
		6.3639	
entropy	2.3611		
mse	0.0547		
N=16		0.3868	0.00
		1.1638	0.7943
		1.9519	1.5977
		2.7602	2.4182
		3.6009	3.2683
		4.4929	4.1670
		5.4722	5.1485
		6.6308	6.2986
entropy	2.4069		
mse	0.0519		

Table 2.6 Positive thresholds and positive output levels for an MSE quantizer (N=2,4,8,16) for a zero mean, unit variance normal probability density function. After Max [46] (mse = mean square error)

		output levels	thresholds
N=2		0.7980	0.00
entropy	1.00		
mse	0.3634		
N=4		0.4528	0.00
		1.510	0.9816
entropy	1.911		
mse	0.1175		
N=8		0.2451	0.00
		0.7560	0.5006
		1.344	1.050
		2.152	1.748
entropy	2.825		
mse	0.0345		
N=16		0.1284	0.00
		0.3881	0.2582
		0.6568	0.5224
		0.9424	0.7996
		1.256	1.099
		1.618	1.437
		2.069	1.844
		2.733	2.401
entropy	3.765		
mse	0.0095		

Table 2.7 Positive thresholds and positive output levels for a MP quantizer (N=2-16) for a zero mean, unit variance Laplacian probability density function (mse = mean square error)

		output levels	thresholds
N=2		1.00	0.00
entropy	1.00		
mse	0.5858		
N=3		0.00	1.2669
		2.4495	
entropy	0.8166		
mse	0.3882		
N=4		0.8183	0.00
		4.0163	2.7193
entropy	1.1491		
mse	0.3744		
N=5		0.00	1.0213
		1.9942	4.3414
		5.7175	
entropy	1.0417		
mse	0.2928		
N=6		0.7371	0.00
		3.2972	2.2191
		7.4655	6.0272
entropy	1.2593		
mse	0.2969		
N=7		0.00	0.9078
		1.7802	3.5842
		4.7376	7.7924
		9.2806	
entropy	1.1716		
mse	0.2466		

Table 2.7, cont.

	output levels	thresholds
N=8	0.6882	0.00
	2.9425	1.9745
	6.2421	5.0278
	11.1214	9.5909
entropy 1.3387		
mse 0.2549		
N=9	0.00	0.83906
	1.6493	3.1963
	4.2342	6.5607
	7.8246	11.4370
	13.0037	
entropy 1.2614		
mse 0.2185		
N=10	0.6545	0.00
	2.7208	1.8226
	5.5928	4.4970
	9.4470	8.1405
	14.9024	13.3038
entropy 1.3997		
mse 0.2279		
N=11	0.00	0.7916
	1.5585	2.9499
	3.9134	5.8867
	7.0302	9.7778
	11.1210	15.2028
	16.8298	
entropy 1.3292		
mse 0.1993		
N=12	0.6293	0.00
	2.5652	1.7164
	5.1716	4.1535
	8.5122	7.3272
	12.8225	11.4467
	18.7686	17.1161
entropy 1.4488		
mse 0.2089		

Table 2.7, cont.

	output levels	thresholds
N=13	0.00	0.7562
	1.4905	2.7756
	3.6856	5.4432
	6.5668	8.8279
	10.0490	13.1566
	14.5613	19.0530
	20.7287	
entropy 1.3832		
mse 0.1851		
N=14	0.6094	0.00
	2.4481	1.6368
	4.8695	3.9076
	7.8878	6.7842
	11.6177	10.3646
	16.3204	14.8893
	22.6971	21.0000
entropy 1.4897		
mse 0.1945		
N=15	0.00	0.7284
	1.4370	2.6438
	3.5130	5.1223
	6.1276	8.1854
	9.3240	11.9454
	13.2273	16.6528
	18.1078	22.9651
	24.6821	
entropy 1.4279		
mse 0.1741		
N=16	0.5932	0.00
	2.3557	1.5740
	4.6388	3.7202
	7.4314	6.3876
	10.7942	9.6242
	14.8614	13.5532
	19.9108	18.4339
	26.6735	24.9387
entropy 1.5246		
mse 0.1832		

Table 2.8 Positive thresholds and positive output levels for a MSE quantizer (N=2,4,8,16) for a zero mean, unit variance Laplacian probability density function. After Adams and Griesler [12] (mse = mean square error)

	output levels	thresholds
N=2	0.1071	0.00
entropy 1.00		
mse 0.500		
N=4	0.4196	0.00
	1.8340	1.1269
entropy 1.7283		
mse 0.1762		
N=8	0.2334	0.00
	0.8330	0.5332
	1.6725	1.2527
	3.0867	2.3796
entropy 2.5654		
mse 0.0545		
N=16	0.1240	0.00
	0.4048	0.2644
	0.7287	0.5667
	1.1110	0.9198
	1.5778	1.3444
	2.1773	1.8776
	3.0169	2.5971
	4.4311	3.7240
entropy 3.4749		
mse 0.0154		

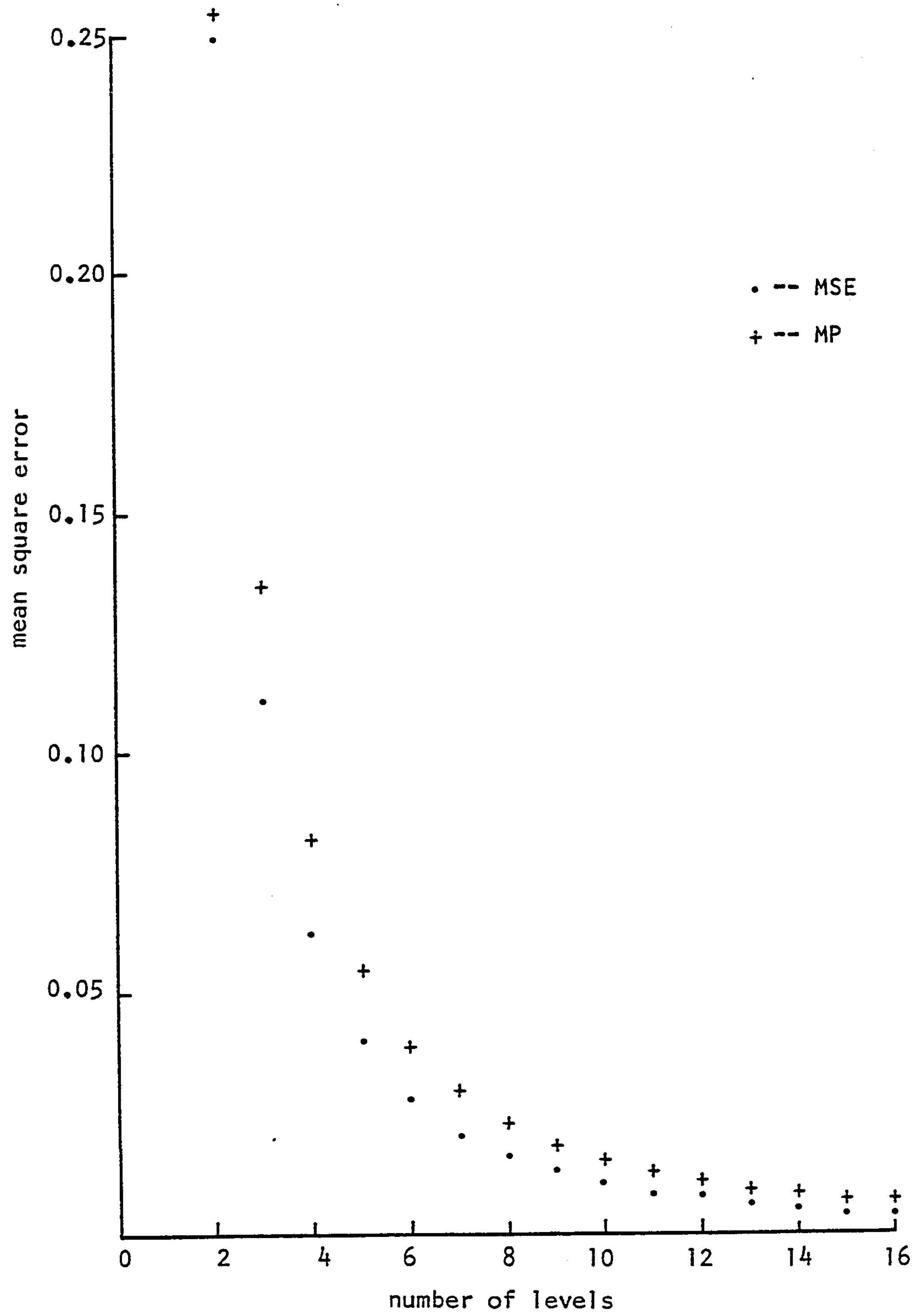

Figure 2.2 Mean square error (mse) vs.number of quantizer levels (N) for both the MSE and MP quantizers for zero mean, unit variance uniform probability distribution.

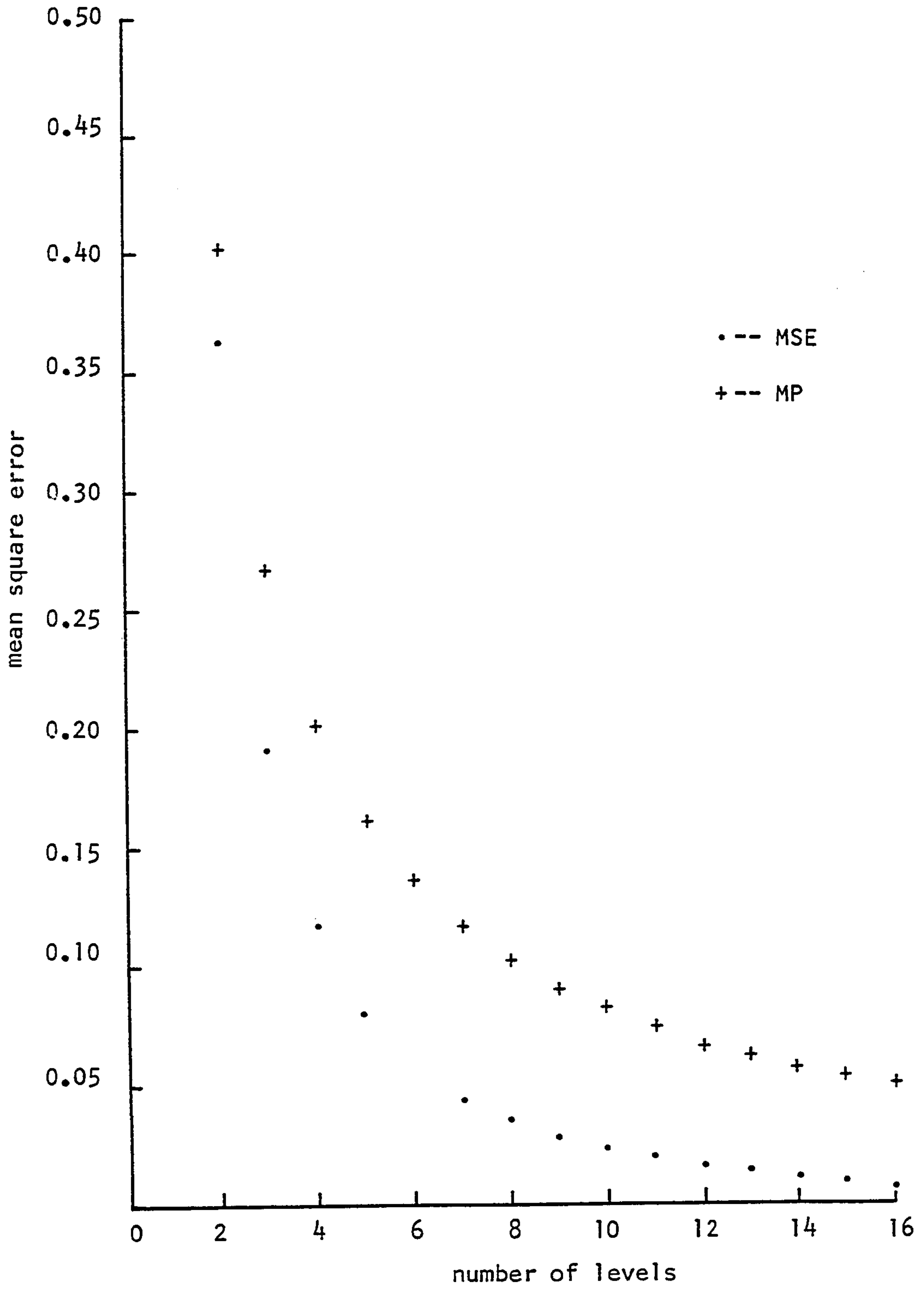

Figure 2.3 Mean square error (mse) vs. number of quantizer levels (N) for both the MSE and MP quantizers for a zero mean, unit variance normal probability distribution.

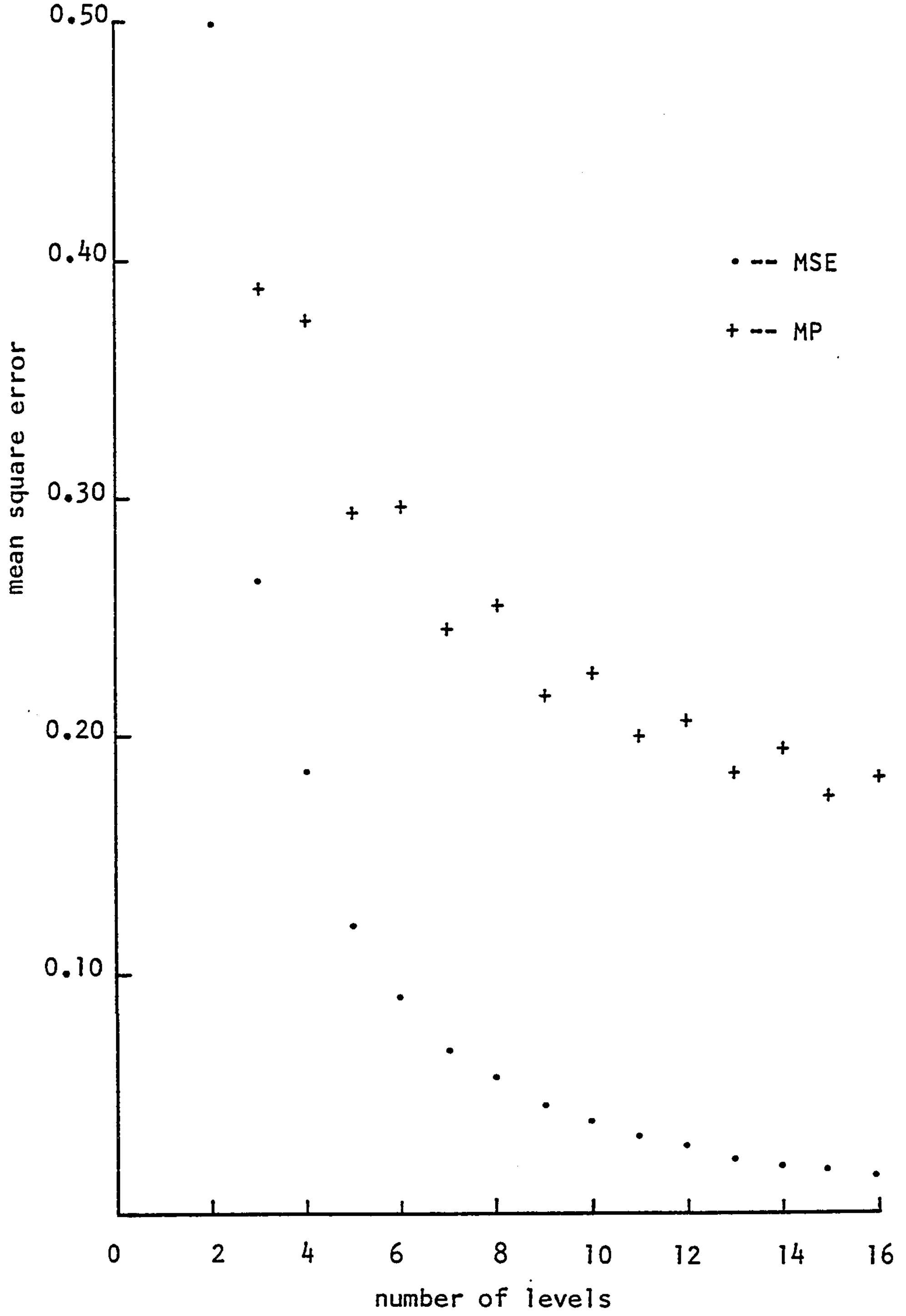

Figure 2.4 Mean square error (mse) vs. number of quantizer levels (N) for both the MSE and MP quantizers for a zero mean, unit variance Laplacian probability distribution.

orthogonal polynomials for a fixed N are denser near the end points [76, p. 311]. The mse error for the uniform MP quantizer decreases with N as shown in Figure 2.2.

The results for the other two density functions on an infinite interval exhibit one of the disadvantages of the MP quantizer; the outputs at y_0 and y_N have a tendency to spread much further then the MSE quantizer. What this says is that the quantizer is assigning output levels that have a very small probability of occurrence. For example the Laplacian MP quantizer for N = 16 assigns levels all the way out beyond 20 standard deviation units as compared to the MSE quantizer which only assigns levels out to 4 standard deviation units. These assignments of low probability output levels are reflected by the low values of the entropy for all three of the MP quantizers. This also points out the fact that it would be very hard to evaluate the MP quantizer for large values of N (say larger than 30) because the output levels would be assigned such low probability of occurrence that one could have problems with machine accuracy. It should also be mentioned that it is no easy task to compute the zeroes of a polynomial of large degree. These types of problems do not manifest themselves in the MSE quantizer due to the type of algorithm used. In fact for the normal distribution case results are available for N as large as 2048. The applicability of using a quantizer with very large number of levels i.e. N > 16 is not significant, in fact usually above N = 8 a uniform quantizer is used.

