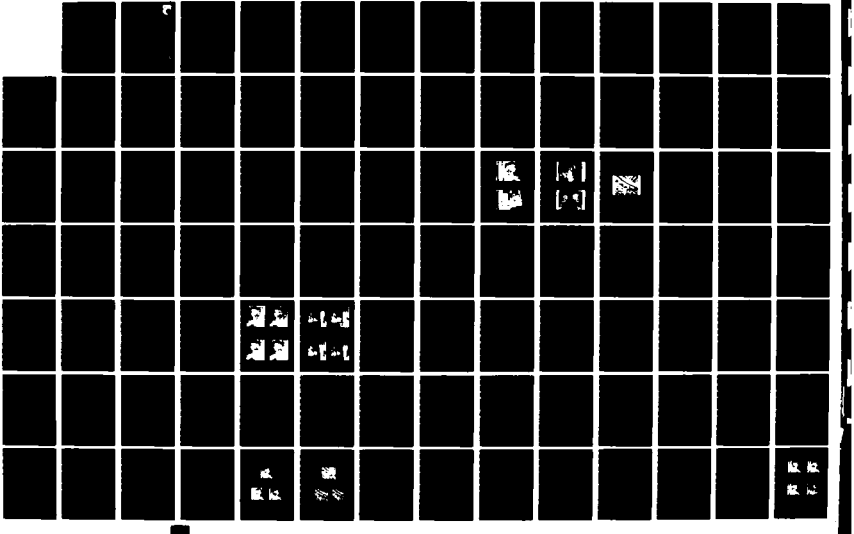
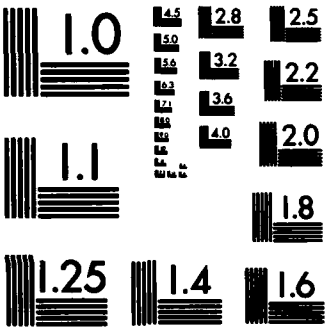
AD-A134 187 BIT WEIGHTING AND SOFT DECISION DEMODULATION FOR IMAGE 1/3
COMPRESSION & ERRO.. (U) TEXAS A AND M UNIV COLLEGE
STATION DEPT OF ELECTRICAL ENGINEE.. J D GIBSON ET AL.
UNCLASSIFIED AUG 83 AFWAL-TR-83-1099 F33615-80-C-1002 F/G 9/2 NL

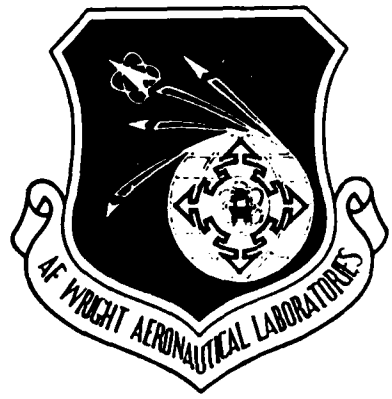




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFWAL-TR-83-1099

12



AD-A134187

BIT WEIGHTING AND SOFT DECISION DEMODULATION FOR IMAGE COMPRESSION & ERROR CONTROL

DRS. J. D. GIBSON AND N. C. GRISWOLD
TEXAS A&M UNIVERSITY
DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE STATION, TEXAS 77843

AUGUST 1983

FINAL REPORT MAY 1979 - DECEMBER 1982

Approved for public release; distribution unlimited.

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

GEORGE A. NEYHOUSE
Electronic Engineer
Information Transmission Branch
Avionics Laboratory

CHARLES C. GAUDER
Chief, Information Transmission Branch
Avionics Laboratory

FOR THE COMMANDER

FRANK A. SCARPINO, Acting Chief
System Avionics Division
Avionics Laboratory

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/AAAI-2. *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFWAL-TR-83-1099	2. GOVT ACCESSION NO. A134187 / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): BIT WEIGHTING AND SOFT DECISION DEMODULATION FOR IMAGE COMPRESSION & ERROR CONTROL	5. TYPE OF REPORT & PERIOD COVERED: FINAL REPORT MAY 1979 TO DEC. 1982
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Dr. J. D. Gibson Dr. N. C. Griswold	8. CONTRACT OR GRANT NUMBER(s): F33615-80-C-1002
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Texas A&M University Dept. of Electrical Engineering College Station, TX 77843	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Avionics Laboratory Air Force Wright Aeronautical Laboratories Air Force Systems Command Wright Patterson AFB, Ohio 45433	12. REPORT DATE: August 1983 13. NUMBER OF PAGES: 208
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified 15a. DECLASSIFICATION/ DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

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

N/A

18. SUPPLEMENTARY NOTES

Computer programs contained herein are theoretical and/or references that in no way reflect Air Force-owned computer software.

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

Image compression, image coding, joint source/channel coding, bit weighting, soft decision demodulation

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

This report documents research concerning image compression at 1 bit/pel over noisy channels. The techniques of bit weighting, soft decision demodulation, and channel coding are compared for reducing or eliminating the effects of channel errors on compressed imagery. The source coding method used is the two-dimensional discrete cosine transform of 16 by 16 blocks. Results indicate that for an independent bit error rate of 10^{-2}, bit weighting alone and soft decision alone provide a noticeable improvement in reconstructed image quality.

(over)

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

S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

O.C.)

Combined bit weighting/soft decision outperforms either technique alone. Judicious channel coding can substantially improve image quality over noisy channels with only a slight degradation in ideal channel performance.

PREFACE

There are numerous operational requirements ~~in the Air Force~~ for remotely sensed imagery. The vast amounts of image data acquired in these applications pose a particular problem for transmission to other locations. In order to achieve real-time operation and to conserve communications channel bandwidth, it is common to compress the image data. Suitable image compression techniques reduce the required transmitted data rate by a factor of approximately 8-to-1, while still maintaining acceptable image quality. Unfortunately, when an image is compressed, it is much more susceptible to bit errors introduced by noisy channels. Since noisy channel applications cannot be avoided, the investigation of image compression system performance over noisy channels is of prime concern.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONTINUED)

LIST OF ILLUSTRATIONS (CONTINUED)

LIST OF ILLUSTRATIONS (CONTINUED)

SECTION I

SUMMARY

The Air Force has operational requirements for the transmission of compressed imagery over noisy channels. Most image data compression studies assume that the channel is ideal, and hence, error-free. Unfortunately, systems designed in this manner are extremely vulnerable to errors in the transmitted data. One method for providing protection against channel errors is to employ forward error correcting (FEC) codes. However, in order to use channel coding at a fixed transmitted data rate, some bits must be allocated to forward error correction, thus reducing the bit rate available for source coding. Therefore, it is desirable to devise methods that provide a reduction in channel errors, or their effects, without reducing the number of bits available for source coding.

A method which satisfies this criterion is called bit weighting (BW). In bit weighting, available signal energy is allocated to transmitted bits according to the relative importance of each bit to the reconstructed image. Bit weighting does not increase the complexity of the receiver and does not reduce the number of bits available for source coding. A second technique, called soft decision demodulation (SDD), tries to reduce the effects of bit errors by identifying errors in significant bits and replacing the affected word with an estimated word which has a smaller reconstruction error. Soft decision demodulation does not impact the transmitter design and does not reduce the number of bits used for source coding. It is noted that soft decision and bit weighting can be combined to improve performance further.

This study investigated the utility of bit weighting alone, soft decision demodulation alone, and combined bit weighting and soft decision demodulation. Additionally, some studies were performed to evaluate the usefulness of forward error correction. The image compression scheme chosen for all of this work is the two-dimensional discrete cosine transform (2D-DCT) over 16 by 16 blocks at

1 bit/pel.

The following paragraphs summarize the efforts under the four tasks.

Task I - Derivation of Information Distribution for Coded Picture Elements

Quantities called A-factors represent the average reconstruction error power in the image contributed by a bit error in a pulse code modulation (PCM) symbol. The A-factors were calculated for Gaussian and Laplacian probability density functions (pdfs), the natural binary code (NBC), the folded binary code (FBC), and the minimum distance code (MDC), and 1-8 bit words. Using these A-factors, optimum thresholds for soft decision demodulation were calculated. Bit repeating with majority logic decoding was found to be an ineffective technique for channel correction. More details on A-factors are presented in Section II, and the results of Task I are employed throughout the remaining sections of the report.

Task II - Modulation and Decoding Synthesis

For bit weighting, the A-factors were used to derive an energy weighted PCM/phase shift keying (PSK) scheme which minimizes the number of bit errors in the most significant bits while maintaining a constant average energy per block. A dynamic programming algorithm was developed to optimize the energy allocation among bits. See Section VI.

For soft decision demodulation it was determined that when a bit is declared unreliable, it is more effective to "smooth" based on the transform coefficients than on the reconstructed image. Both techniques were used, however. When evaluating SDD performance, objective measures such as normalized mean squared error and peak-to-peak signal-to-noise ratio were found to be useful. The final performance criterion used was the subjective quality of reconstructed images. See Section IV.

Task III - Simulation of Error Control Methods of Compressed Images Over Non-Ideal Channels

Numerous Monte Carlo simulation runs were performed to evaluate bit weighting

alone, soft decision alone, and combined soft decision and bit weighting. Also investigated was a forward error correction scheme which used a (7,4) Hamming code to protect the most significant bits in a block (see Section V).

Bit weighting alone and soft decision alone each improve system performance, with combined BW/SDD providing an additional 1.5dB and a clearly improved reconstructed image. The channel coding scheme yields surprisingly good performance over noisy channels with only slight degradation in the ideal channel case when compared with allocating all bits to source coding.

Soft decision demodulation is trivial to implement since only six bits out of 256 per block need to be monitored, and the estimate for an unreliable word is based on coefficients from adjacent blocks. The concept of bit weighting weights the relative signal energy for each PCM symbol in a word by its sensitivity to digital transmission errors. This method applied at the transmitter does effect the transmitter design. To reduce transmitter switching rates, near optimum performance may be achieved by transmitting groups of PCM symbols at the same energy level where the number of groups is less than the number of bits in a PCM word. The digital noise is reduced by allowing more energy to be used for the most significant bits of a PCM word resulting in a smaller bit error probability. This is accomplished at the expense of less energy on the least significant bits. The results indicate even though more total errors are made they are not made on bits which are significant and performance is improved.

Task IV - Spatial Image Coding

The goal of Task IV was to develop an image transmission system simulation based on a spatial image coder which would provide good quality images, low bandwidth requirements and error protection for non-ideal channels. This suggested design utilizes a technique called Block Truncation Coding (BTC) in combination with bit weighting and Quadrature Amplitude Modulation (QAM).

An advantage of Block Truncation Coding is the ease with which it may be matched to (QAM) with bit weighting. This technique matches the probability of transmission errors for a given bit to the relative importance of that bit within a digital word. This task then has developed a modified version of BTC and describes the feasibility of matching this source coder to QAM, with bit weighting. The combined simulation was conducted for binary symmetric channels with Gaussian noise.

SECTION II

A-FACTORS

1.0. INTRODUCTION

The quantities called "A-factors" by Rydbeck and Sundberg [1] represent the average error energy in the reconstructed signal caused by errors in the different bits of a transmitted codeword. The A-factors are useful for both bit-weighting and soft decision since they indicate which bit errors produce the largest reconstruction error.

2.0. A-FACTOR CALCULATION

A block diagram illustrating the particular application of interest for this work is shown in Fig. 2.1. As illustrated by this figure, the transform coefficients are quantized, coded, and transmitted over a noisy channel, and then resynthesized at the receiver. The total error power in the representation of a particular coefficient is given by

$$\varepsilon^2 \triangleq E\{[c - \hat{c}_{i,\ell}]^2\} \tag{2.1}$$

where the expectation is taken over both the source and channel statistics. It can be shown that ε^2 can be written as the sum of quantization noise ε_q^2, clipping noise ε_c^2, and noise due to channel errors ε_a^2, so that

$$\varepsilon^2 = \varepsilon_q^2 + \varepsilon_c^2 + \varepsilon_a^2 \tag{2.2}$$

Focusing on the coefficient error power due to channel errors, the third term in Eq. (2.2) can be written as

$$\begin{aligned} \varepsilon_a^2 &= E_{i,\ell}\{[c - \hat{c}_{i,\ell}]^2\} \\ &= \sum_{\ell} P_\ell \cdot E_i\{[c - \hat{c}_{i,\ell}]^2\} \end{aligned} \tag{2.3}$$

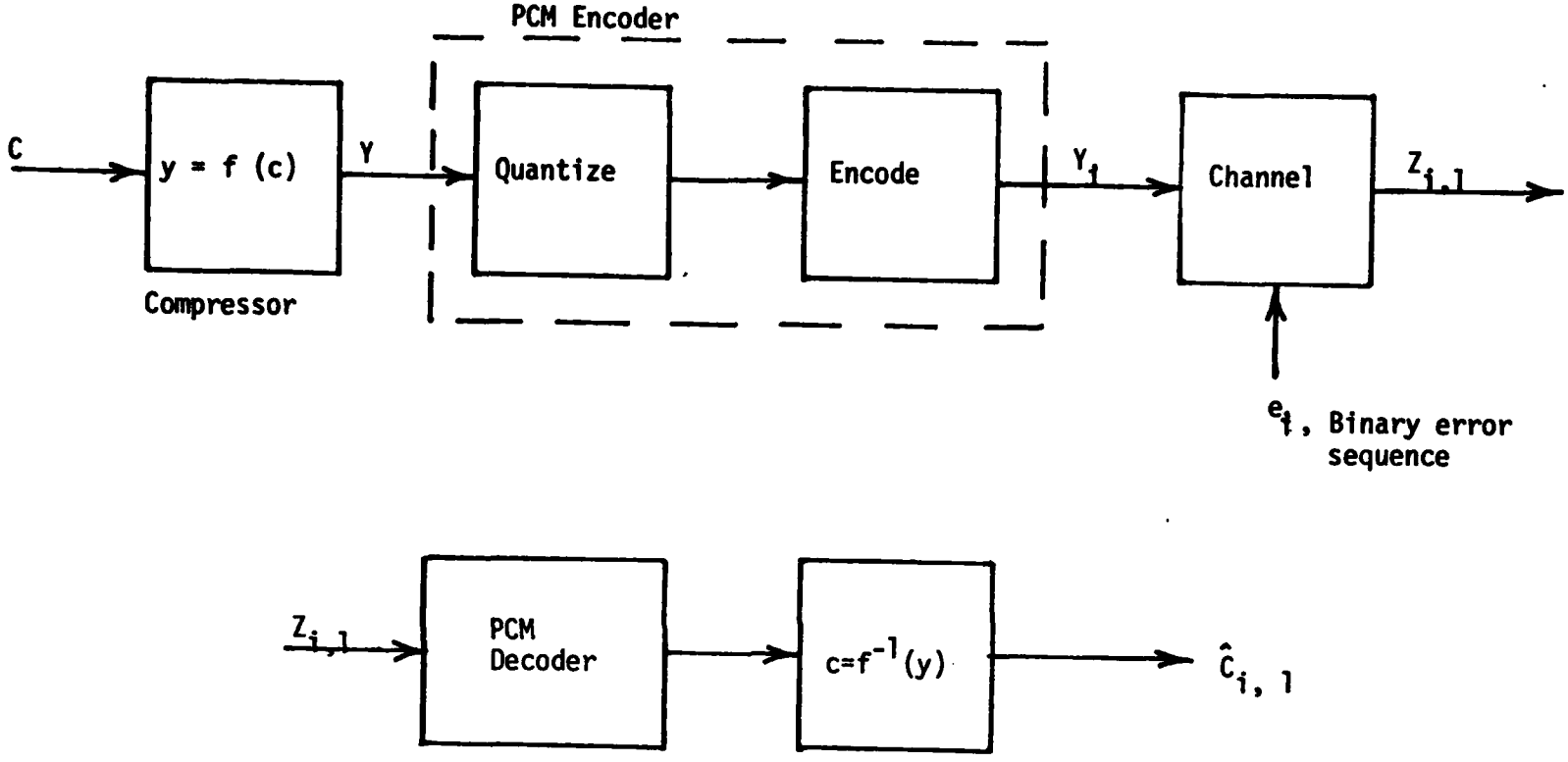


Figure 2.1 Quantization, Coding, and Transmission of Transform Coefficients Over a Noisy Channel

In the above, the subscript "i" denotes the ith quantization level and the subscript "ℓ" indicates the ℓth error sequence, shown as e_ℓ in Fig. 2.1. The quantity P_ℓ is the probability of occurrence of the ℓth error sequence as calculated from the physical channel model. The quantities defined as "A-factors" are the expectations in Eq. (2.3), viz

$$A_\ell \triangleq E_i\{[c - \hat{c}_{i,\ell}]^2\}\ \sigma_s^2 \tag{2.4}$$

Thus, the number A_ℓ denotes the average coefficient error power caused by the ℓth channel error sequence e_ℓ.

For clarification purposes, consider the case on N-bit linear PCM so that the compressor and expandor functions in Fig. 2.1 are straight lines with unity slope. Then there are 2^N-1 possible error sequences, and hence, there are 2^N-1 A-factors, one for each error sequence. For ease of notation, the first N A-factors correspond to the single bit error sequences, and $e_1 = 1\ \underbrace{0\ \ldots\ 0}_{N-1}$ is the error sequence that causes a sign error. Naturally then, $e_2 = 0\ 1\ \underbrace{0\ \ldots\ 0}_{N-2}$ and so on.

Substituting Eq. (2.4) into (2.3) yields

$$\varepsilon_a^2 = \sum_\ell P_\ell A_\ell \tag{2.5}$$

Since we are using coherent binary PSK modulation over an additive white Gaussian noise channel, the channel errors are independent, and therefore, all m-bit error sequences, regardless of their location, are equally probable. As a result, Eq. (2.5) becomes

$$\varepsilon_a^2 = \sum_{m=1}^{N} p^m(1-p)^{N-m}\ \sigma_s^2 \underbrace{\sum_{\ell=m} A_\ell}_{\text{bit A-factor subscripts}} \tag{2.6}$$

Now for a channel probability of error of 10^{-2} or less, Eq. (2.6) can be approximated by

$$\varepsilon_a^2 \cong P\sigma_s^2 \sum_{\ell=1}^{N} A_\ell \tag{2.7}$$

where multiple bit errors have been neglected due to their low probability of occurrence.

The A-factors depend on the input signal pdf, the compressor/expandor functions (quantization), the PCM code assignment, the number of quantization levels, and the channel. Fortunately, for the present application, many of these quantities are known or can be determined. First, for the discrete cosine transform (DCT), the transform coefficients, except for the dc component, can be assumed to be Gaussian, and the dc component has a uniform pdf. Thus, quantizers can be chosen to be matched in the mean squared error sense to a uniform pdf or a Gaussian pdf. Since only single bit errors need be considered, all A-factors for $N \leq 8$ can be computed and tabulated. Next, the channel model can be limited to the binary symmetric channel (BSC), where the probability of bit error is calculated from the physical channel model and the modulation method described previously.

The only parameter not yet chosen is the code assignment. If only fixed-length to fixed-length codes are considered, that is, no entropy coding, then three possible choices are the natural binary code (NBC), the folded binary code (FBC), and the minimum distance code (MDC). The FBC is preferred for the present applications. The single bit A-factors for 1 through 8 bit codewords and the FBC are shown in Tables 2.1 and 2.2 for Gaussian and Laplacian input probability density functions, respectively.

3.0. COMPARISONS OF APPROXIMATIONS

In [2], Sundberg discusses the effect of channel errors in PCM encoded signals and how to improve performance using soft decision demodulation techniques. In his paper, he makes approximations both in choosing the soft decision thresholds and in calculating the overall system performance. It is instructive to review his derivation and approximations, and compare his approximation of system performance to a more exact approximation.

TABLE 2.1

NORMALIZED A-FACTORS FOR THE FBC AND A

GAUSSIAN INPUT

No. of Bits	A-Factors Bit Number 1	2	3	4	5	6	7	8
1	0.25472136E+01							
2	0.35320046E+01	0.11176707E+01						
3	0.38580127E+01	0.15055707E+01	0.37629205E+00					
4	0.39613733E+01	0.18276922E+01	0.45252314E+00	0.10945314E+00				
5	0.39969101E+01	0.20721655E+01	0.50955671E+00	0.12157577E+00	0.29679295E-01			
6	0.40010953E+01	0.27720368E+01	0.69333321E+00	0.17325824E+00	0.43330133E-01	0.10825986E-01		
7	0.40068913E+01	0.33068457E+01	0.82665068E+00	0.20666206E+00	0.51677559E-01	0.12915822E-01	0.32273047E-02	
8	0.40094380E+01	0.38598533E+01	0.96497458E+00	0.24129364E+00	0.60307093E-01	0.15075549E-01	0.37740117E-02	0.94359129E-03

TABLE 2.2

NORMALIZED A-FACTORS FOR THE FBC AND A

LAPLACIAN INPUT

A-Factors

No.of Bits	Bit Number 1	2	3	4	5	6	7	8
1	0.19999555E+01							
2	0.32943518E+01	0.20005214E+01						
3	0.37821250E+01	0.30801225E+01	0.63827842E+00					
4	0.39373693E+01	0.41609163E+01	0.81863296E+00	0.18227927E+00				
5	0.39835272E+01	0.50928936E+01	0.97010863E+00	0.20905438E+00	0.48978470E-01			
6	0.39872208E+01	0.70118918E+01	0.17529594E+01	0.43825439E+00	0.10956930E+00	0.27356079E-01		
7	0.39956300E+01	0.94065208E+01	0.23513219E+01	0.58791393E+00	0.14702968E+00	0.36731541E-01	0.91894809E-02	
8	0.39980862E+01	0.12156384E+02	0.30388882E+01	0.75981718E+00	0.18995896E+00	0.47488842E-01	0.11868138E-01	0.29610712E-02

For an N bit quantizer, there are 2^N-1 different sequences which correspond to possible channel errors. For each of these sequences there is an associated A-factor which is the mean square error resulting from this sequence (averaged over quantizer output statistics). The mean square error due to channel noise is then

$$\epsilon_a^2 = \sum_{i=1}^{2^N-1} A_i P_i \tag{2.8}$$

where P_i is the probability that the error sequence "i" occurs. In [2], this term is approximated by

$$\epsilon_a^2 \approx \sum_{j=1}^{N} A'_j \; P_j \tag{2.9}$$

where A'_j is the A-factor for a single bit error in bit "j", and P is the probability of a bit error (BSC, independent bit errors). Equation (2.9) is exact if the A-factor for a multiple bit error sequence is the sum of the corresponding single bit A-factors. This occurs only for a linear, natural binary quantizer, with uniform input signal density.

As in Sundberg's case, we will consider transmitting the N quantizer bits independently through an additive white Gaussian noise channel using binary antipodal signalling. The M most significant bits of the codeword are monitored, and if they fall within an erasure zone, the entire codeword is rejected and replaced with an estimate. The following notation will be used:

E	Signal energy
No/2	Double sided noise spectral density
N	Number of bits in codeword
M	Number of bits monitored for soft decision
T_i	Normalized threshold of erasure zone i
A_j	A-factor for noise sequence j (2^N-1)

A_i'	A-factor for single bit error (N)
δS^2	Mean square reconstruction error
P	Bit error probability (hard decision)
Pu_i	Probability of undetected error in bit i
Pz_i	Probability that bit i received in erasure zone
Pr	Probability that a codeword is rejected

Straightforward calculation shows the probability terms are

$$P = Q(\sqrt{2E/No}) \tag{2.10}$$

$$Pu_i = Q(\sqrt{2E/No}(T_i+1)) \tag{2.11}$$

$$Pz_i = Q(\sqrt{2E/No}(T_i+1)) - Q(\sqrt{2E/No}(1-T_i)) \tag{2.12}$$

The mean square error due to channel noise for a soft decision demodulation system is

$$\varepsilon_{a_1}^2 = \sum_{j=1}^{2^N-1} A_j P_j + \delta S^2 Pr \tag{2.13}$$

where P_j is the probability that error sequency j occurs and that it is undetected at the receiver. The probability, Pr, is the probability that at least one of the monitored bits of the received codeword falls in the erasure zone. In this exact form, calculation of the mean square error requires evaluating 2^N-1 A-factors and probabilities.

The first simplification in calculating the channel noise requires use of Equation (2.9), which uses only single bit A-factors and assumes the summability of A-factors. The approximation to the noise is

$$\varepsilon_{a_2}^2 = \sum_{i=1}^{M} A_i' Ps_i + \sum_{i=M+1}^{N} A_i Ph + \delta S^2 Pr \tag{2.14}$$

where

Ps = PR {undetected error occurs in bit i, and none of the other M-1 MSB are received as unreliable}

Ph = PR {bit i received in error, and none of the other M-1 MSB are received as unreliable}

The expressions for these probabilities are

$$Ps_i = Pu_i \cdot \prod_{j=1,\ j\neq i}^{M} (1 - Pz_j) \tag{2.15}$$

$$Ph = P \cdot \prod_{j=1}^{M} (1 - Pz_j) \tag{2.16}$$

$$Pr = 1 - \prod_{j=1}^{M} (1 - Pz_j) \tag{2.17}$$

Further simplification of the channel noise can be made by assuming that the terms $(1 - Pz_j)$ are very close to 1 and thus neglected in Equations (2.15) and (2.16). Also the term Pr can be approximated by

$$Pr' = 1 - \sum_{j=1}^{M} Pz_j \tag{2.18}$$

These approximations lead to Sundberg's form of the channel noise

$$\varepsilon_{a_3}^2 = \sum_{i=1}^{M} A_i' \, Pu_i + \sum_{i=M+1}^{N} A_i \, P + \delta S^2 Pr' \tag{2.19}$$

Using (2.19), a closed form solution to the optimum thresholds can be derived and are

$$T_i = \frac{1}{2(2E/No)} \log_e\left(\frac{A_i}{\delta S^2} - 1\right) \tag{2.20}$$

The choice of M, the number of significant bits to monitor, is chosen as the largest number such that $A_M - \delta S^2 > 1$, in which case T_i is always well defined. The optimum threshold for (2.13) and (2.14) cannot be calculated in closed form and require considerable effort in using numerical techniques. For this reason,

we will assume that the thresholds of (2.20) are approximately optimum for (2.13) and (2.14) also.

Example

For the purpose of comparing the three channel noise expressions (2.13), (2.14), (2.19), consider the following example. Assume that the input signal is distributed as N(0,1) and the quantizer is an 8-bit Max quantizer [1]. The total mean square error for the system is

$$\varepsilon^2 = \varepsilon_a^2 + \varepsilon_q^2 + \varepsilon_T^2 \tag{2.21}$$

where ε_q^2 and ε_T^2 are the noise terms introduced by the quantizer. The system performance is described by comparing the output signal-to-noise ratio, SNRO, to the channel signal-to-noise ratio, SNRCH, which are defined by

$$\text{SNRO} = 10 \log_{10} (1/\varepsilon^2) \tag{2.22}$$

$$\text{SNRCH} = 10 \log_{10} (2E/No) \tag{2.23}$$

In Figures 2.2 through 2.5, the system performance is plotted for the different design parameters, δS^2 and 2E/No. The first two figures, with $\delta S^2 = .1$, represent the case when a good estimate of the output is available. Figures 2.4 and 2.5, with $\delta S^2 = .5$, represents the case where only a poor estimate of the output is available. The other design value, 2E/No, represents the signal-to-noise ratio of the channel on which the system is intended to be used. The channel signal-to-noise ratios of 7 and 10 dB, correspond to bit error rates of approximately 10^{-2} and 10^{-3} respectively. Notice that in each of the graphs, the output SNR ratio converges to about 33 dB as the channel SNR ratio becomes large. This value represents the error due to the quantizer alone, with the error being introduced by the channel being negligible. In every case, the exact form of the channel noise gives higher output SNR than the two approximations. The two approximations

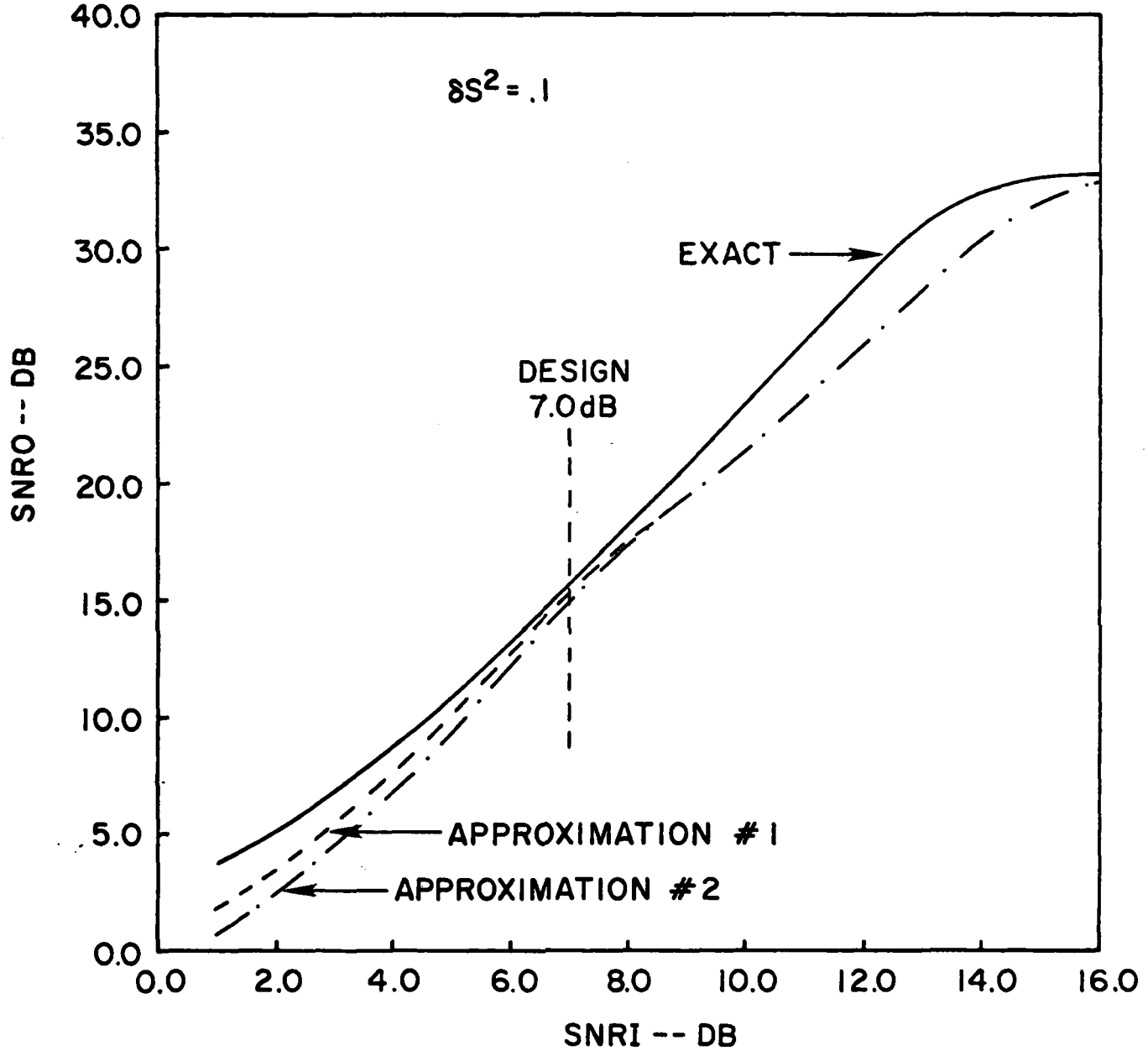


Figure 2.2. Comparisons of Channel Noise Approximations (Design SNRI = 7.0 dB, $\delta S^2 = .1$)

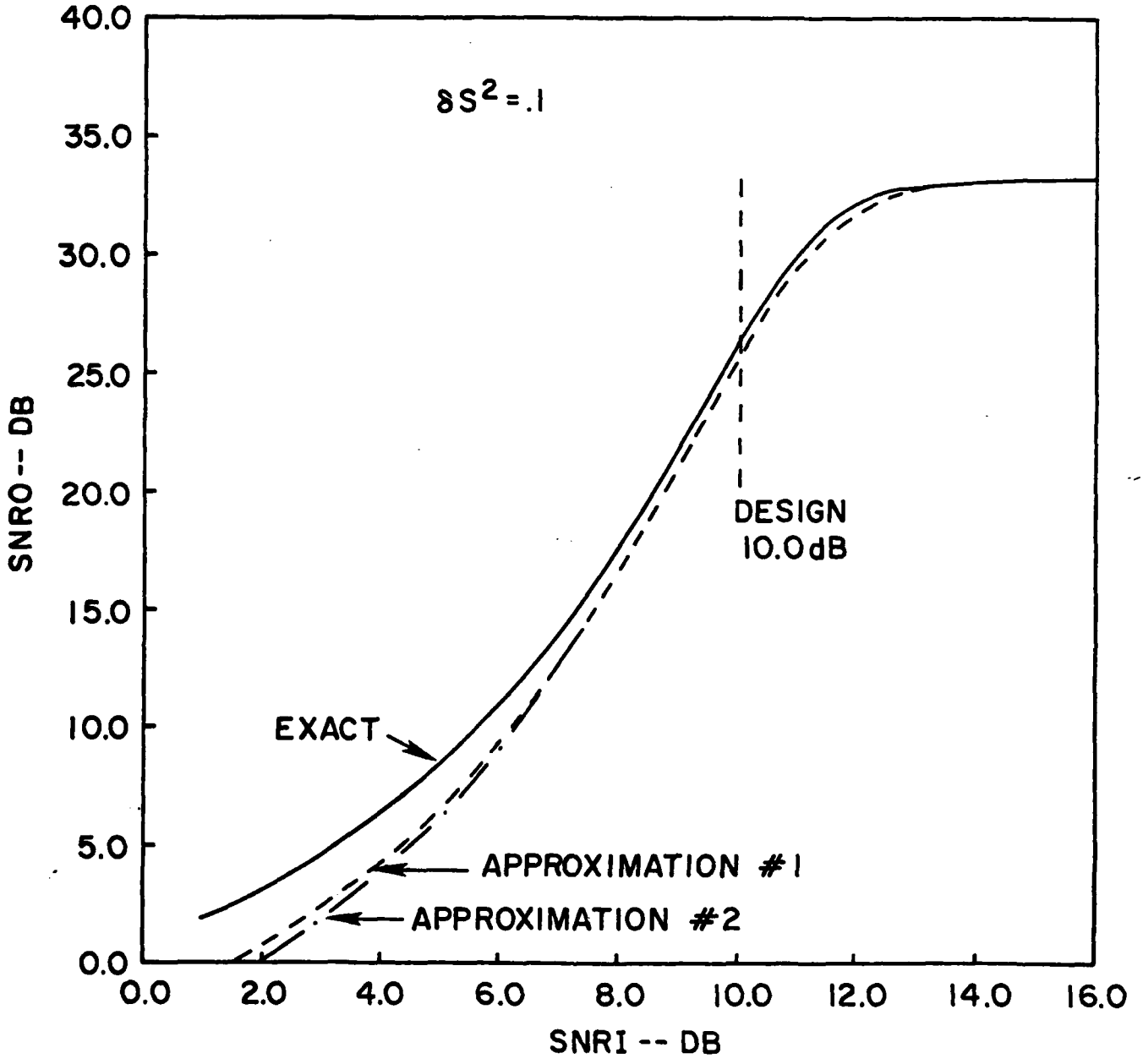


Figure 2.3. Comparisons of Channel Noise Approximations (Design SNRI = 10.0 dB, $\delta S^2 = .1$)

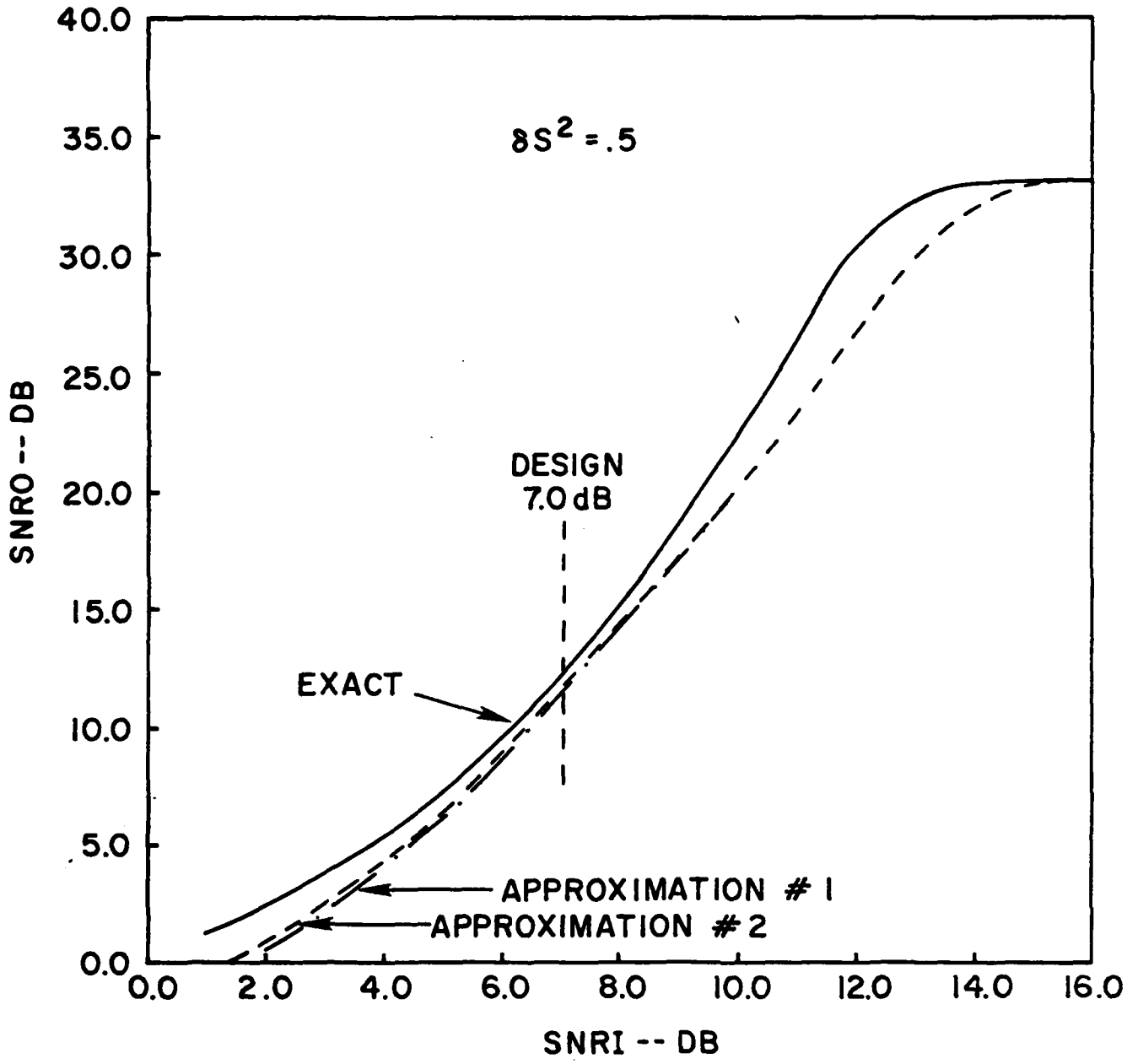


Figure 2.4. Comparisons of Channel Noise Approximations (Design SNRI = 7.0 dB, $\delta S^2 = .5$)

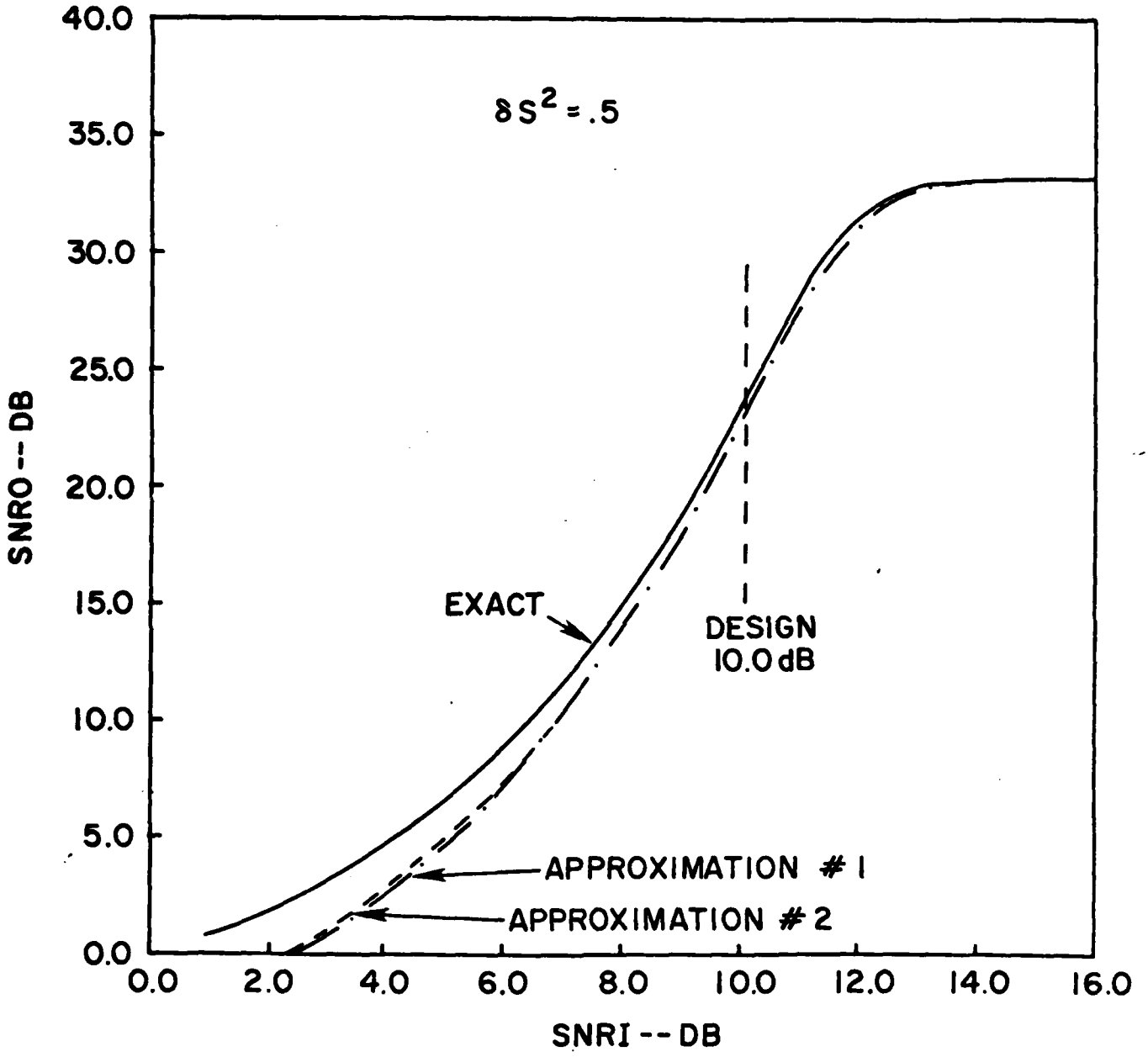


Figure 2.5. Comparisons of Channel Noise Approximations (Design SNRI = 7.0 dB, δS^2 = .5)

tend to converge at high channel SNR and split at low channel SNR, with the difference more pronounced for the good estimator ($\delta S^2 = .1$). Also the approximate representations of the channel noise tend to be closest to the exact representation at and around the design value for the channel SNR.

4.0. CONCLUSIONS

Sundberg's approximation to the channel noise given by Equation (2.19) is not a close approximation to the actual channel noise at all channel SNR's, but is much simpler and faster to calculate than the exact form, and results in a useful lower bound to the output SNR of the system. The approximation given by Equation (2.14) is more difficult to calculate than Sundberg's approximation and does not show much improvement.

SECTION III

DISTRIBUTIONS OF THE TWO-DIMENSIONAL DCT COEFFICIENTS FOR IMAGES

1.0. INTRODUCTION

In image coding systems which use a two-dimensional Discrete Cosine Transform (DCT) [1], there have been several different assumptions on the distributions of the transform coefficients. Pratt [2] conjectured that the DC coefficient should have a Rayleigh distribution since it was the sum of positive values, and, that based on the Central Limit Theorem, the other coefficients should be Gaussian. Netravali and Limb [5] agreed with the above assumption and also stated that the histograms of the non-DC coefficients were roughly bell shaped. On the other hand, Tescher [3] indicated that the non-DC coefficients were not Gaussian, but Laplacian, and most recently, Murakami, et. al. [4] assumed that the DC coefficient was Gaussian and that the non-DC coefficients were Laplacian. These different assumptions have led the authors to perform goodness-of-fit tests on the transform coefficients in order to identify the distribution that best approximates the statistics of the coefficients. In the tests, the Gaussian, Laplacian, Gamma, and Rayleigh distributions were considered.

This section shows that for many images the DC coefficient is best approximated by a Gaussian distribution and non-DC coefficients are best approximated by Laplacian distributions, and that by using Laplacian quantizers for the non-DC transform coefficients, the quality of the reconstructed image can be improved as compared to Gaussian quantizers. This section is organized as follows. Sub-section 2.0 describes the goodness-of-fit test and how it was used with the transform coefficients, and Sub-section 3.0 describes the results of the tests. In Sub-section 4.0, comparisons between the theoretical and actual quantization error for a two-dimensional DCT system are made for different assumptions on the distribution of the coefficients.

2.0. GOODNESS-OF-FIT TEST

A well known test for goodness-of-fit of distributions is the Kolmogorov-Smirnov (KS) test [8,9]. For a given set of data $X = (x_1, x_2, \ldots, x_M)$, the KS test compares the sample distribution function $F_X(\cdot)$ to a given distribution function $F(\cdot)$. The sample distribution function is defined by

$$F_X(z) = \begin{cases} 0, & z < x_{(1)} \\ \frac{n}{M}, & x_{(n)} \leq z < x_{(n+1)}, \; n = 1, 2, \ldots, M-1, \\ 1, & z \geq x_{(M)} \end{cases} \tag{3.1}$$

where $x_{(n)}$, $n = 1, \ldots, M$ are the order statistics of the data X. The KS test statistic, t, is then defined by

$$t = \max_{i=1, 2, \ldots, M} |F_X(x_i) - F(x_i)|. \tag{3.2}$$

The KS test statistic is a distance measure between the sample distribution function and the given distribution function, with the distance defined by the maximum difference between $F_X(\cdot)$ and $F(\cdot)$ evaluated at the sample points x_i. When testing the data against several distributions, the distribution that yields the smallest KS statistic is the best fit for the data.

The KS test was used to test the distributions of the DCT coefficients with block sizes 8, 16 and 32 computed for the five images (Girl, Couple, Moon, X-Ray, and Aerial) shown in Figure 3.1. These images have size 256 x 256 pels, with the gray levels PCM encoded at 8 bits/pel. For each image and block size, the KS goodness-of-fit test was performed on the ten high energy coefficients in the upper left hand corner of the transform block c_{00}, c_{01}, c_{02}, c_{03}, c_{10}, c_{11}, c_{12}, c_{20}, c_{21}, and c_{30}. The data for a given coefficient, 'ij', consisted of the points $c_{ij}(k)$, $k = 1, \ldots, M$, where the index k represents the position of the block in the image, and the number of blocks, M, in a 256 x 256 image is related to the block

(a)

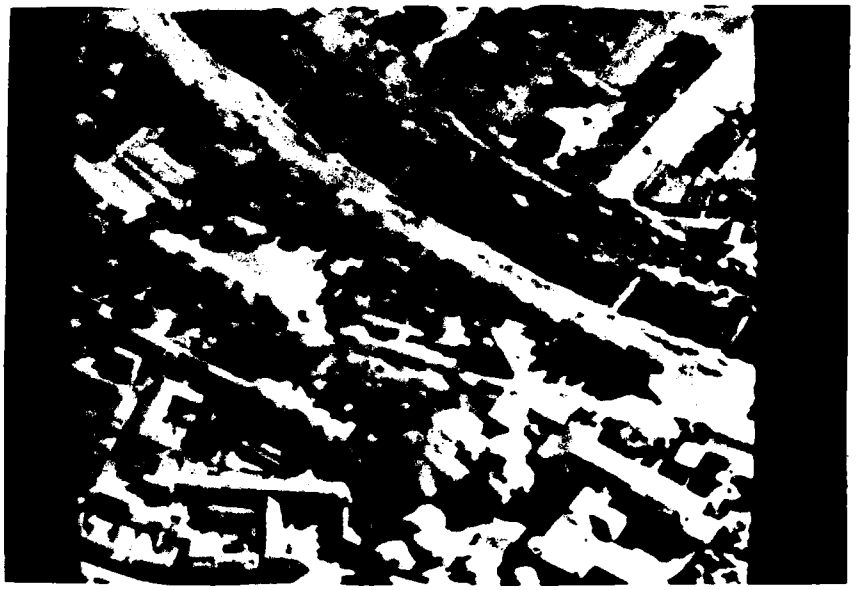
(b)

Figure 3.1. Test Images. (a) Girl, (b) Couple, (c) Moon, (d) X-Ray, (e) Aerial

(c)

(d)

Figure 3.1. (cont.)



(e)

Figure 3.1. (cont.)

size N by $M = (256/N)^2$. For these data points the sample mean and variance $\overline{c_{ij}}$ and S_{ij}^2 were calculated according to

$$\overline{c_{ij}} = \frac{1}{M} \sum_{k=1}^{M} c_{ij}(k) \tag{3.3}$$

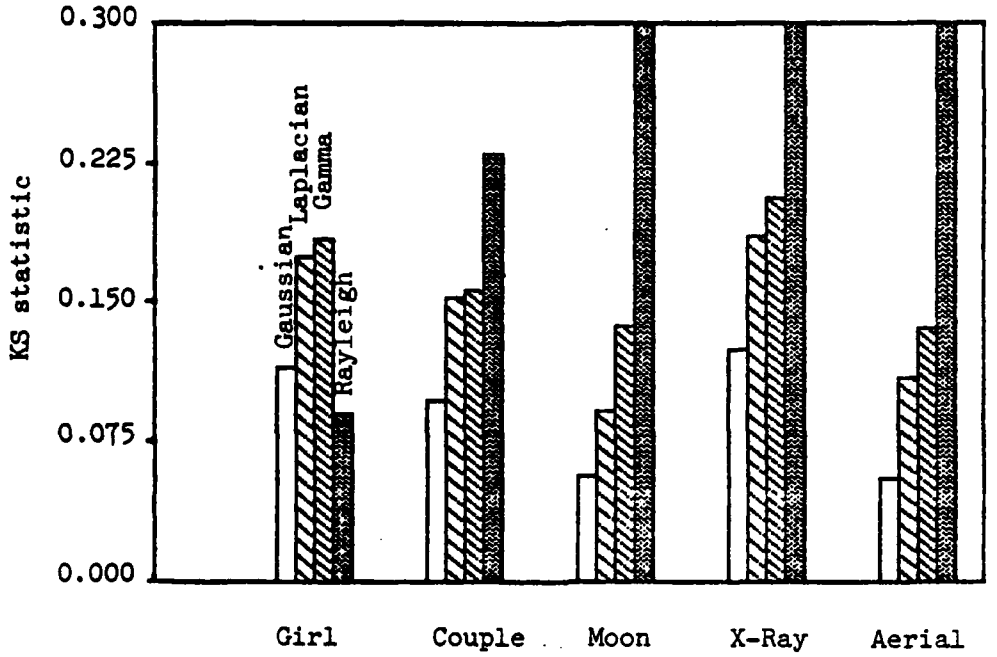
$$S_{ij}^2 = \frac{1}{M} \sum_{k=1}^{M} (c_{ij}(k) - \overline{c_{ij}})^2 \tag{3.4}$$

The data was then tested against the Gaussian, Laplacian, and Gamma distributions which had mean and variance equal to the sample mean and variance, respectively. In addition, for the DC coefficient, c_{00}, the data points were tested against the Rayleigh distribution which had variance equal to the sample variance.

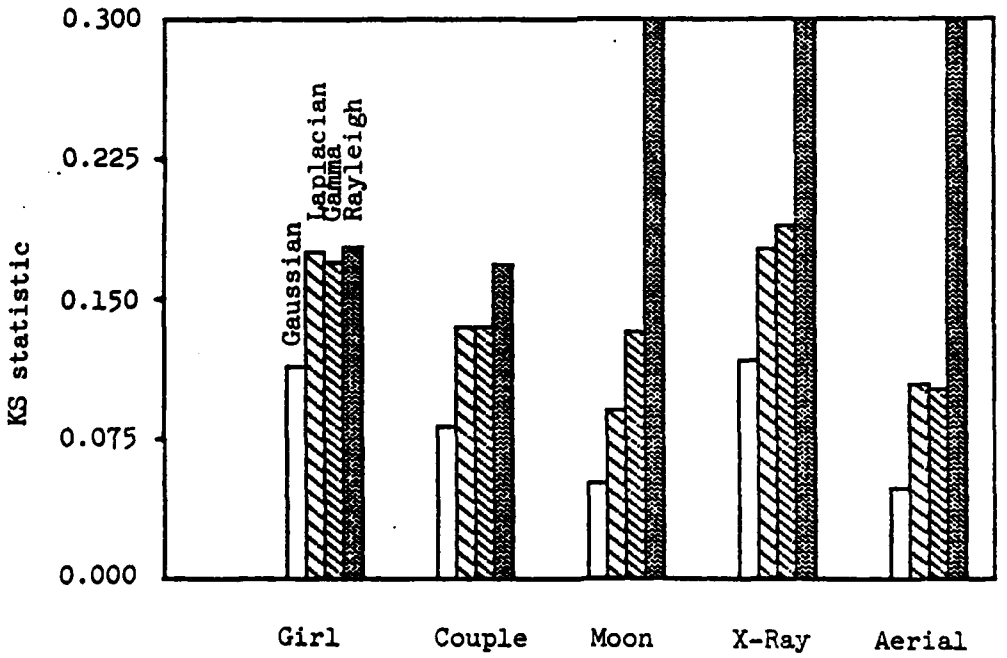
3.0. KS TEST RESULTS

Partial results of the KS tests for coefficients c_{00}, c_{01}, and c_{10} are shown in Figures 3.2, 3.3, and 3.4, respectively, with each figure having a graph for each of the three block sizes 8, 16, and 32. These three coefficients were chosen because they generally have the most effect on image quality. In each graph the x-axis is composed of five discrete points representing the five test images, with bar graphs representing the KS statistic for the given distributions, Gaussian, Laplacian, Gamma, and Rayleigh. The Rayleigh distribution is investigated only for the DC coefficient.

From Figure 3.2, it can be seen that in every case the Gaussian KS statistic is smaller than the Laplacian, Gamma, and Rayleigh statistics. Hence for all of the images, it is reasonable to conclude that the DC coefficient, c_{00}, is Gaussian for all three block sizes in question. In almost all cases the Rayleigh distribution proved to be a very poor choice for modelling the DC coefficient. In most cases for coefficients c_{01} and c_{10}, the Gaussian KS statistic is larger than the



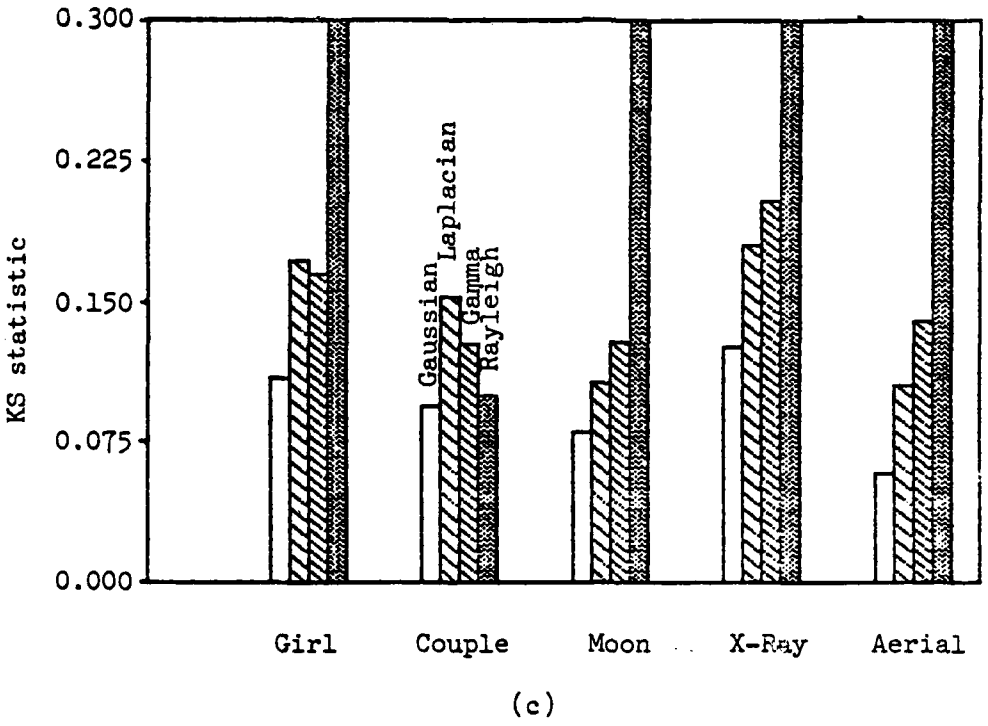
(a)



(b)

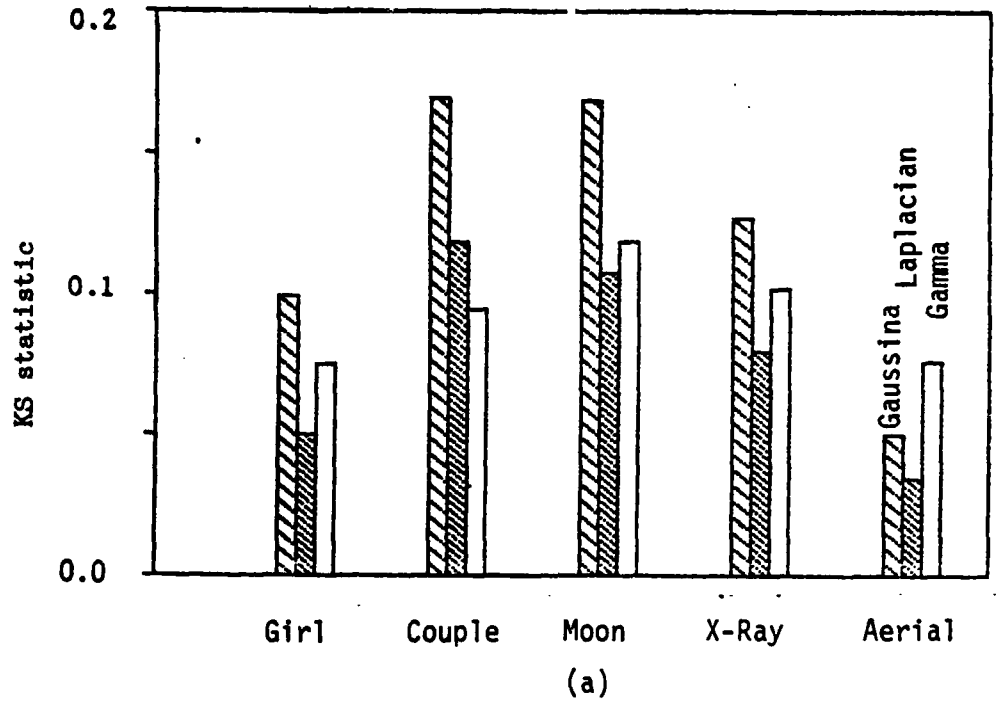
Figure 3.2. KS test statistics for coefficient C_{oo}

(a) N = 8, (b) N = 16, (c) N = 32

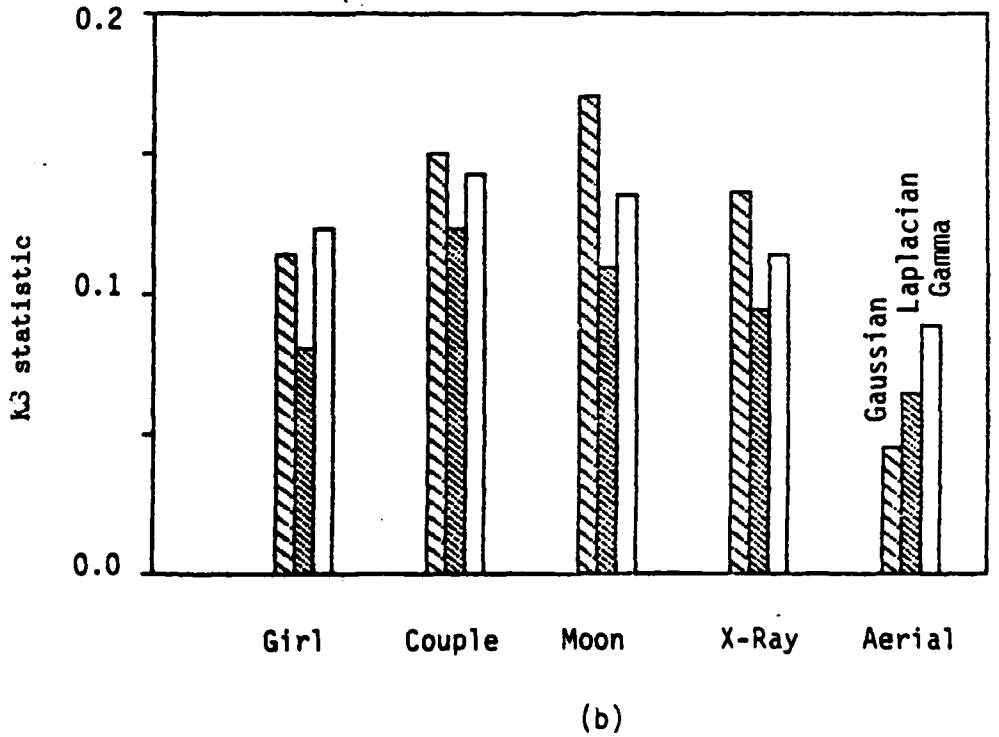


(c)

Figure 3.2. (cont.)

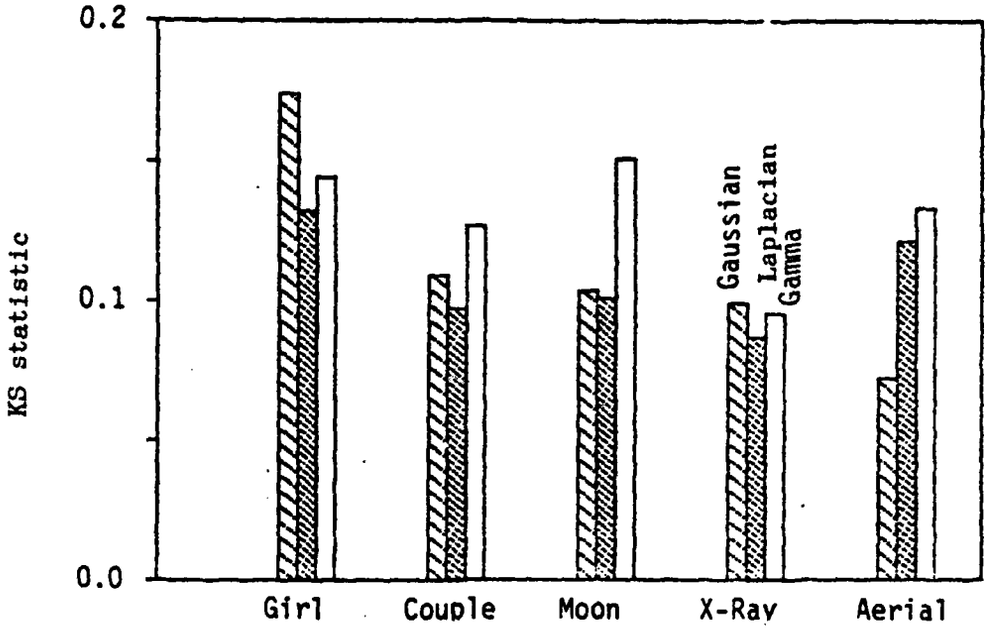


(a)



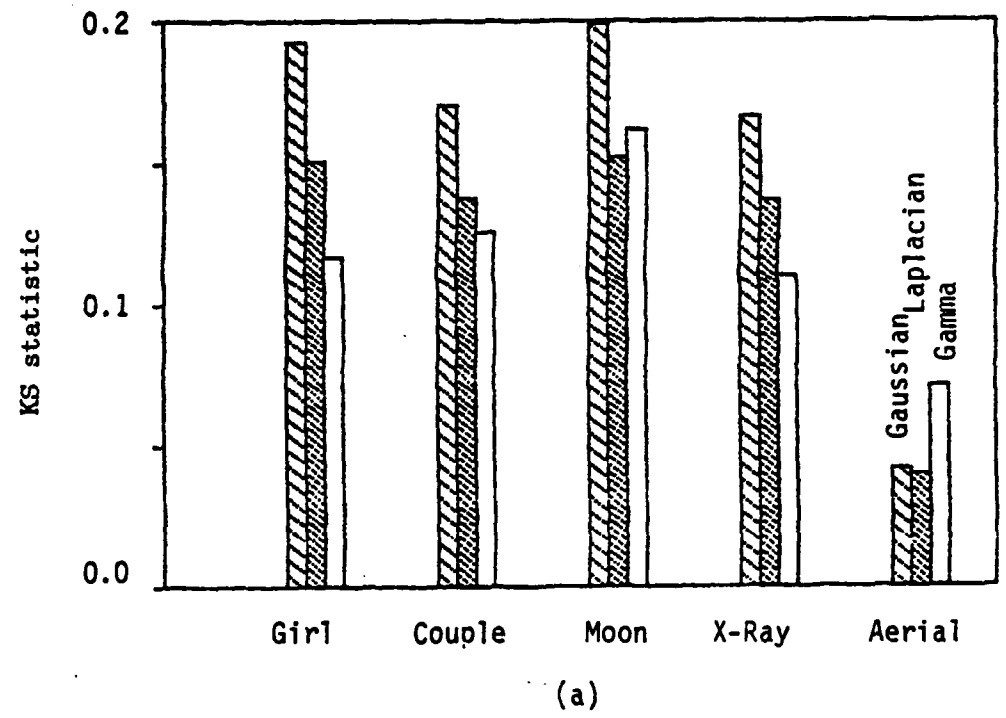
(b)

Figure 3.3. KS test statistics for coefficient c_{01}
(a) N=8, (b) N=16, (c) N=32



(c)

Figure 3.3. (cont.)



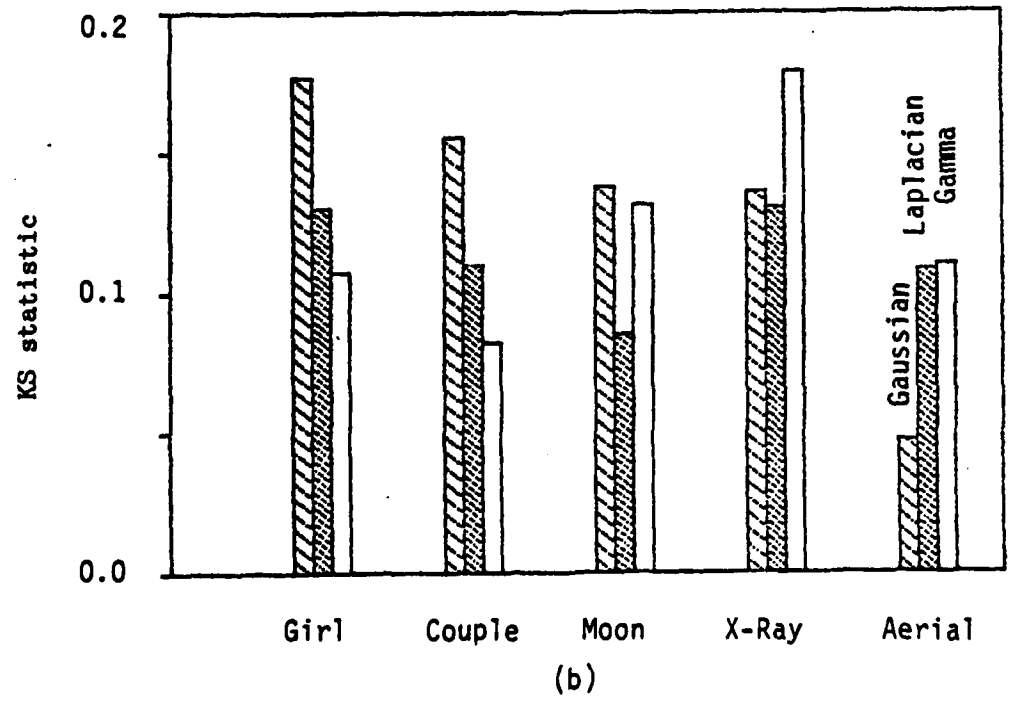
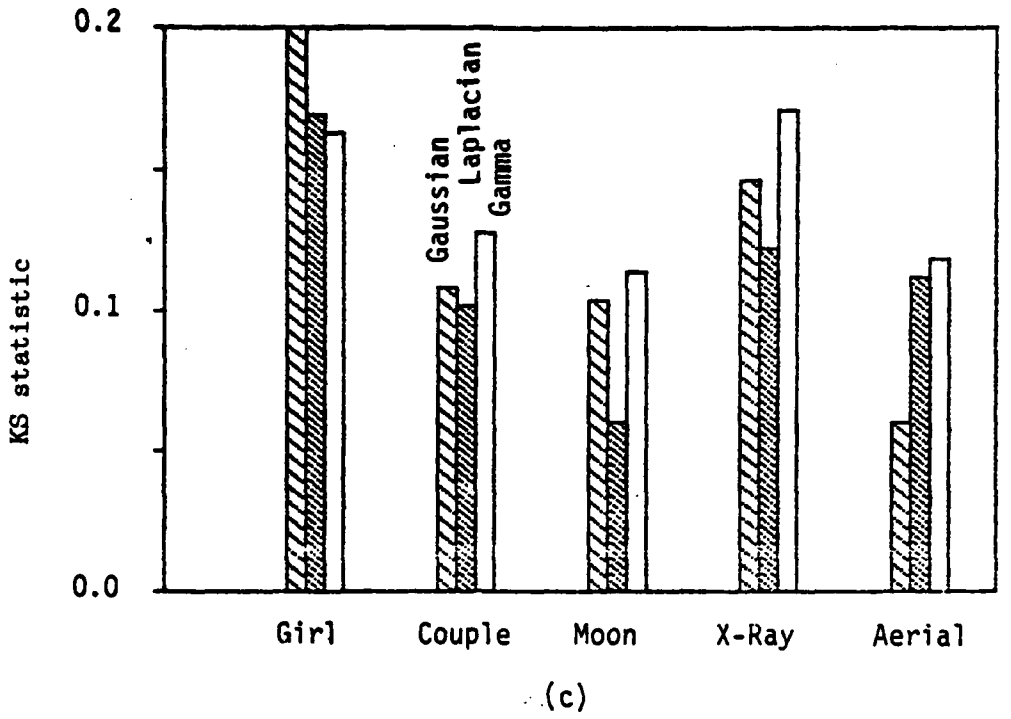


Figure 3.4. KS test statistics for coefficient c_{10}
(a) N=8, (b) N=16, (c) N=32



(c)

Figure 3.4. (cont.)

Laplacian and Gamma, and hence the assumption that these coefficients are Gaussian is unreasonable. The exception is the aerial image, for which the Gaussian KS statistic is the smallest in almost every case, with the Gamma KS statistic always the largest. In Figure 3.3, it can be seen that for coefficient c_{01}, the Laplacian KS statistic is the smallest in most of the cases. In Figure 3.4, there are approximately the same number of cases in which the Laplacian or Gamma KS statistic is the smallest. The results of the KS tests for the other high energy coefficients were similar to the results for coefficients c_{01} and c_{10}, which generally showed that the coefficients were non-Gaussian and tended to be more Laplacian than Gamma. This data indicates that for many images, it would be reasonable to assume that all of the coefficients except c_{00} have a Laplacian distribution. However, the results for the aerial image indicate that the Gaussian assumption is reasonable for very 'busy' images with much detail.

4.0. SIMULATION RESULTS AT ONE BIT/PEL

In light of the results in Section 3.0, for block size N = 16 and average rate of 1 bit per pel, the transform coefficients of the five images were quantized in two different manners. For the first method, it was assumed that the non-DC coefficients were Gaussian, and thus, optimum non-uniform Gaussian quantizers (see Max [6]) were used for all of the non-DC coefficients. In the second method, it was assumed that the non-DC coefficients were Laplacian, and optimum non-uniform Laplacian quantizers were used for those coefficients. In both methods a uniform quantizer was used for the DC coefficient, and the DC coefficient was assumed to be Gaussian. In all cases, the quantizers for the non-DC coefficient were scaled to the sample mean and variance of the coefficients for the image, and the bits allocated to each coefficient were determined by the Wintz-Kurtenbach [7] scheme which is dependent on the sample variance of the coefficients of the image. The theoretical error due to quantization can be expressed by

$$\varepsilon^2 = \varepsilon_q^2 + \varepsilon_t^2 \tag{3.5}$$

where ε_q^2 is the quantization noise and ε_t^2 is the truncation noise which occurs when a coefficient is allocated zero bits, and hence set to zero. If NB_{ij} represents the number of bits allocated to coefficient c_{ij}, the expression for quantization and truncation noise is given by

$$\varepsilon_q^2 = \sum_{\substack{ij \\ NB_{ij} > 0}} nq_{ij} S_{ij}^2, \quad i,j = 0, 1, 2, \ldots, N-1, \tag{3.6}$$

and

$$\varepsilon_t^2 = \sum_{\substack{ij \\ NB_{ij} = 0}} S_{ij}^2, \quad i,j = 0, 1, 2, \ldots, N-1, \tag{3.7}$$

where nq_{ij} is the normalized quantization noise for the optimum NB_{ij} bit quantizer for the given distribution and S_{ij}^2 is the sample variance as in Equation (3.4). The theoretical and simulation performance for the three quantization methods and the five images are presented in Table 3.1. In this table the theoretical signal-to-noise ratio (SNR) is computed from

$$\text{Theoretical SNR} = \sum_{\text{all } i,\, j} S_{ij}^2/\varepsilon^2 \tag{3.8}$$

while the simulation SNR is given by

$$\text{Simulation SNR} = \frac{\sum_{i=1}^{(256)^2} [y_i - \bar{y}]^2}{\sum_{i=1}^{(256)^2} [y_i - \hat{y}_i]^2} \tag{3.9}$$

where y_i represents the ith original pel, $\bar{y}$ is the mean value of the original pels, and $\hat{y}_i$ is the ith reconstructed pel.