The mean square error of the MP quantizers all decrease with N except the Laplacian. For this case it is possible that the mse at N levels is smaller that the mse with N+1 levels (see N = 7 and N = 8); however this has only been observed empirically for pair wise groupings of the mse, that is Figure 2.4 has a definite trend of decreasing. As a final statement we should mention that for probability distributions on an infinite or semi-infinite interval the moment sequence diverges. This will tend to limit the maximum value one can use for N because of machine accuracy of representing large numbers. Here again in a practical application one would usually use a uniform quantizer for large N. In the next section we discuss the convergence of the MP quantizer and the quantization noise.

2.7 Convergence of the MP Quantizer

In this section we will examine the convergence of the MP quantizer for large N. The notation we shall use is that Y_N denotes the random variable at the output of the quantizer with N levels and X is the input random variable. We desire to investigate under what circumstances does:

$$Y_N \xrightarrow[N \to \infty]{} X$$

i.e. does Y_N approach X in some sense when N is large. In particular does Y_N converge to X in mean square (i.e. in $L_F^2(a,b)$) or in distribution? These can be stated as:

1) Y_N converges to X in mean square, denoted as $Y_N \overset{ms}{\to} X$, if $E[(Y_N-X)^2] \to 0$ as $N \to \infty$

2) Y_N converges to X in distribution, denoted as $Y_N \overset{d}{\to} X$, if $F_N(x) \to F(x)$ as $N \to \infty$ where $F_N(x)$ is the distribution of Y_N.

Convergence in mean square guarantees convergence in distribution.

Convergence in distribution seems somewhat attractive for the MP quantizer since, as $N \to \infty$, Y_N and X have the same moments. We know that for each moment sequence

$$m_k(N) = E[Y_N^k] \qquad k = 0,1,2,3,...$$

there exists some integers, say R_k, depending on k, such that if $N > R_k$ then $E[X^k] = m_k(N)$. In other words all the moment sequences are converging hence if the moments of $F_N(x)$ are converging will that imply convergence in distribution? This can be restated by saying under what circumstances will the moments completely characterize the input distribution, i.e., given the moments can the distribution be found? This leads to the classical moment problems of Stieltjes, Hausdorff, and Hamburger. The work of Aheizer [3,6], Krein [3], Shohat and Tamarkin [73] address the moment problem very elegantly. The Stieltjes moment problem is defined on the semi-infinite interval. The Hamburger moment problem is defined on the infinite interval. We will only mention the results for the Hamburger problem; the results are analogous for the Stieltjes problem. On a finite interval (Hausdorff) every probability distribution is characterized by its moments [73,26,76]. This says that on a finite interval we have convergence in distribution. A very remarkable theorem due to Riesz [73, p. 61] states that the orthogonal polynomials associated with F(x) are closed in $L_F^2(a,b)$ if and only if F(x) is characterized by its moments. Since we have already mentioned that the orthogonal polynomials are closed on any finite interval (Theorem 2.1) this implies that any distribution on a finite interval is characterized by its moments. Let us state the Hamburger Moment Problem:

Theorem 2.11 (Hamburger)

For the distribution $F(x)$, $x \varepsilon (-\infty,\infty)$, to be characterized by its moments it is necessary that the Gram determinant:

$$\Delta_n = \begin{vmatrix} m_0 & m_1 & \cdots & m_n \\ m_1 & m_2 & \cdots & m_{n+1} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ m_n & \cdots & & m_{2n} \end{vmatrix}$$

be positive semi-definite for all n, i.e.,

$$\Delta_n \geq 0 \qquad \text{for all } n.$$

The proof is omitted. See [73, p. 5]. In many cases it is difficult to verify if a distribution is characterized by moments by using Theorem 2.11. It is sometimes easier to use the results of Carleman:

Theorem 2.12 (Carleman)

A sufficient condition that the Hamburger moment problem be determined is that:

$$\sum_{i=1}^{\infty} (m_{2i})^{-\frac{1}{2i}} = \infty \tag{2.31}$$

or more generally

$$\sum_{n=1}^{\infty} (\inf_{i \geq n} (m_{2i})^{\frac{1}{2i}})^{-1} = \infty.$$

The proof is omitted. See [73, p. 19]. What Theorem 2.12 does is put conditions on the rate of increase of the even moments of F(x). A much more restrictive sufficient condition, but perhaps more intuitive, is that a distribution is characterized by its moments if its characteristic function is analytic in a neighborhood of the real axis [26]. The above conditions can be extended to the Stieltjes problem. We summarize the above by the following theorem. Thus a sufficient condition for convergence in distribution is:

<u>Theorem 2.13</u>

If the input distribution F(x) is characterized by its moments then the MP quantizer converges in distribution.

Proof:

We have stated above that the moment sequences converge, that is:

$$m_k(N) \to m_k \quad \text{as} \quad N \to \infty \ .$$

so if the input probability distribution F(x) is characterized by its moments we have

$$Y_N \overset{d}{\to} X$$

QED

The three distributions discussed in Section 2.7 are characterized by their moments, hence they converge in distribution.

Mean square convergence is much more difficult. We will only show mean square convergence on a finite interval. We will then state some necessary conditions for mean square convergence on an infinite or semi-infinite interval. For mean square convergence we must show

$$E[(Y_N-X)^2] \to 0 \quad \text{as} \quad N \to \infty .$$

From Equation 2.30 we have

$$E[(Y_N-X)^2] = 2(1 - \sum_{i=1}^{N} y_{iN} \int_{x_{i-1,N}}^{x_{iN}} x \, dF(x))$$

where the output levels and thresholds are also indexed by N. Essentially what we need to show is:

$$\lim_{N\to\infty} \sum_{i=1}^{N} y_{iN} \int_{x_{i-1,N}}^{x_{iN}} x \, dF(x) = \int_a^b x^2 dF(x) = 1 \tag{2.32}$$

$$\text{note } y_{iN} \in [x_{i-1,N}, x_{iN}]$$

We shall assume F(x) is continuous to avoid any problems mentioned by the Separation Theorem. Before proceeding we will state a theorem concerning the distance between consecutive zeroes.

Theorem 2.14 (Szego, [76, p. 112])

Assume F(x) admits a density f(x) on the finite interval [a,b] with $f(x) \geq \nu > 0$. Let $y_{1N} < y_{2N}, < \ldots < Y_{NN}$ be the zeroes of the associated polynomial $\Psi_N(x)$. Let

$$y_{kN} = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\theta_{kN}$$

with $0 < \theta_{kN} < \pi$, $k = 1,2,\ldots,N$

then

$$\theta_{k+1} - \theta_k < K \frac{\log N}{N} \qquad (2.33)$$

with k determined only by ν, a and b. Proof is omitted. Therefore

$$\lim_{N\to\infty} \theta_{k+1,N} - \theta_{kN} = 0$$

Note that the value of K does not depend on f(x). We now can state:

Theorem 2.15

The MP quantizer convergences in mean square on a finite interval.

Proof:

We will require the assumptions of Theorem 2.14 (i.e. $f(x) > = \nu > 0$.) Since the quantizer really represents a formal partition of [a,b] we use Theorems 2.6 and 2.7 and the Separation Theorem (Theorem 2.10). These theorems give us (along with Theorem 2.16):

$$\lim_{N\to\infty} |y_{k+1,N} - y_{kN}| = 0$$

$$\lim_{N\to\infty} |x_{k+1,N} - x_{kN}| = 0$$

for all k

Therefore Equation 2.32 is immediately specified.

QED

For mean square convergence on an infinite interval we need a theorem similiar to Theorem 2.14 plus the additional end point conditions:

$$\lim_{N\to\infty} y_1 \int_{-\infty}^{x_1} x \, dF(x) = 0$$

and

$$\lim_{N\to\infty} y_N \int_{x_{N-1}}^{\infty} x \, dF(x) = 0$$

Hence we need to shown that $y_1 \to -\infty$ slower than $\int_{-\infty}^{x_1} x \, dF(x) \to 0$ and similarly for the other end point. For the semi-infinite interval we have similar conditions at the one end point.

We have not been successful at showing a general results in this case. It is very difficult to make statements concerning the zeroes of general orthogonal polynomials. This difficulty should be compared to the minimum mean square error quantizer (MSE) where convergence in mean square is guaranteed if the input density $f(x)$ is Riemann integrable [81].

As to how well the MSE quantizer preserves moments it can be shown [13] that MSE quantizer always preserves m_1 but in general $E[Y^2] < E[X^2]$. It can also be shown that

$$E[(Y-X)^2] = E[X^2] - E[Y^2]$$

hence convergence in mean square guarantees that the second moment series converges.

Finally before leaving this chapter let us briefly discuss the quantization noise of the MP quantizer. Let

$$\varepsilon_q = Y-X$$

be the error in quantizing X with an N level MP quantizer. ε_q is also known as the quantization noise. So we have

$$Y = X + \varepsilon_q$$

$$\text{hence} \quad E[\varepsilon_q] = 0$$

$$\text{and} \quad E[Y^2] = E[X^2] + 2E[X\varepsilon_q] + E[\varepsilon_q^2]$$

or

$$E[\varepsilon_q^2] = -2E[x\varepsilon_q]$$

$$\text{Thus} \qquad E[X\varepsilon_q] \leq 0$$

$$\text{since} \qquad E[\varepsilon_q^2] \geq 0$$

This says that the quantization noise is negatively correlated with the input. Negative correlation of the quantization noise can be also shown for the MSE quantizer [81]. Before we leave this chapter we should state that the uniqueness properties mentioned in Section 2.5 hold if one insists that a N level MP quantizer preserve 2N-1 moments. It is possible to design a N level MP quantizer that preserves less than 2N-1 moments. In this case it is possible to arrive at more than one formulation of the quantizer.

In the next chapter we will present an application of the MP quantizer in the context of image coding for bandwidth compression.

CHAPTER 3

BLOCK TRUNCATION CODING: AN APPLICATION OF THE MP QUANTIZER

3.1. Introduction

In this chapter we present the non-transform coding technique developed at Purdue University over the past two years called Block Truncation Coding (BTC) [23], [48], [49]. This technique involves the use of a one bit adaptive non-parametric MP quantizer. In this chapter the basic BTC algorithm is presented and compared with some of the other techniques of image compression. Modifications to BTC are presented including some hybrid techniques. The performance of BTC in the presence of channel errors is also included. This technique applies to many situations when images must be either transmitted or stored.

3.2. Basic BTC Algorithm

BTC is a method of using adaptive non-parametric re-quantization over local regions of an image. For the study presented here it will be assumed that the local region of the image will be a 4 x 4 pixel block. It will further be assumed that it is desired to find a two-level (one bit quantizer) rendition of the image in this 4 x 4 block. If one uses classical quantization design such as that by Max [46] which minimizes the mean square error between the input and output of the quantizer, or the MP quantizer developed in Chapter 2 then one must know a priori the probability density function of the pixels in each block. This same a priori knowledge of the input density is also required if the mean absolute error fidelity criteria of Kassam [43] is used. Since it is in general not possible to find adequate density function models for typical imagery, we have chosen the approach of using non-parametric quantizers for our coding schemes. Non-parametric quantizer design using the minimum mean square error fidelity criteria (denoted MSE) and the minimum mean absolute error criteria (denoted MAE) will be presented in Section 3.3. In this section we will develop the basic non-parametric MP quantizer based on preserving the sample moments. The resulting quantizer will be compared to the quantizer developed in Chapter 2.

One proceeds by first dividing the original picture into nxn blocks (we have used n=4 for our examples). Blocks are then coded individually, each into a two level signal. The levels for each block are chosen such that the first two sample moments are preserved. Let $k=n^2$ and let $X_1, X_2, \ldots X_k$ be the values of the pixels in a block of the original picture.

Let

$$\overline{m}_1 = \frac{1}{k} \sum_{i=1}^{k} X_i \text{ be the first sample moment}$$

$$\overline{m}_2 = \frac{1}{k} \sum_{i=1}^{k} X_i^2 \text{ be the second sample moment} \tag{3.1}$$

$\overline{\sigma^2} = \overline{m}_2 - \overline{m}_1^2$ be the sample variance

$\overline{\sigma} = \sqrt{\overline{\sigma^2}}$

As with the design of any one bit quantizer, we find a threshold and two output levels for the quantizer such that:

$$\text{if } X_i \geq X_{th} \qquad \text{output} = y_2 \tag{3.2}$$

$$\text{if } X_i < X_{th} \qquad \text{output} = y_1$$
$$\text{for } i = 1,\ldots.,k.$$

where

X_{th} is the threshold

y_1 and y_2 are the "low" and "high" output levels, respectively.

For our basic non-parametric MP quantizer, we shall set $X_{th} = \overline{m}_1$. This reasonable assumption will be modified in Section 3.4 to achieve a more consistent result with Chapter 2. The output levels y_1 and y_2 for a two-level non-parametric moment preserving quantizer are found by solving the following equations:

let q = number of X_i's greater than X_{th} ($\overline{X}$ in this case)

We then have

$$k\overline{m}_1 = (k-q)y_1 + qy_2 \tag{3.3}$$

$$k\overline{m}_2 = (k-q)y_1^2 + qy_2^2$$

Equation 3.3 is readily solved for y_1 and y_2:

$$y_1 = \overline{m}_1 - \overline{\sigma} \sqrt{\left[\frac{q}{k-q}\right]} \tag{3.4}$$

$$y_2 = \overline{m}_1 + \overline{\sigma} \sqrt{\left[\frac{k-q}{q}\right]}$$

The result of Equation 3.4 should be compared to Equation 2.4 for the parametric quantizer. The result is identical if we assume

$$P_1 = \frac{k-q}{k}$$

$$P_2 = \frac{q}{k}$$

Here the probabilities are replaced by using a relative frequency argument.

Each block is then described by $\overline{m}_1$, $\overline{\sigma}$ and an nxn bit plane consisting of 1's and 0's depending on whether a given pixel is above or below X_{th}. Assigning 8 bits each to $\overline{m}_1$ and $\overline{\sigma}$ results in a bit rate 2 bits/pixel. The receiver (decoder) reconstructs the image block by calculating y_1 and y_2 from Equation 3.4 and placing those values in accordance with the bits in the bit plane.

Let us quickly review the basic BTC encoding algorithm:

a) The image is divided in small non-overlapping blocks such as 4x4.

b) The first and second sample moments are computed.

c) A bit plane is constructed such that each pixel location is coded as a "one" or a "zero" depending on whether that pixel is greater than $\overline{m}_1$.

d) The bit plane, $\overline{m}_1$, and $\overline{\sigma}$ are sent to the receiver.

e) The picture block is reconstructed such that $\overline{m}_1$ and $\overline{\sigma}$ (alternatively $\overline{m}_2$) are preserved. That is, pixels in the bit plane that are "0" are set to "y_1" and the "1"'s are set to "y_2" in Equation 3.4. For example, suppose a 4x4 picture block is given by the following:

$$X_{ij} = \begin{bmatrix} 121 & 114 & 56 & 47 \\ 37 & 200 & 247 & 255 \\ 16 & 0 & 12 & 169 \\ 43 & 5 & 7 & 251 \end{bmatrix}$$

so

$$\overline{m}_1 = 98.75$$

$$\overline{\sigma} = 92.95$$

$$q = 7$$

and

$$y_1 = 16.7 \simeq 17$$

$$y_2 = 204.2 \simeq 204$$

the bit plane is:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The reconstructed block becomes:

$$\begin{bmatrix} 204 & 204 & 17 & 17 \\ 17 & 204 & 204 & 204 \\ 17 & 17 & 17 & 204 \\ 17 & 17 & 17 & 204 \end{bmatrix}$$

and the sample mean and variance are preserved.

We have found that block boundaries are not visible in the reconstructed picture using this technique but two major artifacts are: (1) edge raggedness, (2) misrepresentation of some midrange values due to their assignment to either a high or low value. In the next section we will show methods for reducing these effects and improving the representation even further.

A few comments should be made about the appeal of this method. First is the obvious simplicity of the calculations involved. Only k pixels at a time need be considered, eliminating the need for picture storage and allowing real time coding with a small hardware device. This method does not require the amount of computational overhead necessary in other coding schemes such as transform coding. A recent independent study has been completed indicating that BTC can be realized on an integrated circuit chip [25]. Second is the suitability of this method to the human observer. The largest grey level changes within a block are the ones coded. This is obvious by the way the levels are weighed by the variance. If no large changes are present, the most significant small variations are coded. The human is insensitive to small variations in the presence of large variations this phenomena is known as "masking", so this technique is neglecting the very thing to which the human visual system is insensitive [40]. The third point is that the bit plane preserves the original accuracy of an edge or object location with no blurring of the edge upon reconstruction of the image. If anything, the effect is to enhance boundaries which is again suitable for human observation. An example of two images coded using BTC is shown in Figure 3.1.