Table 3.1.

THEORETICAL AND SIMULATION PERFORMANCE (SNR)
FOR DIFFERENT QUANTIZERS
(DISCRETE COSINE TRANSFORM, 1 BIT/PEL, N = 16)

Image		Gaussian	Laplacian
Girl	Theory	19.48 dB	18.41 dB
	Simulation	16.82	18.25
Couple	Theory	18.51	17.40
	Simulation	14.98	16.62
Moon	Theory	14.57	13.53
	Simulation	12.29	13.45
X-Ray	Theory	11.13	10.07
	Simulation	9.31	9.62
Aerial	Theory	14.26	13.09
	Simulation	13.46	13.50

Notice that when it is assumed that the non-DC coefficients are Gaussian, the theoretical SNR is about 3dB higher than the simulation SNR, but when it is assumed that they are Laplacian, the theoretical SNR is only about 1dB higher than the simulation. In addition, using Laplacian quantizers in simulations resulted in a gain of about 1dB over the Gaussian quantizers, which coincided with slightly better quality in the reconstructed images, due to the Laplacian quantized images having less blocking than the Gaussian quantized images. For the aerial image, the actual SNR for the Gaussian and Laplacian quantizers are about the same, but the theoretical values differ, with the higher theoretical value for the Gaussian quantizer and a lower theoretical value for the Laplacian quantizer. This result is consistent with the data from the KS tests which indicated that the aerial image was best represented by a Gaussian distribution. From these results it is evident that the Laplacian assumption for non-DC coefficients yields a higher simulation SNR and a much better agreement between theory and simulation than the Gaussian assumption.

5.0. CONCLUSIONS

The results given in this section indicate that for a large class of images, the DC coefficient is best modelled by a Gaussian distribution, and the non-DC transform coefficients are best modelled by a Laplacian distribution, which agrees with the assumption of [4]. Assuming that the coefficients are Gaussian will, for most images, result in a higher theoretical performance than can actually be attained. By modelling the transform coefficients as Laplacian and using the appropriate quantizers, not only can simulation performance be improved, but the theoretical performance will be much more indicative of the actual performance. However, as in the aerial image, some images are best represented by Gaussian statistics, and thus care must be taken to correctly classify the images that are to be processed.

SECTION IV

SOFT DECISION DEMODULATION

AND

TRANSFORM CODING OF IMAGES

1.0. INTRODUCTION

In order to transmit binary image data efficiently, it is necessary to employ some method of bandwidth compression in the image coding system. The two most popular methods that have been proposed for efficient coding of images are differential pulse code modulation (DPCM) and transform coding (TC). Due to the different coding techniques used in these systems, channel errors that occur during transmission have different effects on the reconstructed image. In DPCM systems, a channel error causes a streak in the reconstructed image, while in TC systems a channel error is averaged over all the pixels in a block, with the severity of the error depending on the importance of the coefficient in which the error occurred. The channel errors can be virtually eliminated by the use of forward error correcting codes (FEC), [7,8], but the use of these codes increases the bandwidth necessary for transmission, and thus reduces the efficiency of the coding system. Another technique for reducing the effect of channel errors is to employ a receiver that can detect channel errors, and if needed, modify the decoder output. The primary advantages of these type systems is that they do not require an increase in the transmitted data rate, and no special equipment is needed at the transmitter since all of the error detection/correction is performed at the receiver. An example of such a system for DPCM and PCM coded images can be found in Ngan and Steele [2]. In their system, after the image has been decoded, each pixel in the reconstructed image is compared to the previous pixel in that row. If the difference of the pixels is greater than a statistically determined threshold, then the pixel is replaced by an estimate determined by averaging surrounding pixels. In [1],

Sundberg presents a method of improving the performance of a speech coding system using soft decision demodulation, in which certain of the received bits are monitored for reliability, with the decoder rejecting an entire codeword and replacing it with a suitable estimate if one of the monitored bits is received in error. Although both of the systems employ error detection at the receiver, they differ in that Ngan and Steele's method attempts to detect errors on the reconstructed image while Sundberg's method attempts to detect errors at the output of the channel, before the image is decoded. This section presents an image coding system using transform coding for bandwidth compression, and Sundberg's soft decision demodulation technique for error control.

2.0. SYSTEM DESCRIPTION

A block diagram of the coding system using transform coding and soft decision demodulation is shown in Figure 4.1. The input, x_i, is an N x N block of pixels and H is a two-dimensional discrete cosine transform (DCT). Each of the coefficients in the N x N block at the output of the transform is quantized independently by a Gaussian quantizer scaled to the mean and variance of the coefficient, with the number of bits allocated to each coefficient determined by the Wintz-Kurtenbach bit allocation scheme [3]. In this scheme, for a given number of bits per block, the highest energy coefficients are allocated the most bits and the lowest energy coefficient allocated the least bits (truncated coefficient). These bit allocations are held fixed for each of the N x N transform blocks of the image. Notice that no error correction/detection techniques are implemented at the transmitter. At the receiver, the codeword representing a coefficient is decoded to the corresponding reconstructed coefficient c_i, but in addition, the most significant bits (MSB) of the highest energy coefficients are monitored for reliability. If it is determined that one of the MSB is unreliable, then the coefficient corresponding to that codeword is rejected and replaced by an estimate. The

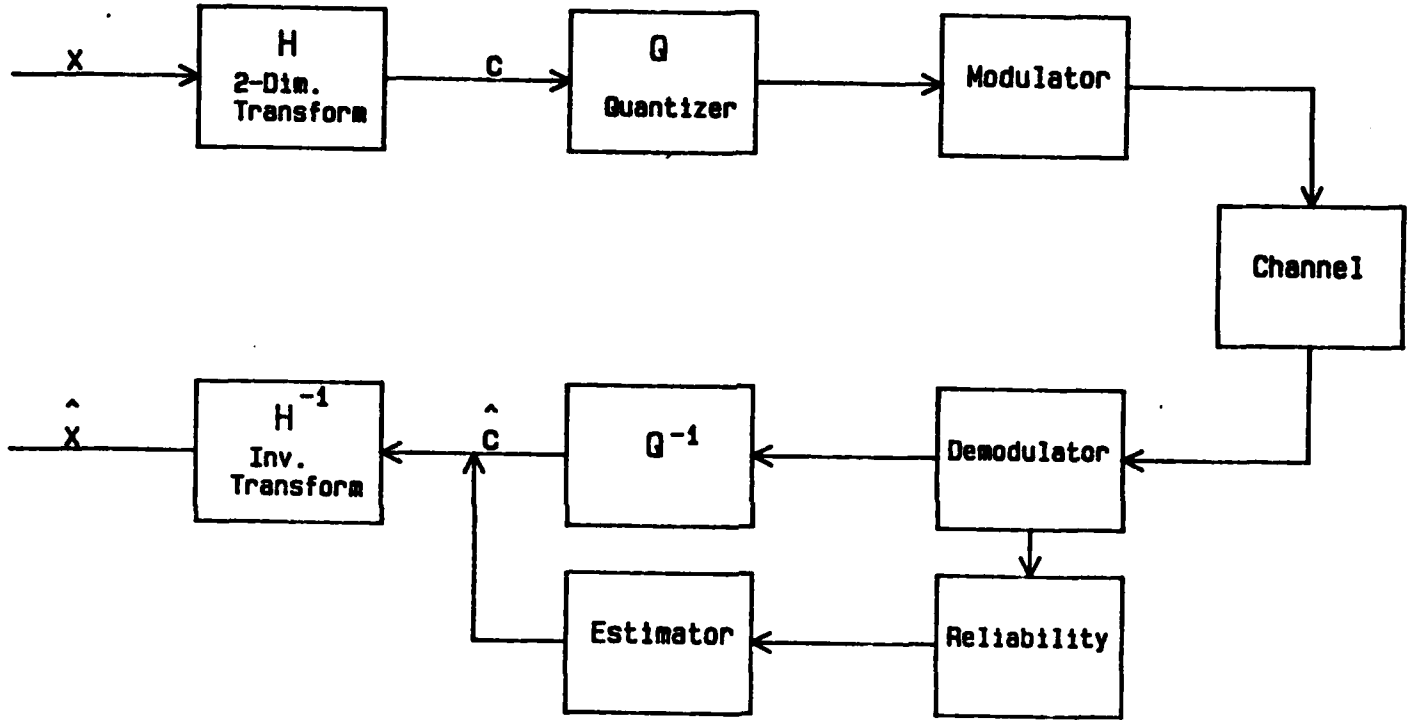


Figure 4.1. Soft Decision Demodulation System

reconstructed image, $\hat{x}_i$, is then obtained by the inverse DCT operation, H^{-1}.

In this type of system, there are three types of events that can occur:

1. A channel error occurs, but the receiver does not detect it or
2. A channel error occurs, and the receiver detects it and replaces the affected coefficient with an estimate, or
3. No channel error occurs, but the receiver thinks one has, and it replaces the affected coefficient with an estimate.

In designing a soft decision system it is desirable to minimize Type 1 events (maximize Type 2 events) when channel errors have occurred and to minimize Type 3 events when no channel errors have occurred. Since increasing the probability of a Type 2 event given that a channel error occurs also increases the probability that a Type 3 event will occur, the goals stated above are conflicting, and some trade off must be made between the different types of errors. The criterion used here in optimizing the system is the familiar minimum mean squared error between the original and reproduced image. Although minimizing mean squared error does not necessarily coincide with the best perceptual image, it is useful in that it gives a basic framework for the initial design of the decoder.

3.0. SOFT DECISION DEMODULATION

To simplify notation, the N x N matrix of coefficients will be considered as a 1 x N row vector, in which c_1 - c_N is the first row of the N x N block, c_{N+1} - c_{2N} is the second row of the N x N block, and so on. Consider coefficient i, c_i, which has average energy σ_i^2 and has been allocated NB_i bits. The mean squared error in reproducing c_i at the receiver is

$$\varepsilon_i^2 = [\varepsilon_q^2(NB_i) + \varepsilon_c^2(NB_i) + \varepsilon_a^2(i)] \cdot \sigma_i^2 \qquad (4.1)$$

where ε_q^2 and ε_c^2 represent the quantization and clipping noise introduced at the transmitter and ε_a^2 represents the noise introduced by the channel. Each of these quantities are normalized, assuming that the input to the NB bit quantizer has unit variance. The terms ε_q^2 and ε_c^2 depend on the number of bits allocated to the quantizer and the probability density function of the input to the quantizer. The sum $\varepsilon_q^2 + \varepsilon_c^2$ for unit variance optimum Gaussian quantizers of size one to eight bits can be found in Table 4.1 [4]. The channel noise term, ε_a^2, depends on the noise on the channel and the type of coding and modulation that is used.

Since the basis vectors of the DCT are approximately the same as the principal components, the mean squared error in the coefficients is approximately the same as in the reproduced image, and since the basis vectors are virtually uncorrelated, the mean squared error in the coefficients is the sum of the terms in (4.1). Thus, the mean squared error in the reproduced image can be expressed by

$$\varepsilon_T^2 = \sum_{j=1}^{N^2} \varepsilon_j^2 \tag{4.2}$$

For a system with no error correction/detection, the channel noise can be expressed as

$$\varepsilon_a^2(i) = P \cdot \sum_{j=1}^{NB_i} A(j, NB_i) \tag{4.3}$$

where P is the probability of a channel bit error. The term $A(j, NB_i)$ is the so called A factor, which represents the mean squared error in reproducing a quantized codeword with a single bit error in bit j of an NB_i bit quantizer (averaged over quantizer output statistics) [1]. The A factors and quantization noise for folded binary Gaussian quantizers of size one to eight bits are shown in Table 4.1 [6]. Following the development in Sundberg [1], we assume that the bits of a codeword are transmitted independently as binary antipodal signals of

TABLE 4.1. QUANTIZATION/CLIPPING NOISE AND SINGLE BIT A-FACTORS FOR 1 - 8 BIT GAUSSIAN QUANTIZERS

		Single Bit A-Factors							
NB	$\varepsilon_q^2 + \varepsilon_c^2$	1	2	3	4	5	6	7	8
1	.363	2.54							
2	.117	3.53	1.12						
3	.0454	3.86	1.51	.376					
4	9.49×10^{-2}	3.96	1.83	.453	.109				
5	2.50×10^{-2}	3.99	2.07	.509	.126	.0296			
6	1.21×10^{-2}	3.98	2.25	.563	.141	.0351	.0088		
7	6.23×10^{-3}	3.98	2.25	.563	.141	.0352	.0088	.0022	
8	4.69×10^{-3}	3.98	2.25	.563	.141	.0352	.0088	.0022	.0005

energy E, and that the additive noise has double sided noise spectral density $N_0/2$. At the receiver the most significant bits of the highest energy coefficients are monitored for reliability. If the received signal for bit j falls in the erasure zone $(-T_{ij}\sqrt{E}, T_{ij}\sqrt{E})$, $0 < T_{ij} < 1$, then that bit is considered unreliable, and hence the entire codeword is rejected. The number of MSB that must be monitored and their associated thresholds T_{ij}, are determined by the A factors and the channel signal-to-noise ratio, $2E/N_0$.

The following derivation of optimum threshold levels is an adaptation of the derivation in Sundberg [1], but it is included here because of the "two-dimensional" notation and for clarity. The expression for the channel noise with a soft decision receiver is

$$\varepsilon_a^2(i) = \sum_{j=1}^{M_i} A(j, NB_i)\, Pu(i, j) + P \sum_{j=M_i+1}^{NB_i} A(j, NB_i) + \Delta\sigma_i^2\, Pr_i \qquad (4.4)$$

In (4.4) M_i is the number of most significant bits that are monitored, Pu(i, j) is the probability that bit j is in error but the received signal is not in the erasure zone, P is the channel error probability, Pr_i is the probability that at least one of the M_i most significant bits were received unreliably, and $\Delta\sigma_i^2$ is the normalized mean square estimation error for coefficient i. The expressions for the probabilities in (4.4) are

$$P = Q(\sqrt{2E/N_0}) \qquad (4.5)$$

$$Pu(i, j) = Q(\sqrt{2E/N_0}\,(T_{ij} + 1)) \qquad (4.6)$$

where $Q(\cdot)$ is the Q function defined in [5]. To calculate Pr_i, it is useful to introduce Pz_{ij}, which is the probability that bit j of coefficient i is received in the erasure zone

$$Pz_{ij} = Q(\sqrt{2E/N_0}\ (1 - T_{ij})) - Q(\sqrt{2E/N_0}\ (1 + T_{ij})) \tag{4.7}$$

The probability, Pr_i, can now be expressed as

$$Pr_i = 1 - \prod_{j=1}^{M_i} (1 - Pz_{ij}) \approx \sum_{j=1}^{M_i} Pz_{ij} \tag{4.8}$$

Optimizing (4.4) for M_i and T_{ij} using (4.5) - (4.8) yields the optimum thresholds

$$T_{ij} = \frac{1}{2(2E/N_0)} \log_e \left[\frac{(A(j, NB_i)}{\Delta\sigma_i^2} - 1 \right] \tag{4.9}$$

and M_i is the largest j such that T_{ij} is defined and greater than zero.

The proposed estimate of c_i given that a soft decision error has occurred, is simply the average of the corresponding coefficients in neighboring blocks. For example, assume that coefficient c_1 in block (ℓ, m) has been determined to be unreliable, then it is rejected and replaced by

$$\hat{c}_1(\ell, m) = (1/3)\ [\hat{c}_1(\ell - 1, m) + \hat{c}_1(\ell - 1, m - 1) + \hat{c}_1(\ell, m - 1)]. \tag{4.10}$$

These three neighboring blocks will have already been decoded, and it is assumed that they have been decoded correctly. In addition to the soft decision demodulation, an averaging process is implemented on the high energy coefficients in order to correct some of the errors that are undetected by the soft decision receiver. For each block that is not on the edge of the image, the sample mean and variance of a coefficient in the eight neighboring blocks are calculated. If the variance is less than $20\sigma_i^2$ then the coefficient is replaced by the mean of the neighboring blocks. The graphic location of these neighboring blocks is shown in Figure 4.2. The blocks marked with an S are those used in the soft

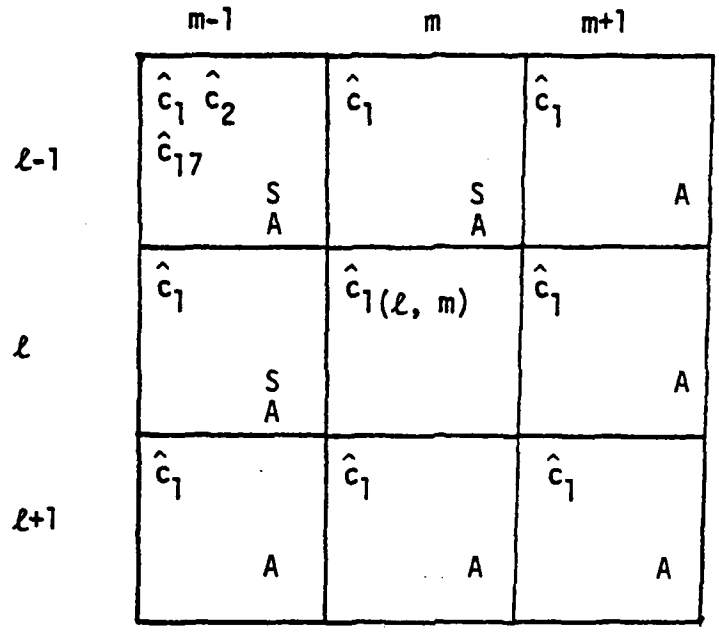


Figure 4.2. Location of Neighboring Blocks Used for Soft Decision Estimation and Averaging.

decision estimator, those marked with an A are used in the averaging procedure. This averaging technique allows the correction of some of the errors that have gone undetected (Type 1), especially in areas of the image that are constant, but has no effect on blocks in regions where the intensity is changing rapidly.

4.0. SIMULATION

The system proposed above was tested by computer simulation. The images used were of size 256 x 256 pixels and they were divided into 256 16 x 16 blocks for processing (N = 16). The three highest energy coefficients, 1, 2, and 17, were selected for soft decision monitoring. These coefficients are the three coefficients in the upper left hand position of each block. Coefficient 1, also known as the DC coefficient, contains the information on the overall intensity of the transformed block, and has the highest energy (most variation) of all the coefficients. Coefficient 2 contains the information of vertical edges in the block and coefficient 17 contains information of the horizontal edges in the block, and they have approximately the same energy.

The choice of the $\Delta\sigma_i^2$'s for the three coefficients were obtained experimentally by comparing the visual effects on the image. Since coefficient 1 is more highly correlated across surrounding blocks than coefficients 2 and 17, the reconstruction error $\Delta\sigma_1^2$ is less than $\Delta\sigma_2^2$ or $\Delta\sigma_{17}^2$. Due to the symmetrical location of coefficients 2 and 17, the reconstruction errors for these coefficients were always assigned the same value.

5.0. RESULTS

The effects of channel errors and soft decision demodulation on two images is shown in Figures 4.3 and 4.4. In these figures, the (a) images are the output of the transform coding system with transmission rate 1 bit per pixel and no error detection/correction or channel noise. When channel noise is introduced into the system with probability of error 10^{-2}, the (b) images are the result. The (c)

(a) 1 bit/pel 2 Dimensional DCT

(b) Channel Errors

(c) Soft Decision 1

(d) Soft Decision 2

Figure 4.3. Simulation Results for Channel Errors and Soft Decision (Girl)

(a) 1 bit/pel 2 Dimensional DCT

(b) Channel Errors

(c) Soft Decision 1

(d) Soft Decision 2

Figure 4.1. Simulation Results for Channel Errors and Soft Decision (Couple)

and (d) images are a result of introducing soft decision demodulation and the averaging procedure described previously to the same channel noise that produced the (b) images. The parameters of the soft decision system that produced images (c) and (d) are in Table 4.2. For the (d) images, the $\Delta\sigma_i^2$ parameter of coefficients 2 and 17 was decreased from that used in the (c) images, to increase the width of the erasure zones for the bits of these coefficients. This change increases the probability of Type 2 and 3 events, and decreases the probability of Type 1 events that occur in these coefficients. In both (c) and (d) systems only the two MSB of the three coefficients are monitored for reliability. Thus, only 6 of the 256 bits received for each block of data need be monitored for reliability.

In comparing Figures 4.3(b), (c) and (d) it can be seen that in this case, the implementation of soft decision demodulation resulted in a very much improved image. In Figure 4.3(c) six of the blocks where a DC error occurred in Figure 4.3(b) have been detected by the soft decision receiver and restored to very near their original levels. An error in the DC coefficient can be recognized by a uniform change in the intensity of all the pixels of the block in which the error occurred. The most common result of an error in the DC coefficient is a block that is all white or all dark, and has lost all of its detail. For example, in the dark block above the flower on her left shoulder, the error in the DC coefficient was not detected (Type 1) and hence the entire block has lost its detail. In Figure 4.3(c) there are two blocks in the upper right part of the image that have contours introduced by the channel noise. The vertical contour in the block near the top of the picture was caused by a Type 1 event in the second MSB of coefficient 2, and the norizontal contour in the block near the center of the picture was caused by a Type 1 event in the second MSB of coefficient 17. It should also be noted that in this latter block, an error occurred in one of the least significant bits of the DC coefficient, causing it to darken as seen in Figure 4.3(b). However, the

Image	Coefficient	$\Delta\sigma_i^2$	Bit	Threshold	Averaging
(c)	1	.3	1 2	.231 .172	yes
	2	.8	1 2	.127 .055	no
	17	.8	1 2	.127 .055	no
(d)	1	.3	1 2	.231 .172	yes
	2	.5	1 2	.179 .115	yes
	17	.5	1 2	.179 .115	yes

TABLE 4.2. PARAMETERS FOR SOFT DECISION SYSTEM

averaging procedure was able to restore the proper intensity of this block since all of the neighboring blocks are approximately the same intensity. With the wider erasure zones used for coefficients 2 and 17 in the system used for Figure 4.3(d), the errors that caused these contours were detected (Type 2) and the contours eliminated.

The images in Figure 4.4 demonstrate some of the trade-offs involved in a soft decision system. While most of the intensity errors in Figure 4.4(c) were detected using soft decision, the demodulator was unable to restore all the blocks to the original level. Only the black block above the man's left shoulder in 4.4(b) was restored correctly. A DC error was detected in the block in the upper right hand corner, but since this block is decoded first and there were no previous blocks from which to base an estimate, the DC coefficient of this block was replaced with the mean value, which results in the light block of Figures 4.4(c) and (d). More interesting in this image is the introduction of errors in the soft decision process (Type 3). Notice that two errors appear in Figures 4.4(c) and (d) that are not in Figure 4.4(b). The first of these is in the block on the man's (right) shoulder, in which a false vertical contour appears. In this block, the second bit of coefficient 2 was decoded as unreliable although it was not in error. Due to the edge occurring in this block, it is dissimilar with the neighboring blocks previously decoded, and thus the estimate for this coefficient is poor. The other error introduced by the soft decision receiver is in a block near the center of the image. In this block, the receiver decoded the first bit of the DC coefficient as unreliable, and replaced the coefficient with an estimate. Although the error is noticeable, it is not too severe since there is not a sudden change in the intensity of neighboring blocks and thus a good estimate could be obtained.

In Figures 4.5 - 4.7, the performance curves for the soft decision systems are plotted. In these figures, the x axis, SNRI, represents the channel signal-

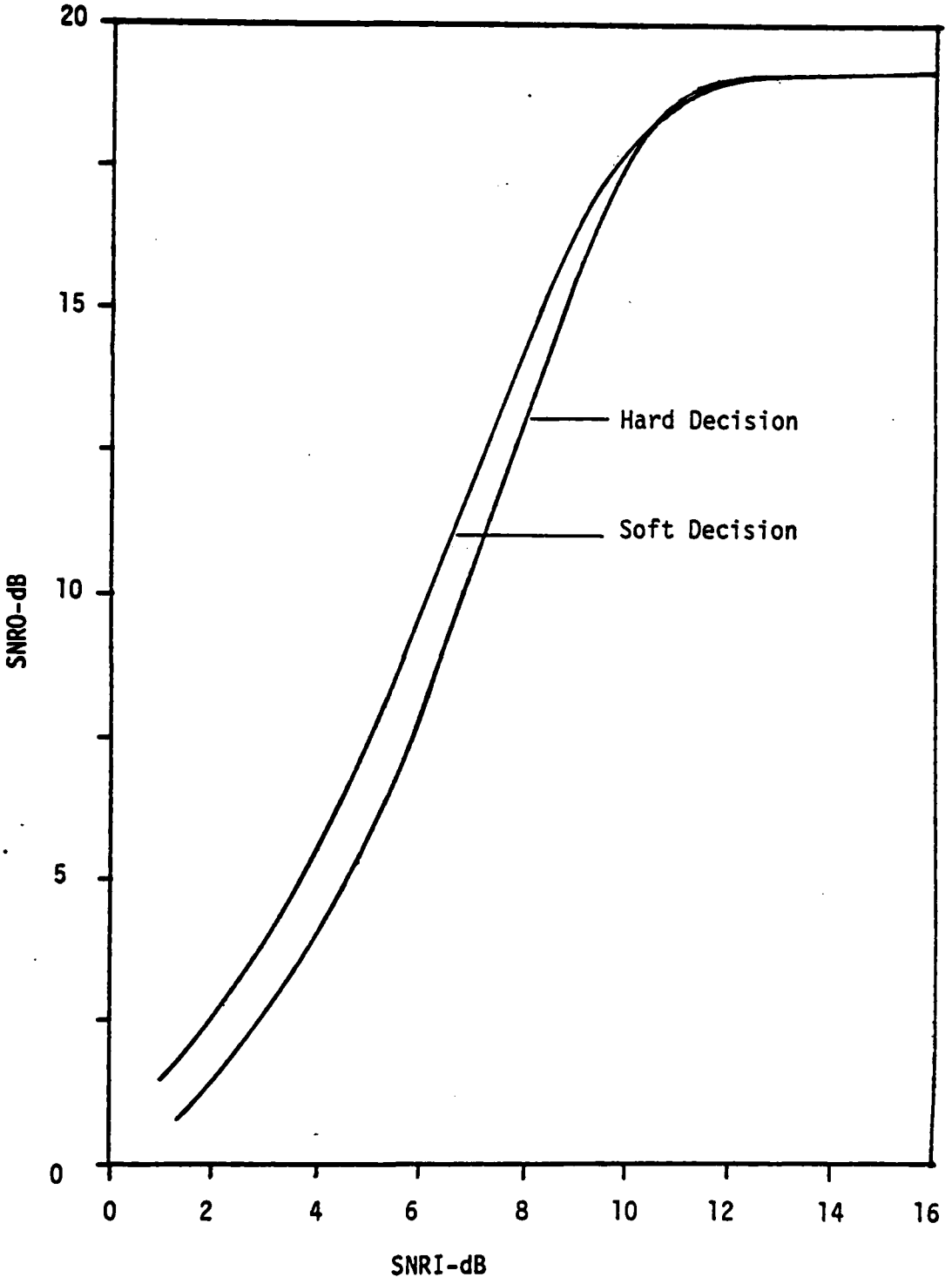


Figure 4.5. Expected Performance for Hard and Soft Decision Receivers

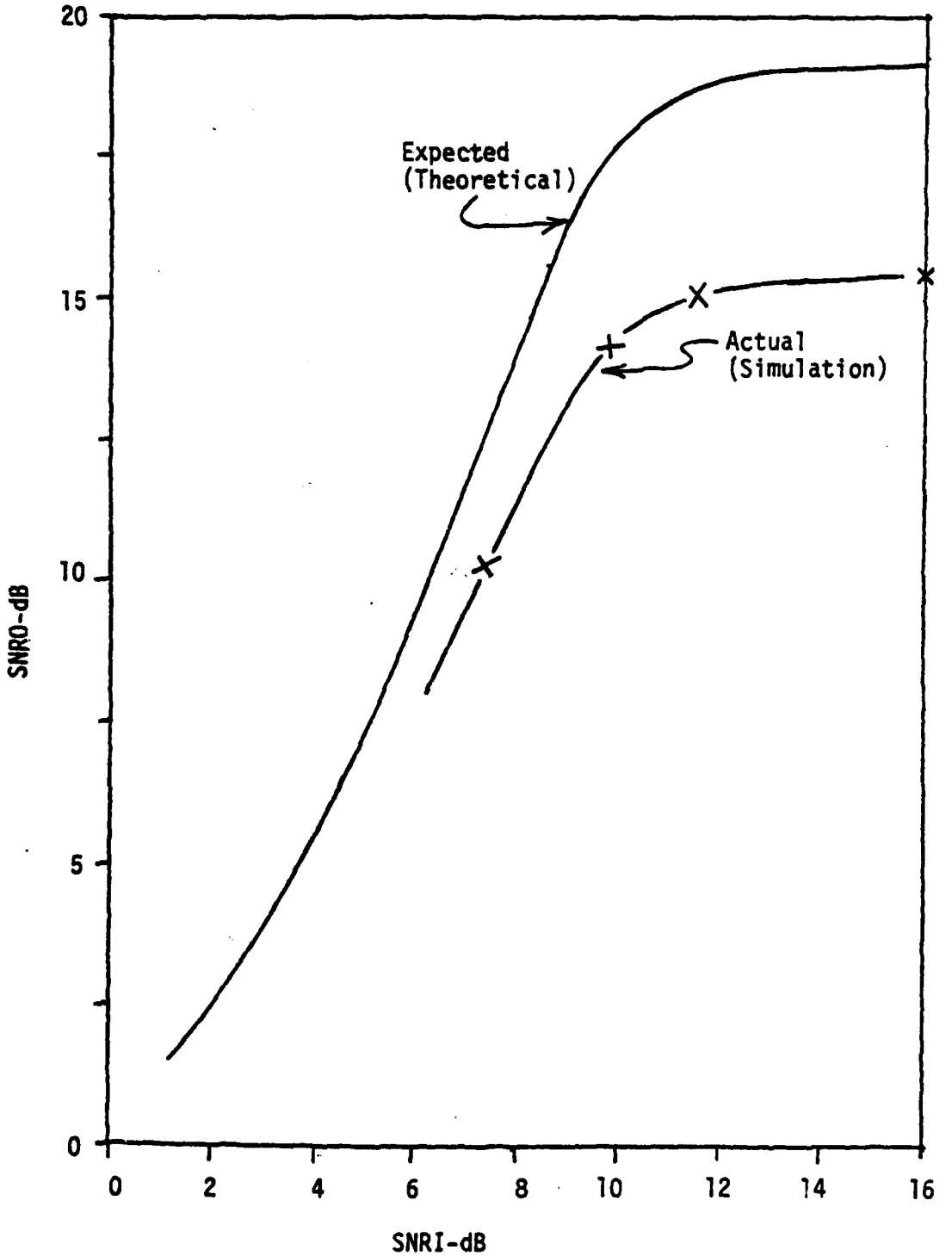


Figure 4.6. Comparison of Expected and Experimental Performance for Soft Decision Receiver (Assuming Gaussian Coefficients)

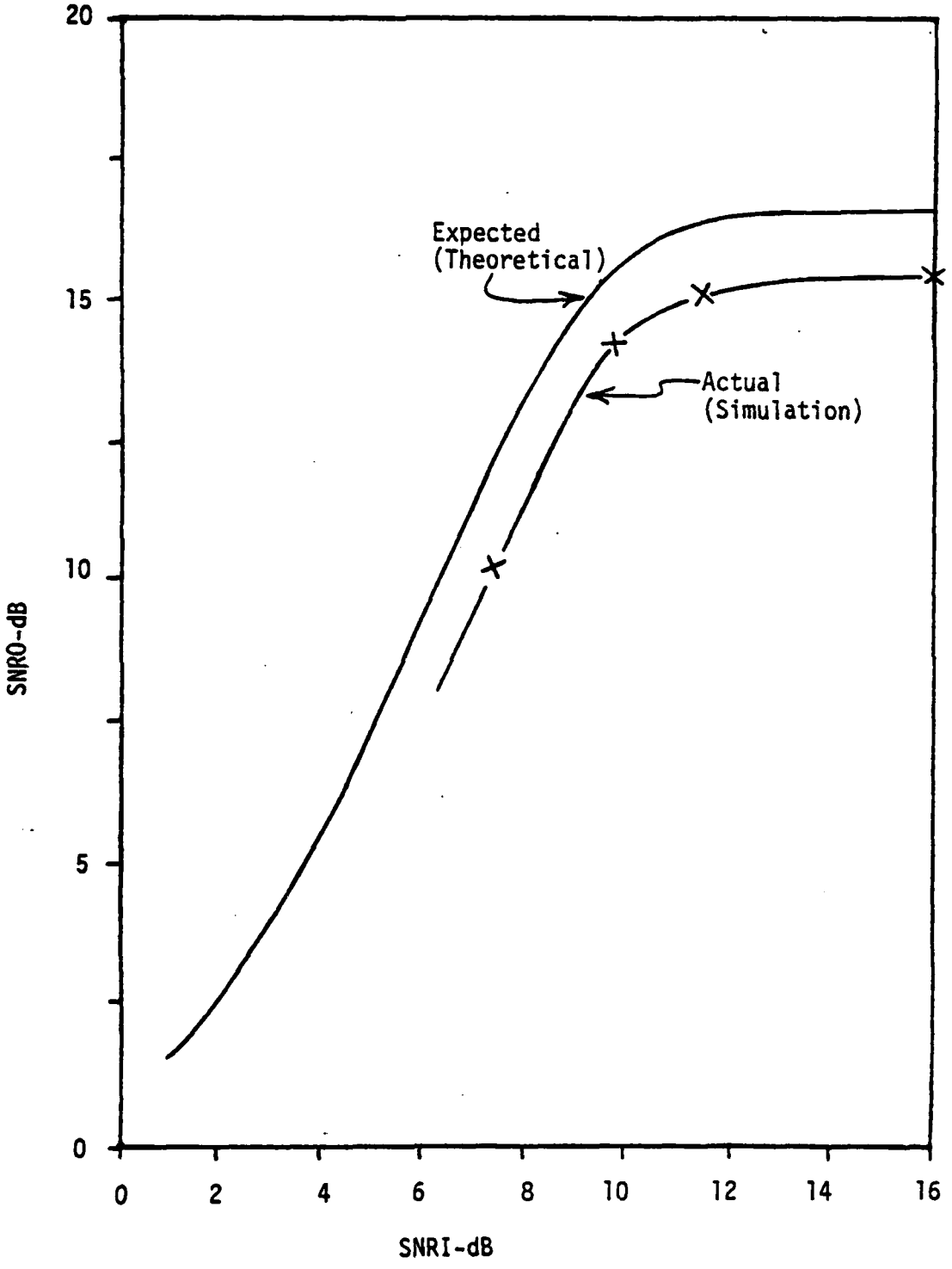


Figure 4.7. Comparison of Expected and Experimental Performance for Soft Decision Receiver (Assuming Laplacian Coefficients)

to-noise ratio $2E/N_0$, and the y axis, SNRO, represents the output signal-to-noise ratio defined by

$$SNRO = \frac{\langle (x_i - \bar{x})^2 \rangle}{\langle (x_i - \hat{x}_i)^2 \rangle} \tag{4.11}$$

where $\bar{x}$ is the mean value of the image pixels and the $\hat{x}_i$'s are the reconstructed pixels.

In Figure 4.5, the expected performance for a hard and soft decision receiver for a 1 bit/pixel system are plotted. The soft decision system has the parameters as described in system (d) of Table 4.3, with the soft decision thresholds fixed at their optimum values as in Equation (4.9) for an expected channel SNR of 7.35 dB ($Pe = 10^{-2}$). The output SNR for the soft decision system was calculated from Equations (4.1), (4.2), (4.4), and (4.5) - (4.8), and the output SNR for the hard decision receiver was calculated using Equations (4.1), (4.2), (4.3), and (4.5). From Figure 4.5, it can be seen that soft decision has better performance than hard decision if the channel SNR is less than 10dB. At the design value, SNRI = 7.35dB, the soft decision receiver shows about 2dB improvement over the hard decision receiver. For SNRI greater than 10dB, the hard decision receiver is only slightly better than the soft decision receiver, and at very high SNRI, they both level off at SNRO = 19.2dB. This leveling off represents the quantization noise introduced at the transmitter, since at high channel SNR the probability of a channel error goes to zero. These curves are in agreement with the images of Figures 4.3 and 4.4 which showed improvement when the hard decision receiver for the noisy channel was replaced by a soft decision receiver.

In Figure 4.6, the expected performance for the soft decision system is compared to experimental results of a Monte Carlo simulation of the system (see Appendix A.4). It is clear from this figure that the expected (theoretical) per-

formance curve does a very poor job in representing the actual system performance. Notice that at high SNRI, the curves differ by more than 4dB, and since the quantization noise is dominant in this region, it can be inferred that the differences in the curves arise from mismatched quantizer statistics. Thus, the original assumption that the DCT coefficients are Gaussian distributed may be invalid for a 16 x 16 transform block. Further investigation indicated that a Laplacian distribution is a much better fit to the statistics of the DCT coefficients than a Gaussian distribution, which is in contradiction with past practice [9, 10]. The expected performance for the soft decision system was recalculated assuming that the coefficients were Laplacian and the result is shown in Figure 4.7 along with the experimental results.

Since the A factors and quantization noise terms used in calculating the system performance are dependent on the probability distribution of the input to the quantizer, it was necessary to calculate new A factors and quantization noise terms in order to obtain the expected (theoretical) performance curve in Figure 4.7. In the Monte Carlo simulation, the system used optimum Gaussian quantizers for the DCT coefficients. The expected performance for this system, with the assumption that the input to the quantizer has a Laplacian distribution, is obtained by calculating new A factors and quantization noise terms for the Gaussian quantizers with an input that has a unit variance Laplacian distribution. From Figure 4.7, it is seen that the expected performance for the Laplacian assumption is a much better indicator of the actual performance than the original Gaussian assumption. For high channel SNR, the expected performance levels off a 16.5 dB which is just over 1 dB greater than the simulation results.

The results above confirm that the DCT coefficients are much closer to Laplacian statistics than to Gaussian. This implies that the system performance could be improved by redesigning the quantizers to match the Laplacian statistics.

6.0. CONCLUSIONS

In soft decision demodulation, if certain bits of a received codeword are unreliable, the codeword is replaced by an estimate. By monitoring only the two most significant bits of the three highest energy DCT coefficients, the reconstructed image can be considerably improved for a system transmitting over a noisy channel. The main tool used in analyzing soft decision systems is the A factor. The A factors are a function of the spacing of the quantizer, the code assigned to the quantizer, and the probability density of the input to the quantizer. The invalid assumption that the DCT coefficients were Gaussian distributed resulted in incorrect A factors for the system and led to erroneous results for expected system performance. The expected performance was corrected by assuming that the DCT coefficients were Laplacian distributed.

APPENDIX A.4

This appendix contains the details of the Monte Carlo simulation from which the experimental performance curve in Figures 4.6 and 4.7 was obtained. For the simulation, the transmitted data rate was fixed at 1 bit/pixel and the block size was N = 16. The soft decision thresholds were fixed at the values of the (d) system described in Table 4.2. The system was tested using the girl image for channel error rates of 10^{-2}, 10^{-3}, and 10^{-4}, which correspond to channel SNR's of 7.35 dB, 9.8 dB, and 11.4 dB respectively. For each of the three error rates, 25 channel simulations were performed. A simulation consisted of coding the DCT coefficients into their binary representation using optimum gaussian quantizers scaled to the mean and variance of the coefficients. The number of bits allocated to the quantizer of each coefficient are in Figure A-1. For each coded bit of the image, an independent gaussian random variable was generated. The variance of this random variable was set so that $Q(1/\sigma) = Pe$, where Pe is one of the three channel error rates that was tested. These gaussian random variables determined whether a bit was decoded correctly, incorrectly, or within the soft decision erasure zone. In each of the 25 simulations a different sequence of random variables was used, so that 25 different noisy reconstructed images were obtained for each of the three error rates. The experimental output SNR for an error rate was obtained by averaging the mean squared error of the 25 reconstructed images.

8 8 5 5 4 4 3 3 2 2 2 1 1 1 1 1

8 5 4 3 3 3 2 2 2 1 1 1 0 0 0 0

6 4 4 3 3 2 2 1 1 1 1 1 1 0 0 0

5 3 3 3 2 2 2 1 1 1 1 0 0 0 0 0

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

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

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

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

2 2 1 2 1 1 1 1 0 0 0 0 0 0 0 0

2 1 2 1 1 1 1 1 0 0 0 0 0 0 0 0

2 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0

2 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0

2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Figure A.4-1. Bits allocated to the DCT coefficients in the 16 x 16 transform block.

SECTION V

HAMMING CODING OF DCT-COMPRESSED IMAGES OVER NOISY CHANNELS

1.0. INTRODUCTION

In the transmission of images over a noisy channel using transform source coding, reconstructed image quality is substantially degraded by channel errors. As a result, for noisy channel applications it is necessary to correct the channel errors or to devise methods for reducing the effects of the errors. Efforts in this latter category include the work by Ngan and Steele [1], Mitchell and Tabatabai [2], and Reininger and Gibson [3]. The research described in the present section is concerned with the former approach, namely, forward error correction of transmission errors. Previous work on forward error correcting (FEC) codes used in conjunction with the discrete cosine transform (DCT) over noisy channels has been performed by Duryea [4] and Modestino, Daut, and Vickers [5].

Duryea [4] conducted theoretical and simulation studies of three convolutional codes and three block codes. The three block codes studied were the (3, 1) repetition or majority vote code, the (7, 4) Hamming code, and the (23, 12) Golay code. For his simulation studies, Duryea uses a bit error rate of 10^{-3} and considers only two error protection schemes with the (7, 4) Hamming code. In the first scheme, a (7, 4) Hamming code was applied to all bits, while in the second method, only a 6 by 6 square block of the lowest frequency DCT coefficients were protected by the (7, 4) Hamming code. While the mean squared error was reduced, the quality of the reconstructed image was not clearly improved.

Modestino, Daut, and Vickers [5] primarily investigate convolutional codes, although they briefly consider an (8, 4) Hamming code and a (24, 12) Golay code. They consider the three options of coding all bits of each coefficient the same, coding each bit of a specified coefficient the same with variation between coefficients, and coding each bit of each coefficient differently. Further, their work

emphasizes rate 1/n short-constraint length convolutional codes which allow the use of Viterbi decoding (short constraint length) but limit channel coding flexibility (rate 1/n). Their results indicate that for noisy channels there is a distinct advantage to allocating additional channel bandwidth to channel coding rather than to source coding.

The research described in the present section is an extensive study of using Hamming codes with the two-dimensional DCT (2D-DCT) at a transmitted data rate of 1 bit/pixel over a binary symmetric channel (BSC). The system configuration of interest is shown in Fig. 5.1. The combination of Hamming codes with the 2D-DCT is logical since both methods are block oriented. The (7, 4), (15, 11), and (31, 26) Hamming codes are used to protect the most important bits in each transformed block, where the most important bits are determined by calculating the mean squared reconstruction error contributed by a channel error in each individual bit. A theoretical expression is given which allows one to compute the number of protected bits to achieve minimum mean squared reconstruction error for each code rate. By comparing these minima, the best code and bit allocation can be determined. The design bit error rate of interest is 10^{-2}. Monte Carlo simulation results and reconstructed images are presented to demonstrate the utility of the method.

2.0. TWO-DIMENSIONAL DCT

The monochrome images used for this work consist of 256 by 256 pixels with each pixel represented by an 8-bit word. The two-dimensional DCT (2D-DCT) is a popular transform for image compression at 1 bit/pixel [6], and it is considered exclusively in this work. The 2D-DCT is defined by

$$F(u,v) = \frac{2}{N} c(u)\, c(v) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) \cdot \cos\left[\frac{(2j+1)\pi u}{2N}\right] \cos\left[\frac{(2k+1)\pi v}{2N}\right]; \qquad (5.1)$$

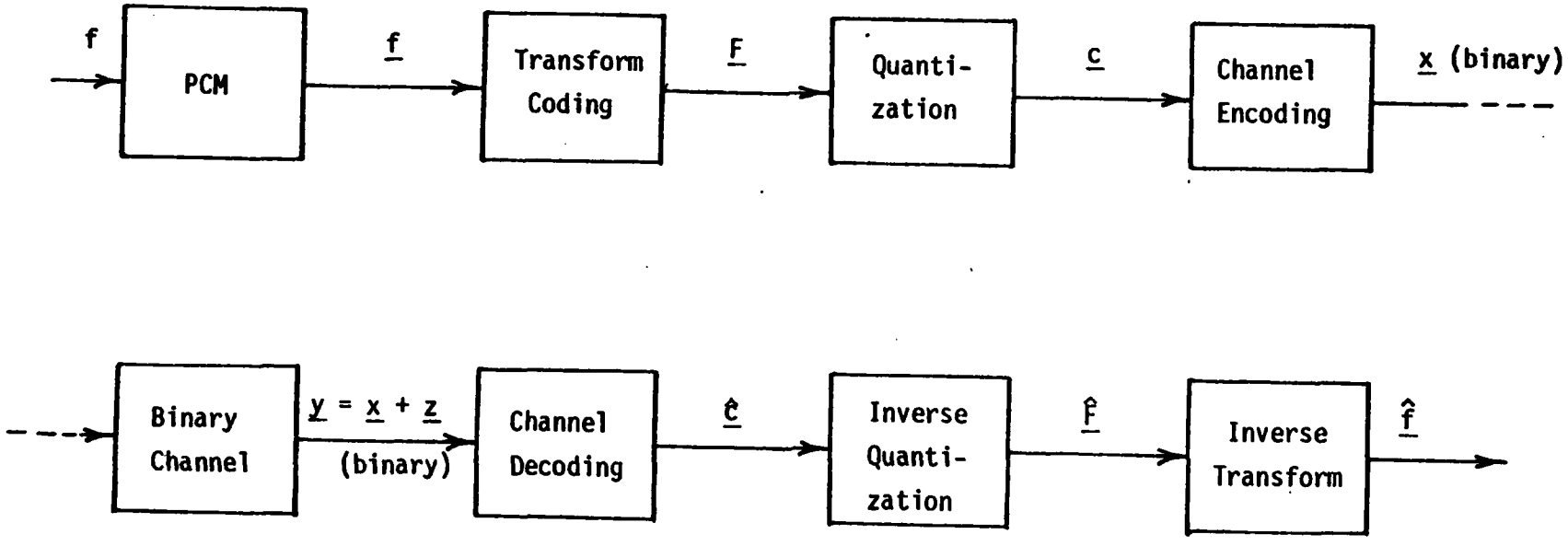


Figure 5.1. Block Diagram of Digital Image Processing System with Transform Source Coding and Channel Coding

for u, v = 0, 1, ..., N - 1, c(0) = $1/\sqrt{2}$, and c(u) = 1 for u = 1, 2, ..., N - 1. The inverse 2D-DCT is

$$f(j,k) = \left(\frac{2}{N}\right) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)\, c(v)\, F(u,v) \cos\left[\frac{(2j+1)\pi u}{2N}\right] \cdot \cos\left[\frac{(2k+1)\pi v}{2N}\right], \qquad (5.2)$$

for j, k = 0, 1, ..., N - 1. One advantage of the 2D-DCT is that it can be computed using "fast" algorithms.

To use the 2D-DCT in a data compression system, F(u, v) in Eq. (5.1) is calculated over an N by N block (N = 16 for this paper), the lowest energy coefficients are discarded, and the highest energy coefficients are quantized and coded. The scheme used for bit allocation is due to Wintz and Kurtenbach [7]. The coefficients were quantized using minimum mean squared error (MMSE), nonuniform, Gaussian-assumption quantizers for up to 36-levels [8] and using MMSE Gaussian-assumption, uniform quantizers for more than 36-levels. The quantizer output levels are represented digitally by the folded binary code (FBC). In the absence of channel errors, this method produces good quality reconstructed images at a rate of 1 bit/pixel.

3.0. CHANNEL CODING

A bit error rate (BER) of 10^{-2} causes substantial degradation in the 2D-DCT coded images. To reduce or remove these channel error effects, (7, 4), (15, 11), and (31, 26) Hamming codes are investigated. In order to apply these codes, it it necessary to determine how many bits to protect and which bits to protect. The latter question is answered first by calculating the mean squared reconstruction error contributed by each of the 256 bits in a block and then ranking these bits from largest to smallest error.

Since the DCT is an unitary transform, the mean squared error (MSE) for the reconstructed image in the spatial domain is the same as the MSE in the transform domain. The MSE between an uncoded DCT coefficient F(i, j) and its received version $\hat{F}(i, j)$ is given by

$$\varepsilon^2(i,j) = E\{[F(i,j) - \hat{F}(i,j)]^2\} \tag{5.3}$$

where the expectation is taken over the source and channel probability measures. Equation (5.3) can be separated into three components,

$$\varepsilon^2(i,j) = [\varepsilon_a^2(i,j) + \varepsilon_q^2(i,j) + \varepsilon_c^2(i,j)]\ \sigma^2(i,j) \tag{5.4}$$

where ε_a^2 is the MSE contributed by the channel, ε_q^2 is the mean squared quantization error, ε_c^2 is the mean squared clipping error, and σ^2 is the mean squared value of the coefficient. Each of the terms in brackets in Eq. (5.4) is normalized to one. The mean squared quantization and clipping errors are dependent on the number of bits assigned to the particular coefficient and the probability density of the coefficient. The sum $\varepsilon_q^2 + \varepsilon_c^2$ is given in Table 5.1 for unit variance, nonuniform Gaussian quantizers with one through eight bits.

The MSE due to the channel is given by

$$\begin{aligned}\varepsilon_a^2(i,j) &= E\{[F(i,j) - \hat{F}(i,j)]^2\} \\ &= \sum_{\ell} P_\ell(z = z_\ell)\ E\{[F(i,j) - \hat{F}(i,j)]^2 \mid z = z_\ell\} \\ &= \sum_{\ell} P_\ell(z = z_\ell)\ A_\ell,\end{aligned} \tag{5.5}$$

where A_ℓ is called the A-factor associated with error sequence $z_\ell[10]$. The A-factor is the average reconstruction error power caused by the digital error sequence z_ℓ for a given quantizer and binary code. For $P_e \leq 10^{-2}$, the probability of two or more independent channel errors in z_ℓ is small, so Eq. (5.5) can be simplified to

TABLE 5.1. QUANTIZATION PLUS CLIPPING ERROR

AND SINGLE BIT A-FACTORS FOR 1-8 BIT GAUSSIAN QUANTIZERS

No. of Bits	$\epsilon_q^2 + \epsilon_c^2$	Single Bit A-Factors							
		1	2	3	4	5	6	7	8
1	3.63×10^{-1}	2.547							
2	1.17×10^{-1}	3.532	1.117						
3	3.45×10^{-2}	3.858	1.506	.3763					
4	9.50×10^{-3}	3.961	1.828	.4525	.1095				
5	2.50×10^{-3}	3.997	2.072	.5096	.1216	.02968			
6	1.04×10^{-3}	4.001	2.772	.6933	.1733	.04333	.01083		
7	3.04×10^{-4}	4.007	3.307	.8267	.2067	.05168	.01292	.003227	
8	8.88×10^{-5}	4.009	3.860	.9650	.2413	.06031	.01508	.003774	.0009436

$$\varepsilon_a^2(i,j) \cong P_e \sum_{\ell=1}^{M} A_\ell \tag{5.6}$$

where the channel errors are assumed independent with equal probability P_e and the first M A-factors are defined to correspond to single-bit errors. Table 5.1 lists the single bit A-factors for a Gaussian quantizer and the FBC.

Using Table 5.1 and Eq. (5.6) in conjunction with the optimum bit allocation for a particular image, the relative importance of each bit in terms of its effect on mean squared reconstruction error can be computed.

For those bits protected by channel coding, the probability of bit error becomes P_{ec} so the channel MSE expression becomes

$$\varepsilon_a^2(i,j) = P_{ec} \sum_{\ell=1}^{r} A_\ell + P_e \sum_{\ell=r+1}^{M} A_\ell \tag{5.7}$$

where r bits are assumed protected and errors with coding are independent and equally likely. Since the DCT coefficients are approximately uncorrelated, the total MSE for an N by N block is given by

$$\varepsilon_t^2 = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \varepsilon^2(i,j) \tag{5.8}$$

The normalized mean squared reproduction error (NMSE) is the total error in Eq. (5.8) divided by the sum of the variances of the coefficients.

The question of how many bits to protect involves a tradeoff between bits allocated to source coding and bits allocated to channel coding with the overall rate constrained to 1 bit/pixel. By using Eq. (5.8) with Eqs. (5.4) and (5.7), the NMSE as the number of channel coding bits is increased can be computed. Figures 5.2 - 5.4 for the girl image and Figures 5.5 - 5.7 for the aerial image show how the NMSE varies as a function of the number of bits protected at a design BER of 10^{-2}. The minimum value of each of these curves is listed in

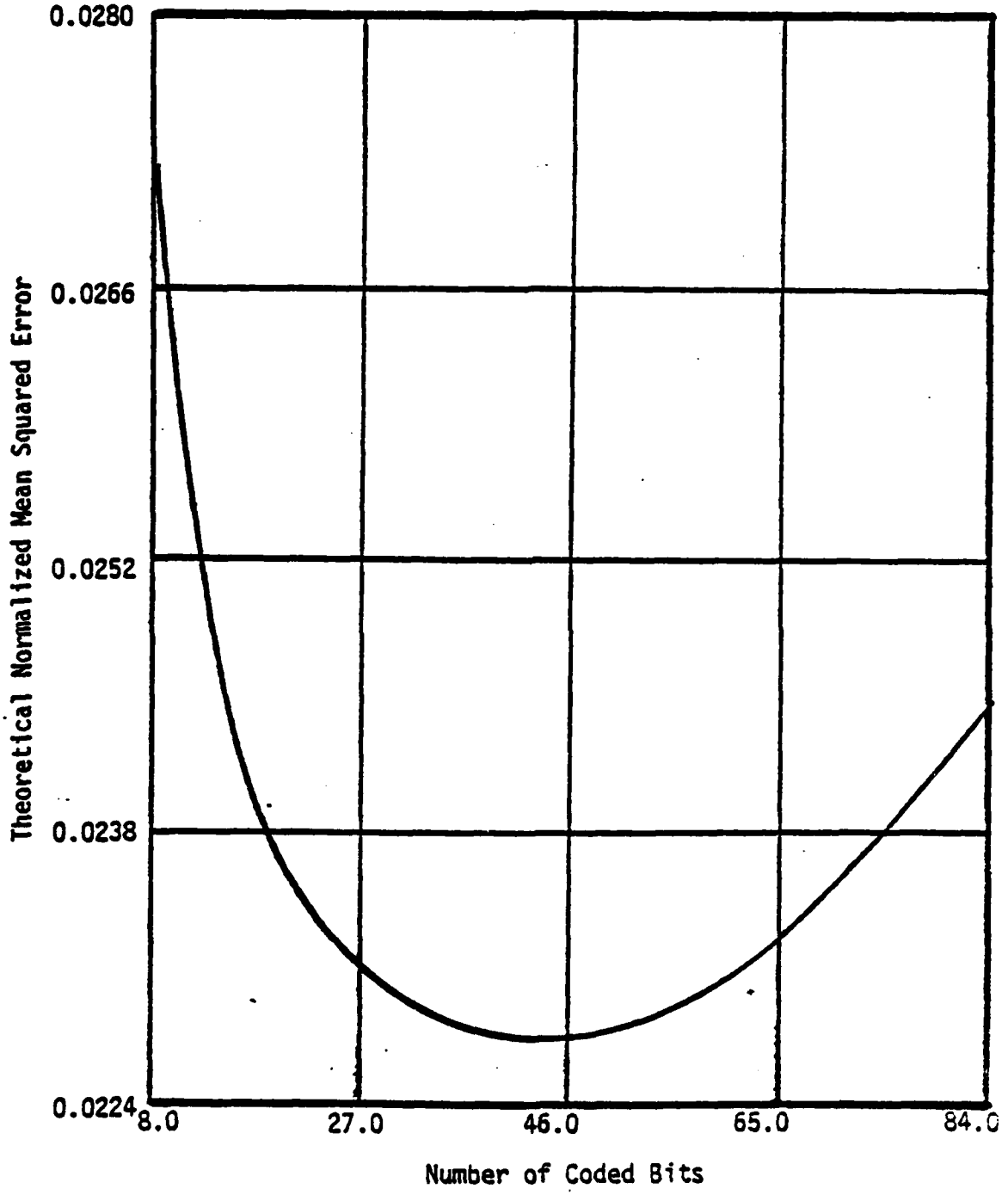


Figure 5.2. Normalized Mean Squared Error Versus Number of Bits Protected by (7,4) Hamming Code for Girl Image with 10^{-2} Error Rate.

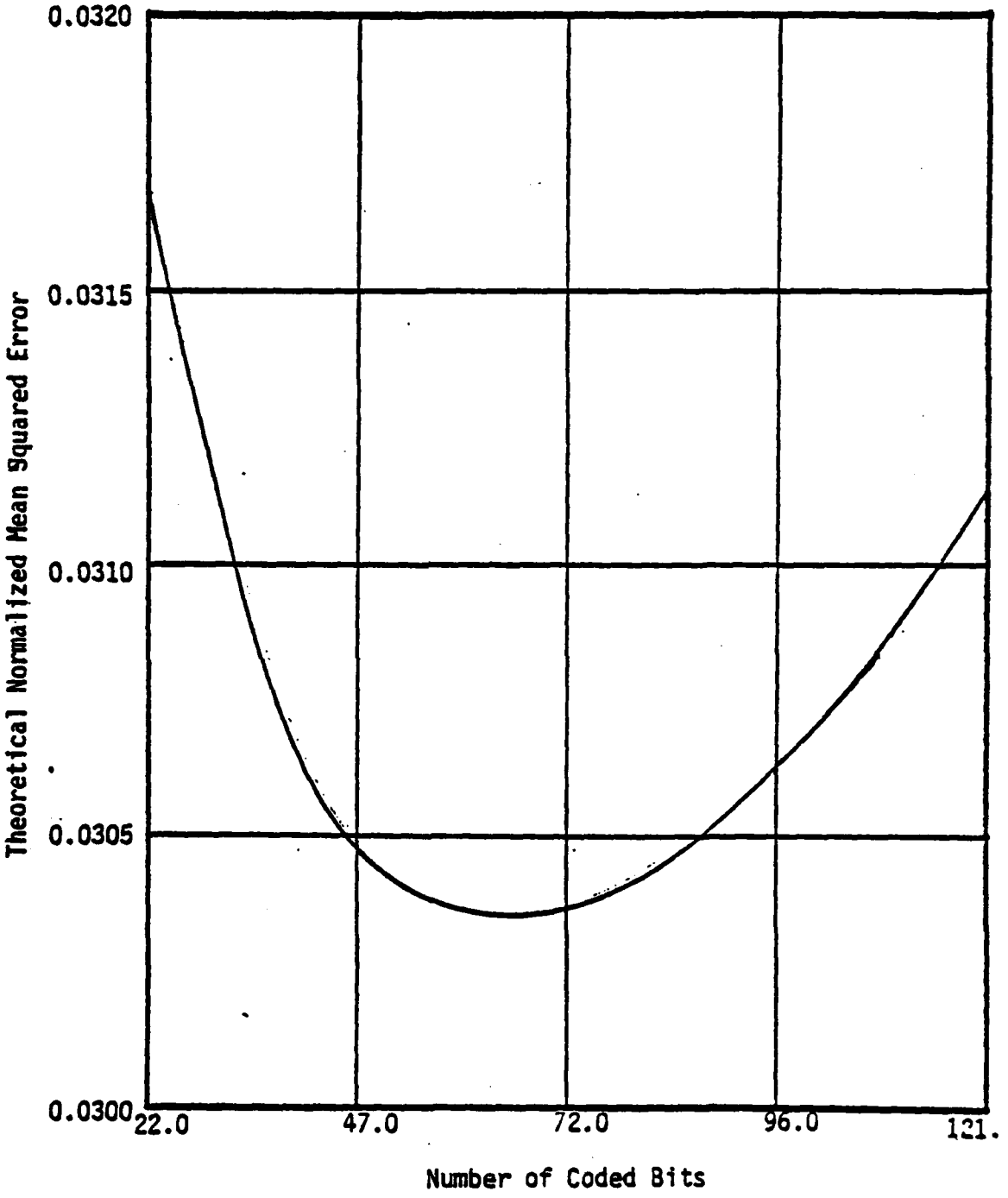


Figure 5.3. Normalized Mean Squared Error Versus Number of Bits Protected by (15,11) Hamming Code for Girl Image with 10^{-2} Error Rate

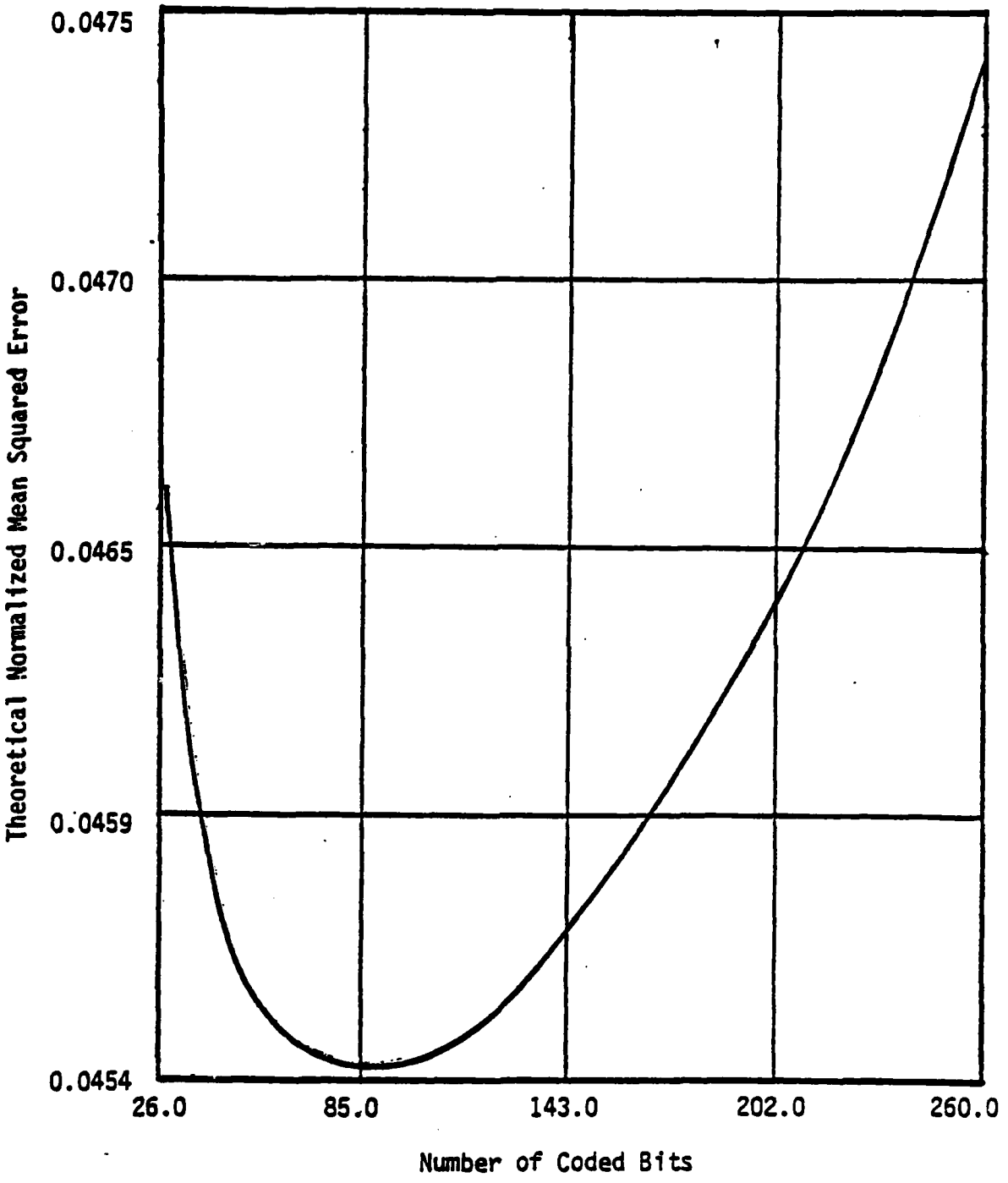


Figure 5.4. Normalized Mean Squared Error Versus Number of Bits Protected by (31,26) Hamming Code for Girl Image with 10^{-2} Error Rate

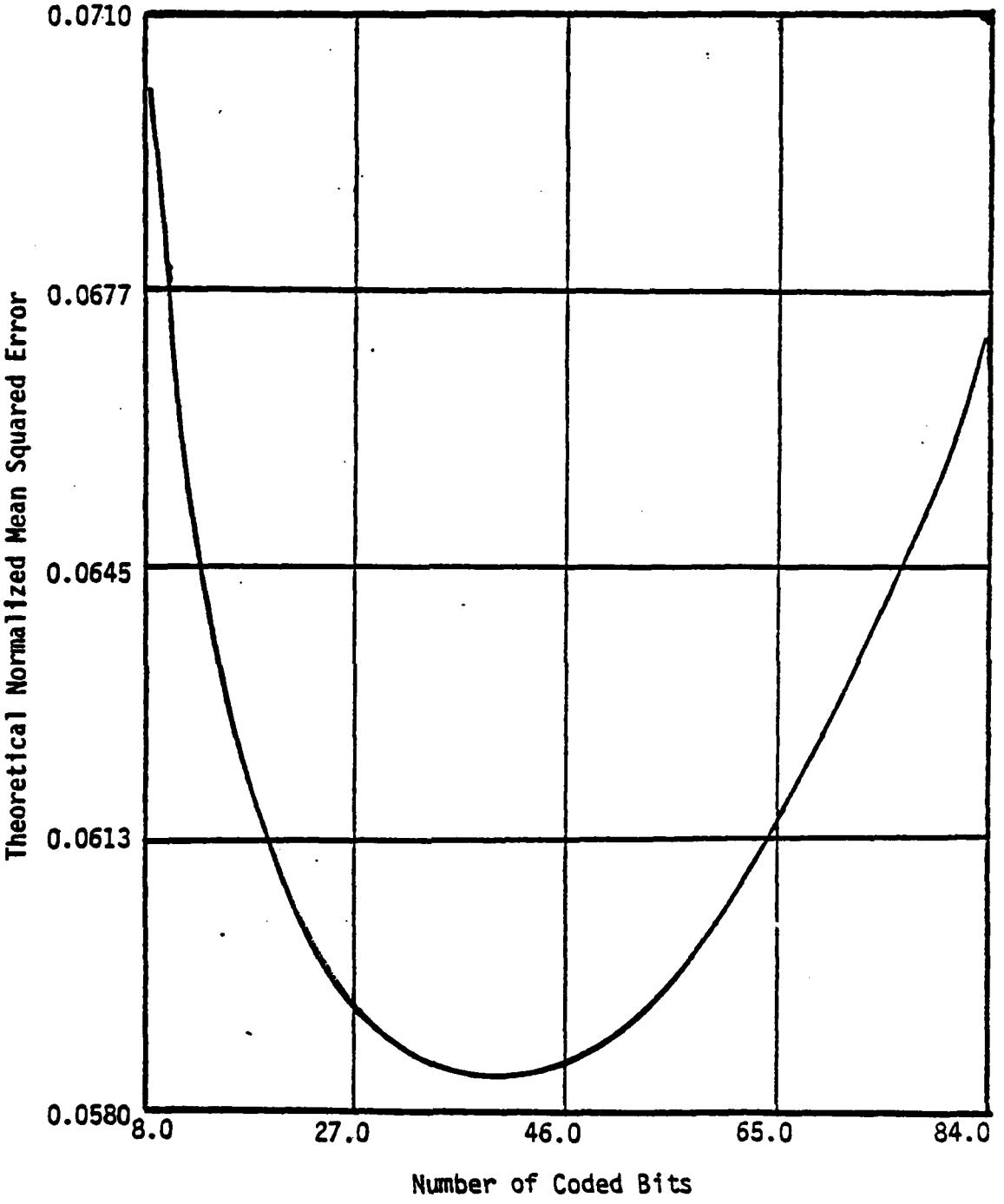


Figure 5.5. Normalized Mean Squared Error Versus Number of Bits Protected by (7,4) Hamming Code for Aerial Image with 10^{-2} Error Rate.

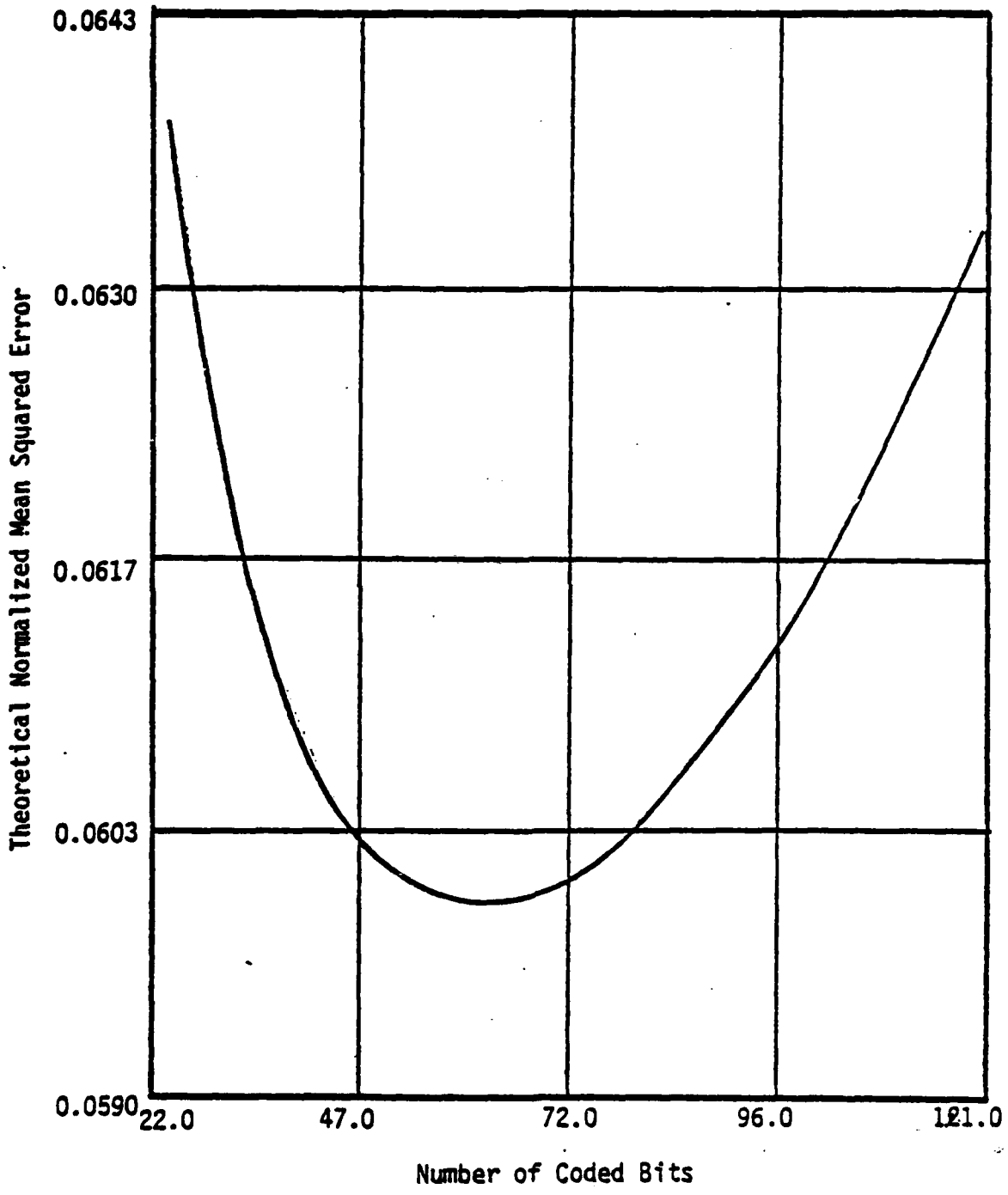


Figure 5.6. Normalized Mean Squared Error Versus Number of Bits Protected by (15,11) Hamming Code for Aerial Image with 10^{-2} Error Rate.