3.3. Other Non-Parametric Quantizer Schemes

As mentioned in Section 3.2, other techniques could be used to design (or fit) a one bit non-parametric quantizer. The use of rate-distortion theory seems theoretically attractive but somewhat impracti-

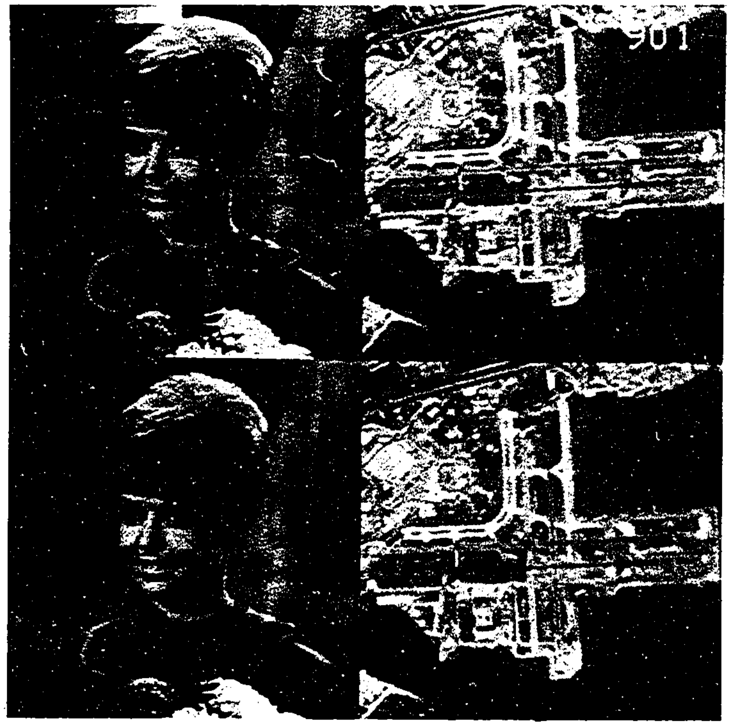

Figure 3.1 Results using the basic BTC algorithm. The original images (top) are 256x256 pixels with nominal 8 bits gray level resolution. The coded images (bottom) have a data rate of 2 bits/pixel.

cal for real imagery [11], [54]. In this section we will discuss the use of the minimum mean square error (MSE) and minimum mean absolute error (MAE) fidelity criteria for non-parametric quantizers.

To use the MSE fidelity criterion, one proceeds by first constructing a histogram of the X_i's (i.e., sorting the X_i's). Let Z_1, Z_2, ... Z_k be the sorted X_i's; i.e., $Z_1 \leq Z_2 ... \leq Z_k$. Again let q be the number of X_i's greater than X_{th}. Then y_1 and y_2 are found by minimizing:

$$J_{MSE} = \sum_{i=1}^{k-q-1} (Z_i - y_1)^2 + \sum_{i=k-q}^{k} (Z_i - y_2)^2 \tag{3.5}$$

where

$$y_1 = \frac{1}{k-q} \sum_{i=1}^{k-q-1} Z_i$$

$$y_2 = \frac{1}{q} \sum_{i=k-q}^{k} Z_i$$

In general it is impossible to solve this equation in closed form for X_{th}, y_1 and y_2. One obvious way to solve this problem is to try every possible threshold (there are at most k-1 thresholds) and pick the one with smallest J_{MSE}. Assuming y_1 and y_2 have 8-bit resolution, this gives a data rate of 2 bits/pixel. The encoding operation would be similar to that of Section 3.2.

The problem of using the MAE fidelity criterion is very similar to the MSE. The values y_1 and y_2 are found by minimizing:

$$J_{MAE} = \sum_{i=1}^{k-q-1} \left|Z_i - y_1\right| + \sum_{i=k-q}^{k} \left|Z_i - y_2\right| \qquad (3.6)$$

where

y_1 = median of $(Z_1, Z_2, \ldots, Z_{k-q-1})$

y_2 = median of $(Z_{k-q}, \ldots, Z_k)$

Here again the non-parametric quantizer is arrived at by an exhaustive search. Results using these adaptive non-parametric quantizers and BTC are shown in Figures 3.2 and 3.3. Table 3.1 has the computed mean square error and mean absolute error measures for each image. As anticipated the MSE quantizer has the smallest computed mean square error measure and the MAE quantizer has the smallest computed mean absolute error measure. The performance of BTC is quite good when compared to these standard fidelity criteria. The advantage of using a MP non-parametric quantizer is that the quantizer formulation is available in closed form. Once one knows $\overline{m}_1$, $\overline{m}_2$ and q the quantizer is immediately specified. This greatly reduces the computational load.

As discussed in Chapter 2 it is possible to use a parametric quantizer once the probability density function of the pixels is known (or guessed). In Figures 3.2 and 3.3 results are presented for a parametric MSE quantizer where the pixels are assumed to be uniformly distributed over each block. The computed error measures are shown in Table 3.1.

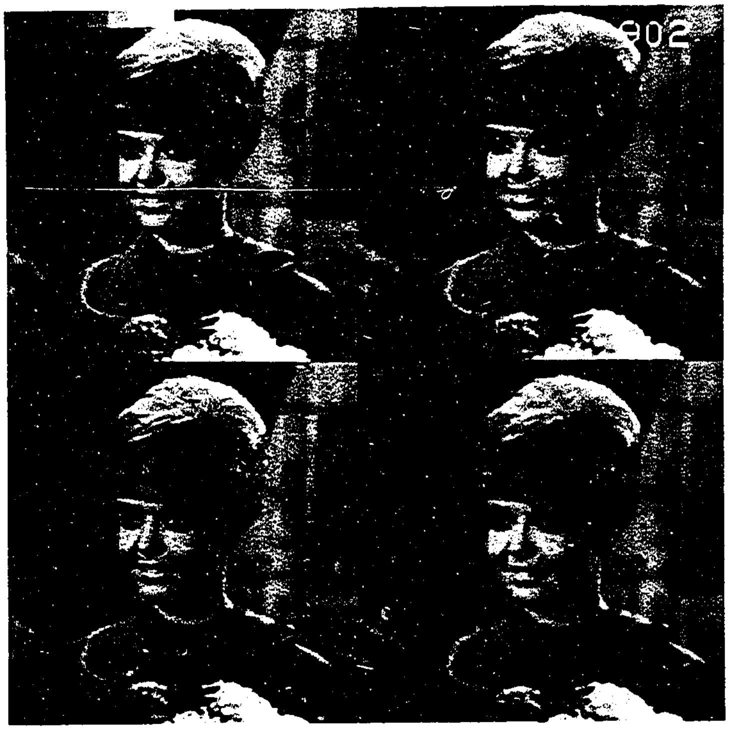

Figure 3.2 Results using various fidelity criterion. All representations are 2.0 bits/pixel. Upper left: minimum mean square error; Upper right: minimum mean absolute error; Lower left: moment preserving; Lower right: minimum mean square error and also assuming image data uniformly distributed each block.

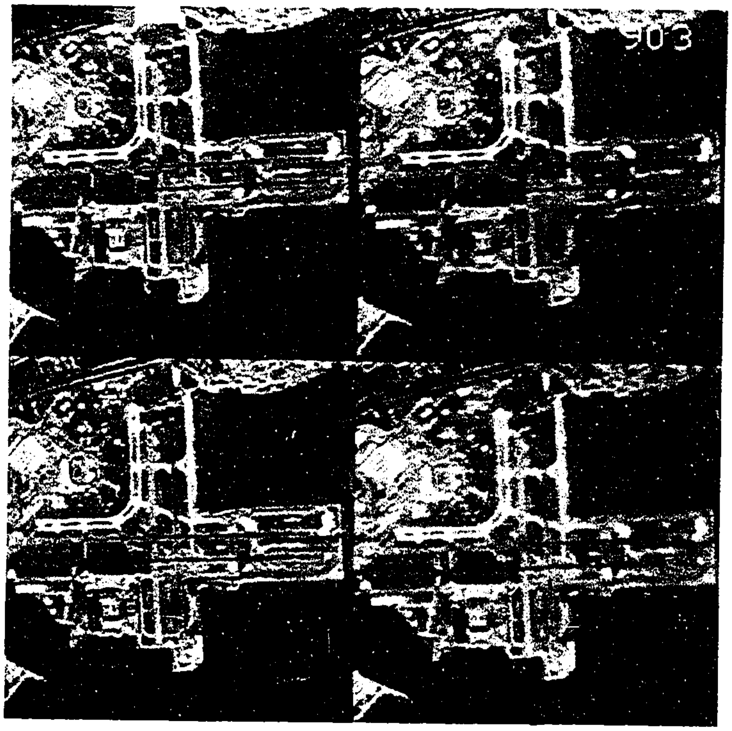

Figure 3.3 These four pictures were produced as described in Figure 3.2 using the other original.

Table 3.1 Computed Mean Square Error and Mean Absolute Error measures for various quantization schemes.

	Mean Square Error	Mean Absolute Error
Data from Figure 3.2:		
Using MSE quantizer	32.94	3.54
Using MAE quantizer	37.13	3.28
Using BTC	40.89	3.91
Using parametric MSE quantizer and assumed uniform density	44.64	4.23
Data from Figure 3.3:		
Using MSE quantizer	47.14	4.39
Using MAE quantizer	53.22	4.10
Using BTC	58.34	4.85
Using parametric MSE quantizer and assumed uniform density	64.02	5.42

3.4. BTC Modifications

One of the disadvantages of BTC is that the compression achieved corresponds to a data rate of only 2 bits/pixel. In many image coding schemes it is desired to obtain data rates lower than this.

As mentioned in Section 3.2, it is necessary to transmit some overhead information for the quantizer in each block. This overhead information tell the quantizer how to adapt the levels y_1, y_2 for each block. The information usually transmitted is $\overline{m}_1$ and $\overline{\sigma}$. One obvious way of decreasing the image representation is to assign less than 8 bits to $\overline{m}_1$ and $\overline{\sigma}$. Experimental evidence has indicated that it is possible to code $\overline{m}_1$ with 6 bits and $\overline{\sigma}$ with 4 bits. This allows for some savings and few perceivable errors upon reconstruction and a bit rate of 1.63 bits/pixel. While allowing only 6 bits for $\overline{m}_1$ is not acceptable in some cases it is possible to jointly quantize $\overline{m}_1$ and $\overline{\sigma}$. This is done by allowing 10 bits for $\overline{m}_1$ and $\overline{\sigma}$ where $\overline{m}_1$ is assigned more bits in blocks where $\overline{\sigma}$ is small and fewer bits where $\overline{\sigma}$ is large.

To allow for more savings in coding, there are various ways of coding the bit plane. A typical bit plane image for the girl's face is shown in Figure 3.4. This image was obtained by setting all the 1's in the various bit planes to white and 0's to black. This image is of course a binary image. The literative is very well developed in the area of coding binary (or two-tone) pictures [38], [55]. The entropy of the bit plane image has been approximated by using three different run-length coding models. The models were that of 1) a one dimensional runlength code differentiating between 0's and 1's, 2) a two dimensional Markov runlength code having 16 states and 3) a two dimensional Markov runlength code having 16 states and where the maximum likelihood state prediction error is coded; this is a modification of Preuss's TUH code [55]. The TUH code is one of the most sophisticated coding techniques used for binary images. The results are summarized in Table 3.2. Typical values indicate that it is necessary to allow about 0.90 bits/pixel in the bit plane instead of the nominal 1 bit/pixel. The fact that a large coding gain was not obtained can be understood by observing that in Figure 3.4 there are not large black and white regions indicating long white and black run lengths. Due to the poor performance of these codes in the presence of noise and the fact that the gain in compression would not be that great, (not to mention the overhead in calculation), the bit plane was not entropy coded to lower the compression.

By choosing the threshold of the quantizer at $\overline{m}_1$, it has been observed that partitioning of the pixels leads to some "unnatural" appearance of the data. For high resolution imagery, this manifested itself by some unacceptable coding artifacts. It would be desirable if the fidelity criterion allowed for a threshold choice. To be consistent with Chapter 2 we shall force the quantizer to preserve the third sample moment. To use this technique it is necessary to first construct a histogram of the pixel values in each block. Let

$$\overline{m}_3 = \frac{1}{k} \sum_{i=1}^{k} X_i^3 = \frac{1}{k} \sum_{i=1}^{k} Z_i^3 \qquad (3.7)$$

be defined as the third sample moment. The problem then is finding y_1, y_2, and q to preserve $\overline{m}_1$, $\overline{m}_2$, and $\overline{m}_3$. Since q specifies the number of

Table 3.2 Run length coding results (in bits/pixel) using the bit planes of Figure 3.4

Coding Type	Original BTC Algorithm	Third Moment Preserving BTC Algorithm
one dimensional run length coding	0.916	0.937
two dimensional run length coding (assuming 16 state Markov model)	0.877	0.903
TUH code (error predictor assuming a 16 state Markov model)	0.90	0.927

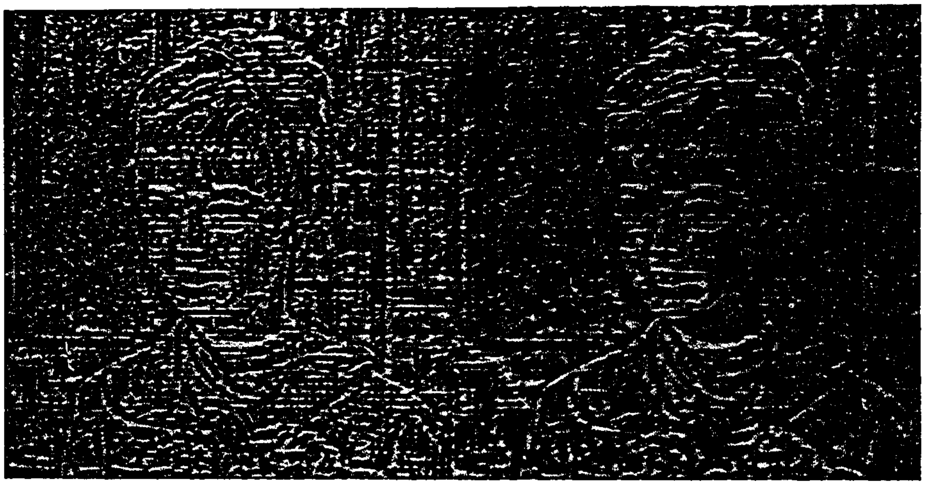

Figure 3.4 Bit plane images for the girl's face image. The original BTC algorithm is on the left. The third-moment preserving bit plane is on the right.

X_i's greater than X_{th}, by finding q, one has the threshold. It is necessary to sort the data because of the way q is defined. Equation 3.3 now becomes:

$$k\overline{m}_1 = (k-q)y_1 + qy_2$$

$$k\overline{m}_2 = (k-q)y_1^2 + qy_2^2 \qquad (3.8)$$

$$k\overline{m}_3 = (k-q)y_1^3 \div qy_2^3$$

These equation should be compared to Equation 2.3. After some algebra, the solution to Equation 3.8 is obtained:

$$y_1 = \overline{m}_1 - \sigma\sqrt{\left[\frac{q}{k-q}\right]}$$

$$y_2 = \overline{m}_1 + \overline{\sigma}\sqrt{\left[\frac{k-q}{q}\right]} \qquad (3.9)$$

$$q = \frac{k}{2}\left[1 + A\sqrt{\frac{1}{A^2+4}}\right]$$

where

$$A = \frac{3\overline{m}_1\overline{m}_2 - \overline{m}_3\ 2(\overline{m}_1)^3}{\overline{\sigma}^3}$$

$$\overline{\sigma} \neq 0.$$

If $\overline{\sigma} = 0$, then $y_1 = y_2 = \overline{m}_1$.

A problem with this solution is that in general the q arrived at using Equation 3.9 will be a non-integer, however, in practice q is rounded to the nearest integer. Equation 3.9 indicates that the threshold is nominally the sample median and biased one way or the other by the value of A. This is consistant with the interpretation of Equation 2.4. This method of threshold selection requires no extra computation by the receiver (decoder); however the transmitter (encoder) must now construct a histogram of each nxn block. In practice, it is very easy to sort 16 numbers (n=4) efficiently. This then does not increase the computational load for BTC significantly. It should be mentioned that this method of threshold selection is far easier than the non-parametric quantizers discussed in Section 3.3 since an exhaustive search is not necessary to find y_1, y_2, and q.

Figure 3.5 shows results using this new threshold selection. This new threshold technique improved the subtle features (such as near edges) of the image that are usually important in analysis of aerial photography imagery. This improvement will be discussed further in the following section.

The non-parametric MP quantizer described by Equation 3.9 is consistent with the results obtained in Chapter 2. This result can be generalized to a discrete orthogonal polynomial problem similar to Chapter 2 by a relative frequency interpretation of the output levels y_i's. This analysis will not be presented here because the application, BTC, is that of bandwidth compression where the number of output levels need to be relatively small. Also to maintain the small computational load of the algorithm it would not help if it were necessary as in the case

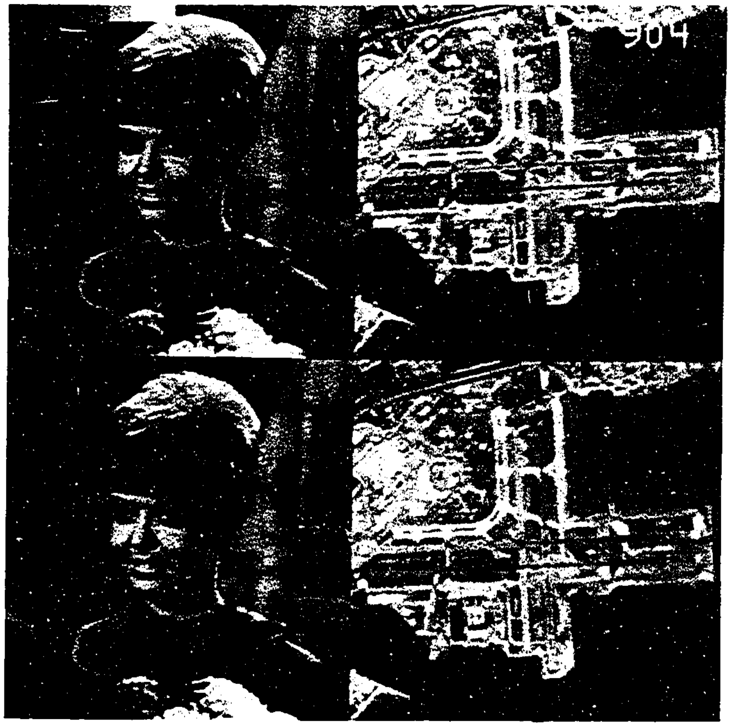

Figure 3.5 Results of BTC using third moment preserving threshold selection. Original images are at the top. Coded images are at the bottom. Data Rate is 2.0 bits/pixel.

of N=4 to compute the zeroes of a fourth order polynomial for every block. Although the zeroes of a fourth order polynomial can be written in closed form it would degrade the computational advantage of the algorithm. At data rates greater than 2 bits/pixel it would be easier to use DPCM with a parametric quantizer.