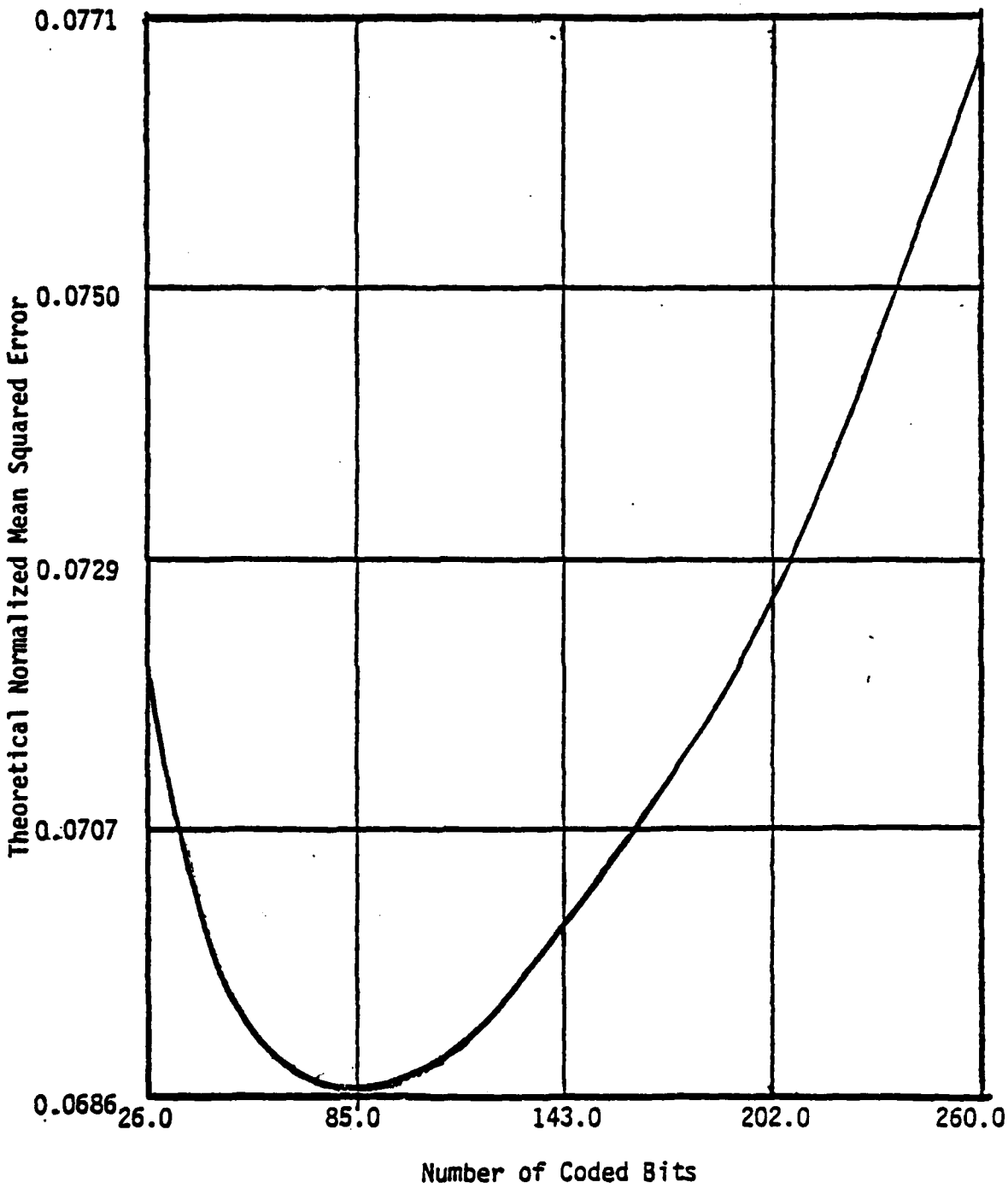


Figure 5.7. Normalized Mean Squared Error Versus Number of Bits Protected by (31,26) Hamming Code for Aerial Image with 10^{-2} Error Rate.

Table 5.2. Since the statistics of the girl and aerial images are different, their bits are ranked differently. As can be seen from Table 5.2, the (7, 4) code yields the best performance and the optimal number of bits to code for the girl image is forty-four, while for the aerial image the optimal number of bits is forty. Since the NMSE does not vary much over this range, it was decided to protect forty-four bits using the (7, 4) Hamming code.

Since the total transmitted data rate is fixed at 1 bit/pixel, using 33 bits per block for channel coding reduces the number of bits available for source coding to 223. It is important to ascertain how much degradation in image quality is imposed by this reduction of source coding bits when the channel is error-free. Figure 5.8(a) shows the original girl image using 8 bits/pixel. Figure 5.8(b) is the reconstructed image at 1 bit/pixel with no bits allocated to channel coding and a zero BER, while Fig. 5.8(c) is the compressed image at 1 bit/pixel with 33 bits per block allocated to channel coding. As is evident, Figs. 5.8(b) and 5.8(c) do not differ substantially, and hence, allocating bits to channel coding does not seriously reduce error-free system performance at 1 bit/pixel. Figure 5.9 illustrates the same behavior for the aerial image.

In Figs. 5.10 and 5.11, the receiver output signal-to-noise ratio (SNR), the inverse of NMSE, is plotted versus BER for systems with and without channel coding and for both images. At the chosen design error rate of 10^{-2}, the channel error protection provides an improvement over the no channel coding system of 5.18 dB for the girl image and 2.5 dB for the aerial image. As is expected, as the BER gets small, allocating bits to channel coding reduces the output SNR.

TABLE 5.2. NO. OF BITS CODED THAT ACHIEVES MINIMUM NMSE

Girl Image

Code rate	Error rate	No. bits coded	NMSE
4/7	10^{-2}	44	2.273×10^{-2}
11/15	10^{-2}	66	3.036×10^{-2}
26/31	10^{-2}	78	4.544×10^{-2}

Aerial Image

Code rate	Error rate	No. bits coded	NMSE
4/7	10^{-2}	40	5.850×10^{-2}
11/15	10^{-2}	55	5.997×10^{-2}
26/31	10^{-2}	78	6.869×10^{-2}

Original Girl Image

(a)

Compressed Image without
Channel Coding

(b)

Compressed Image with
Channel Coding

(c)

Figure 5.8. Original Girl Image and Data Compressed Reconstructed Images with and without Channel Coding (P_e =0)

Original Aerial Image

(a)

Compressed Image without Channel Coding

(b)

Compressed Image with Channel Coding

(c)

Figure 5.9. Original Aerial Image and Data Compressed Reconstructed Images with and without Channel Coding (P_e=0).

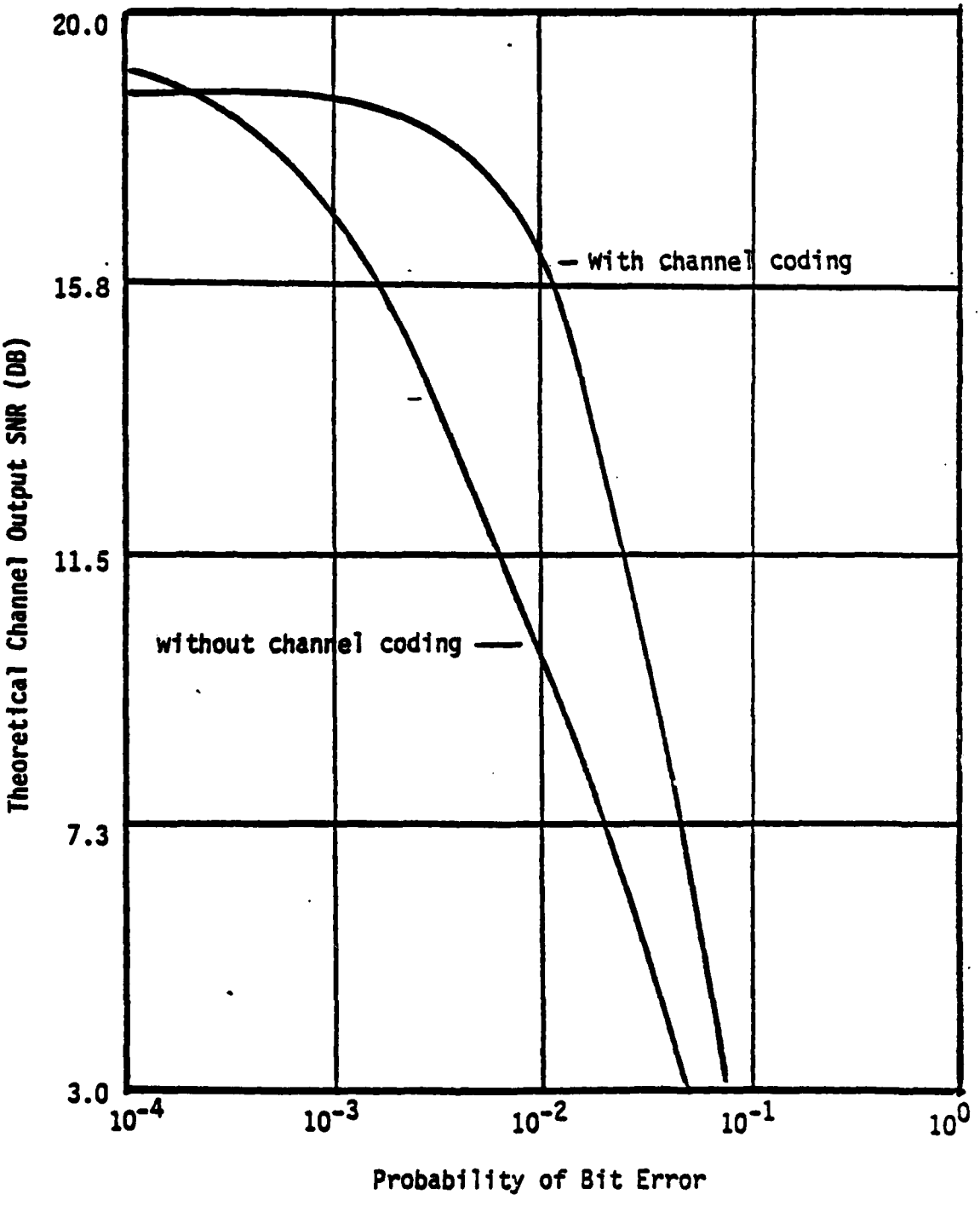


Figure 5.10. SNR vs. Prob. of Bit Error for Girl Image with 44 Bits Protected per Block by (7,4) Hamming Code and without Channel Coding

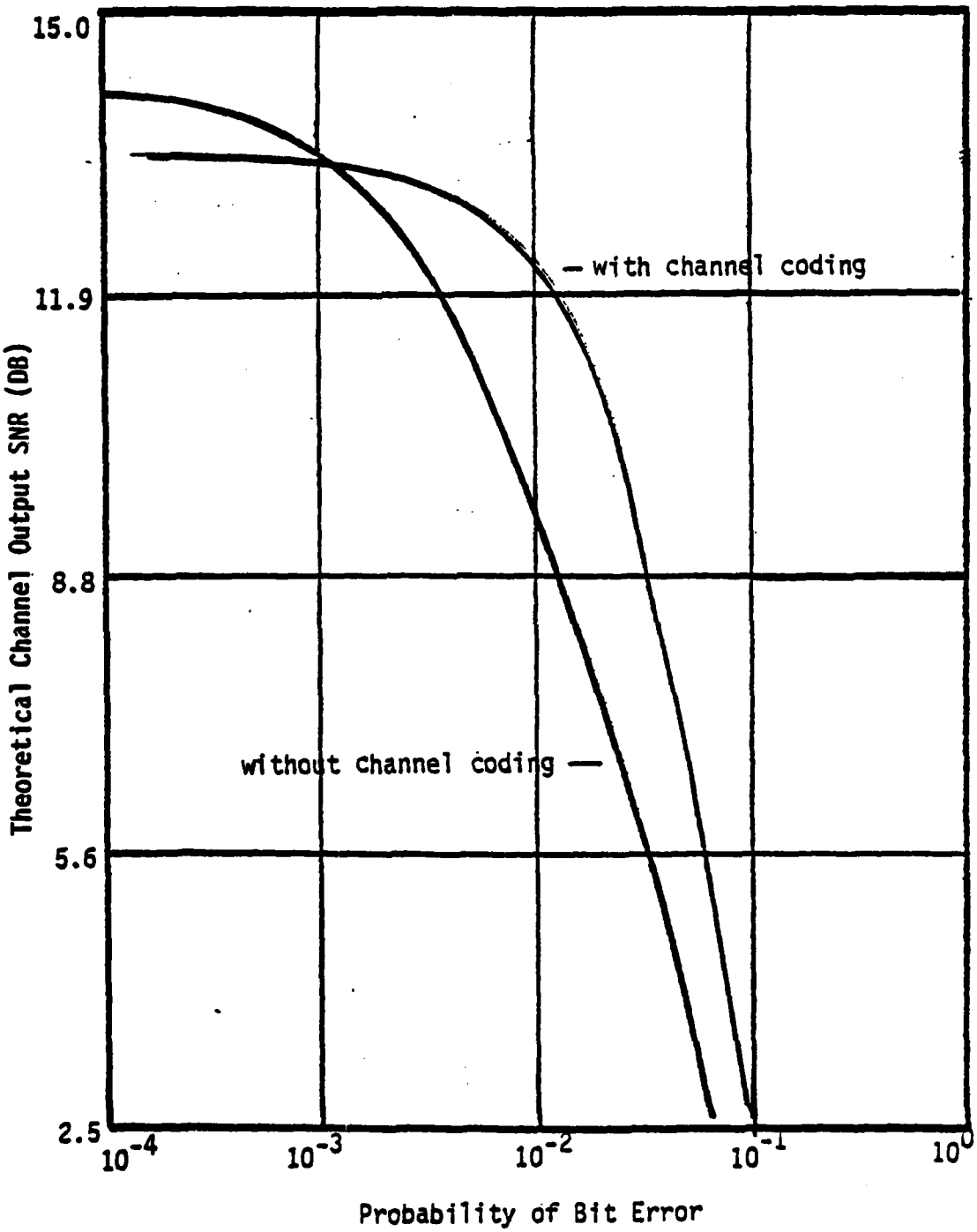


Figure 5.11. SNR vs. Prob. of Bit Error for Aerial Image with 44 Bits Protected per Block by (7,4) Hamming Code and without Channel Coding

4.0. SIMULATION RESULTS

Monte Carlo simulation results were obtained for each BER of interest by processing each image 25 times with a different random error sequence for each run. The average output SNR over each set of 25 runs is then computed for system evaluation. Simulations were performed both for systems with channel coding and without channel coding at BER's of 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1}. The simulation results are plotted in Figs. 5.12 and 5.13 for the girl and aerial images, respectively. At the design error rate of 10^{-2}, channel coding provides a 3.2 dB advantage for the girl image and a 1.7 dB advantage for the aerial image.

Figures 5.14 and 5.15 compare simulation and theoretical results for the system with channel coding. For BER $\leq 10^{-2}$ on the girl image, theory and simulation show a substantial disagreement. For the aerial image, simulation results and the theory are in better agreement. The reason for this discrepancy seems to be the assumption used for the theoretical calculations that the 2D-DCT coefficients are Gaussian. Recent studies [9] indicate that for the girl image the DCT coefficients are more nearly Laplacian, but for the aerial image the DCT coefficients are nearer a Gaussian distribution.

Of course, the final important question is how much channel coding improves the reconstructed image visual quality. To provide an indication of the full range of possible reconstructed images over the many Monte Carlo runs, a subjective selection of the worst and best images was made. Figures 5.16(a) and (b) show the worst and best girl images without channel coding at a BER of 10^{-2}, and Figs. 5.16(c) and (d) show the worst and best images, respectively, for channel coding at a BER of 10^{-2}. Figure 5.17 presents the same results for the aerial image. Clearly, the channel coding scheme proposed here provides a substantial, noticeable improvement in reconstructed image quality.

The visible errors in Fig. 5.16(c) are due to the inducement of more than one channel error in the bits of a codeword, which, since, the (7, 4) Hamming

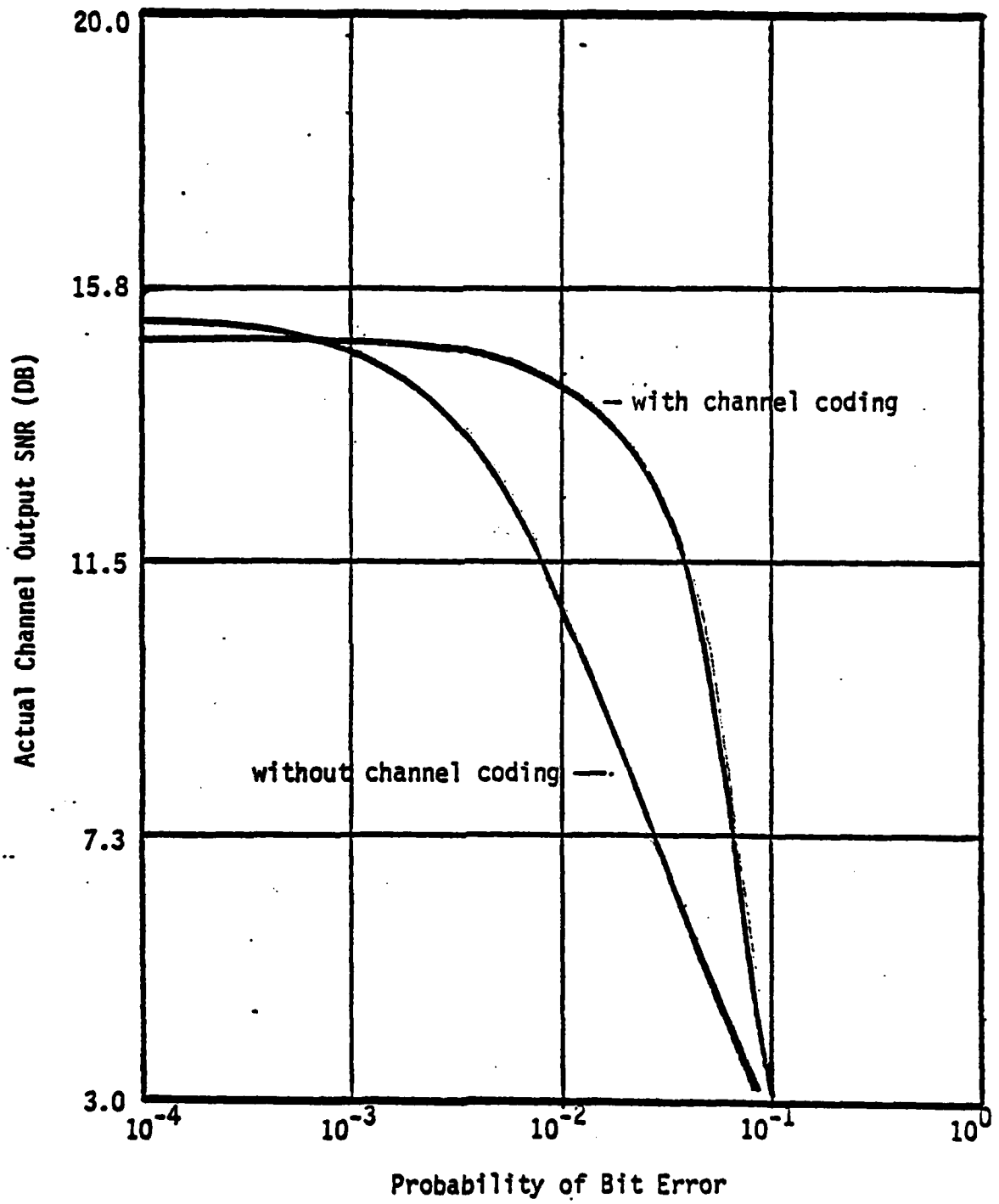


Figure 5.12. Actual SNR vs. Prob. of Bit Error for Girl Image w/ 44 Bits Protected per Block by (7,4) Hamming Code and w/o Channel Coding.

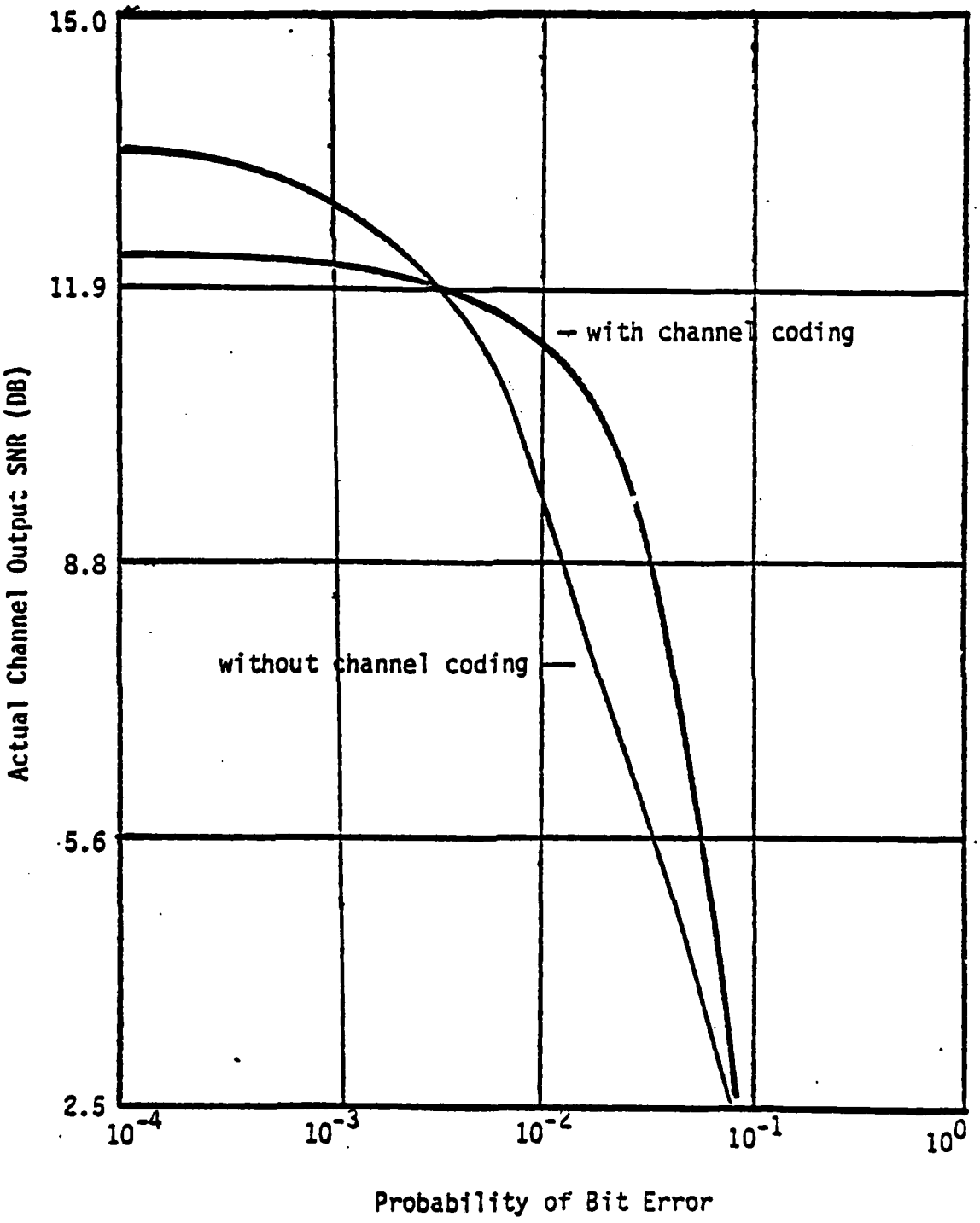


Figure 5.13. Actual SNR vs. Prob. of Bit Error for Aerial Image w/ 44 Bits Protected per Block by (7,4) Hamming Code and w/o Channel Coding.

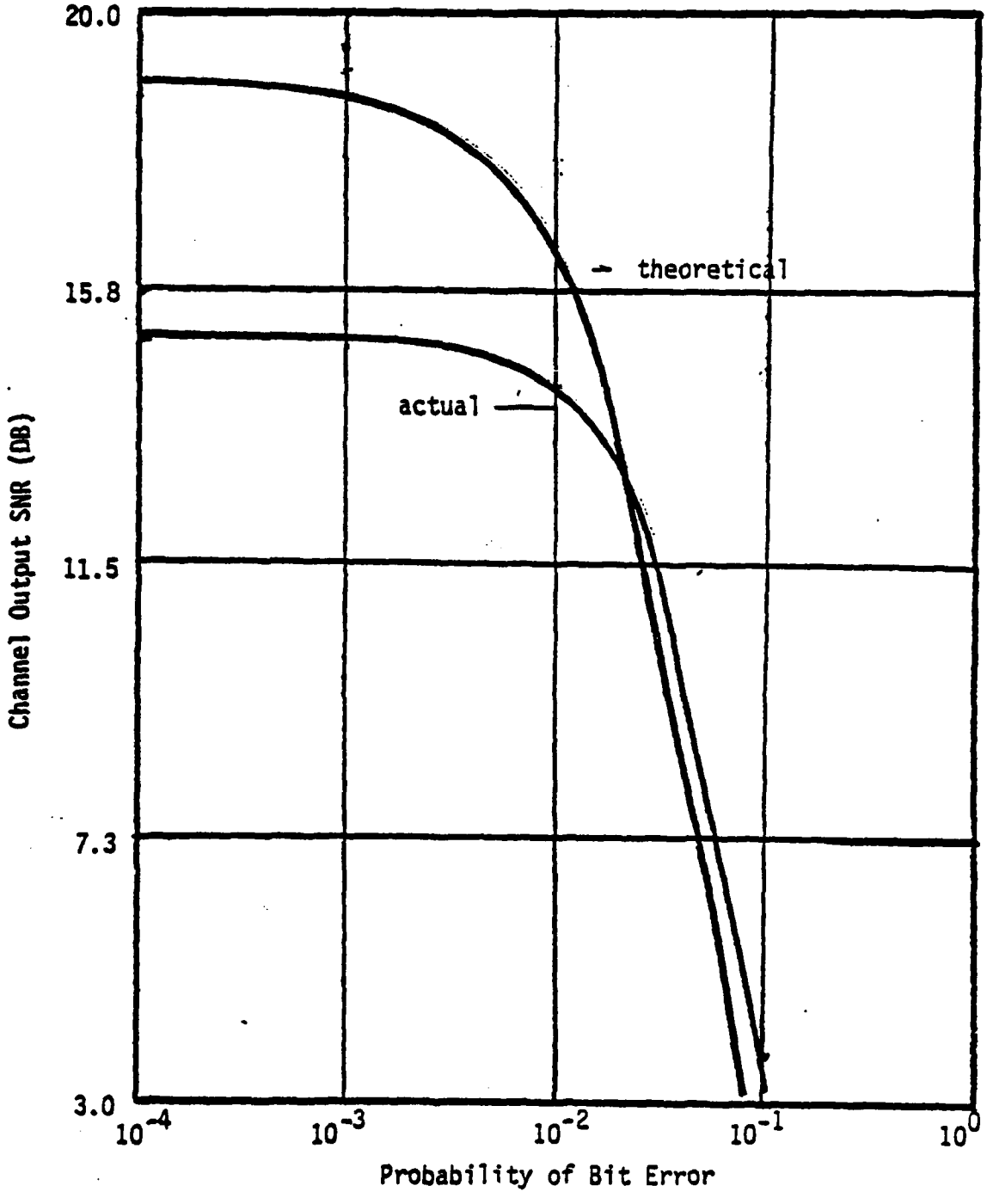


Figure 5.14. Theor. and Actual SNR vs. Prob. of Bit Error for Channel Coded Girl Image.

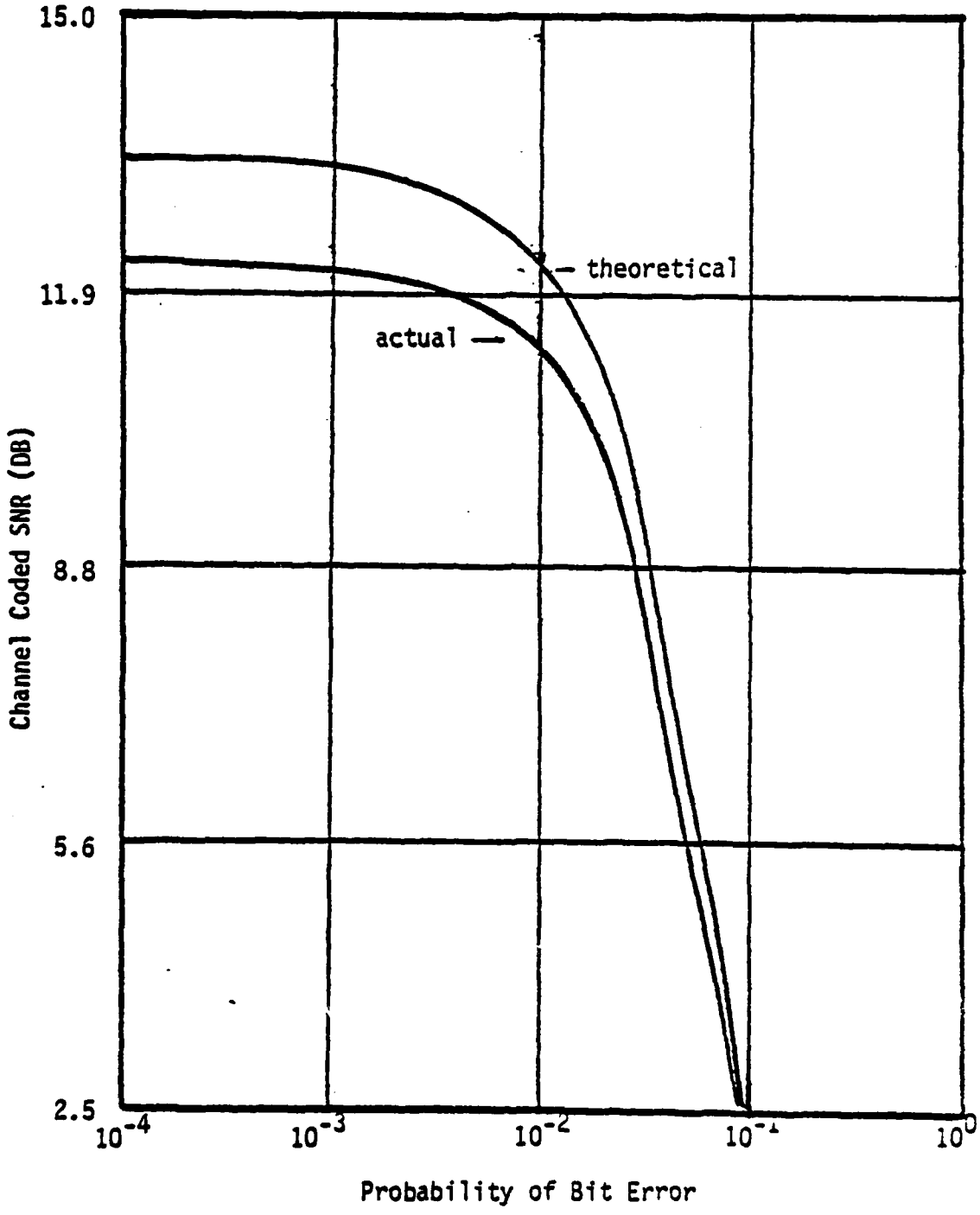


Figure 5.15. Theor. and Actual SNR vs. Prob. of Bit Error for Channel Coded Aerial Image.

Without Channel Coding

(a)

Without Channel Coding

(b)

With Channel Coding

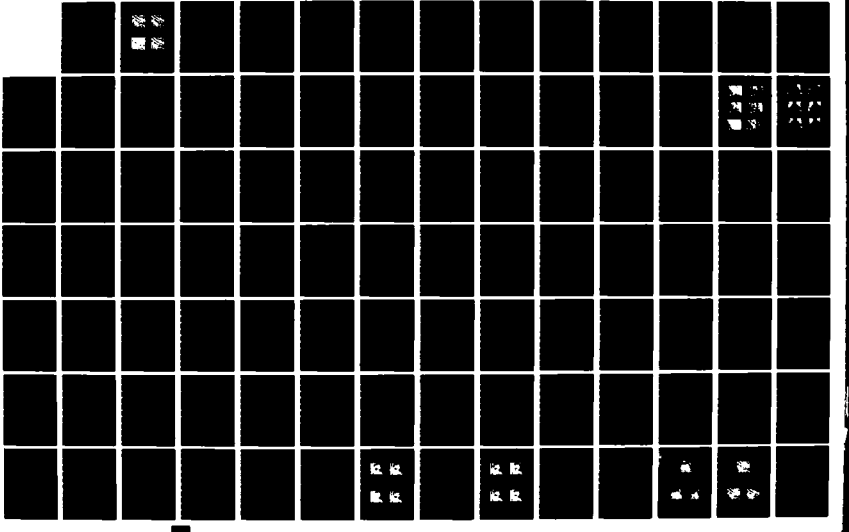
(c) Worst Case

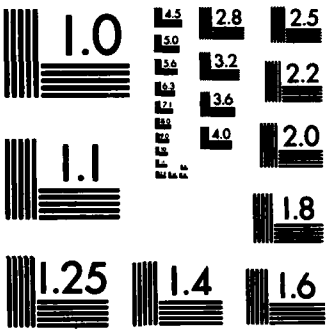
With Channel Coding

(d) Best Case

Figure 5.16. Worst and Best Reconstructed Noisy Girl Images without and with Channel Coding ($P_e=10^{-2}$).

AD-A134 187 BIT WEIGHTING AND SOFT DECISION DEMODULATION FOR IMAGE 2/3
COMPRESSION & ERRO.. (U) TEXAS A AND M UNIV COLLEGE
STATION DEPT OF ELECTRICAL ENGINEE.. J D GIBSON ET AL.
UNCLASSIFIED AUG 83 AFWAL-TR-83-1099 F33615-80-C-1002 F/G 9/2 NL





Without Channel Coding

(a)

Without Channel Coding

(b)

With Channel Coding

With Channel Coding

(d) Best Case

Figure 5.17. Worst and Best Reconstructed Noisy Aerial Images without and with Channel Coding ($P_e=10^{-2}$).

code is single-error-correcting, causes a decoding error. There are many other errors in Figs. 5.16(c) and (d) which are not evident upon visual inspection since these errors occur in the lower energy DCT coefficients.

5.0. CONCLUSIONS

The results presented in this section indicate that using the (7, 4) Hamming code to protect the most important 2D-DCT coefficients can substantially improve reconstructed image quality at a BER of 10^{-2}. A surprising and important result which runs counter to conventional "wisdom" is that the allocation of 33 out of the 256 bits per block to channel coding does not noticeably degrade reconstructed image quality in the absence of channel errors. This fact seems to be due primarily to the property of transform coding systems which "averages" source coding errors over the entire block.

Mean squared error proved to be a useful design criterion even though it is well known that subjective image quality and mean squared error are not always in agreement. Comparisons between theoretical and simulation results indicate that a good estimate of the probability density function of the DCT coefficients is necessary for the theoretical results to be accurate. While the standard Gaussian assumption on the coefficients proved reasonably accurate for the aerial image, the Gaussian assumption produced a large discrepancy between theory and simulation for the girl image.

This work demonstrates that the judicious combination of source and channel coding methods can produce a data compression system which has both the quality and robustness necessary for realistic applications.

Section VI

An Optimized Weighting Algorighm for Variations in PCM Energy Levels

1.0 Introduction

The concept of weighted Pulse Code Modulation (PCM) was first introduced by Bedrosian [1]. Sundberg [2] has derived signal sets for Pulse Code Modulation of speech signals. This application weights the relative signal energy for each PCM symbol in a word by its sensitivity to digital transmission errors. Near optimum performance may be achieved by transmitting groups of PCM symbols at the same energy level where the number of groups is less than the number of bits in the PCM word.

The total energy for each transmitted word remains constant. The digital noise power in an arbitrary PCM system, assuming independent bit errors, may be approximated by (6.1), see [2 - 5].

$$\varepsilon_a^2 \cong P \cdot \sum_{i=1}^{N} A_i \qquad 6.1$$

Digital noise power is the mean square noise associated with making a digital error in bit i with a total of N bits in the PCM word. In this formulation, A_i, is called the A-Factor for a single error in bit i. It represents the noise power averaged over the input signal statistics caused by a single error in PCM symbol i, where i = 1, 2, ..., N. The values of the A-factors vary with input signal densities, the particular PCM code, number of bits per PCM word, and companding law; see [3 - 5]. P is the average bit error probability for a memoryless transmission channel and N

is the number of bits in a PCM word. When transmission is assumed to occur over an additive white Gaussian channel with spectral density N_o (double sided), and the modulation is binary antipodal, P is given by [6] to be:

$$P = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{\frac{E_b}{N_o}}}^{\infty} e^{-\frac{t^2}{2}} d\tau = Q(\sqrt{\frac{E_b}{N_o}}) \tag{6.2}$$

where E_b is the signal energy. For the average channel signal-to-noise ratio, E_b/N_o average, the minimum digital noise is given by

$$\varepsilon_a^2 \cong N \cdot A_o \, Q(\sqrt{\frac{E_b}{N_o}}) \tag{6.3}$$

where the constant A_o is the geometric mean of the single error A-factors [2].

A simpler near optimum performance can be obtained by grouping bits into J groups. Each group of symbols is transmitted at the same energy level, with the same bit error probability. The digital noise is reduced by allowing more energy to be used for the most significant bits of a PCM word (resulting in a smaller bit error probability). This is accomplished at the expense of less energy on the least significant bits (increasing the probability of error on these symbols). The corresponding minimum digital noise of J groups is given by Sundberg [2] to be:

$$\varepsilon_a^2 = N \cdot A_{oJ} \, R(\sqrt{\frac{E}{N_o}}) \tag{6.4}$$

In this expression $R(\cdot)$ is given by

$$R(x) = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{x} e^{-\frac{x^2}{2}} , \tag{6.5}$$

N is the number of bits in a word, and A_{oJ} is the geometric mean realative to J groups. Hence A_{oJ} may be expressed as

$$A_{oJ} = \sqrt[N]{\left(\frac{a_1}{n_1}\right)^{n_1} \cdot \left(\frac{a_2}{n_2}\right)^{n_2} \cdots \left(\frac{a_j}{n_j}\right)^{n_j}} \tag{6.6}$$

where a_j is the sum of the A factors in group j

n_j is the number of bits included in this j^{th} group

N is the total number of bits/word such that $n_1+n_2+\ldots+n_J = N$.

To minimize (6.4) implies minimizing the geometric mean A_{oJ} based upon having derived or being given the A-factors. [2 - 3].

This paper then presents an efficient algorithm for optimizing the number of bits allocated to J energy levels in order that the digital noise is minimal (in the mean square error sense). The bit assignment (which bit is assigned to what energy level) is accomplished solely on A-factors or the bits relevance based on position. The particular relative energy level may then be calculated from the bit assignment, see [2],

2. Dynamic Programming Application to Minimize A_{oJ}

The optimization problem may be stated as allocating the resources available (the PCM symbols) to several relative energy levels while minimizing the digital noise. Dynamic programming is used to develop an algorithm to minimize (6.6).

Let the number of groups J represent n particular stages in the assignment process. So for a two level assignment, two stages may be filled with bits until all bits have been assigned. But depending on the A-factors and the size of the PCM word it may be more advantageous to assign an odd number of symbols to one stage and even number of symbols to the other stage. Three level or three stage assignment will depend on what is optimum for the two stage assignment. The algorithm must therefore be optimum at every stage. It is also reasonable to constrain every stage to having at least one bit; otherwise there would be no reason for having that level at all. We seek, therefore, a function which minimizes A_{oJ}, the geometric mean, to the constraint that each stage must have at least one bit (i.e. zero stages or zero bits to any stage is not allowed). The recursion formula then becomes:

$$A_{oJ} = f_n(s) = \begin{cases} \min\limits_{1 \le d \le s} \{[G_n(d)]^d \cdot [f_{n-1}(s-d)]^{s-d}\}^{1/s} & \text{for } n \ge 1 \\ \min\limits_{1 \le d \le s} \{[G_n(d)]^d\}^{1/s} & \text{for } n = 1 \end{cases} \qquad (6.7)$$

$f_n(s)$ is the minimum A_{oJ} overall possible values of d.

n is the current number of stages or energy levels n = 1, 2, ..., j.

s is the total number of bits to be assigned.

d is a variable number of bits between 1 and s assigned to stage n and is incremented until $f_{n-1}(s-d)$ is $f_{n-1}(1)$ (i.e. d = s-1) since $f_{n-1}(0)$ would represent 0 bits to the (n-1) stage and is not allowed.

s-d is the number of bits remaining to be assigned to stage n-1 after d bits are assigned to stage n.

The functions $G_n(d)$ or $f_{n-1}(s-d)$ are simply the sum of the A-factors for those PCM symbols or bits divided by the number of symbols used at stage n or (n-1). Therefore

$$G_n(d) = \frac{\sum_{i=1}^{d} a_i}{d} \; ; \; f_{n-1} = \frac{\sum_{j=1}^{s-d} a_j}{(s-d)} \tag{6.8}$$

where a_1 represents the A-factor of the least significant bit of those assigned to stage n.

a_2 represents the A-factor for the next significant bit of those assigned to stage n.

⋮

a_d is the A-factor for the most significant bit of those assigned to stage n.

Stage 1 in this procedure (i.e. $(f_1(s))$ will always be the arithmetic average. Since $f_0(s-d)$ cannot exist, by the zero stage constraint, the minimization function is reflected in equation (6.7) for n=1. Now since $N = n_1 + n_2 + \ldots + n_j$ for the geometric mean in (6.6), $f_1(s)$, $n_2 = n_3 = \ldots = n_j = 0$, so $N = n_1 = d = s$. Hence there is really no minimization over d for $f_1(s)$ since d must equal s. However this represents the initialization of the algorithm where the first stage is always the stage containing the least significant bits. An arithmetic average is then calculated for each d between 1 and s. Each $f_1(d)$ is changed only by the addition of a more significant A-factor corresponding to including the next significant bit in the first stage until all bits are utilized.

For a two energy level (2 stage case) example the minimization function is decided over the range of d given $f_1(s)$. Consider a 4 bit PCM word being assigned two different energy levels to minimize A_{oJ}.

$$f_2(4) = \min \text{ of } \begin{cases} \{[G_2(1)]^1 \cdot [f_1(3)]^3\}^{1/4} \\ \{[G_2(2)]^2 \cdot [f_1(2)]^2\}^{1/4} \\ \{[G_2(3)]^3 \cdot [f_1(1)]^1\}^{1/4} \end{cases} \tag{6.9}$$

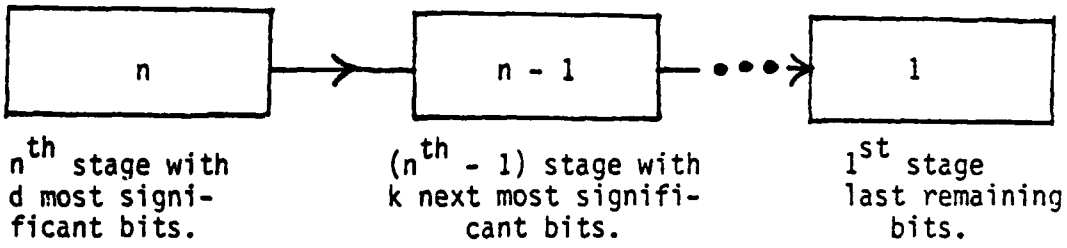
The combinations represented are the most significant bit in stage 2 with the three least significant bits in stage 1; or alternately two bits in stage 2, two bits in stage 1; or third the choice of three most significant bits in stage 2 with the least significant bit in stage 1. The optimum solution consists of the two most significant bits in stage 2, and the two least significant bits in stage 1. For 4 bits and three levels of energy the decomposition yields 1, 2, 1. This represents the most significant bit in stage 3, the two next most significant bits in stage 2, the least significant bit in stage 1.

To see how this result is obtained examine the recursion equation (6.7). If the functional notation is replaced by the appropriate summations we have:

$$A_{oJ} = f_n(s) = \min_{1 \le d \le s} \{[\frac{\sum_{i=1}^{d} a_i}{d}]^d \cdot [\frac{\sum_{j=1}^{s-d} a_j}{(s-d)}]^{s-d}\}^{1/s} \tag{6.10}$$

for n>1 levels.

In general the formulation of equation (6.10) is depicted in Figure (6.1).



Typical Stage Decomposition

Figure 6.1

Having calculated or being given the A-factors*, see [6.2], we initialize the algorithm. Let the A-factors be tabulated in decreasing order:

A_1	weight of most significant bit	8.71606
A_2	weight of next most significant bit	1.99617
A_3	weight of third bit	.62266
A_4	weight of least significant bit	.180237

For n=1:

$$f_1(s) = \min_{1 \le d \le s} \{[\frac{\sum_{i=1}^{d} a_i}{d}]^d\}^{1/s} \qquad (6.11)$$

Therefore, since $f_1(s)$ represents the last s bits remaining to be assigned to the first stage, the first stage being represented as containing the least significant bits, $f_1(s)$ is the arithmetic average of s bits by (6.11).

*Explicit conditions and assumptions for the A-factors are given in Appendix 6.1.

$$f_1(1) = \min_{1 \le d \le 1} \{[\frac{\sum_{i=1}^{1} a_i}{1}]^1\}^1 = A_4$$

$$f_1(2) = \min_{1 \le d \le 2} \{[\frac{\sum_{i=1}^{2} a_i}{2}]^2\}^{1/2} = \frac{A_4+A_3}{2} \qquad (6.12)$$

$$f_1(3) = \cdots\cdots\cdots\cdots = \frac{A_4+A_3+A_2}{3}$$

$$f_1(4) = \cdots\cdots\cdots\cdots = \frac{A_4+A_3+A_2+A_1}{4}$$

Using the aforementioned A-factors we then substitute the calculated functions $f_1(3)$, $f_1(2)$, $f_1(1)$ into equation (6.10). Equation (6.10) may then be used directly to calculate the minimum $f_2(4)$ as given in equation (6.9). Specifically this is:

$$f_2(4) = \text{min of} \begin{cases} \{(8.71606)^1 \cdot (.933024)^3\}^{1/4} \\ \{(5.356117)^2 \cdot (.401450)^2\}^{1/4} \\ \{(3.77829)^3 \cdot (.180237)^1\}^{1/4} \end{cases} \qquad (6.13)$$

which is:

$$f_2(4) = \text{min of} \begin{cases} 1.63117 \\ 1.4664 \\ 1.7657 \end{cases}$$

By decomposing the minimum value (i.e. tracing backward those elements which determined $f_2(4)$ to be minimum) $f_2(4)$ is minimum when $f_2(2)$ and $f_1(2)$ are used. This result corresponds to the claim that 2 stages with 4 bits will have minimum digital noise when the two most significant bits are assigned to stage 2 and the two least significant bits are assigned to the first stage.

Confirmation of the three stage case may be found using the same A-factors. For clarification we shall use only the functional notation of equation (6.7). Since we are limited to 4 bits and the constraint that each stage must have at least one bit assigned the following combinations are possible.

$$f_3(4) = \min \text{ of } \begin{cases} \{[G_3(1)]^1 \cdot [f_2(3)]^3\}^{1/4} \\ \{[G_3(2)]^2 \cdot [f_2(2)]^2\}^{1/4} \end{cases} \qquad (6.14)$$

$$\text{but } f_2(3) = \min \text{ of } \begin{cases} \{[G_2(1)]^1 \cdot [f_1(2)]^2\}^{1/3} \\ \{[G_2(2)]^2 \cdot [f_1(1)]^1\}^{1/3} \end{cases}$$

$$\text{and } f_2(2) = \min \text{ of } \{[G_2(1)]^1 \cdot [f_1(1)]^1\}^{1/2}$$

We notice however that $f_1(1)$, $f_1(2)$ were already calculated. Intermediate levels of $f_2(2)$ and $f_2(3)$ corresponding to two stages with the two least significant bits and two stages with the three least significant bits must be calculated. The respective minimums may then be substituted into (6.14) enabling calculation of a minimun for $f_3(4)$, three stages and four bits. Numerically (6.14) becomes:

$$f_3(4) = \min \text{ of } \begin{cases} \{(8.7160616)^1 \cdot (.6760842)^3\}^{1/4} \\ \{(5.35611755)^2 \cdot (.3349929)^2\}^{1/4} \end{cases} \qquad (6.15)$$

Again decomposition of (6.14) yields the minimum when $f_3(1)$ and $f_2(3)$ are used. $f_2(3)$ was minimum when the first stage contained the least significant bit and the second stage contains the next two significant bits. Therefore $f_3(4)$ corresponds to $f_3(1)$, $f_2(2)$ and $f_2(1)$ respectively. This confirms the placement of bits in three energy levels as 1, 2, 1.

This recursive optimization conforms to the optimal policy that whatever the initial states or decisions are, the remaining decisions also constitute an optimal policy with regard to the states resulting from the previous decisions. This allows rapid decomposition in computing while calculating only two stages at a time. That is, for a large PCM word and several energy levels stage n is considered relative to the optimization of stage (n-1). Stage (n-1) may be decomposed into the two stages that made that decision optimal and stage (n-2) may also be decomposed. The algorithm then becomes general and computationally efficient when more than one energy level is needed and the PCM symbols represent a large word. Figure 2 depicts the algorithm decomposition for 3 levels of energy with an 8 bit PCM word. The calculations are slightly more involved but follow the same process as with 4 bits. Returning to the functional form of equation (6.7) we seek the minimum of $f_3(s)$ where s is 8 bits. By including our constraints for each stage:

$$f_3(8) = \min \text{ of } \begin{cases} \{[G_3(1)]^1 \cdot [f_2(7)]^7\}^{1/8} \\ \{[G_3(2)]^2 \cdot [f_2(6)]^6\}^{1/8} \\ \{[G_3(3)]^3 \cdot [f_2(5)]^5\}^{1/8} \\ \{[G_3(4)]^4 \cdot [f_2(4)]^4\}^{1/8} \\ \{[G_3(5)]^5 \cdot [f_2(3)]^3\}^{1/8} \\ \{[G_3(6)]^6 \cdot [f_2(2)]^2\}^{1/8} \end{cases}$$

$G_3(i)$ where i = 1 to 6 is readily calculated. Functions $f_2(2)$ through $f_2(7)$ are intermediate levels and must be calculated but each must be a minimum for all possible two stage combinations. The A-factor weights for this example are tabulated in Appendix 6.1.

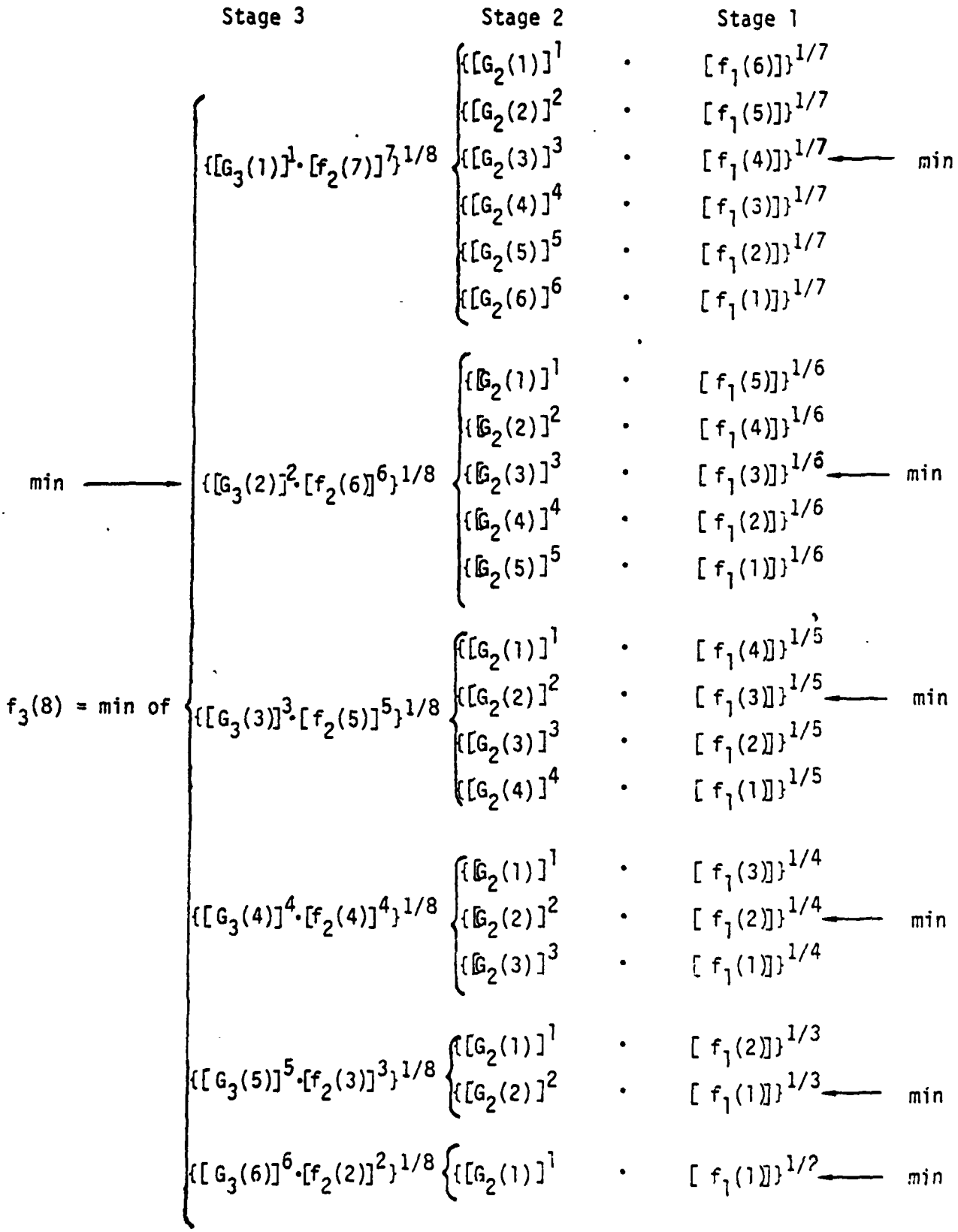


Figure 6.2. Three stage decomposition for an 8 bit PCM word

As seen by figure (6.2), a minimum for 3 levels and 8 bits occurs for 2 bits (most significant) residing in the third stage while 6 bits are allocated to the n-1 (2) remaining stages. From calculation of intermediate levels the minimum digital noise is achieved when the 3 least significant bits are grouped in stage 1 and the three next most significant bits are allocated in stage 2.

Application of this technique to minimize equation (6.6) has been accomplished for image processing. In this application, the transmitted words have variable bit assignments and this algorithm efficiently calculates the number of bits to be assigned to any number of levels. Table 6.1 demonstrates the versatility by listing several PCM word lengths and the desired number of energy levels (stages) for which bits are assigned. Note that the value of A_{oJ} is independent of the (E_b/N_o) relationship depending only on the A-factors in each group level. The A-factors however are dependent on signal densities, coding technique, modulation, etc. For methods of deriving the A-factors see [2-5].

3.0. Image Processing Example of Dynamic Programming Selection of Minimal A_{oJ} and Resulting Performance

The chosen application for this technique was that of bandwidth compression of images. The concept includes the partitioning of an N x N pixel (data point) image into smaller blocks of M x M pixels where N is an integer multiple of M. These smaller blocks are sequentially mapped to a frequency or sequency space. The transformed coefficients are then coded to minimize the mean squared error. These techniques have been thoroughly documented and are independent of this investigation, see [7 - 9].

Table 6.1
Output of Dynamic Programming
for
Minimization of A_{oj}

Number of bits	Number of Levels	Bits/Level MSB→LSB	A_{oj} Minimum
14	2	7 7	0.1497×10^{-1}
13	2	6 7	0.2315×10^{-1}
12	2	6 6	0.3493×10^{-1}
11	2	5 6	0.5481×10^{-1}
10	2	5 5	0.8378×10^{-1}
9	2	4 5	$0.1342 \times 10^{+0}$
8	2	4 4	$0.2088 \times 10^{+0}$
7	2	3 4	$0.3443 \times 10^{+0}$
6	2	3 3	$0.5510 \times 10^{+0}$
5	2	2 3	$0.1124 \times 10^{+1}$
4	2	2 2	$0.1466 \times 10^{+1}$
3	2	1 2	$0.1942 \times 10^{+1}$
2	2	1 1	$0.2357 \times 10^{+1}$
14	3	4 5 5	0.4361×10^{-2}
13	3	4 5 4	0.7500×10^{-2}
12	3	4 4 4	0.1258×10^{-1}
11	3	3 4 4	0.2216×10^{-1}
10	3	3 3 4	0.3890×10^{-1}
9	3	3 3 3	0.6634×10^{-1}
8	3	2 3 3	$0.1199 \times 10^{+0}$
7	3	2 3 2	$0.2175 \times 10^{+0}$
6	3	2 2 2	$0.3801 \times 10^{+0}$
5	3	1 2 2	$0.8992 \times 10^{+0}$
4	3	1 2 1	$0.1281 \times 10^{+1}$
3	3	1 1 1	$0.1721 \times 10^{+1}$
14	4	3 3 4 4	0.2562×10^{-2}
13	4	3 3 3 4	0.4606×10^{-2}
12	4	3 3 3 3	0.8138×10^{-2}
11	4	2 3 3 3	0.1510×10^{-1}
10	4	2 3 3 2	0.2319×10^{-1}
9	4	2 2 2 3	0.5181×10^{-1}
8	4	2 2 2 2	0.9331×10^{-1}

Table 6.1 - Output of Dynamic Programming for Minimization of A_{oj}

Number of bits	Number of Levels	Bits/Level MSB→LSB	A_{oj} Minimum
7	4	1 2 2 2	$0.1776 \times 10^{+0}$
6	4	1 1 2 2	$0.3421 \times 10^{+0}$
5	4	1 2 1 1	$0.8369 \times 10^{+0}$
4	4	1 1 1 1	$0.1182 \times 10^{+1}$
14	5	2 3 3 3 3	0.1895×10^{-2}
13	5	2 3 3 3 2	0.3595×10^{-2}
12	5	2 2 3 3 2	0.6761×10^{-2}
11	5	2 3 2 2 2	0.1257×10^{-1}
10	5	2 2 2 2 2	0.2304×10^{-1}
9	5	1 2 2 2 2	0.4426×10^{-1}
8	5	1 2 2 2 1	0.8615×10^{-1}
7	5	1 2 2 1 1	$0.1667 \times 10^{+0}$
6	5	1 1 1 1 2	$0.3176 \times 10^{+0}$
5	5	1 1 1 1 1	$0.7899 \times 10^{+0}$

This coding, however, usually results in a variable number of bits being assigned to different coefficients. Thus a new dimension to the optimization problem of assigning bits to variable energy levels as proposed by Sundberg [2] must be achieved. The efficient algorithm proposed here was utilized to determined the optimum allocation of bits for a given number of energy levels.

Two original test images were each compressed in dimensionality from 8 bits/pixel to an average bit rate of 2 bits/pixel and 1 bit/pixel. Standard dimensionality reduction techniques using a cosine transform were used, see [10]. Binary Symmetric transmission over an additive white Gaussian noise channel utilizing Monte Carlo cimulations was performed at a bit error rate of 10^{-2}.

Performance evaluation was made by absolute error (ABSE), (6.17), peak signal-to-noise ratio (PSNR), (6.18) and visual inspection as criteria, see [6.9].

$$\text{ABSE} = \frac{1}{(256)^2} \left| \sum_{i=1}^{N} \sum_{J=1}^{N} (x_{iJ} - \hat{x}_{iJ}) \right| \tag{6.17}$$

$$\text{PSNR} = -10 \log_{10} \frac{\left[\frac{1}{N^2} \sum_{i=1}^{N} \sum_{J=1}^{N} (x_{iJ} - x_{iJ})^2\right]}{(256)^2} \tag{6.18}$$

where x_{iJ} are the pixels of the original image and $\hat{x}_{iJ}$ are the reconstructed pixels.

The single error A-factor used were those given in Appendix I for a folded binary code. Since each set of transformed coefficients exhibits a variable length code it was arbitrarily chosen to select four energy levels for each transformed coefficients being coded with four bits or greater while any code word less than four bits would use the same number of energy levels as bits allocated to that coefficient. Figures [6.3-6.5]

Signal-to-noise ratio as a
Function of E/N (db) for
various word sizes.

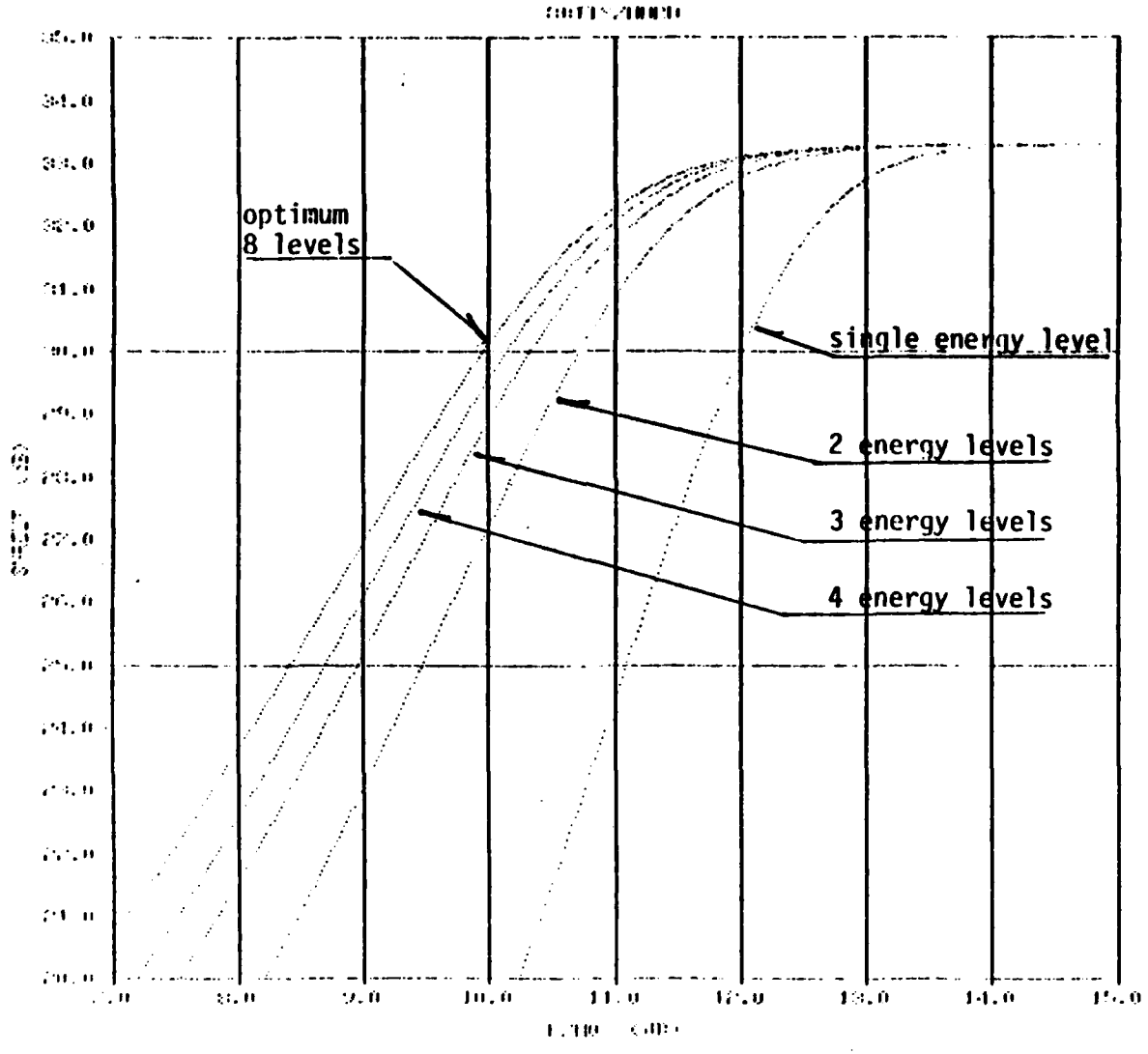


Figure 6.3. Normalized signal-to-noise Ratio output as a function of E/N_o for variable energy levels: 8 bit PCM word.

Signal-to-noise ratio as a
Function of E/N (db) for
various word sizes.

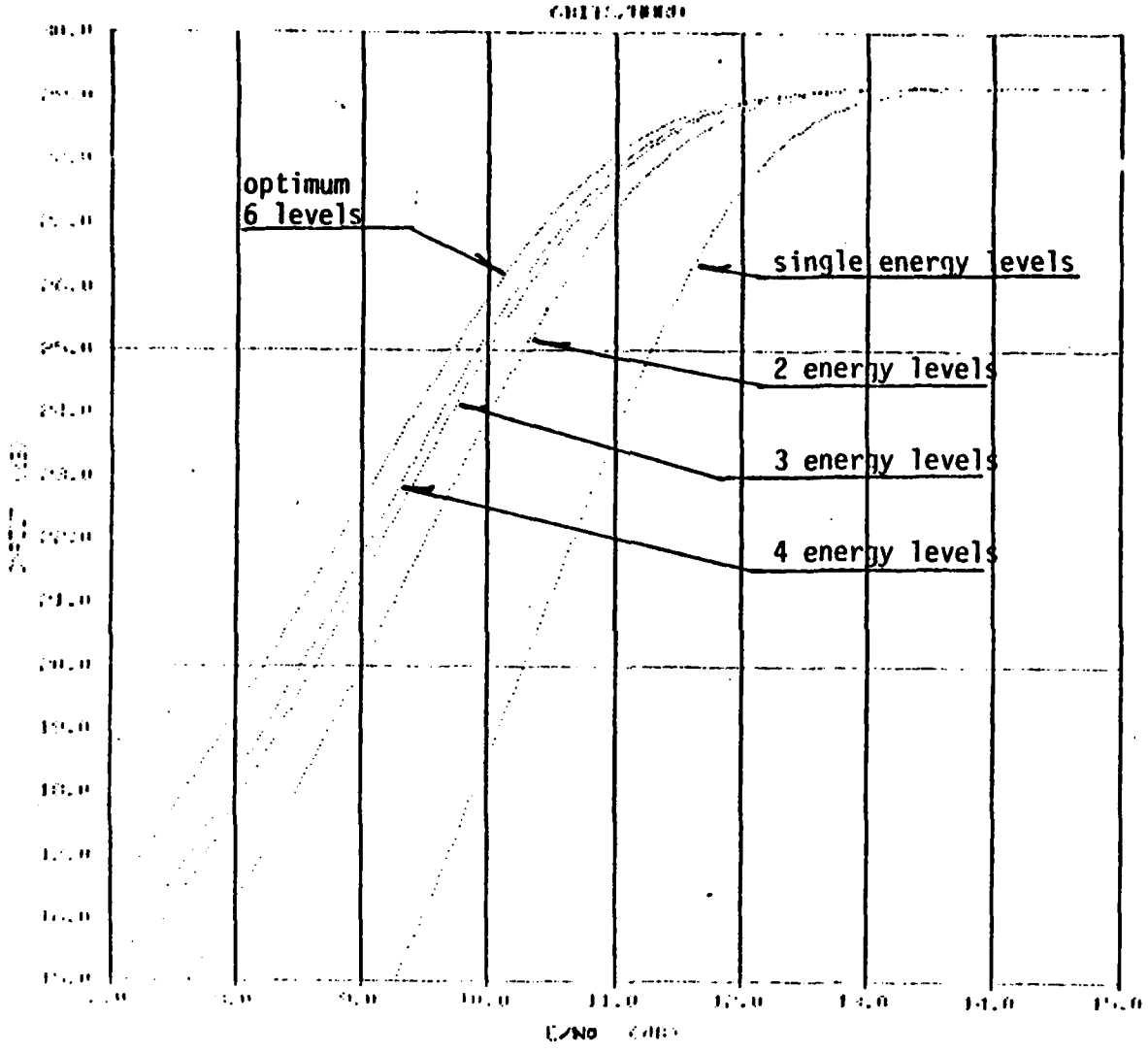


Figure 6.4: Normalized signal-to-noise Ratio output as a function of E/N_o for variable energy levels: 6 bit PCM word.

Signal-to-noise ratio as a
Function of E/N (db) for
various word sizes.

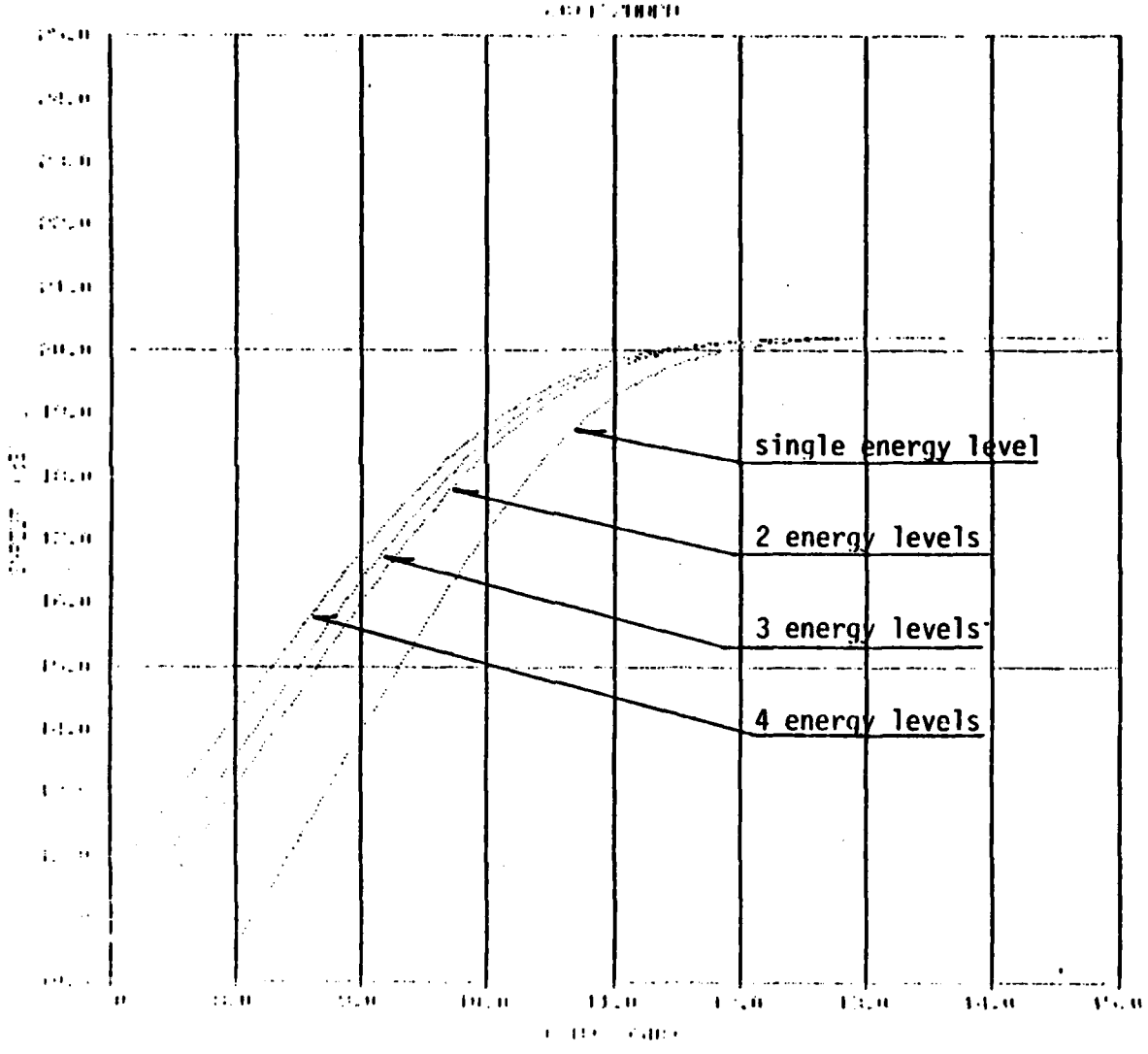


Figure 6.5. Normalized signal-to-noise Ratio output as a function of E/N_o for variable energy levels: 4 bit PCM word.

are examples of the normalized signal-to-noise ratio output as a function of average energy to noise ratio in db for 4, 3, 2 and 1 energy levels. The improvement in signal-to-noise ratio by choosing four levels where possible is apparent and is nearly optimum [2]. The number of bits to be allocated at each energy level or stage was determined by the previously defined algorithm. The relative energy for these bits or groups of bits at a particular level is given by Sundberg to be (6.19):

$$\sqrt{e_i} = \frac{1}{\sqrt{\frac{E}{N_o}}} \; R^{-1}\left(\frac{n_J}{a_J} \; A_{oJ} \; R\left(\sqrt{\frac{E}{N_o}}\right)\right) \qquad (6.19)$$

where n_J & a_J are the number of bits in the J^{th} level and a_J is the sum of these respective A-factors. E is defined as the average energy and $E_i = e_i E$ is the energy of the i^{th} level.

Table [6.2] gives the numerical results obtained in this simulation. It is noted that both ABSE and PSNR are improved utilizing bits allocated to multiple stages or energy levels by this algorithm. As predicted more errors occurred at lower relative energy levels or on bits placed in the first stage while fewer errors occur in the most significant stages (higher energy levels). The total number of errors was actually higher for the image in which multiple energy levels were used but did not hinder performance because these errors occurred for less significant bits.