3.5. Performance Evaluation of BTC

Recently Purdue University has been engaged in studying the coding of aerial reconnaissance images for transmission over noisy channels [50], [51], [52]. In this study, various image coding techniques, including BTC, were evaluated by professional photo analysts. The coding techniques included transform, Hybrid [33] and two spatial techniques.

Although BTC did not do as well as transform coding in the photo interpreters evaluation, it proved superior to the other spatial techniques. In the presence of many channel errors, BTC was rated superior by the photo analysts to all of the other techniques. Some images used in the study are shown in Figures 3.6-3.9. In these figures BTC is compared with the Chen and Smith [16] method of transform coding and Hybrid coding. The computed mean square errors and mean absolute errors are shown in Table 3.3. Our study has verified the known phenomena that the mean square error and mean absolute error measure cannot be easily correlated with photo analysts' evaluations [66]. In some cases, images with greatly larger mean square errors were evaluated higher by the photo interpreters than images with smaller mean square errors. It should be noted that BTC requires a significantly smaller computational load and much less memory than transform or Hybrid coding. For instance, the Chen and Smith method besides requiring the two-dimensional Cosine

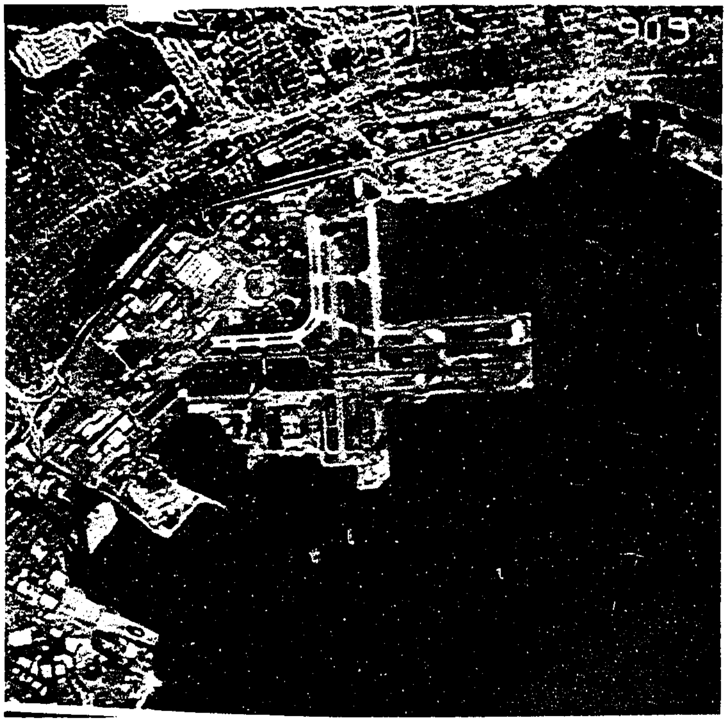

Figure 3.6 Original image used in comparison study. Image is 512 x 512 pixels with nominal 8 bits gray level resolution.

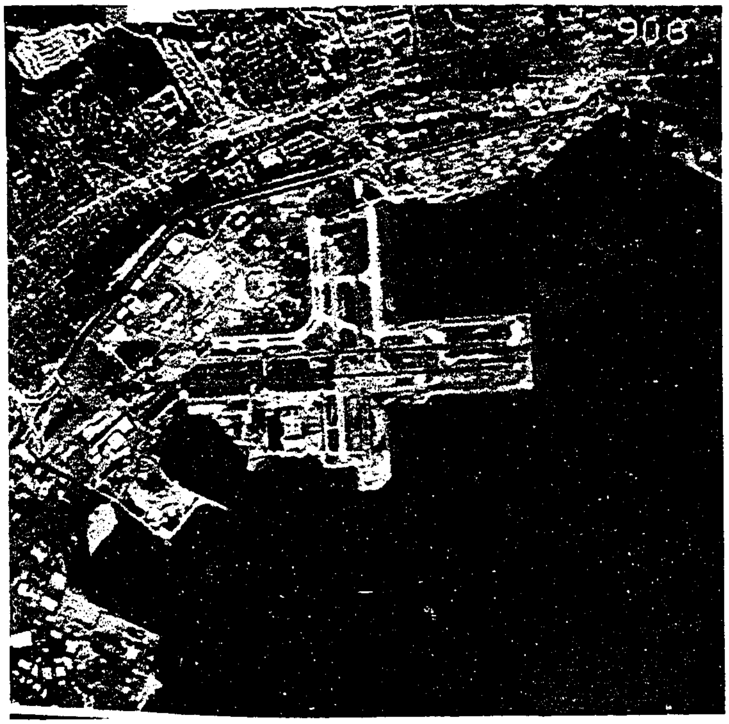

Figure 3.7 Results of coding original (Figure 3.6) using BTC with third moment preserving threshold selection. Data rate is 1.63 bits/pixel.

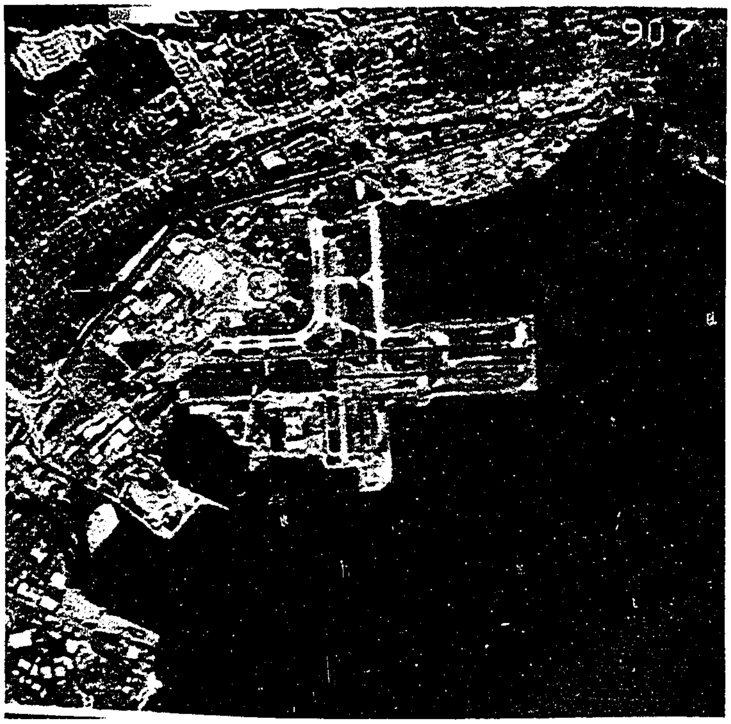

Figure 3.8 Results of coding original (Figure 3.6) using Chen and Smith [16] method of Cosine Transform coding. Data rate is 1.63 bits/pixel.

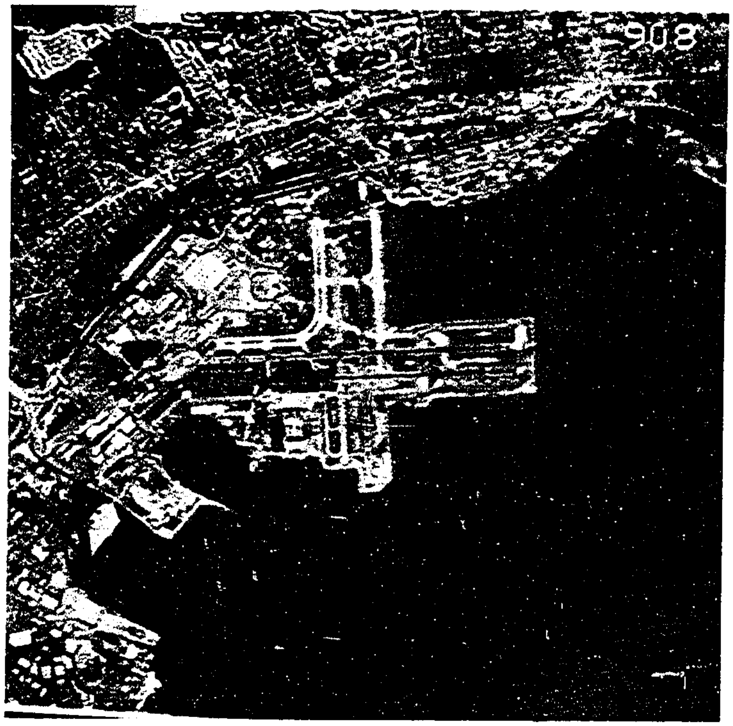

Figure 3.9 Results of coding original (Figure 3.6) using Hybrid coding [33]. Data rate is 1.63 bits/pixel.

Table 3.3 Computed Mean Square Error and Mean Absolute Error measures for comparison images shown in Figures 3.7-3.9 and Figures 3.13-3.16.

	Mean Square Error	Mean Absolute Error
Figure 3.7	84.22	5.94
Figure 3.8	67.13	6.32
Figure 3.9	125.84	6.12
Figure 3.13	115.09	6.29
Figure 3.14	115.31	7.06
Figure 3.15	140.33	6.67
Figure 3.16	74.67	5.72

transform over every 16x16 image block also requires multiple passes through the transform data to collect various statistics about the transform coefficients. It should also be mentioned that BTC requires no sophisticated error protection as do the other coding methods evaluated. Figures 3.10-3.12 show the difference pictures obtained by subtracting the coded images of Figures 3.7-3.9 from the original of Figure 3.6. These pictures give an indication of the relative coding artifacts manifested by each coding scheme. A medium gray indicates no coding error. Figures 3.13-3.15 show the results of each coding method in the presence of channel errors. The channel was assumed to be binary symmetric with the probability of a bit error of 10^{-3}.

As with all non-information preserving image coding, coding artifacts are produced in the image. It became apparent very early in this study that BTC produces artifacts that are very different than transform and hybrid coding. These artifacts are usually produced in regions around edges and in low contrast areas containing a sloping gray level. As mentioned above, BTC does produce sharp edges; however, these edges do have a tendency to be ragged. Transform coding usually produces edges that are blurred and smooth. The second problem in low contrast regions is due to inherent quantization noise in the one bit quantizer. Here sloping gray levels can turn into false edges. Preliminary experiments have indicated that pre and post processing of the image can reduce the effects of both these artifacts while simultaneously reducing the mean square error and mean absolute error. For example the computed mean square error dropped by one half for Figure 3.7 when a simple circularly symmetric moving average low pass filter was used in a post-

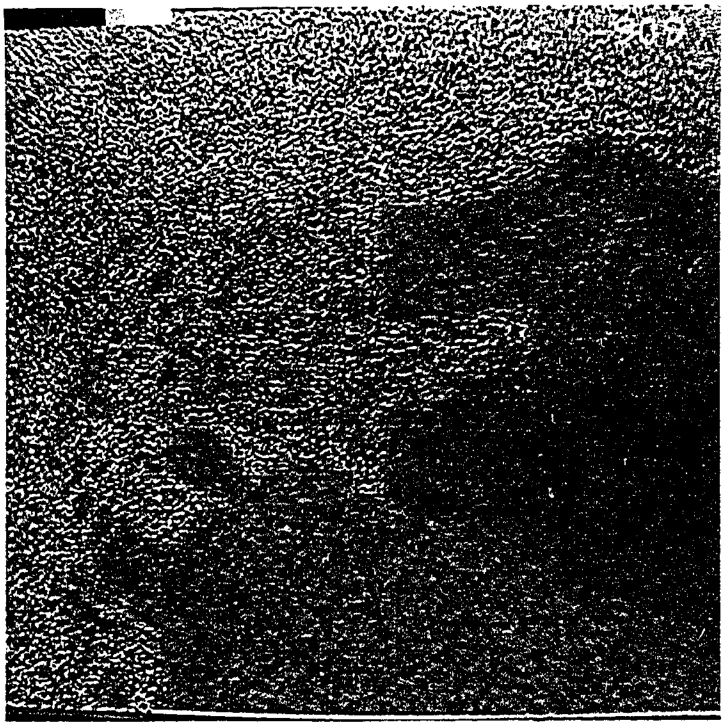

Figure 3.10 The difference picture for BTC (Figure 3.7). Medium gray corresponds to no coding artifacts. Gray scale expanded by factor of 5.

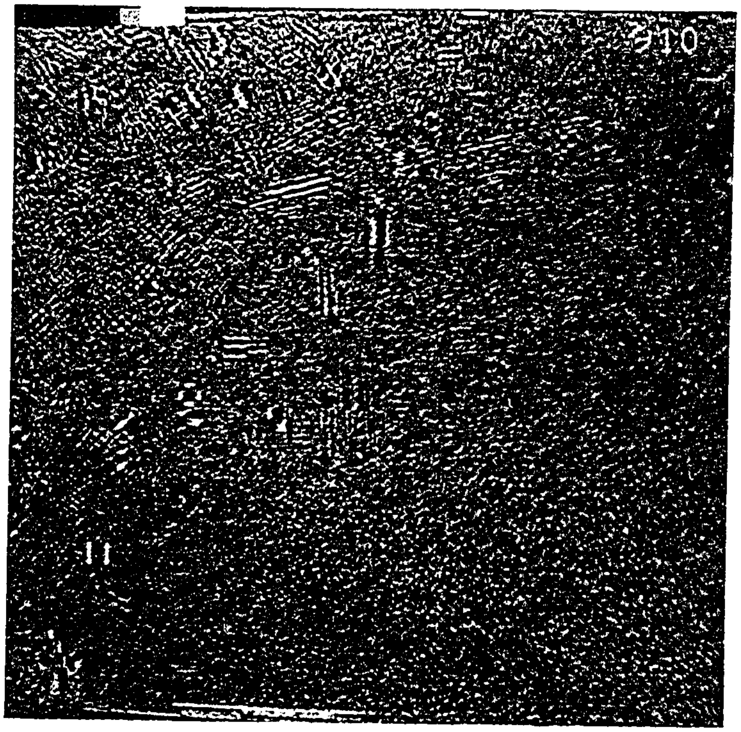

Figure 3.11 The difference picture for Chen and Smith (Figure 3.8). Medium gray corresponds to no coding artifacts. Gray scale expanded by factor of 5.

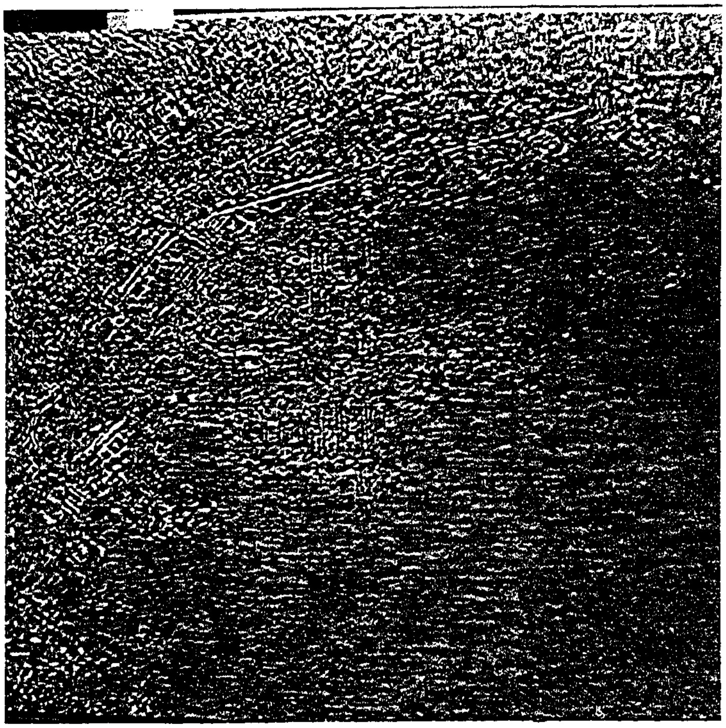

Figure 3.12 The difference picture for Hybrid coding (Figure 3.9). Medium gray corresponds no coding artifacts. Gray scale expanded by factor of 5.

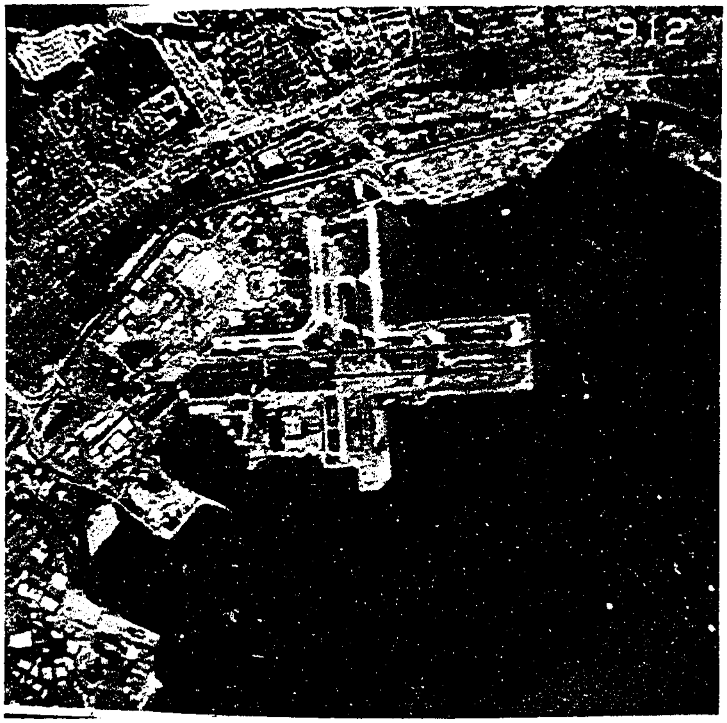

Figure 3.13 BTC coding with channel (bit) error probability of 10^{-3}. Data rate is 1.63 bits/pixel.