A comparison of the variable energy encoding and single energy encoding for binary antipodal modulation may be made by the signal to noise ratio (SNR) required for a 10^{-2} bit error rate (BER). The SNR for the single energy encoding is 7.347 db, assuming a spectral density of N_o (double sided). The variable energy encoding distributes the energy relative to the average level resulting in each energy level having its own BER. The

TABLE 6.2

NUMERICAL PERFORMANCE OF TEST IMAGES

		Single Energy Level					Multiple Energy Levels				
bit rate	Image	Errors Stage 1	Errors Stage 4	Total Errors	PSNR (db)	ABSE	Errors Stage 1	Errors Stage 4	Total Errors	PSNR (db)	ABSE
1 bit/pixel	moon	378	39	668	26.56	7.43	1020	5	1212	29.59	5.76
2 bit/pixel	moon	661	70	1305	28.38	5.85	2089	4	2571	31.64	4.34
1 bit/pixel	xray	428	31	668	21.59	14.44	873	2	1020	23.41	12.28
2 bit/pixel	xray	649	45	1305	23.85	10.64	1778	2	2212	26.12	8.74

average SNR required for the variable energy encoding was 6.95 db, 6.22 db, 4.57 db, for 4 bit, 6 bit, and 8 bit words respectively. These average SNR's correspond to 10^{-2} BER for the single energy level case. Table 6.3 summarizes the relative energy, the BER and corresponding SNR for 4, 6, 8 bit words.

Perceptual improvement is demonstrated in Figure [6.6 - 6.7]. The high bit error rate of 10^{-2} significantly affects compressed images. This agrees with the intuitive concept of the less redundancy there is, the higher the probability of error. However, by the judicious placement of bits to multiple stages under the constraint of minimizing digital noise power, visual perception is improved for high error rates.

IV. Conclusion

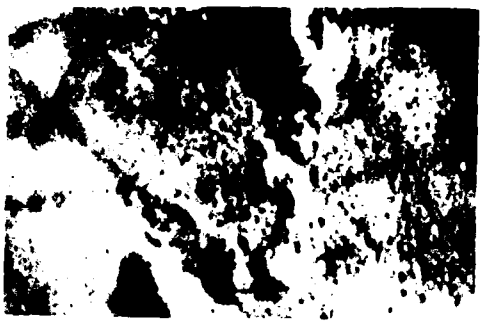
A recursive optimzation algorithm for minimizing digital noise power at various energy levels has been formulated. The application of this Technique to speech and image processing affords rapid efficient variational energy coding with variable bit assignments for each PCM word at any number of energy levels. The perceptual and quantitative improvements are demonstrated by a bandwidth compression example where the transmitted image is subjected to a high bit error rate in an additive white Gaussian noise channel. This method therefore provides a robust approach for transmission of coded images without increasing the transmission rate for a non-ideal channel.

TABLE 6.3

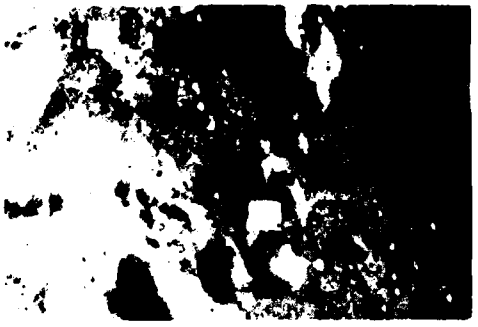
COMPARISON OF SINGLE AND VARIABLE ENERGY LEVELS
IN TERMS OF SNR AND BER. (4 LEVEL PCM)

Bits/word		Level 1	Level 2	Level 3	Level 4
4	Relative Energy	.451332	.803351	1.1655	1.6468
	SNR (db)	3.892	6.396	8.012	9.514
	BER	.59142E-01	.186E-01	.6034E-02	.1423E-02
	Equivalent SNR for single energy 10^{-2} BER 6.95 db.				
6	Relative Energy	.24427	1.00848	1.6088	2.1725
	SNR (db)	1.23	7.3837	9.4122	10.716
	BER	.125E+00	.9774E-02	.159E-02	.305E-03
	Equivalent SNR for single energy 10^{-2} BER 6.22 db.				
8	Relative Energy	.03929	.569041	1.4252	2.439
	SNR (db)	-6.7099	4.898	8.89	11.219
	BER	.3224	.397E-01	.275E-02	.14119E-03
	Equivalent SNR for single energy 10^{-2} BER 4.57 db.				

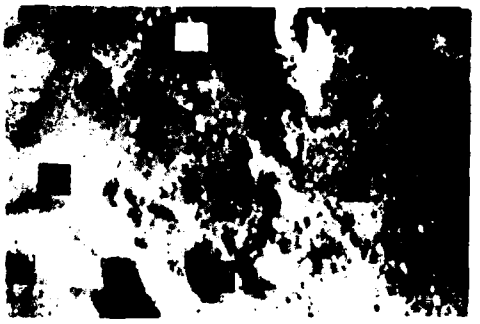
Reconstructed transform Coded Image
average rate of 2 bits/pixel
ideal channel.



Reconstructed transform Coded Image
average rate of 1 bit/pixel
ideal channel.

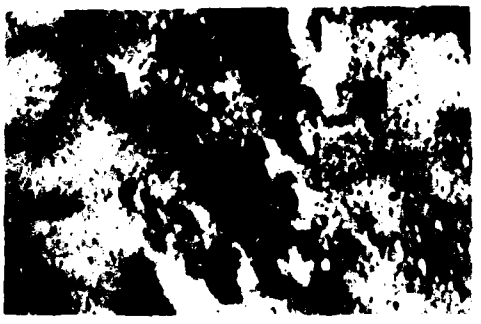


Reconstructed transform Coded Image
at 2 bits/pixel, BSC, AWGN and 10^{-2} BER



Reconstructed transform Coded Image
at 1 bit/pixel, BSC, AWGN and 10^{-2} BER

Reconstructed transform Coded Image
at 2 bits/pixel, BSC, AWGN, 10^{-2} BER
variable energy levels.



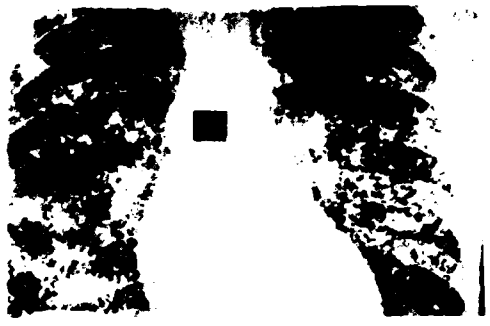
Reconstructed transform Coded Image
at 1 bit/pixel, BSC, AWGN, 10^{-2} BER
variable energy levels.

Reconstructed moon image for 2 bits/pixel and 1 bit/pixel for: a) ideal channel, b) BSC channel with AWGN at 10^{-2} BER and c) variable energy levels with optimum bit allocation.

Figure 6.6

Reconstructed transform Coded Image
average rate of 2 bits/pixel
ideal channel.

Reconstructed transform Coded Image
average rate of 1 bit/pixel
ideal channel.



Reconstructed transform Coded Image
at 2 bits/pixel, BSC, AWGN and 10^{-2} BER

Reconstructed transform Coded Image
at 1 bit/pixel, BSC, AWGN and 10^{-2} BER

Reconstructed transform Coded Image
at 2 bits/pixel, BSC, AWGN, 10^{-2} BER,
variable energy levels.

Reconstructed transform Coded Image
at 1 bit/pixel, BSC, AWGN, 10^{-2} BER,
variable energy levels.

Reconstructed XRay Image for 2 bits/pixel and 1 bit/pixel for: a) ideal channel, b) BSC channel with AWGN at 10^{-2} BER and c) variable energy levels with optimum bit allocation.

Figure 6.7

Section VII

Task IV: Spatial Image Coding for Non-ideal Channels

1.0 INTRODUCTION

The goal of the research presented in this task is to develop an image transmission scheme based on a spatial image coder which will provide good quality images, low bandwidth requirements, and error protection for non-ideal channels. The complexity of the hardware required to realize this coding and transmission scheme is considered, and emphasis is placed on developing a system which requires a relatively simple hardware implementation.

The problem of developing an efficient image transmission system is of concern in fields such as broadcast and relay television, remote image sensing, fascimile transmission, biomedical imaging and surveillance. The efficiency of an image transmission system may be defined in terms of the quality of the reproduced images, the time and channel bandwidth required for transmission, the performance of the system in the presence of channel errors, and the complexity of the hardware required to realize the system.

A general block diagram of an image transmission system is presented in Figure 7.1.

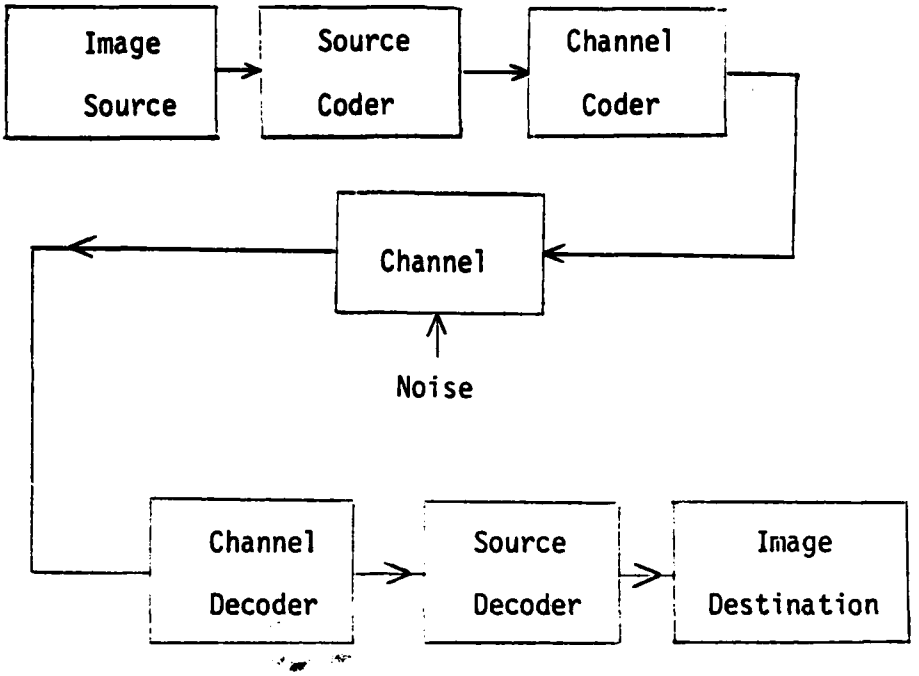


Figure 7.1. General Image Transmission System Block Diagram

The model contains a source of images obtained from some fascimile scanner, television scanner, or other imaging device. For the purposes of this research, it is assumed that the source image is monochrome, and is presented in digital form as an array of discrete points of picture elements or "pixels". Each pixel is coded to 8 bit resolution in the source image, so that each pixel may take on one of 256 different shades of gray.

Given a digitized source image, the source coder transforms the image data into a form with minimal transmission requirements. It is the source coder which is the primary subject of the research described in this paper.

The data output from the source coder is converted by the channel coder to a format suitable for transmission. This step involves modulation of the transmission carrier and possibly the addition of error correcting codes to the source data in order to provide protection against channel errors. The channel decoder and source decoder operate on the received signal, inverting the coding process and producing the reconstructed images. The image destination may consist of an image display or recording device, which often serves to present the data to the human viewer.

As noted above, the majority of the research discussed in this paper is concerned with the development of an image source coder. Most of the image coders developed to date may be classified as belonging to one of the following three categories: frequency domain, spatial domain, and hybrid. In general, frequency domain coders provide a high degree of data compression but require large amounts of storage and complex, high-speed hardware for their implementation. Spatial domain coders often require only small amounts of storage and relatively simple hardware implementations, but result in a lower compression rate for a given level of image quality [1]. Hybrid coders, such as the DPCM scheme discussed by Habibi [2], attempted to achieve a compromise between frequency and spatial domain coders with respect to implementation speed, hardware complexity, and image quality.

When a complete image transmission system is being considered, the image (source) coder must also be evaluated in terms of error susceptibility. Traditionally, the frequency domain coders, also known as transform coders, have proven to be much more robust in the presence of channel noise than

spatial domain or hybrid coders [3]. Thus, the frequency domain coders which provide high compression rates and good performance in the presence of noise are often computationally intensive, while the spatial coders which impose a light computational load suffer from low compression capability and error susceptibility.

A new spatial coding technique has recently been developed by Delp and Mitchell [4] which has the advantages of providing a good match to the human visual system and producing immediate source code with data rates in the 1.5 to 2.0 bits per pixel range. This technique, called Block Truncation Coding (BTC), uses a two-level nonparametric quantizer which adapts to local properties of the image. As is true in general for spatial coding schemes, BTC is computationally simple and requires only one pass through the image. A further advantage of BTC is that it is easily matched to a standard digital modulation method such as Quadrature Amplitude Modulation (QAM).

QAM is an attractive modulation method because of its high theoretical efficiency (4 bits/s/Hz) [5] , [6] and its wide acceptance. As Sundberg [7] has suggested, QAM is also attractive because it may be easily modified through a technique known as bit weighting. This technique matches the probability of transmission errors for a given bit to the relative importance of that bit within a digital word, while keeping the total transmitter energy per word constant. Bit weighting is applicable to the source code produced by BTC and would provide error protection with no increase in bit rate.

The research described in this paper consisted of developing a modified version of BTC though the application of DPCM and an unsupervised learning algorighm. The feasibility of matching this source coder to QAM with bit weighting was demonstrated, and the combined source and

channel coding system was simulated over binary symmetric channels with Gaussian noise. The performance of this system was compared to that of a scheme using standard cosine transform coding with binary antipodal modulation.

The metric used throughout the research for the purpose of evaluating image quality was mean square error (MSE). In the discrete domain, the MSE between an image F(i,j) and its coded reconstruction F'(i,j) is defined to be

$$E_{MSE} = \frac{\sum_{j=1}^{J} \sum_{k=1}^{K} [F(j,k) - F'(j,k)]^2}{J \cdot K} \qquad (7.1)$$

where J and K are the dimensions of the images in pixels. Although MSE is a mathematically tractible metric, Pratt [8] and others have found that MSE sometimes correlates poorly with subjective image evaluations. For this reason, visual results as well as MSE figures will be presented at key points in the report.

2. BLOCK TRUNCATION CODING

Block Truncation Coding (BTC) is a spatial domain coding scheme recently developed at Purdue University [4]. This source coding technique shows the potential for achieving good performance with respect to image quality and excellent efficiency in terms of computational requirements. The algorighm operates on small blocks of the image, usually 4 X 4 pixels. The local block moments are determined, and the mean (first moment) and variance of the block are quantized and coded. A threshold for a one-bit quantizer is determined based on the sample moments, and the individual pixel values are then quantized. Pixel values above the threshold are assigned a code of "1", while values below the threshold are assigned a "0", resulting in a binary bit map for the image block. The block is reconstructed from the bit map and the coded quantities for the mean and variance.

The choice of a quantization scheme is a critical factor in BTC. This choice influences coder performance in terms of image reproduction quality, computational load, and hardware complexity. In order to utilize the classical quantizer design of Max [9], which minimizes mean square error, the probability density function of the data to be quantized must be known or approximated. With respect to BTC, this would require knowledge of the probability density function of the pixel values within each block. This stringent requirement, which also exists for the minimum absolute error quantizer of Kassam [10], led Delp and Mitchell to the choice of a nonparametric quantizer for their coding scheme [11].

Nonparametric quantizers as a general class are those which may be formulated without a priori knowledge concerning the distribution parameters of the data to be quantized. Parametric quantizers incorporate the distribution parameters into the quantizers formulation, and thus require knowledge (or estimates) of these parameters for the realization of the quantizer. Typical distribution paramaters which may be considered on the formulation of parametric quantizers are the mean and variance for a Gaussian distribution, or the variance for a Laplacian distribution.

The use of a nonparametric quantizer elimates the requirement of prior knowledge of the pixel value probability density functions. A nonparametric quantizer may be designed to minimize mean square error (MSE) in a BTC scheme as detailed below; alternatively, the quantizer could be designed to minimize mean absolute error (MAE) or preserve the sample moments.

In general the threshold which defines the boundary between the upper and lower quantizer levels may be either fixed or variable. In the fixed threshold case, the threshold is defined to be some function (the sample mean or nth sample moment, for example) of the input data statistics. In the case of a variable threshold, the threshold is not specified a priori, but is considered as a variable in the quantizer formulation and may be selected along with the quantizer output levels in order to satisfy some metric such as minimum MSE.

A one bit minimum MSE quantizer may be developed for BTC in the following manner [4]. Assuming that the BTC algorithm operates on image blocks of n X n pixels. Them $m=n^2$ pixels will be contained within each block. Given a set of data points (pixel values) $\left[X_1, X_2, \ldots, X_m\right]$, a

threshold X_{TH} may be indirectly defined by picking some number of the data points to lie above the threshold. If the data values are sorted from least to greatest, and if q is defined to be the number of X_i's greater than X_{TH}, the quantizer output levels a and b may be found by minimizing

$$J_{MSE} = \sum_{i=1}^{m-q-1} (X_i-a)^2 + \sum_{i=m-q}^{m} (X_i-b)^2 \qquad (7.2)$$

where

$$a = \frac{1}{m-q} \sum_{i=1}^{m-q-1} X_i \qquad (7.3)$$

$$b = \frac{1}{q} \sum_{i=m=q}^{m} X_i \qquad (7.4)$$

This quantizer may be optimized by solving the first equation for all possible values of q, and then using the value of q which results in the minimum J_{MSE}.

Thus, the implementation of a nonparametric minimum MSE quantizer would require an exhaustive search for the optimum value of q within each n X n image block. If n=4, $m=n^2=16$, and since q may take on m-1 values for the two-level quantizer, the equation for J_{MSE} must be solved 15 times for each block.

The development of a nonparametric minimum MAE quantizer also leads to the need for an exhaustive search of all possible thresholds in order to determine the optimum quantizer for each block. In either the minimum MSE or minimum MAE case, the quantizer formulation is not available in closed form. The lack of a closed form solution leads to a heavy computational demand through the need for exhaustive searches.

As an alternative to minimum MSE or minimum MAE quantizers, Delp and Mitchell [12] investigated the development and performance of a one-bit nonparmetric quantizer based on the fidelity criterion of preserving the sample moments of the input data. Their investigation led to the determination of several desireable properties of such quantizers with respect to image coding. The development of such a quantizer is presented below.

For image blocks of n X n pixels, $m=n^2$ pixel values are to be quantized, each into one of two levels. For the original pixel values $\left[X_1, X_2, \ldots, X_m\right]$, it is desired to preserve the first two sample moments, defined by

$$M_1 = \frac{1}{m} \sum_{i=1}^{m} X_i \tag{7.5}$$

$$M_2 = \frac{1}{m} \sum_{i=1}^{m} X_i^2 \tag{7.6}$$

In general,

$$M_i = E[X^i] \tag{7.7}$$

where $E[\cdot]$ is the expectation operator. Two output levels a and b and a threshold X_{TH} are defined for the quantizer, such that

$$\begin{aligned} &\text{if } X_i > X_{TH} \quad \text{output} = a \\ &\text{if } X_i < X_{TH} \quad \text{output} = b \\ &\text{if } X_i = X_{TH} \quad \text{output} = b \\ &\qquad \text{for } i=1,2,\ldots,m. \end{aligned} \tag{7.8}$$

The threshold X_{TH} may be set equal to the first moment in order to simplify quantizer formulation, or may be evaluated as a variable in order to enhance performance.

For the case in which $X_{TH}=M_1$, if q is defined as the number of X_i's greater than X_{TH}, then to preserve M_1 and M_2,

$$\begin{aligned} mM_1 &= (m - q)a + qb \\ mM_2 &= (m - q)a^2 + qb^2 \end{aligned} \tag{7.9}$$

The above equations may be solved for the quantizer output levels a and b (see [7.4]), with the result that

$$a = M_1 - (M_2 - M_1^2) \quad \frac{q}{m - q} \tag{7.10}$$

$$b = M_1 + (M_2 - M_1^2) \quad \frac{m - q}{q} \tag{7.11}$$

Note that the quantity $M_2 - M_1^2$ is the variance σ^2 of the input data, and thus each block may be represented by the values of M_1, σ^2, and an n X n bit plane consisting of 1's and 0's indicating whether the given pixel value fell above or below X_{TH}.

For the case in which X_{TH} is evaluated as a variable, it is possible to preserve the third sample moment as well as the first and second moments. The third moment is defined as

$$M_3 = \frac{1}{m} \sum_{i=1}^{m} X_i^3. \tag{7.12}$$

The quantizer formulation then involves solving the equations

$$\begin{aligned} mM_1 &= (m - q)a + qb \\ mM_2 &= (m - q)a^2 + qb^2 \\ mM_3 &= (m - q)a^3 + qb^3 \end{aligned} \tag{7.13}$$

for the variables a, b, and q. The value of q obtained, rounded to the nearest integer, defines X_{TH} since it specifies the number of X_i's greater than X_{TH}.

The system (7.8) has the solutions (see [13]):

$$a = M_1 - (M_2 - M_1^2) \frac{q}{m - q} \tag{7.14}$$

$$b = M_1 + (M_2 - M_1^2) \frac{m - q}{q} \tag{7.15}$$

$$q = \frac{m}{2} \cdot \left[1 + A \left(A^2 + 4 \right)^{-1/2} \right], \tag{7.16}$$

where

$$A = \frac{3 \cdot M_1 \cdot M_2 - M_3 - 2 \cdot (M_1)^3}{\sigma^3} \tag{7.17}$$

if σ is not equal to zero. If $\sigma=0$, equations (7.10) and (7.11) imply that $a = b = M_1$.

Note that for either the fixed or the variable threshold case, the image block is represented by the block mean, the standard deviation, and an n X n bit map. If the mean and standard deviation are each assigned 8 bits, the average data rate for either case is 2 bits/pixel.

Note also that for either the fixed or variable threshold case, the moment preserving quantizer formulation is available in closed form. The fact that a closed form solution exists for this class of quantizers greatly reduces the computational load required for implementation. Furthermore, investigations by Mitchell and others [3], [4], [11] have shown that BTC with a moment preserving quantizer performs well in subjective evaluations.

The performance of BTC relative to a high-quality cosine transform coder was considered by Delp and Mitchell [4] and by Goeddel and Bass [1]. These studies compared the performance of BTC to that of the Chen and Smith [14] two-dimensional cosine transform coder at bit rates of 1.63 and 1.875 bits/pixel respectively. The Chen and Smith coder is considered to be among the best of the published frequency domain coders developed to date [1]. Both groups concluded that under error free conditions, BTC did not perform as well as the Chen and Smith coder. Both subjective eveluations by professional photo interpreters and mean square error figures were used in arriving at this conclusion. However, both groups concluded that a high channel error rate (10^{-3} to 10^{-2}) has a greater effect on the transform coding scheme than on BTC.

An example provided from the study sponsored by Delp and Mitchell dealt with an aerial scene coded at an average rate of 1.63 bits/pixel. With no channel errors, the MSE resulting from BTC and Chen and Smith

coding were 84.22 and 67.13 respectively. Under an average bit error rate of 10^{-3}, BTC produced an image with a MSE of 115.09, while the transform coding yielded a MSE of 115.31. The Goeddel and Bass study provided similar results; see [1].

Furthermore, BTC imposes a significantly lighter computational demand than the transform coder. Rough arithmetic counts [1] indicate that the Chen and Smith coder performs three passes through the data, with a total of $5.6 \cdot 10^6$ additions and $0.4 \cdot 10^6$ multiplications. BTC also imposes smaller memory demands, since only n^2 (typically 16) pixels must be stored at a given time, while the cosine transform technique requires that the entire frame of 512^2 pixels be stored.

An additional advantage of BTC is that it provides a good match to the human visual system [4]. This effect is due to the fact that BTC codes the largest intensity changes within a block. If no large changes are present, the most significant small variations are coded. The human visual system is also insensitive to small luminance variations in the presence of large variations [15]. BTC thus takes advantage of the noise masking property of human vision.

Bit Weighting

Weighted pulse code modulation was introduced by Bedrosian [16] in 1958. This technique, later modified and extended by others [17]-[19], is based on the idea that typically the individual bits within a pulse code modulation (PCM) word are of different importance to the

reconstructed signal, since each bit denotes the presence or absence of a different power of the base 2. Weighting PCM redistributes the energy used to trnasmit a PCM word so as to minimize the effect of transmission errors.

For an arbitrary PCM system operating over a memoryless transmission channel with independent single-bit errors of probability P, the digital noise is approximately [19]

$$\varepsilon_a^2 \simeq P \cdot \sum_{i=1}^{n} A_i \tag{7.18}$$

where n is the word length in bits. The terms A_i, where i=1,2,...,n, are called the A-factors for a single error in bit i. Each A-factor A_i represents the average noise power caused by a single error in bit i, where the average is formed over the input signal statistics [19]. The values of the A-factors vary with input signal density, the particular PCM code used, companding law, and a number of bits per PCM word.

Sundberg [7] derived the signal sets for weighting PCM of speech signals utilizing binary antipodal modulation over an additive white Gaussian channel with noise spectral density N_0. For an average signal to noise ratio of E/N_0, the bit error probability is

$$P = Q\left(\sqrt{\frac{E}{N_0}}\right) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{\frac{E}{N_0}}}^{} e^{-t^2/2} dt \tag{7.19}$$

where $Q(\cdot)$ is the standard Q function [20]. The optimum weighting scheme provides n different energy levels for each word of n bits. In this case,

the energy assigned to each bit is matched exactly to the importance of that bit within the digital word. The digital noise for the weighted system becomes

$$\varepsilon_a^2 \simeq \sum_{i=1}^{n} A_i \cdot Q\left(\sqrt{\frac{E_i}{N_o}}\right) \tag{7.20}$$

where E_i is the energy in the binary antipodal signal used to transmit symbol i. The total energy per PCM word is kept unchanged so that

$$n \cdot E = \sum_{i=1}^{n} E_i \tag{7.21}$$

where E is the average energy per bit.

The energy levels E_i to be used in particular application are determined by minimizing (7.18) subject to the energy constraint (7.19).

Suboptimum schemes may be developed in which the number of energy levels is less than the number of bits per word. Such a scheme would have the advantage of reducing the energy switching rate required at the transmitter for a given bit rate. In these cases, n bits are grouped into J groups, where $J < n$. All PCM symbols within a single group are transmitted with equal energy, and thus equal probability of error. The expression for the digital noise for the general suboptimum case is given by Sundberg [7] to be

$$\varepsilon_a^2 = n \cdot A_{0J} \cdot R\left(\sqrt{\frac{E}{N_o}}\right) \tag{7.22}$$

where

$$R(x) = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{x} \cdot e^{-x^2/2} , \tag{7.23}$$

n is the number of bits per word, and A_{0J} is the geometric mean of the particular grouping of the A-factors into J groups. Thus A_{0J} may be expressed as

$$A_{0J} = \sqrt[n]{\left(\frac{a_1}{n_1}\right)^{n_1} \cdot \left(\frac{a_2}{n_2}\right)^{n_2} \cdots \left(\frac{a_J}{n_J}\right)^{n_J}} \tag{7.24}$$

where a_j is the sum of the A-factors in group j

n_j is the number of bits in the j^{th} group

n is the total number of bits per word.

The groupings of the a_i's within the J groups must be optimized to yield a minimum value of A_{0J} in order to obtain the minimum value of the digital noise.

The relative energy levels for this scheme are given by

$$\sqrt{e_i} = \frac{1}{\sqrt{\frac{E}{N_0}}} \cdot R^{-1}\left[\frac{A_{0J} \cdot n_J}{a_j} \cdot R\left(\sqrt{\frac{E}{N_0}}\right)\right] \tag{7.25}$$

where a_j, n_j, and A_{0J} are defined in (7.22) and $R(\cdot)$ is defined in (7.21). The actual energy levels may be found from the relative energy levels through the relation

$$E_i = e_i \cdot E \tag{7.26}$$

where E is the average bit energy.

For a given set of A-factors, the digital noise ε_a^2 may be plotted as a function of the number of allowed energy levels J, and the average signal to noise ratio E/N_0. Figure 7.2 (page 127) provides an example of

this type of plot. In this case, the A-factors given by Sundberg [7] for 8 bit folded binary code with μ-law companding (μ = 100) have been used, and the resulting digital noise is plotted for suboptimum schemes J=2,3,4, and for the unweighted case (J=1) and optimum (J=8) cases.

Note that the use of two levels provides a significant reduction in the digital noise when compared to the unweighted case. The beneficial effect of increasing J is reduced for larger values of J, with the result that for an 8 bit word, 4 energy levels provides performance just slightly below the optimum 8 level case.

The above considerations have been concerned with binary antipodal modulation. The concept of bit weighting may be applied to a wide variety of modulation methods, however. Sundberg [7] has outlined the application of bit weighting to 16 level Quadrature Amplitude Modulation (QAM). This application will be considered in detail in Chapter IV, where a weighted QAM scheme is derived for the modified BTC technique developed in Chapter III.

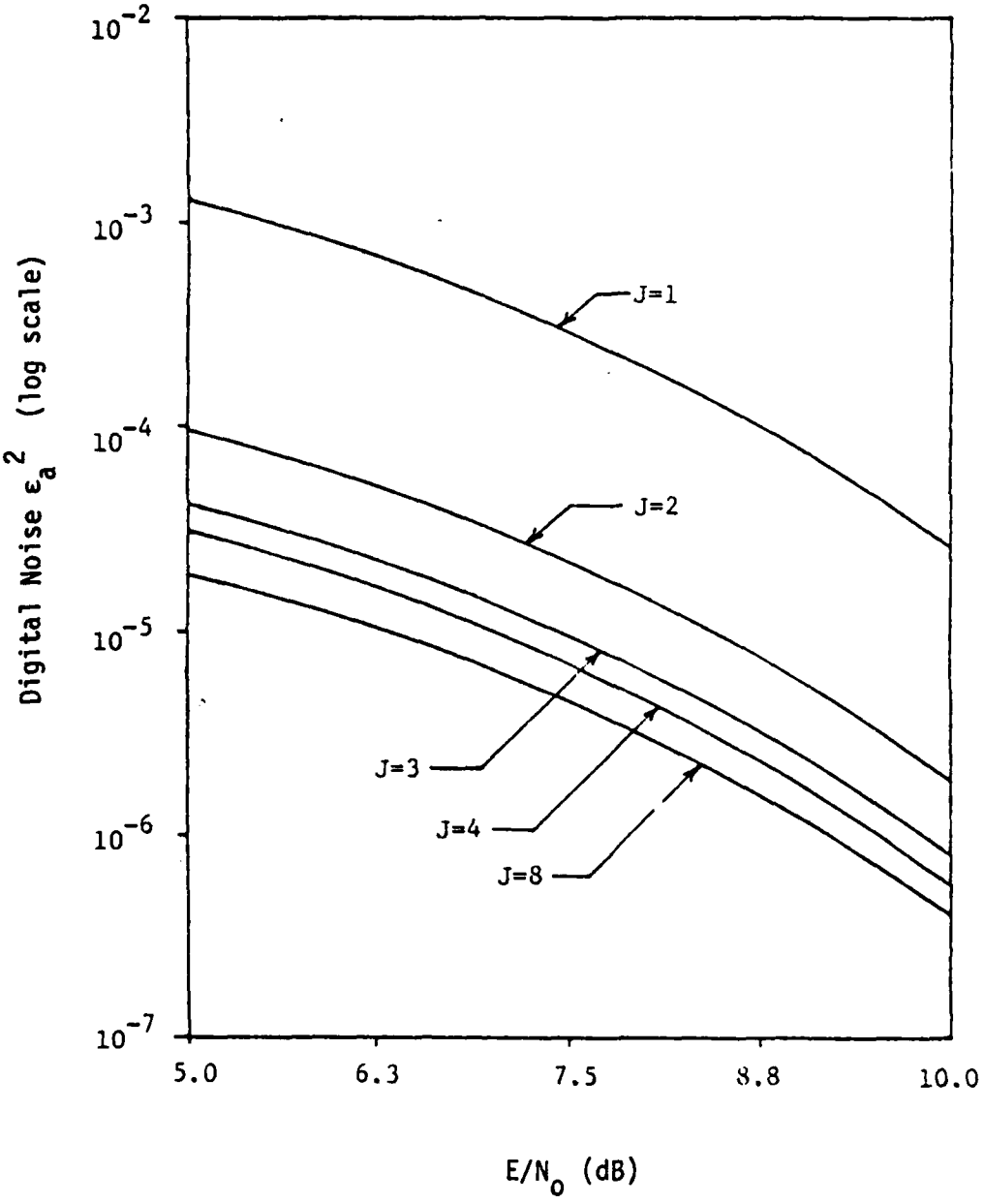


Figure 7.2. Digital Noise vs. Input SNR as a Function of the Number of Allowed Energy Levels

3. INVESTIGATION AND MODIFICATION OF BTC

The Block Truncation Coding technique described in Section 2 was chosen as the basis of the source coder to be used in this research for the following reasons:

1) it provides relatively good compression rates (1.5 to 2.0 bits/pixel) for a spatial coder;
2) it imposes relatively small computational and storage requirements;
3) it suffers less from the effects of channel errore than many other coding techniques.

BTC thus seems well suited for applications which demand good performance in the presence of noise, a moderate degree of compression, and a minimum degree of implementation complexity.

This chapter will describe the primary investigation which were performed on the BTC technique, leading to the development of a modified form of this coding.

Implementation of Basic BTC

The basic BTC coder as described by Delp and Mitchell [11] was first simulated at a data rate of 2 bits/pixel over an ideal (error free) channel. This simulation served as a benchmark with which to compare all future versions of the coder, and provided a measure of the performance of this technique relative to the standard two-dimensional cosine transform coding.

As discussed in the previous chapter, the implementation of BTC requires the determination, within each block, of the mean, the standard deviation, and the n X n bit map. Each Block is then reproduced from these three parameters. The threshold used to quantize and code the actual pixel values may be either fixed or variable. For either method of threshold selection, however, the block means and standard deviations are constant for a given image.

Three test images will be considered throughout this paper: a Girl image, a Moon image, and an Aerial image. The BTC block means and standard deviations which result from these images are presented in the form of histograms in Figures 7.3 through 7.8 (pages 130 through 135).

For the initial 2 bits/pixel simulation, the block means and block standard deviations were each coded with 8 bits. The block size used for all simulations was 4 X 4 pixels. The bit map thus required 16 bits, resulting in an average bit rate of (8 + 8 + 16) bits/(16) pixels = 2 bits/pixel.