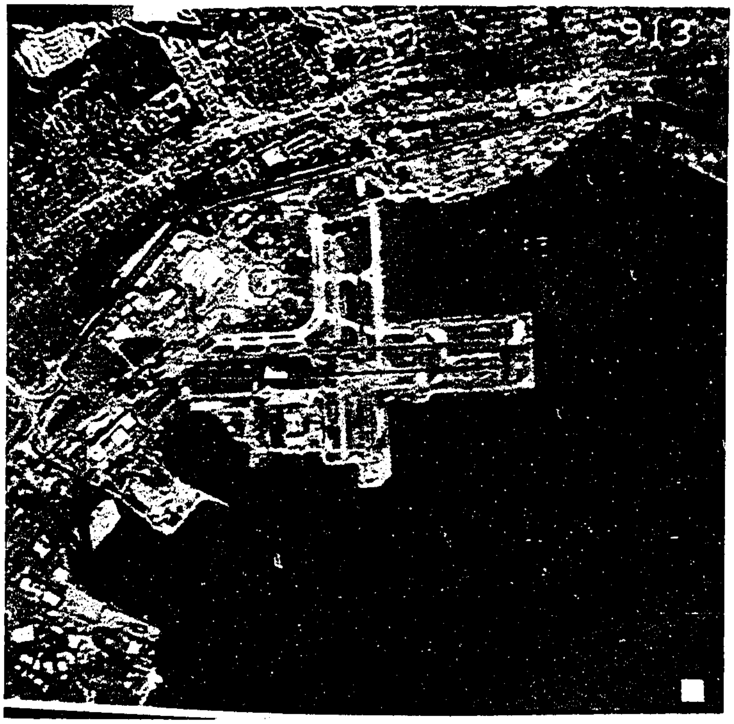

Figure 3.14 Chen and Smith coding with channel (bit) error probability of 10^{-3}. Data rate is 1.63 bits/pixel.

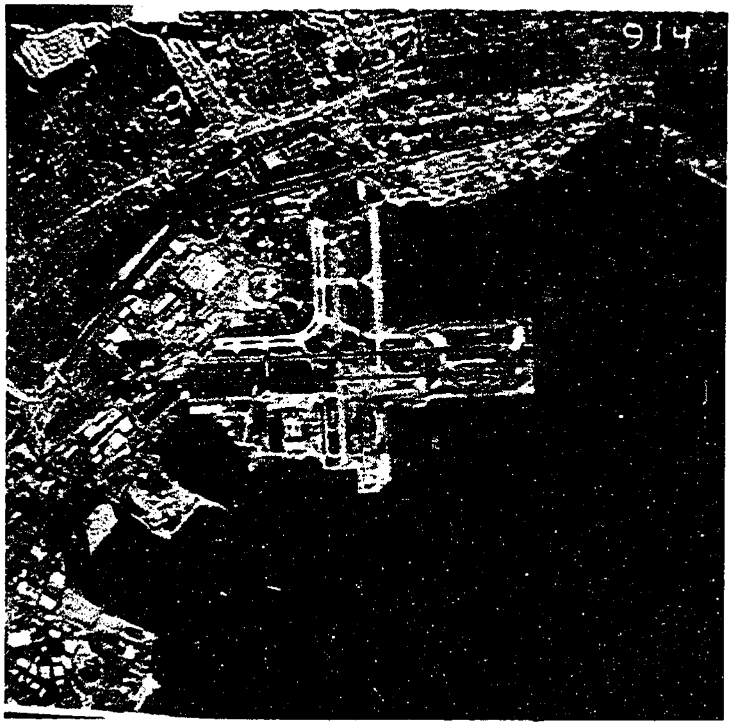

Figure 3.15 Hybrid coding with channel (bit) error probability of 10^{-3}. Data rate is 1.63 bits/pixel.

coding scheme. This type of filtering reduces the quantization noise at the cost of slightly blurring the image. It should be emphasized that the above coding artifacts are problems in high resolution aerial reconnaissance images where man-made objects are important (i.e., edges). These coding artifacts usually are not any problem in typical "head and shoulders" imagery.

One of the advantages of BTC is that of edge preservation. Figure 3.16 shows an enlarged section of an image coded using the Chen and Smith method and BTC. The edges using the transform coding method are not as sharp as the BTC image.

3.6. Hybrid Formulation of BTC

One of the problems that BTC has is that it is really a one-dimensional quantization technique. In no way does BTC exploit the two-dimensional nature of the image within each block as do most other forms of image coding. For example in two-dimensional transform coding the transform coefficients contain information about variations in the picture in both directions. BTC only uses the moment information which gives no insight into the two dimensional variations within the picture. Also BTC generally has a poor response near the spatial frequency of 1/2 cycle per block.

One method to improve both of the problems above is a hybrid formulation. First a highly compressed Cosine transform coded image is subtracted from the original image. For the results presented here the transform picture was obtained by taking the two-dimensional Cosine transform over 16 x 16 pixel blocks. Only the eight non d.c. coefficients in the low frequency section of each block were retained. This

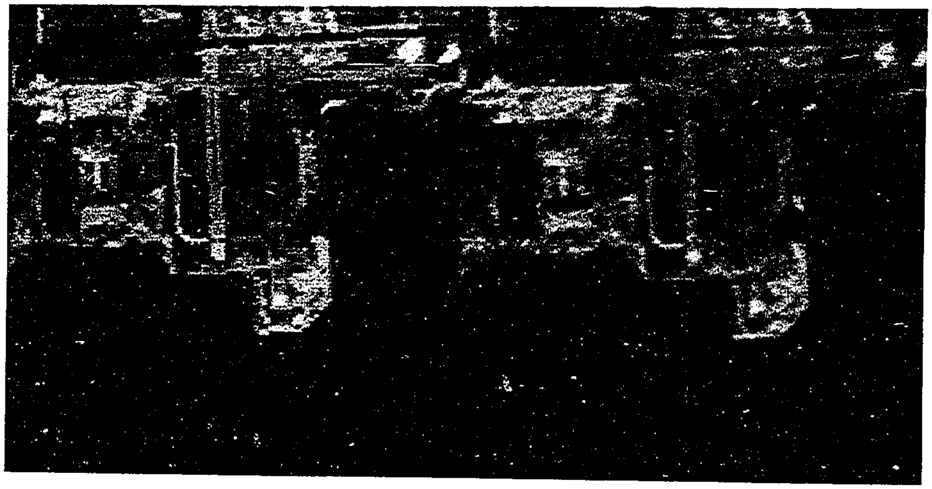

Figure 3.16 Enlarged section of image from Figure 3.7 (left) and Figure 3.8 (right).

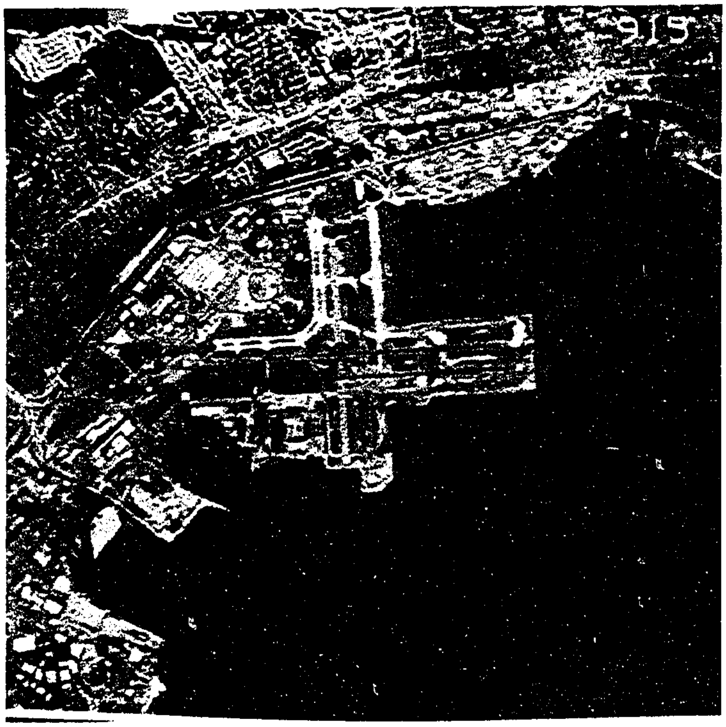

Figure 3.17 Results of Hybrid Formulation of BTC. Data rate is 1.88 bits/pixel.

corresponds to a zonal filtering method with a representation of 0.25 bits/pixel for the highly compressed image. BTC was then used on this difference picture and the recombination formed at the receiver. While this did increase the computational load, the improvement seemed to be significant enough to give this method further attention. Figure 13.17 presents results of this hybrid method. Table 3.3 has the mean square error and mean absolute error measures for Figure 3.17. This technique exploits the edge preservation of BTC and helps in the low contrast regions of the image by improving the frequency response.

CHAPTER 4

IMAGE MODELING

4.1. Introduction

In recent years much work has gone into obtaining reasonable spatial models for images [27], [68], [79], [10]. This has led to some very good results in areas such as image enhancement and coding [31], [74], [60], [82], [56]. For example the Cosine transform which is widely used in image transform coding (as compared to BTC in Chapter 3) can be shown to be nearly optimal if the image is a sample picture of a Markov field [4]. Various studies using rate distortion theory for examining coding performance usually assume a Gauss-Markov model for images [19], [34], [62]. In this context studies of the nature of the human observer in image systems have been undertaken to improve the image model.

In this chapter we will examine Gauss-Markov image modeling from the point of view of classical time series analysis using the method of least squares [12], [42]. A seasonal one-dimensional model is obtained and used to regenerate test images. This approach has shown to have some promise in the area of texture modeling [47], [78]. We will use these models to generate background scenes and for texture synthesis for use in an image coding application. The suitability of Gauss-Markov image models will be demonstrated from these synthesized scenes. The model is also used to introduce differential pulse code modulation scheme (DPCM) to be presented in Chapter 5.

4.2. The Image Model

Suppose each NxN discrete image is described as a matrix y(i,j). If the image is assumed to be a sample picture from a two-dimensional discrete homogeneous Gauss-Markov field, one can show that the pixel at y(i,j) has representation [83]:

$$y(i,j) = \sum_{\substack{m=0 \\ m=n\neq 0}}^{a} \sum_{n=0}^{b} \theta_{mn}\, y(i-m,j-n) + u(i,j) \tag{4.1}$$

where

a) $E[u(i,j)\, y(i-m,j-n)] = 0$; $m,n \geq 0$, $m+n > 0$

b) $E[u(i,j)u(k,\ell)] = \sigma^2 \delta_{ik} \delta_{j\ell}$

c) $E[u(i,j)] = 0$

d) $E[\bullet]$ is the expectation operator

e) $\delta_{ik} = 1$; if $i=k$,

$= 0$; if $i \neq k$

f) u(i,j) is the noise driving process

g) θ_{mn} weights to be estimated

The initial conditions for this model consists of the first a rows and b columns. Equation 4.1 indicates that the pixel gray level at (i,j) is related to the pixels in the recursion region weighted by the respective θ's plust some Gaussian white noise u(i,j). Condition of Equation 4.1 indicates the noise term u(i,j) is uncorrelated with the pixel values in the recursion region. Conditions b and c indicate that u(i,j) is zero and white (uncorrelated). Therefore u(i,j) is a zero mean independent

Gaussian random field. The θ's are sometimes called the regression coefficients. Models of this type are referred to as "quarter-plane" models. There are more general recursion regions that include more neighbors but the problem of obtaining accurate parameter estimates is more complicated. This model says that a sample picture can be generated by driving a two-dimensional recursive digital filter with Gaussian white noise. What we shall do is show how a real image can be modeled by the two dimensional random field described by Equation 4.1.

The special case of a Gauss-Markov field with a and b equal to one is sometimes assumed for the picture:

$$y(i,j) = \theta_1 y(i-1,j) + \theta_2 y(i-1,j-1) + \theta_3 y(i,j-1) + u(i,j) \qquad (4.2)$$

By row-concatenation one can obtain a <u>one-dimensional</u> formulation of Equation 4.2:

$$\text{let} \quad (i,j) \rightarrow k = (i-1)N+j$$

then

$$y(k) = \theta_1 y(k-N) + \theta_2 y(k-N-1) + \theta_3 y(k-1) + u(k) \qquad (4.3)$$

This formulation of $y(\cdot)$ and the labeling of the pixels is suggestive of horizontal line scanning. A similar transformation could also be used to obtain a vertical scan type of formulation. The formulation of Equation 4.3 is shown in Section 4.3 to work quite well for parameter estimation of the θ's and σ^2.

Except for the handling of the initial conditions, Equation 4.3 can easily be recognized as a seasonal autoregressive time series [42] where the seasonal period is N. One can obtain a more compact version of Equation 4.3:

$$y(k) = \underline{\theta}^T \underline{Z}(k-1) + u(k) \qquad (4.4)$$

where

$$\underline{\theta}^T = [\theta_1, \theta_2, \theta_3]$$

$$\underline{z}^T(k-1) = [y(k-N), y(k-N-1), y(k-1)]$$

The vector $\underline{z}(k-1)$ is called the "past history" of the process.

Throughout the rest of this chapter, the image will be assumed to be described by the Gauss-Markov field of Equation 4.1 and through the change of variables discussed above a seasonal autoregressive time series will be obtained.

4.3. Parameter Estimation

To fit the model given in Equation 4.4 to a particular picture one has to obtain estimates of the regression parameters $\underline{\theta}$ and white noise variance σ^2. The method of least squares was chosen to obtain the parameter estimates for the model. Given Equation 4.4 one wishes to find $\underline{\theta}$ and σ^2 such that the squared error between the actual pixel value and the pixel value generated by the model is minimum. The squared error is given by

$$J_N(\underline{\theta}) = \sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} \left[y(k) - \underline{\theta}^T\underline{Z}(k-1)\right]^2$$

where I.C. = {index of initial condition set} (4.5)

It is easily shown that the values of $\underline{\theta}$ and σ^2 which minimize $J_N(\underline{\theta})$ are

$$\underline{\hat{\theta}}(N_1) = \left[\sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} (\underline{Z}(k-1)\underline{Z}^T(k-1))\right]^{-1} \sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} y(k)\underline{Z}(k-1) \qquad (4.6)$$

$$\hat{\sigma}^2 = \frac{1}{N_1} \sum_{\substack{k=1 \\ k \neq I.C.}}^{N^2} [y(k) - (\underline{\hat{\theta}}(N_1))^T \underline{Z}(k-1)]^2$$

where $N_1 = N^2$ - (number of points in initial condition set)

Since it is assumed the process u(•) is Gaussian, the results obtained in Equation 4.6 are the same as the conditional maximum likelihood estimates of $\underline{\theta}$ and σ^2 [42].

The term

$$w(k) = y(k) - (\underline{\hat{\theta}}(N_1))^T\underline{Z}(k-1) \qquad (4.7)$$

is called the residual of the model. If the image is actually described by Equation 4.4 and if $\underline{\theta} = \underline{\hat{\theta}}(N_1)$, then w(k) would of course be the white noise driving process. In practice a complex image cannot fully be described by a simple model as in equation 4.4 and therefore the residuals can be used as an indication of how good the model actually fits the particular picture. In general then, it would be desirable that the residuals represent a zero-mean white Gaussian sequence. Various statistical tests can be performed on the residual sequence to verify the above properties [12], [42]. One very simple test that can illustrate the validity of the model is to let $\underline{\theta} = \underline{\hat{\theta}}(N_1)$ and $\sigma^2 = \hat{\sigma}^2$. Then regenerate the image according to Equation 4.4 with the initial conditions and an independent Gaussian random number generator. The quality of the regenerated image of course would be the final test.

4.4. Image Regeneration and Applications

The model given in Equation 4.2 was used for this study. Due to the gross non-homogeneous nature of large images we separated the image into NxN subpictures and fitted the same model type to each subpicture of a large picture. In other words, all the models for each large image had the same form but a different parameter set $\underline{\hat{\theta}}(N_1)$ and $\hat{\sigma}^2$ for each subpicture.

Results were obtained for four large pictures each of which were 256x256 pixels. The subpicture size was chosen to be 16x16 pixels, so that each 256x256 scene had 256 parameter sets associated with it. After the parameter set was obtained each image was regenerated as described in Section 4.3 with a Gaussian random number generator (provision was made for the cases where the residual was not zero-mean). Results are shown in Figures 4.1 and 4.2 for the model of Equation 4.2. The texture images are that of cork and wood. The results obtained are interesting in that the picture was generated from the initial conditions and the parameter set along with the model description and a random number generator. One should remember that in each subpicture the initial conditions used were the actual pixel values for those rows and columns respectively.

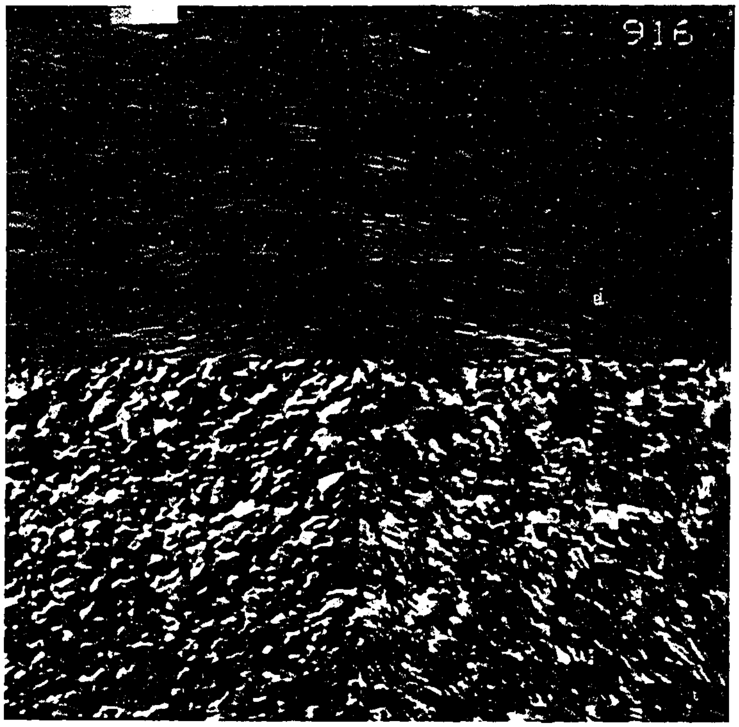

Figure 4.1 Original images are on the left. The regenerated images are on the right. The textures are wood (top) and cork (bottom).

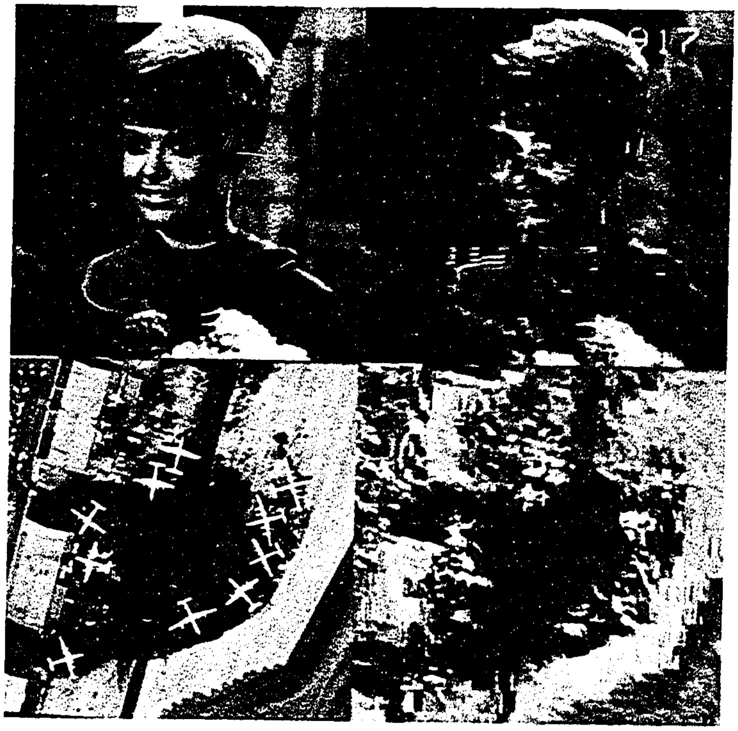

Figure 4.2. Original images are on the left. The regenerated images are on the right.

There is no guarantee using the method of least squares that the model obtained will be stable. In fact, in the girl's face seen in Figure 4.2 one can see subpicture blocks which have an unstable model formulation. One can observe this by noticing the white and black streaking in the regenerated scene toward the bottom of the picture. The model of Equation 4.2 is of course low-pass in nature and any complex image properties, such as edges, are not accounted for by the model.

An obvious application of the model is synthetic texture generation. While the textures shown in Figure 4.1 are generated using the model of Equation 4.2 the data rate achieved is really not that small. This is due to the fact that for each 16x16 subpicture the initial conditions used are the first row and first column of the original image. This amounts to 31 initial conditions. If the original image gray level resolution is 8 bits/pixel and if we assume that θ_1, $\hat{\theta}_2$, $\hat{\theta}_3$, and $\hat{\sigma}^2$ can be represented adequately at 8 bits. This results in a data rate of 1.09 bits/pixel. This is not a very good rate since other coding schemes such as transform coding would give a better representation of the texture. If the initial condition set could be reduced then the compression would be greater. In fact if the initial condition set could be eliminated the data rate would be 0.125 bits/pixel.

We have regenerated the textures of Figure 4.1 using a smaller initial condition set consisting of only using the original initial conditions in every fourth block. In the other blocks a synthesized initial condition set is obtained by using the regenerated pixel values in the blocks to left and above the block being regenerated. In other words we update the regeneration procedure by using the true initial conditions

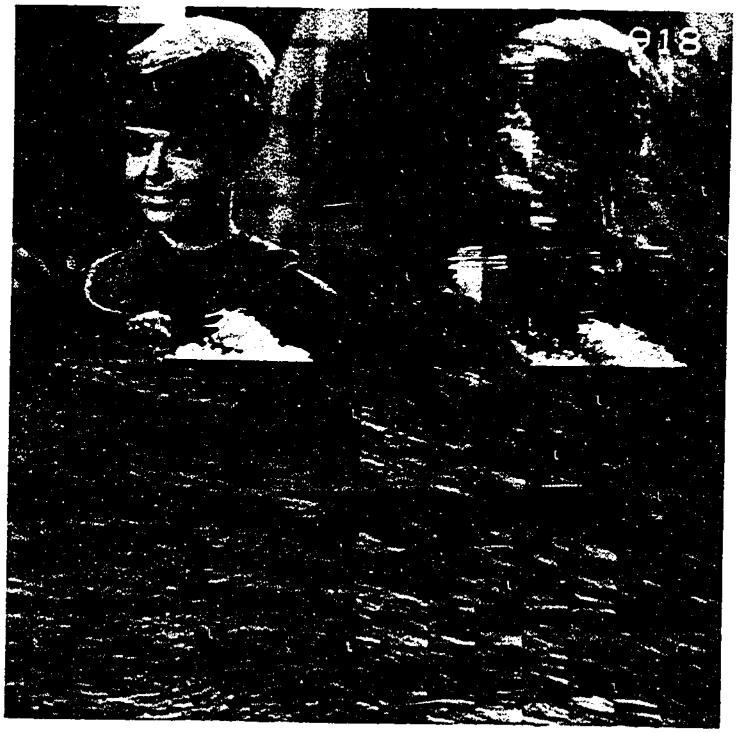

Figure 4.3. Results of image regeneration using modified initial condition set. The original images are on the left. The regenerated images are on the right.

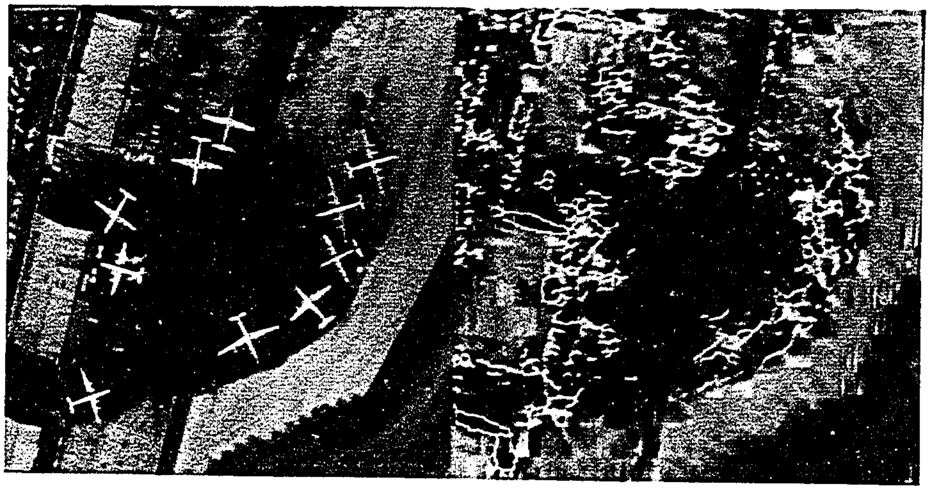

Figure 4.4. A regenerated background scene with edges displayed on top. Original image is on the left.

in every fourth 16x16 subpicture and using synthesized initial conditions for the other blocks. These results are shown in Figure 4.3 for two of the four scenes. The data rate becomes 0.37 bits/pixel. These results suggest that Gauss-Markov models are adequate for textures. In an image coding scheme the texture model would only have to be transmitted to the receiver along with periodically updated initial conditions and the receiver could regenerate the texture locally.

An application would be that of generating highly compressed and relatively crude background scenes for use in real time displays such as remotely piloted vehicle guidance. The background would be generated very crudely by the Gauss-Markov model and then geometric features such as edges would be displayed on top of the regenerated background. The edge features could be used for target identification or classification. The background being supplied to the human observer would help with visual cues as to how the target fits in the scene. A demonstration of this application is shown in Figure 4.4. The edges were obtained using modified Frei and Chen edge detection [28] and edge thinning using an algorithm due to Reeves [67].

These results indicate that Gauss-Markov models do not describe complex scenes fully but these models can possibly be used for many applications. Results using other types of models for image synthesis are presented in [21].

CHAPTER 5

DIFFERENTIAL BLOCK TRUNCATION CODING

5.1 Introduction

Differential Pulse Code Modulation (DPCM) has been widely used in image bandwidth compression schemes [17], [58]. In DPCM the difference between the actual pixel gray level and a predicted gray level is quantized and transmitted (encoded). At the receiver (decoder) the quantized difference signal and predictor model are then used to reconstruct the image. The performance of DPCM is based on the predictor model and the difference quantizer. In this chapter we will examine the use of a non-parametric MP quantizer for the difference. We call this method Differential Block Truncation coding (DBTC). We shall assume that the predictor is fixed and that the quantizer is block adaptive i.e., the quantizer will be allowed to change only between non-overlapping blocks in the picture. The design of DPCM quantizers has been discussed widely in the literature [45], [77], [41]. The study by Sharma and Netravali [71] presents quantizer design based on the squared error fidelity criterion and weighting function relative to a visual threshold function. The quantizers were found using a geometric search (an exhaustive search). In particular we examine the case of coded pictures having representations of 1-1.5 bits/pixel where the quantizers have only two levels (one bit). The quantizers will have both a non-parametric and parametric formulation based only on the sample statistics of the difference signal in each block. It has previously been shown in Chapter 3 that the one bit non-parametric moment preserving quantizer can be written in closed form.

5.2 Moment Preserving Quantizers in DPCM

In this section the use of moment preserving (MP) quantizers will be discussed in DPCM. The classical DPCM block diagram is shown in Figure 5.1. The MP quantizer used for the difference signal preserves the moments of only the difference signal. In particular for a one bit MP quantizer, where the threshold of the quantizer is assumed to be the mean of the error, only the first two moments of the error are preserved. The question then is if the first two moments of the error are preserved are the first two moments of the input signal preserved in the reconstructed signal, i.e., does

$$\begin{array}{c} E[e_n] = E[\tilde{e}_n] \\ E[e_n^2] = E[\tilde{e}_n^2] \end{array} \longrightarrow \begin{array}{c} E[Y_n] = E[\tilde{Y}_n] \\ E[Y_n^2] = E[\tilde{Y}_n^2] \end{array} \tag{5.1}$$

where

e_n = nth difference signal

$\tilde{e}_n$ = nth quantized difference signal

Y_n = nth input signal

$\tilde{Y}_n$ = nth reconstructed signal

The predictor as depicted in Figure 5.1 can be either one or two dimensional. For the analysis presented in this section we will index the predictor by only a single variable without loss of generality. It is

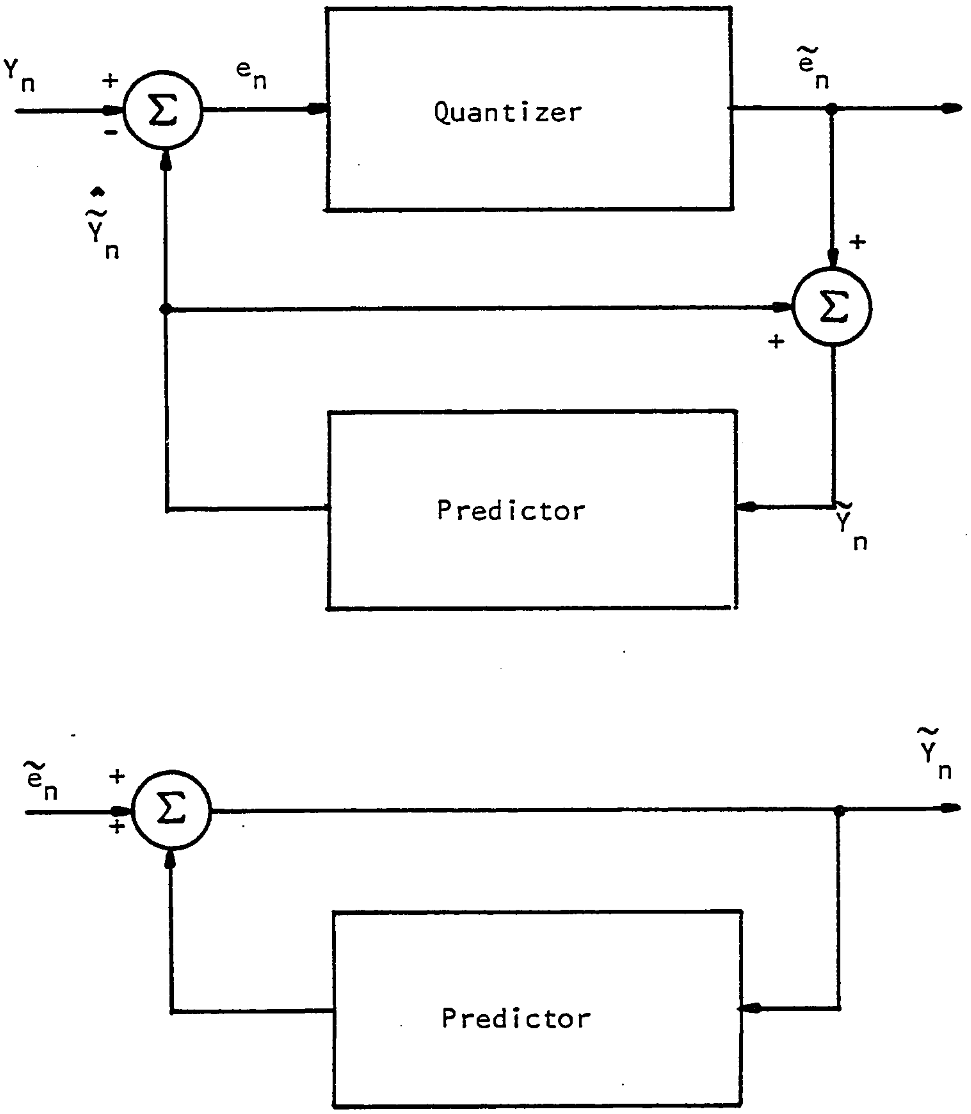

Figure 5.1 The Differential Pulse Code Modulation Block Diagram. Transmitter (encoder) is on the top. Receiver (decoder) is on the bottom.

easy to show:

$$\tilde{e}_n - e_n = \tilde{Y}_n - Y_n = E_n \qquad (5.2)$$

where E_n = quantization error in nth sample. This enables us to state a lemma.

Lemma 5.1. The first moment of reconstructed signal is preserved if the first moment of the error is preserved.

Proof: Using Equations 5.1 and 5.2 we have

$$E[\tilde{e}_n] - E[e_n] = E[\tilde{Y}_n] - E[Y_n]$$

but $\qquad E[\tilde{e}_n] = E[e_n]$

hence $\qquad E[\tilde{Y}_n] = E[Y_n]$

QED

This result will be used in the following theorem.

Theorem 5.1. Given that $E[\tilde{e}_n^2] = E[e_n^2]$ and Lemma 5.1. The second moment of the reconstructed picture is preserved ($E[Y_n^2] = E[\tilde{Y}_n^2]$) if and only if $E[\hat{\tilde{Y}}_n E_n] = 0$.

Proof:

a) ("if")

rewriting Equation 5.2 as

$$\tilde{e}_n = (\tilde{Y}_n - Y_n) + e_n$$

$$E[\tilde{e}_n^2] = E[(\tilde{Y}_n - Y_n)^2] + 2E[e_n(\tilde{Y}_n - Y_n)] + E[e_n^2]$$

$$0 = E[\tilde{Y}_n^2] - 2E[\tilde{Y}_n Y_n] + E[Y_n^2] + 2E[e_n(\tilde{Y}_n - Y_n)]$$

since $E[\tilde{e}_n^2] = E[e_n^2]$

$$E[\tilde{Y}_n^2] = -E[Y_n^2] + 2E[\tilde{Y}_n Y_n - e_n(\tilde{Y}_n - Y_n)]$$

but $e_n = Y_n - \hat{\tilde{Y}}_n$

where $\hat{\tilde{Y}}_n$ = nth predicted quantized signal.

hence

$$E[\tilde{Y}_n^2] = E[Y_n^2] + 2E[\hat{\tilde{Y}}_n E_n] \qquad (5.3)$$

Therefore

$$E[\tilde{Y}_n^2] = E[Y_n^2]$$

if $E[\hat{\tilde{Y}}_n E_n] = 0$ (5.4)

b) ("only if")

Since Equation 5.3 is in general true if $(E[\tilde{e}_n^2] = E[e_n^2])$ this part of the proof proceeds backwards from Equation 5.4.

QED

Equation 5.4 can be alternatively written as (using Equation 5.2):

$$E[\hat{\tilde{Y}}_n(\tilde{e}_n - e_n)] = 0 \tag{5.5}$$

One could argue heuristically that if the quantization error was uncorrelated with input that the condition of Equation 5.4 could be met. Alternatively another heuristic argument could be put forth the assuming $\tilde{Y}_n$ is the optimal predictor and $\tilde{e}_n \approx e_n$ hence using the orthogonality principle we have the desired condition. Unfortunately neither of the arguments can be successfully applied. Since the quantization error is negatively correlated with the input signal and since our application involves a one bit quantizer the quantization error is relatively large. This result could obviously be extended to any number of moments, however, Theorem 5.1 is sufficient to indicate that preserving the moments of the difference signal usually will not preserve the moments in the reconstruction.

5.3 Some Preliminaries Relative to DBTC

Throughout this chapter we shall assume a two-dimensional predictor identical to the Gaussian-Markov model developed in Equation 4.3 of Chapter 4:

$$\hat{y}(i,j) = \theta_1 y(i-1,j) + \theta_2 y(i-1,j-1) + \theta_3 y(i,j-1) \tag{5.6}$$

It is possible to first fit this prediction model to the image as in

Chapter 4 and [21]. However, because we are interested in bandwidth compression with a relatively low data rate we shall fix the predictor coefficients at $\theta_1 = .8$, $\theta_2 = -.6$, and $\theta_3 = .8$. This will eliminate the overhead information that would be necessary if the predictor model adapted at every block. While we make no claim of optimality by assuming a fixed predictor we are only interested in investigating the performance of the quantizer. This given fixed predictor has no "leak" associated with it, i.e. the predictor model is marginally stable. This statement will be modified later.

The prediction error in the absence of quantization then is identical to the Gaussian-Markov residual of Equation 4.7 (i.e. $e_n = w(n)$). The prediction error is quantized using the feedback arrangement of Figure 5.1 to keep the quantization error (noise) E_n from propagating. This is evident from Equation 5.2. It should be pointed out that this method of one bit DPCM should not be confused with Delta Modulation. In Delta Modulation the sampling rate of the input signal is increased much greater than the Nyquist rate in order to achieve high correlation between the samples. In our formulation of DPCM the sampling rate is identical to that used for the other coding schemes discussed in Section 3.6. The basic philosophy of DPCM is that the difference, e_n, will have an inherently lower variance and hence it will require fewer bits for a given degree of accuracy.

From the results of Chapter 4 it is obvious that the difference signal contains the relatively high frequency content of the input signal. For the application of image coding this would indicate that the edge information is contained in the difference signal. From Chapter 3 we know that BTC works quite well on these types of data. In the next section we will compare this differential scheme using the non-parametric MP quantizer to results previous published for the model of the probability density function of the error signal.

5.4 Comparison Study of DBTC

The study presented is not meant to be a comprehensive review of DPCM; we will only investigate the feasibility of using the MP quantizer. O'Neal [58] has shown that the difference signal can be modeled as having a Laplacian density function. In this section we will compare the use of the parametric MSE one bit quantizer for a Laplacian density (Table 2.5) with that of a non-parametric MP quantizer. The later system is of course DBTC. We will use the basic non-parametric MP algorithm; i.e., that of preserving only the first two sample moments. We will base our comparison on computed mean square error and subjective evaluation.

The coding scheme used is that of Figure 5.1 where the quantizer adapts at every block of pixels. For this study nxn block size will be 8x8. All of the quantizer schemes require that sample moments be calculated in order that quantizer parameters be obtained. For the parametric quantizer we shall assume that the sample moments of error signal represent a "reasonable" estimator of the moments. These sample moments are obtained by first doing an open-loop prediction of the error signal; the quantizer parameters are obtained and then the loop is closed for closed-loop prediction and quantization. A better method would be to estimate the quantizer parameters at each step (i.e. a truly adaptive scheme).

For each quantizer scheme, a bit plane and open-loop sample mean and variance are obtained (k = nxn):

$$\overline{m}_e = \frac{1}{k} \sum_{i=1}^{k} e_k$$
$$\overline{\sigma}_e^2 = \frac{1}{k} \sum_{i=1}^{k} (e_k - \overline{m}_e)^2 \qquad (5.7)$$

As previously mentioned in Chapter 4 it is necessary to send some initial conditions to get the predictor started. For this study initial conditions were used only in the image blocks along the left and upper edges of the total image. Empirical evidence indicates it is necessary to assign $\overline{m}_e$ only 4 bits and $\overline{\sigma}_e$ only 3 bits; this leads to a data rate of 1.18 bits/pixel. This should be contrasted to the data rate obtained by straight BTC of 1.63 bits/pixel. The predictor model tends to add a little smoothing to the reconstructed image which makes the image cosmetically more pleasing. Results using these two quantizers are shown in Figures 5.2-5.3 along with their difference pictures in Figures 5.4-5.5. The computed mean square errors and mean absoluteness are shown in Table 5.1. While the error measures are not significantly different the DBTC coded image of Figure 5.2 looks sharper where the other coded image looks slightly blurred. The blurring in Figure 5.3 can be seen by observing the edge information that is "lost" as evident in Figure 5.5. The mean square errors are large because in general the open loop prediction causes the apparent average gray level in each block to shift. Only a small shift in average gray level will cause a relatively large change in the computed mean square error. Figures 5.6-5.7 show results in the presence of channel errors. These reconstructed channel error images are interesting in that the 45° bias in the predictor is

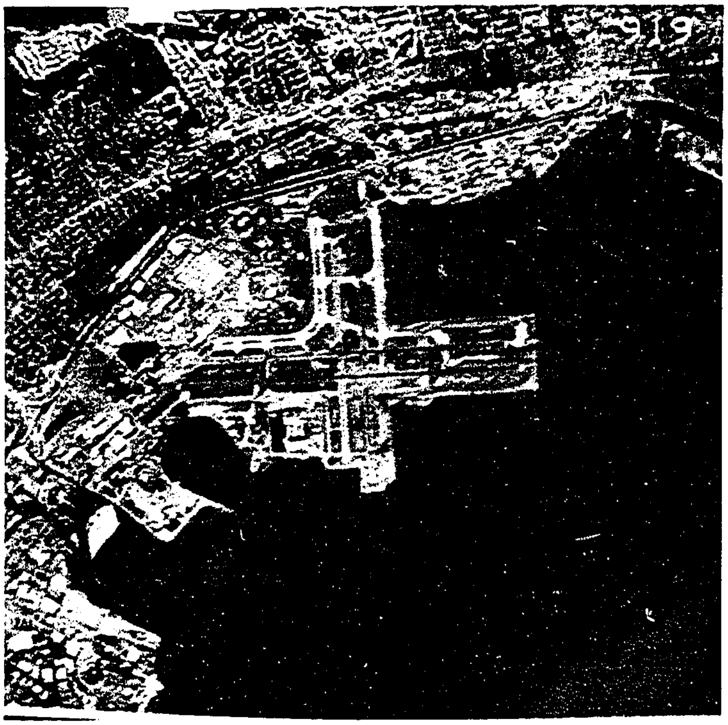

Figure 5.2 Results using DBTC with bit rate of 1.18 bits/pixel.

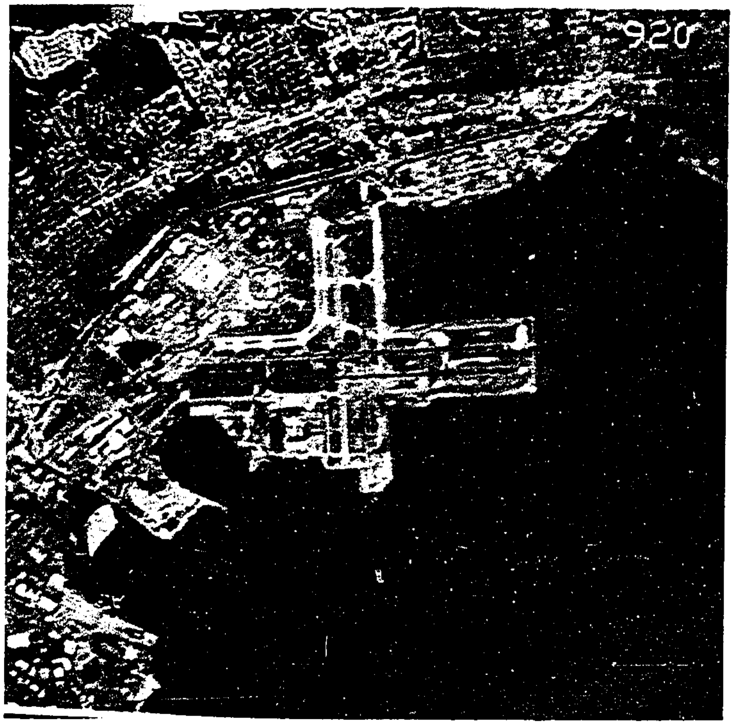

Figure 5.3 Results using parametric mean square error quantizer with bit rate of 1.18 bits/pixel.

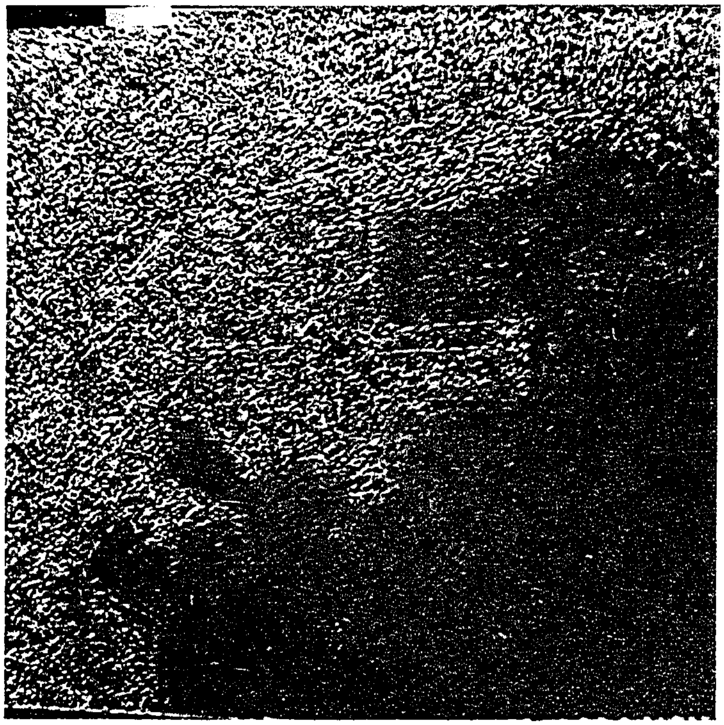

Figure 5.4 The difference picture for DBTC (Figure 5.2). Medium gray corresponds to no coding artifacts. Gray scale expanded by factor of 5.

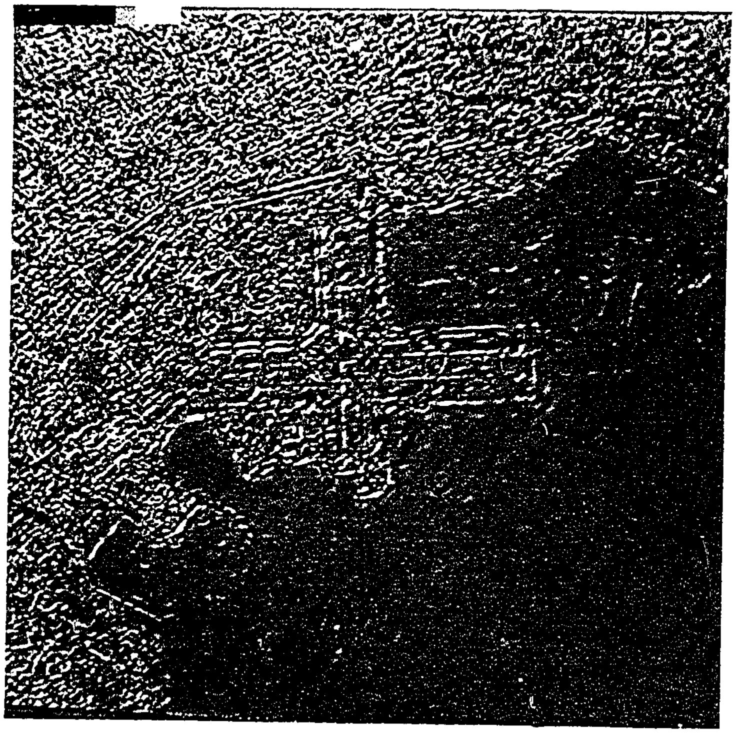

Figure 5.5 The difference picture for parametric mean square error quantizer (Figure 5.3). Medium gray corresponds to no coding artifacts. Gray scale expanded by factor of 5.

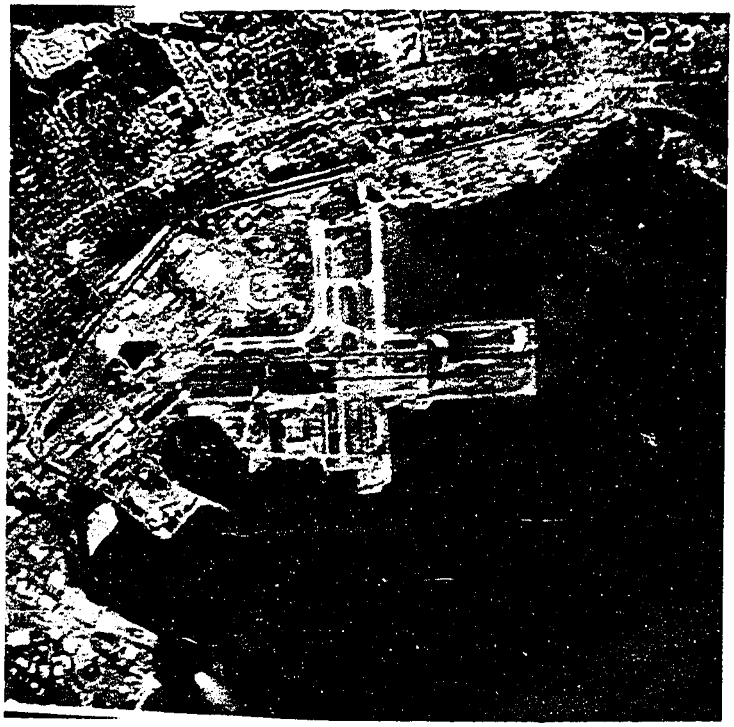

Figure 5.6 Results using DBTC coding with channel (bits) error probability of 10^{-3}. Data rate is 1.18 bits/pixel.

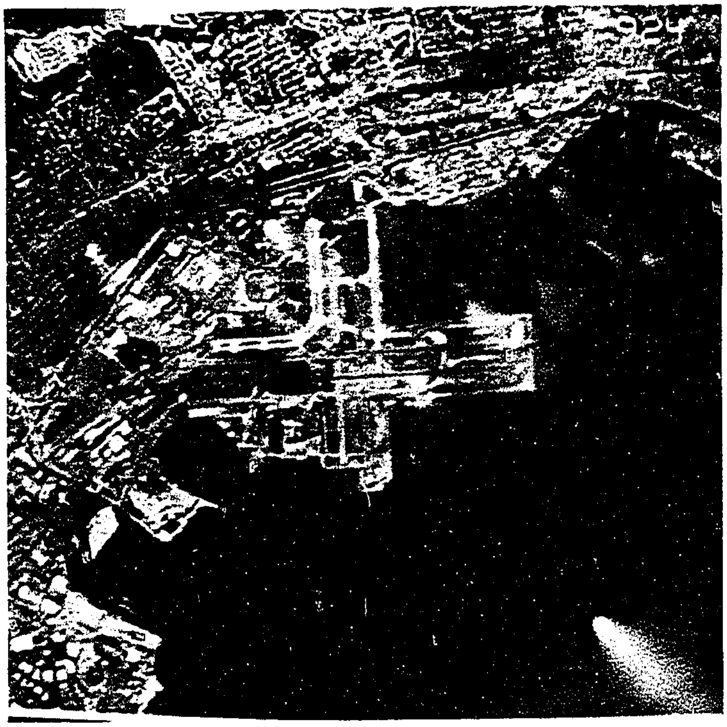

Figure 5.7 Results using parametric mean square error quantizer with channel (bit) error probability of 10^{-3}. Data rate is 1.18 bits/pixel.

Table 5.1 Computed Mean Square Error and Mean Absolute Error measures for the two DPCM coding schemes.

	Mean Square Error	Mean Absolute Error
Figure 5.2	113.01	6.59
Figure 5.3	122.86	6.17

quite apparent. Errors propagate along the 45° axis because the predictor has no "leak" in that direction. In that horizontal or vertical direction the predictor has "leak" therefore error propagation is not as evident.

These results indicate that MP quantizer can be used quite satisfactorily in DPCM and in fact DBTC compliments BTC; i.e., when data rates less than 1.5 bits/pixel are desired it is better to use the differential scheme (DBTC) than to use a straight PCM scheme (BTC). The main reason why it is not possible to use BTC at data rates much (if at all) below 1.6 bits/pixel is the relatively large amounts of overhead information needed by $\overline{m}_1$ and $\overline{\sigma}$ (i.e. 10 bits). If the overhead information was not needed then the minimum data rate is 1 bit/pixel. This is not attainable since the adaptive quantizer needs to know how to change. In DBTC we can take advantage of two changes from BTC. Here it is only necessary to assign 7 bits to $\overline{m}_e$ and $\overline{\sigma}_e$ but the real improvement is the ability to go to a larger block size (8x8). This larger block size minimizes the effect of quantizer overhead and allows for a lower data rate. Unfortunately these results were not evaluated by the professional photo analysts. It is believed that these results would compete very favorably with transform coding. DBTC still retains the relative simplicity of BTC although the prediction process adds some complications to the encoding and decoding. Taking into account these extra operations DBTC would still be far easier than the Chen and Smith Cosine transform method. The effects of the prediction algorithm need to be investigated along with a comparison between DBTC and "true" adaptive DPCM.

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

6.1 Summary of Results

We have presented a new criteria for designing quantizers -- that of preserving moments of the input probability distribution. We have shown that output levels and input thresholds of the quantizer can be found by obtaining the zeroes of the Nth degree orthogonal polynomial associated with the input distribution. One of the advantages of the MP quantizer is that the quantizer can be written in closed form when the input density is symmetric and the number of levels is relatively small. A distinct disadvantage of the MP quantizer is that the output levels tend to be spread further apart whereby certain levels have a relatively low probability of occurring. In many cases obtaining the zeroes of a polynomial represent a much greater computational load that the iterative method of Max [46] for obtaining a minimum mean square error quantizer.

We have demonstrated an interesting application of the MP quantizer for image coding called Block Truncation Coding. This technique competes quite well with transform coding at data rates of 1.63 bits/pixel. At lower rates (1.18 bits/pixel) a differential form of BTC has been described. BTC has relatively robust performance in the presence of noise and the coded images appear to be slightly enhanced. A distinct advantage of BTC is the relatively small computational load of the coding algorithm. A disadvantage is that data rates for BTC or DBTC cannot be obtained below 1.0 bits/pixel. Image modeling has been discussed using a seasonal autoregressive time series for generating synthetic textures and demonstrating that such simple models, that are usually assumed for pictures, in fact do not represent typical images.

6.2 Suggestions for Further Work

The following is a brief listing of possible further research in the MP quantizer - BTC area:

1. The mean square convergence of MP quantizer needs to be further investigated and generalized to the infinite and semi-infinite interval.

2. Source error correction of BTC and DBTC needs to be investigated. A possible approach is that of block and/or boundary matching.

3. Pre and post filtering, either linear or nonlinear, should be explored to minimize the quantization noise effects in the reconstructed images.

4. A benchmark test should be performed comparing the moment preserving, minimum mean square error and mean absolute error quantizers with quantizer levels greater than two. Subjective evaluations of real data should be used.

5. The applicability of BTC and DBTC to coding multi-level graphics, such as bank checks, seems very attractive and should be

explored.

6. The image modeling techniques should be investigated further for better synthetic texture representation with the possibility of going to a higher level stable model.

7. Alternate DPCM predictors should be explored that include more prediction points to help reduce the prediction bais problem.

8. A comparison between DBTC and true adaptive DPCM should be performed.

9. The applicability of coding another types of data, such as speech, should be investigated using MP/BTC methods.

10. The applicability of true two dimensional MP quantization should be examined for image coding.

LIST OF REFERENCES

LIST OF REFERENCES

1. N. Abramson, Information Theory and Coding, New York: McGraw-Hill, 1963.

2. W. C. Adams and C. E. Giesler, "Quantizing Characteristics for Signals Having Laplacian Amplitude Probability Density Function," IEEE Trans. on Communications, Vol. COM-26, No. 8, pp. 1295-1297, August 1978.

3. N. I. Aheizer and M. Krein, Some Questions in the Theory of Moments, Amer. Math. Soc., Translations of Math. Monographs, Vol. 2, 1962.

4. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," IEEE Trans. on Computers, pp. 90-93, Jan. 1974.

5. N. Ahmed and K. R. Rao, Orthogonal Transforms for Digital Signal Processing, New York: Springer-Verlag, 1975.

6. N. I. Akhiezer, The Classical Moment Problem, (translated by N. Kemmer), New York: Hafner, 1961.

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

8. H. C. Andrews, "Two-Dimensional Transforms," in Picture Processing and Digital Filtering, T. S. Huang (editor), New York: Springer-Verlag, 1975.

9. R. Askey, Orthogonal Polynomials and Special Functions, Philadelphia, SIAM, Regional Conference Series, Nov. 21, 1975.

10. M. S. Bartlett, The Statistical Analysis of Spatial Pattern, London: Chapman Hall, 1975.

11. T. Berger, Rate Distortion Theory, Englewood Cliffs: Prentice-Hall, 1971.

12. G. E. P. Box and G. M. Jenkins, Time Series Analysis-Forecasting and Control. San Francisco: Holden-Day, 1970.

13. J. A. Bucklew and N. C. Gallagher, "A Note on Optimum Quantization," to appear IEEE Trans. on Info. Theory.

14. Z. L. Budrikis, "Visual Fidelity Criterion and Modelling," Proc. IEEE, Vol. 60, pp. 771-779, July 1972.

15. S. J. Campanella and G. S.Robinson, "A Comparison of Orthogonal Transformations," IEEE Trans. on Communication Technology, Vol. COM-19, pp. 1045-1050, Dec. 1971.

16. W-H. Chen and C. H. Smith, "Adaptive Coding of Monochrome and Color Images," IEEE Trans. on Communications, Vol. COM-25, pp. 1285-1292, Nov. 1977.

17. D. J. Connor, R. F. W. Pense, and W. G. Scholes, "Television Coding Using Two-Dimensional Spatial Prediction," BSTJ, Vol. 50, pp. 1049-1061, March 1971.

18. P. J. Davis, Interpolation and Approximation, New York: Ginn, 1963.

19. L. D. Davisson, "Rate-Distortion Theory and Application," Proc. IEEE, Vol. 60, pp. 800-808, July 1972.

20. E. J. Delp, "Special Analysis and Synthesis Using Walsh Functions," M.S. Thesis, University of Cincinnati, June 1975.

21. E. J. Delp, R. L. Kashyap, O. R. Mitchell, and R. B. Abhyankar, "Image Modelling with A Seasonal Autoregressive Time Series with Applications to Data Compression," Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing, May 31-June 2, 1978, Chicago, pp. 100-104.

22. E. J. Delp and O. R. Mitchell, "Some Aspects of Moment Preserving Quantizers," accepted for presentation at the IEEE Communications Society's International Conference on Communications (ICC), Boston, June 1979.

23. E. J. Delp and O. R. Mitchell, "Image Compression Using Block Truncation Coding," to appear IEEE Trans. on Communications.

24. E. J. Delp and O. R. Mitchell, "Differential Block Truncation Coding," Purdue University - Purdue Research Foundation, Record and Disclosure of Invention, dated 6 Feb. 1979.

25. W. L. Eversole, D. J. Mayer, F. B. Frazee, and T. F. Cheek, "Investigation of VLSI Technologies for Image Processing," Proceedings: Image Understanding Workshop, Pittsburgh, PA, Nov. 14-15, 1978. Sponsored by the Defense Advanced Research Projects Agency (DARPA), pp. 191-195. (Copies available from the Defense Documentation Center (DDC) under Accession NO. ADA 052903.)

26. W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 2, New York: John Wiley, 1971.

27. L. E. Franks, "A Model for the Random Video Process," *BSTJ*, pp. 609-630, April 1966.

28. W. Frei and C. C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," *IEEE Trans. on Computers*, Vol. C-26, No. 10, pp. 988-998, Oct. 1977.

29. R. G. Gallager, *Information Theory and Reliable Communication*, New York: John Wiley, 1968.

30. R. C. Gonzalez and P. A. Wintz, *Digital Image Processing*, Reading: Addison-Wesley, 1977.

31. A. Habibi, "Two-Dimensional Bayesian Estimate of Images," *Proc. IEEE*, Vol. 60, pp. 878-883, July 1972.

32. A. Habibi, "Survey of Adaptive Image Coding Techniques," *IEEE Trans. on Communications*, Vol. COM-25, No. 11, pp. 1275-1284, Nov. 1977.

33. A. Habibi, "Hybrid Coding of Pictorial Data," *IEEE Trans. Computers*, Vol. COM-22, pp. 614-624, May 1974.

34. J. F. Hayes, A. Habibi, and P. A. Wintz, "Rate Distortion Function for a Gaussian Source Model of Images," *IEEE Trans. on Information Theory*, Vol. IT-16, No. 4, pp. 507-509.

35. T. S. Huang, W. F. Schreiber, and O. J. Tretiak, "Image Processing," *Proc. IEEE*, Vol. 59, No. 11, pp. 1586-1609, Nov. 1971.

36. T. S. Huang, "Easily Implementable Suboptimum Runlength Codes," Conference Record, *1975 IEEE Communications Society's International Conference on Communications*, Vol. I, June 16-18, 1975, pp. 7-8-7-11.

37. T. S. Huang and O. J. Tretiak (editors), *Picture Bandwidth Compression*, New York: Gordon and Breach, 1972.

38. T. S. Huang, "Coding of Two-Tone Images," *IEEE Trans. on Communications*, Vol. COM-25, No. 11, Nov. 1977, pp. 1406-1424.

39. D. Jackson, *Fourier Series and Orthogonal Polynomials*, Amer. Math. Soc., Carus Math. Monographs No. 6, 1941.

40. D. Jameson and L. M. Hurvich (editors), *Handbook of Sensory Physiology: Visual Psychophysics*, New York: Springer-Verlag, 1972.

41. N. S. Jayant, "Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers," Proc. IEEE, Vol. 62, May 1974, pp. 611-632.

42. R. L. Kashyap and A. R. Rao, Dynamic Stochastic Models from Empirical Data, New York: Academic Press, 1976.

43. S. A. Kassam, "Quantization Based on the Mean-Absolute-Error Criterion," IEEE Trans. on Communications, Vol. COM-26, pp. 267-270, Feb. 1978.

44. V. I. Krylov, Approximate Calculation of Integrals, (translated by A. H. Stroud), New York: MacMillian, 1962.

45. J. O. Limb and F. W. Mounts, "Digital Differential Quantizers for Television, BSTJ, Vol. 48, pp. 2583-2599, Sept. 1969.

46. J. Max, "Quantizing for Minimum Distortion," IRE Trans. on Info. Theory, Vol. IT-6, pp. 7-12, March 1960.

47. B. H. McCormick and S. N. Jayarmamurthy, "Time Series Model for Texture Synthesis," International Journal of Computer and Information Sciences, Vol. 3, No. 4, pp. 329-343, 1974.

48. O. R. Mitchell, E. J. Delp, and S. G. Carlton, "Block Truncation Coding: A New Approach to Image Compression," Conference Record, 1978, IEEE Communications Society's International Conference on Communications, Vol. I, June 4-7, 1978, pp. 12B.1.1-12B.1.4.

49. O. R. Mitchell and E. J. Delp, "Image Compression Using Block Truncation," Purdue University - Purdue Research Foundation, Record and Disclosure of Invention, dated 8 April 1977.

50. O. R. Mitchell, S. C. Bass, E. J. Delp, and T. W. Goeddel, "Coding of Aerial Reconnaissance Images for Transmission over Noisy Channels," issued as Rome Air Development Center (RADC) Technical Report, Griffiss Air Force Base, NY. Report No. RADC-TR-78-210. (This report is available from the National Technical Information Service, Springfield, VA 22151, Accession No. ADA 061539.)

51. O. R. Mitchell, S. C. Bass, E. J. Delp, T. W. Goeddel, and A. Tabatabai, "Improvements in Some Image Compression Techniques for Aerial Reconnaissance Analysis," to be issued as Rome Air Development Center (RADC) Technical Report, Griffiss Air Force Base, NY.

52. O. R. Mitchell, S. C. Bass, E. J. Delp, T. W. Goeddel, and T. S. Huang, "Image Coding for Photo Analysis," to appear Proceedings of the Society for Information Display.

53. J. M. Morris and V. D. Vandelinde, "Robust Quantization of Discrete-Time Signals with Independent Samples," IEEE Trans. Communications, Vol. COM-22, pp. 1897-1902, Dec. 1974.

54. J. L. Munnos and D. J. Sakrison, "The Effects of a Visual Fidelity Criterion on the Encoding of Images," *IEEE Trans. on Information Theory*, Vol. IT-20, pp. 525-536, July 1974.

55. H. G. Musmann and D. Preuss, "Comparison of Redunancy Reducing Codes for Facsimile Transmission of Documents," *IEEE Trans. on Communications*, Vol. COM-25, No. 11, Nov. 1977, pp. 1425-1433.

56. N. E. Nahi, "Role of Recursive Estimation in Statistical Image Enhancement," *Proc. IEEE*, Vol. 60, No. 7, pp. 872-877.

57. I. P. Natanson, *Theory of Functions of a Real Variable*, Vol. 1, (translated by L. F. Bora), New York: Ungar, 1961.

58. J. B. O'Neal, "Predictive Quantizing Systems (differential pulse code modulation) for the Transmission of Television Signals," *BSTJ*, Vol. 45, May/June 1966, pp. 689-721.

59. J. B. O'Neal, "Entropy Coding in Speech and Television PCM Systems," *IEEE Trans. on Info. Theory*, Vol. IT-17, pp. 758-761, Nov. 1971.

60. D. P. Panda and A. C. Kak, "Recursive Least Squares Smoothing of Noise in Images," *IEEE Trans. on ASSP*, Vol. ASSP-25, pp. 520-524, Dec. 1977.

61. P. F. Panter and W. Dite, "Quantization Distortion in Pulse-Count Modulation with Nonuniform Spacing of Levels," *Proc. IRE*, Vol. 39, Jan. 1951, pp. 44-48.

62. J. Pearl, H. C. Andres, and W. K. Pratt, "Performance Measures for Transform Data Coding," *IEEE Trans. on Communications*, Vol. COM-20, No. 6, pp. 411-415, June 1972.

63. W. K. Pratt, *Digital Image Processing*, New York: John Wiley, 1978.

64. W. K. Pratt, W-H. Chen, and L. R. Welch, "Slant Transform Image Coding," *IEEE Trans. on Communications*, Vol. COM-22, pp. 1075-1093.

65. W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard Transform Image Coding," *Proc. IEEE*, Vol. 57, pp. 58-68, Jan. 1969.

66. L. R. Rabiner and J. A. Johnson, "Perceptual Evaluation of the Effects of Dither on Low Bit Rate PCM Systems," *BSTJ*, Vol. 51, Sept. 1972, pp. 1487-1494.

67. A. P. Reeves, "A Systematically Designed Binary Array Processor," submitted for publication to IEEE Trans. on Computers.

68. R. P. Roesser, "A Discrete State-Space Model for Linear Image Processing," *IEEE Trans. on Auto. Control*, Vol. AC-20, pp. 1-10, Feb. 1975.

69. A. Rosenfeld and A. C. Kak, Digital Picture Processing, New York: Academic Press, 1976.

70. D. J. Sakrison, "On the Role of the Observer and a Distortion Measure in Image Transmission," IEEE Trans. on Comm., Vol. COM-25, pp. 1251-1267, Nov. 1977.

71. D. J. Sharma and A. N. Netravali, "Design of Quantizers for DPCM Coding of Picture Signals," IEEE Trans. on Communications, Vol. COM-25, No. 11, Nov. 1977, pp. 1267-1274.

72. J. A. Shohat (editor), A Bibliography on Orthogonal Polynomials, National Research Council Bulletin No. 103, 1940.

73. J. A. Shohat and J. D. Tamarkin, The Problems of Moments, Amer. Math. soc., Math. Surveys No. 1, 1943.

74. M. G. Strintzis, "Comments on 'Two-Dimensional Bayesian Estimate of Images'," Proceed. of IEEE, Vol. 64, pp. 1255-1257, Aug. 1976.

75. T. G. Stockham, "Image Processing in the Context of a Visual Model," Proceed. of IEEE, Vol. 60, pp. 828-842.

76. G. Szego, Orthogonal Polynomials, Amer. Math. Soc., Vol. 23, 1975.

77. W. Thoma, "Optimizing the DPCM for Video Signals Using a Model of the Human Visual System," Proc. 1974 Int. Zurich Seminar on Digital Communications, pp. (3(1)-(3-(7), March 12-15, 1974.

78. J. T. Tou, D. B. Kao, and Y. S. Chang, "Pictorial Texture Analysis and Synthesis, Proceedings Third International Joint Conference on Pattern Recognition, Nov. 8-11, 1976, pp. 590-590.

79. P. Whittle, "On Stationary Processes in the Plane," Biometrika, Vol. 41, Part 3 and 4, pp. 434-449, 1954.

80. P. A. Wintz, "Transform Picture Coding," Proc. IEEE, Vol. 60, No. 7, pp. 809-820, July 1972.

81. R. C. Wood, "On Optimum Quantization," IEEE Trans. on Info. Theory, Vol. IT-5, March 1969, pp. 248-252.

82. J. W. Woods and C. H. Rademan, "Kalman Filtering in Two Dimensions," IEEE Trans. on Info. Theory, Vol. IT-23, pp. 473-482, July 1977.

83. J. W. Woods, "Two-Dimensional Discrete Markovian Fields," IEEE Trans. on Info. Theory, Vol. IT-18, pp. 232-240, March 1972.

VITA

VITA

Edward J. Delp III was born in Cincinnati, Ohio on January 1, 1949. He received the B.S.E.E. (cum laude) and M.S. degrees from the University of Cincinnati, Cincinnati, Ohio, in 1973 and 1975, respectively. From 1969-1971 Mr. Delp was employed as a co-op student at the U.S. Naval Ship Research and Development Center in Washington, D.C. During the summer of 1973, Mr. Delp was employed as a research assistant at the National Radio Astronomy Observatory in Greenbank, WV. Mr. Delp was associated with the Laboratory for Information and Signal Processing (LISP) at Purdue University. His research interests include communication and information theory and digital signal processing. Mr. Delp has also consulted for various companies in the area of digital image processing. Mr. Delp is a member of Eta Kappa Nu, Tau Beta Pi, Sigma Xi, and Phi Kappa Phi. He was married in June 1974 to Maureen Bonney of Stratford, Connecticut.