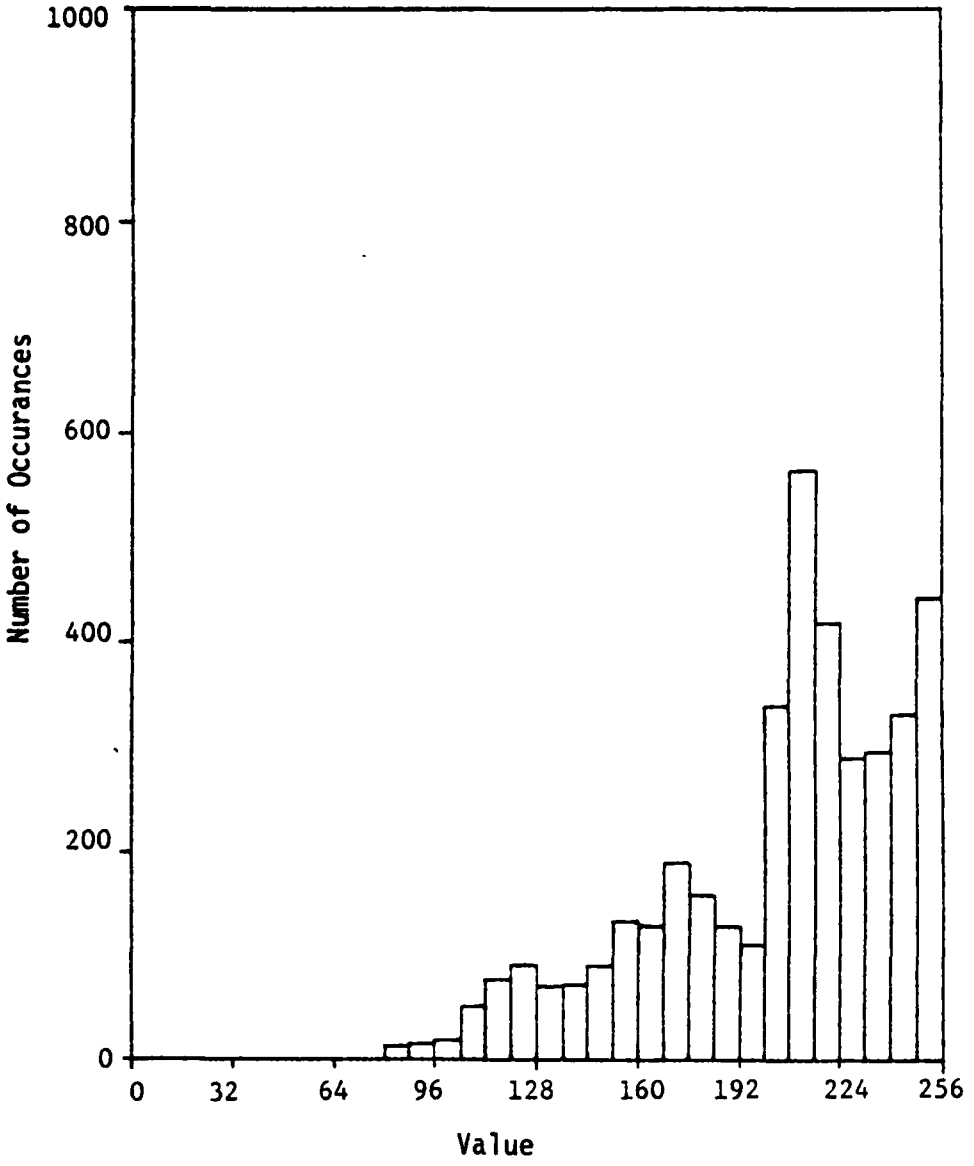


Figure 7.3. Histogram of Block Means, Girl Image

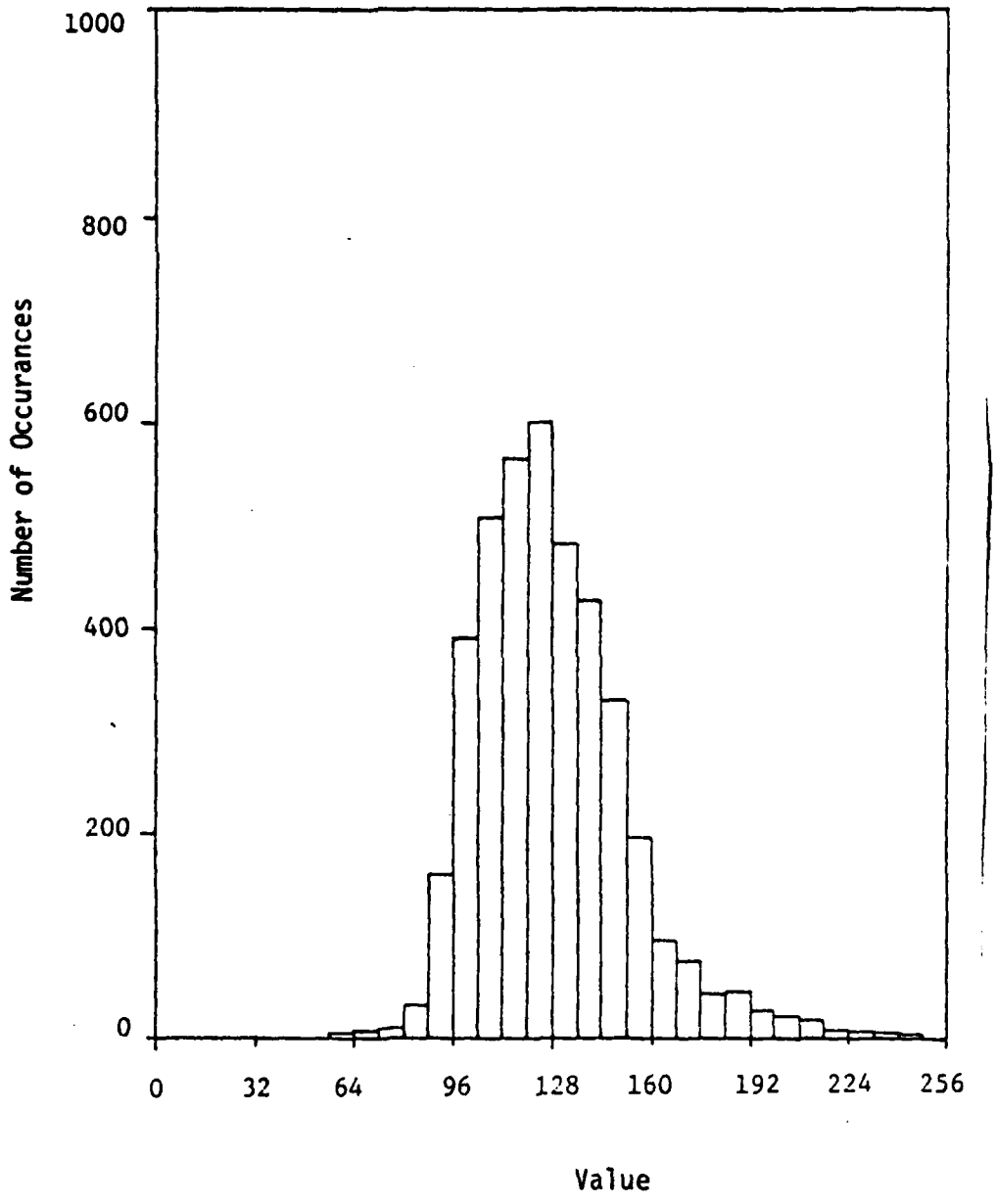


Figure 7.4. Histogram of Block Means, Moon Image

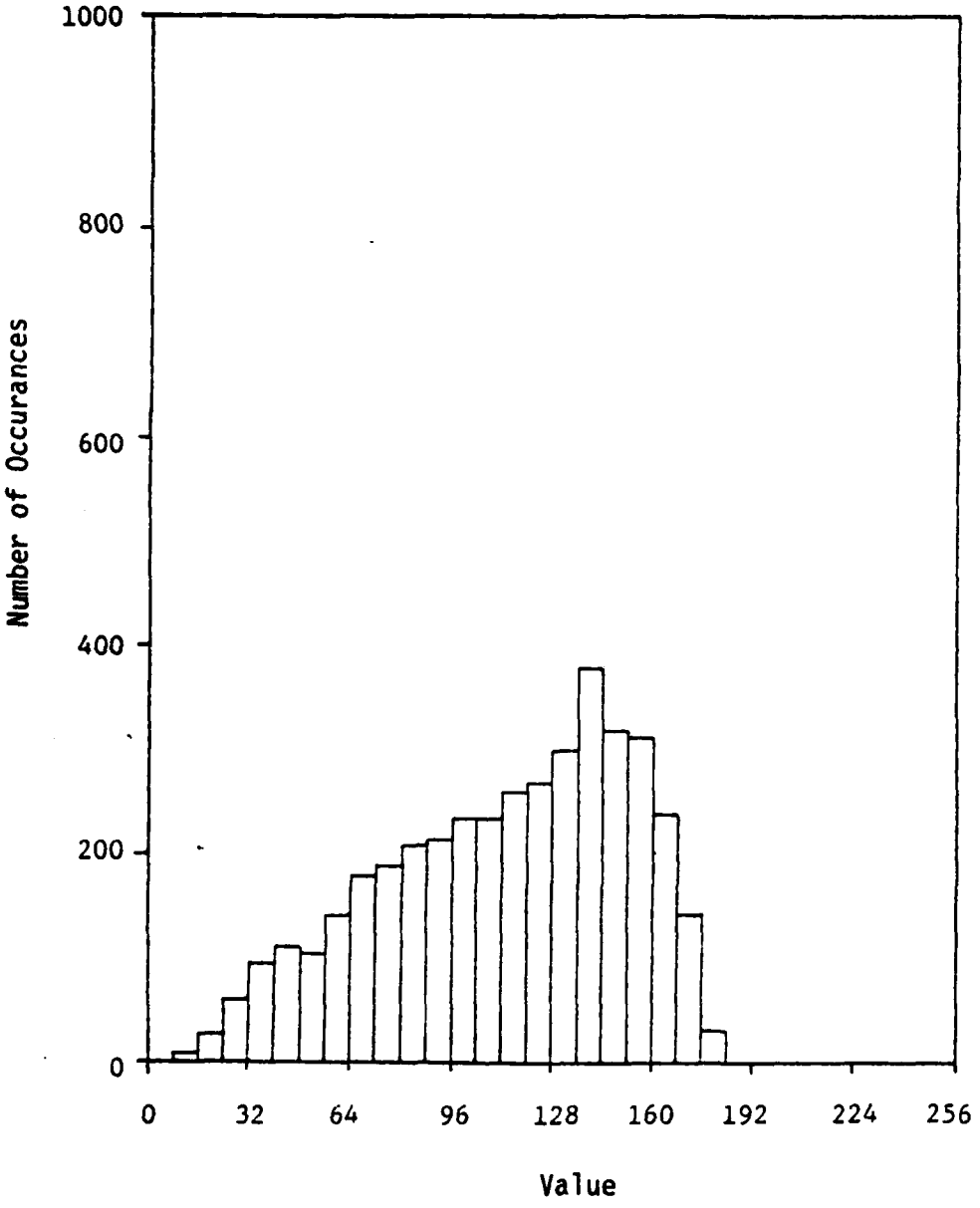


Figure 7.5. Histogram of Block Means, Aerial Image

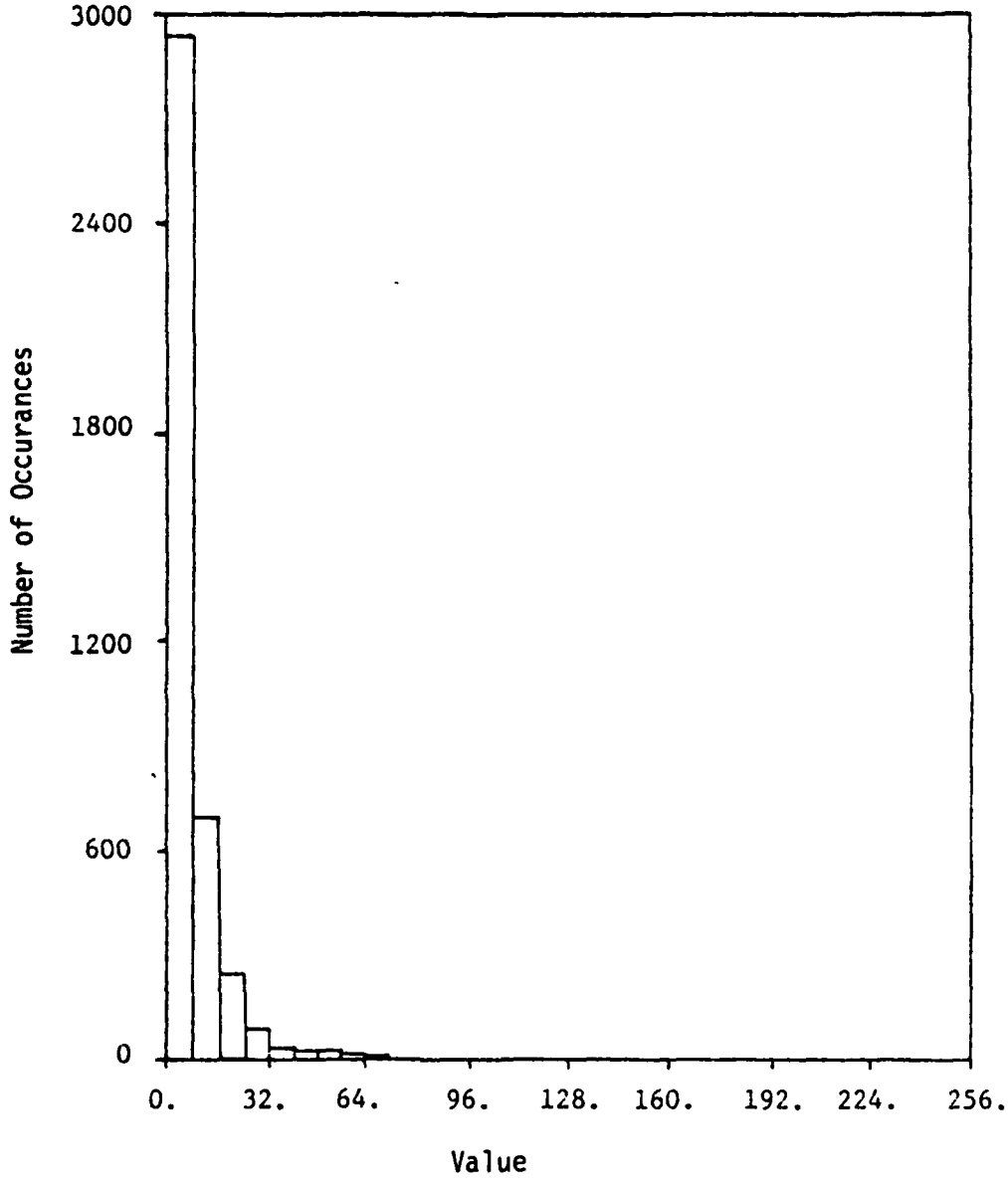


Figure 7.6. Histogram of Block Standard Deviations, Girl Image

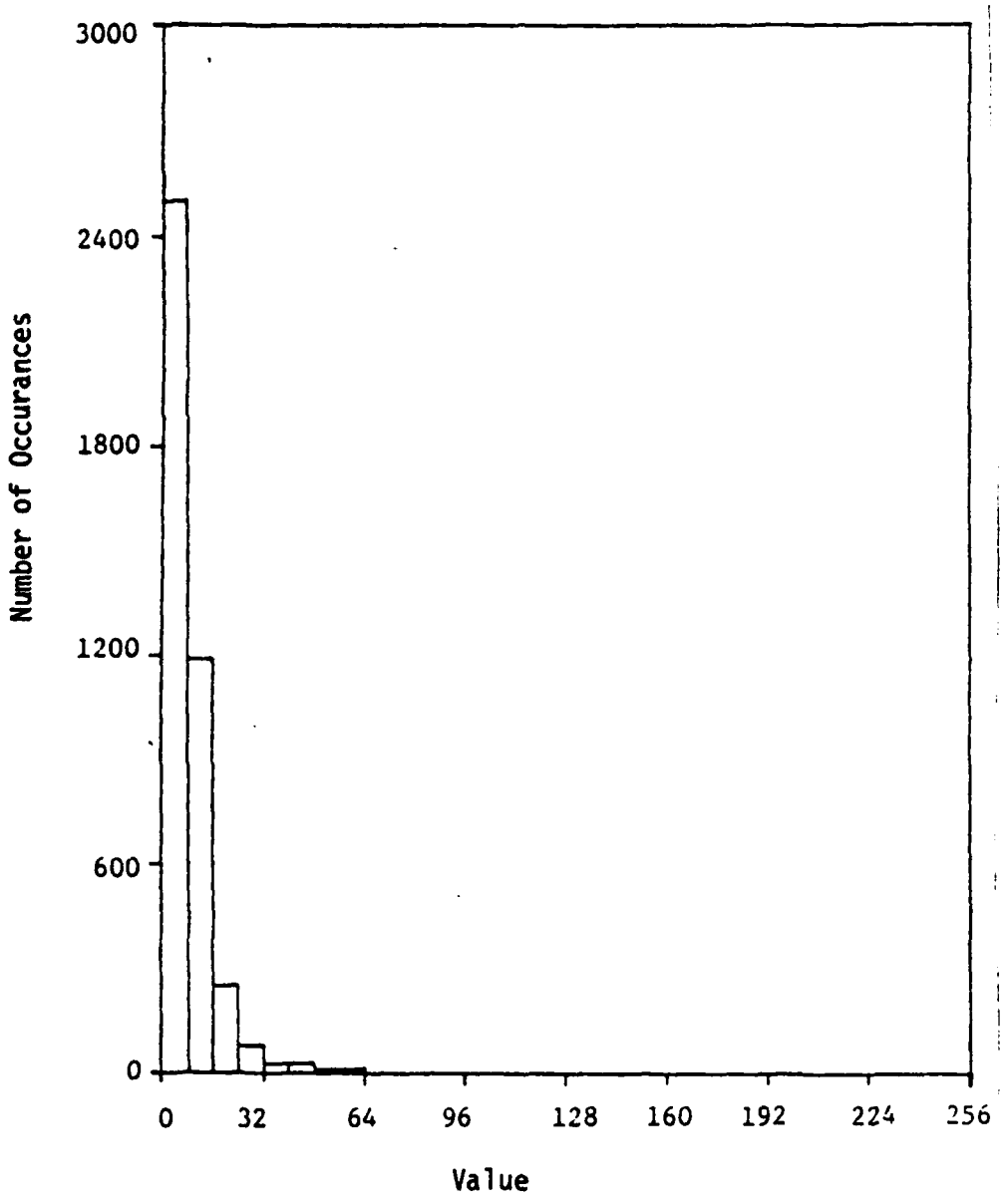


Figure 7.7. Histogram of Block Standard Deviations, Moon Image

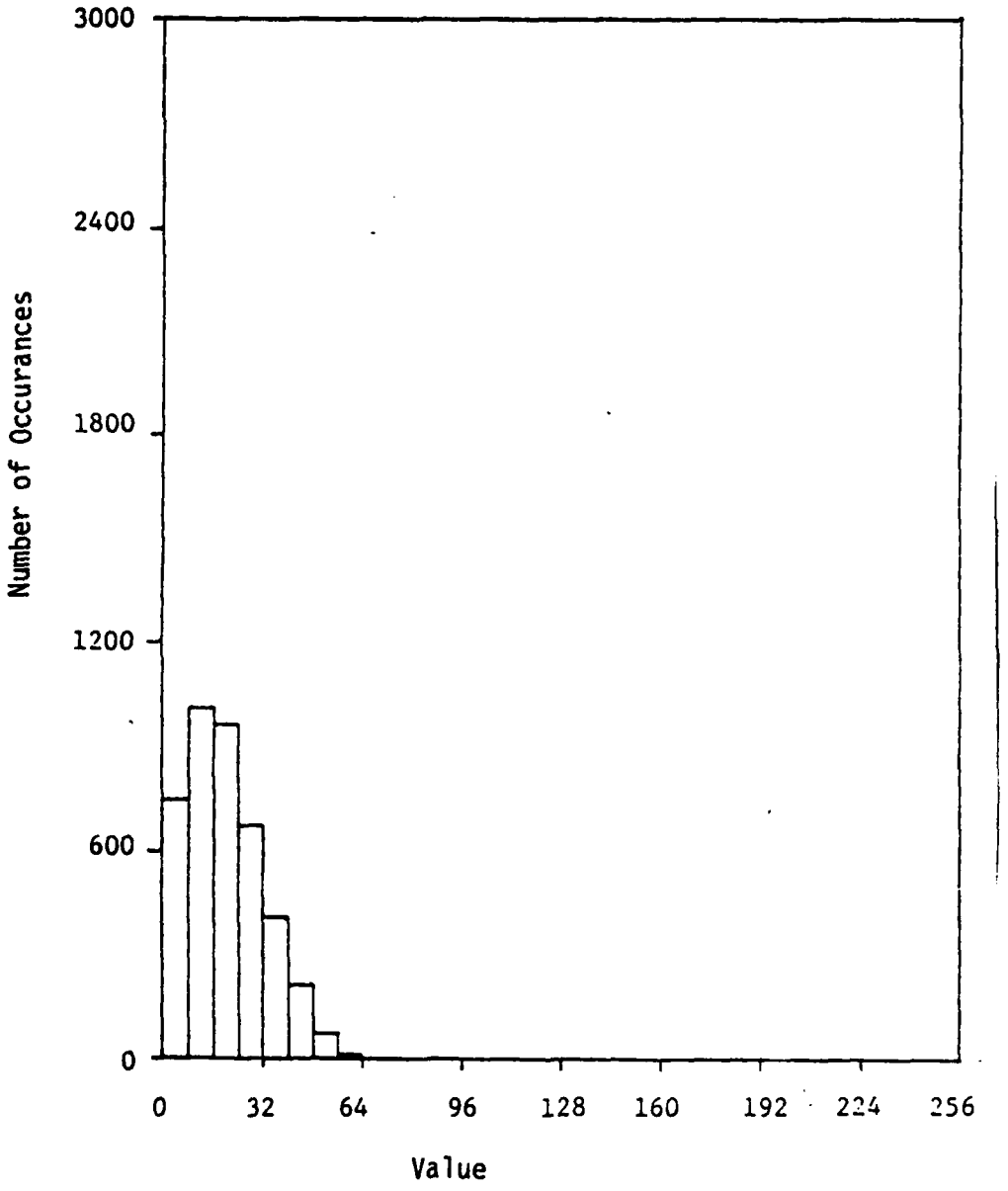


Figure 7.8. Histogram of Block Standard Deviations, Aerial Image

The results of this initial simulation, for both the fixed and the variable threshold implementations, are presented in Table 7.1.

TABLE 7.1. MSE FOR IMAGES PROCESSED BY BTC AT 2 BITS/PIXEL

Image	Threshold Selection	
	Fixed	Variable
Girl	35.83	35.07
Moon	38.06	39.53
Aerial	137.98	130.84

The results presented above demonstrate that for the three images of concern here the variable threshold coder provided only a questionable level of improved performance over the fixed threshold coder. For the sake of consistency, however, all of the following simulations will be performed using the variable threshold scheme.

In order to reduce the data rate below 2 bits/pixel, fewer bits must be allocated to code the block means and standard deviations. The following section is concerned with the effect of varying these bit allocations.

Sensitivity of BTC to Quantizer Coarseness

In order to determine the sensitivity of BTC to the coarseness of the mean and standard deviation quantizers, a variable-bit uniform

quantizer simulation was developed. Uniform quantizers were used for both the block means and block standard deviations in this simulation for the reasons detailed below.

The distribution of the block means can take on many different forms (see pages 130 through 132), and in general a particular mean may take on a value in the range 0.0 to 255.0 when the original image pixels are allowed to take on values from 0 to 255. The distribution of the block standard deviations takes on a roughly exponential form for many types of images (see pages 133 through 136), but this distribution is also highly variable. One common characteristic of the block standard deveations is that values greater than 96.0 were vary rare, and no standard deviation greater than 128.0 was found to exist for any of the images studied.

Therefore due to the lack of a general form for the distributions of the means and standard deviations, uniform quantizers were used in the determination of the sensitivity of BTC to quantizer coarseness. The quantizer for the block means was defined over the range 0.0 to 256.0; for an allocation of m bits, this quantizer will consist of 2^m input ranges, each with a corresponding output value. The size of the input ranges, Δ_m, as a fuction of bits allocated is thus

$$\Delta_m = \frac{256}{2^m} \quad . \tag{7.27}$$

The quantizer for the standard deviations was defined over the range 0.0 to 128.0. For an allocation of s bits, this quantizer will contain 2^s input ranges of size Δ_s, where Δ_s is given by

$$\Delta_s = \frac{128}{2^s} \quad . \tag{7.28}$$

The sensitivity of BTC to the coarseness of the individual quantizers was determined by performing the BTC simulation assuming an ideal channel while varying the bit allocations for each quantizer. To determine the effect of the coarseness of the means quantizer, the standard deviation quantizer was assigned a fixed allocation of 8 bits (s=8), and simulations were performed for m=8,7,6,5,4. Thus, Δ_s=0.5, while Δ_m was varied over the range $1<\Delta_m<16$. Similarly, to determine the effect of the coarseness of the standard deviations quantizer, the mean quantizer was fixed at m=8 (Δ_m=1), while the input range size for the standard deviation quantizer was varied over the range $0.5<\Delta_s<16$. For all simulations, the variable threshold option was used.

The results of the initial sensitivity tests for the Girl images are provided in Table 7.2. The upper section of this table is concerned with the effect of the coarseness of the quantizer for the means. The MSE between the iput and output of the means quantizer is tabulated (QMERR), as is the MSE of the standard deviation quantizer (QSERR). The values of QMERR and QSERR provide an estimate of the relative error due to each quantizer.

As can be seen from Table 7.2, in terms of MSE, the effect of a given level of coarseness in one quantizer is quite similar to the effect of the same level of coarseness in the other quantizer. The visual effects are noticeably different, however. The result of coarsely quantizing the block means is that a number of false contours appear in low variance regions of the image. Conversely, coarse quantization of the standard deviations leads to a grainy effect distributed over the entire image. Both effects are detrimental to image quality. From

TABLE 7.2. RESULTS OF THE QUANTIZER SENSITIVITY TESTS FOR THE GIRL IMAGE

Δ_m	Δ_s	Image MSE	QMERR	QSERR
1.0	0.5	35.06566	0.08368	0.02161
1.0	1.0	35.14143	0.08368	0.08417
1.0	2.0	35.57709	0.08368	0.34712
1.0	4.0	36.14406	0.08368	1.19090
1.0	8.0	39.50471	0.08368	4.50202
1.0	16.0	59.43402	0.08368	22.92423
1.0	0.5	35.06566	0.08368	0.02161
2.0	0.5	35.33759	0.34517	0.02161
4.0	0.5	36.25166	1.27751	0.02161
8.0	0.5	39.91125	4.95567	0.02161
16.0	0.5	55.52716	20.60512	0.02161

these preliminary sensitivity investigations, it can be seen that an attempt to reduce the bit rate substantially by reducing the number of bits allocated to either of the quantizers will result in image degradation when independent uniform quantizers are used.

The effect of reducing the bit allocation to both quantizers simultaneously was also investigated. Simulations were performed in which quantizers were allocated 7 bits, 6 bits, 5 bits, and 4 bits. These simulations correspond to average data rates of 1.875, 1.75, 1.625, and 1.50 bits/pixel respectively. Table 7.3 contains the results of these simulations for the Girl image.

TABLE 7.3. EXTENDED RESULTS OF THE QUANTIZER SENSITIVITY TESTS FOR THE GIRL IMAGE

Bits/pixel	Δ_m	Δ_s	Image MSE	QMERR	QSERR
1.875	2.0	1.0	35.48068	0.34517	0.08417
1.750	4.0	2.0	36.57942	1.27751	0.34712
1.625	8.0	4.0	41.02132	4.95567	1.19090
1.500	16.0	8.0	60.35036	20.60512	4.50202

The results provided in Table 7.3 demonstrate that an attempt to achieve a bit rate of 1.5 bits/pixel using the BTC scheme described thus far will result in a substantial sacrifice in image quality.

Healy and Mitchell [21] have proposed an alternate method of quantizing the block means and standard deviations of BTC. Noting that grey level quantization error is most visible in low variance regions of an image, they proposed a two-dimensional quantizer which simultaneously codes the mean and standard deviation using 10 bits. The quantizer is designed so that the quantization error for both inputs increases as the standard deviation increases. The results of using this quantizer scheme are compared in Table 7.4 to the results obtained by separately quantizing the mean and standard deviation using five bits each.

TABLE 7.4. COMPARISON OF THE 10 BIT TWO-DIMENSIONAL QUANTIZER TO TWO INDEPENDENT 5 BIT QUANTIZERS (MSE)

Image	Simultaneous Quantizer	Independent Quantizers
Girl	37.00244	41.02132
Moon	42.05714	46.20068
Aerial	138.65907	137.73419

The results given in Table 7.4 indicate that the two-dimensional quantizer generally provides a slight improvement in a MSE sense. A reduction of the false contours artifact is also provided through the use of this quantizer, though in some images (the Aerial image, for example), there is little perceivable visual improvement. Though the two-dimensional quantizer of Healy and Mitchell does usually provide some degree of improvement over the independent quantization scheme; it is not apparent if a more efficient 10 bit two-dimensional quantizer may be designed, and there is no provision in the design of this quantizer which would allow it to adapt to varying distributions of the input parameters. Also, the data rate is fixed at 1.625 bits/pixel with this quantizer; operation at a different data rate would require that the quantizer be redesigned.

Based on the simulation results detailed above, it was apparent that modifications of BTC beyond the introduction of a two-dimensional quantizer were required in order to achieve acceptable image quality at data rates in the 1.50 to 1.75 bits/pixel range. It was determined that

the introduction of a Differential Pulse Code Modulation (DPCM) loop into the BTC scheme provided enhanced performance at these low data rates. The details of this modification and the results it provided are the subject of the following section.

Application of DPCM to BTC

A block diagram of a generalized DPCM system is shown in Figure 7.9 (page 143). For every input signal x_N, the linear predictor generated a prediction value $\hat{x}_N$ which is calculated from N-1 preceeding samples according to the relation

$$\hat{x}_N = \sum_{i=1}^{N-1} a_i x_{N-1} \tag{7.29}$$

Only preceeding transmitted samples are used for the prediction, so that the receiver is also able to calculate $\hat{x}_N$. The predictor coefficients a_i are optimized to yield a predictor error

$$e_N = x'_N - \hat{x}_N \tag{7.30}$$

with minimum variance. This error value is then quantized. The reconstructed input signal x_N is created at the receiver and the transmitter by adding $\hat{x}_N$ to the quantized prediction error e'_N. The reconstructed value x'_N thus differs from the original sample by the quantization error

$$q_N = e_N - e'_N = x_N - \hat{x}_N - (x'_N - \hat{x}_N) = x_N - x'_N. \tag{7.31}$$

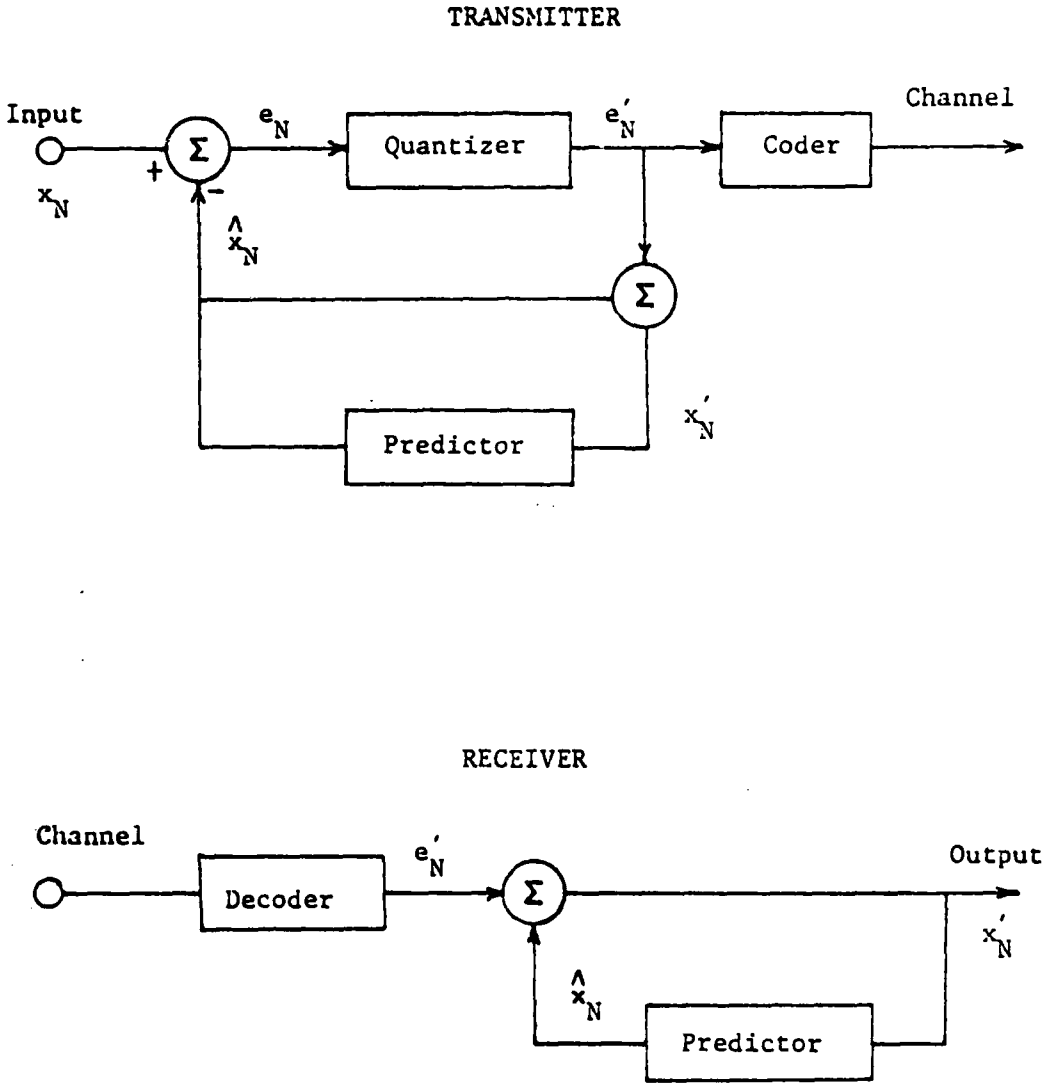


Figure 7.9. Block Diagram of a General DPCM System

When operating on data samples in which there exists a degree of linear statistical dependency, the prediction errors will have a smaller variance than the original sample values [27]. Due to the smaller variance of the signal to be quantized, coded and transmitted, the amplitude range of the quantizer may be reduced, a smaller number of quantizing levels may be used, and fewer coding bits are required for a given signal to noise ratio than in a non-predictive system.

Thus, DPCM systems may be used advantageously in situations in which there exists a degree of correlation between data samples. With respect to BTC, it seems reasonable to assume that, in general, successive samples of the block mean will be correlated, as will successive samples of the block standard deviations.

The form of DPCM chosen for use with the BTC technique utilizes a first order linear predictor. Following (7.27),

$$\hat{x}_N = a_1 x_{N-1} \cdot \tag{7.32}$$

The prediction for the current data sample is based simply on a linear function of the preceeding sample. The effect of different values of the predictor constant a_1 will be discussed later.

A number of images were sampled to determine general distribution functions for the differences of the block means and block standard deviations. The shape of these distributions was of interest in choosing the quantizer to be used in the DPCM loop. Histograms of these differents for the three dimages discussed previously (Girl, Moon, and Aerial) are provided in Figures 7.10 through 7.15 (pages 145 through 150). All distributions are of Laplacian form, with means essentially equal to zero. The variance of a particular distribution was found to depend on the data

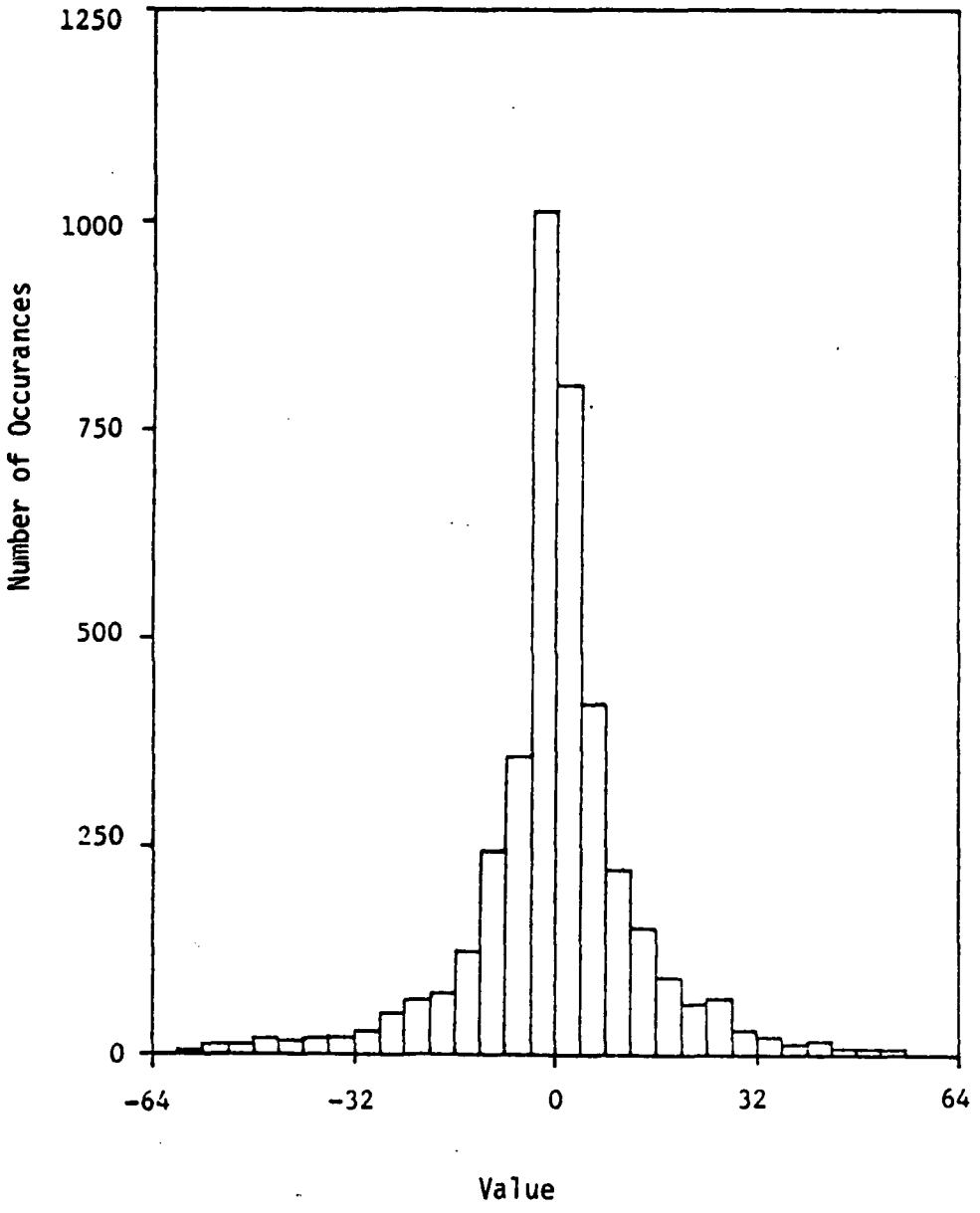


Figure 7.10. Histogram of Differences of Block Means, Girl Image

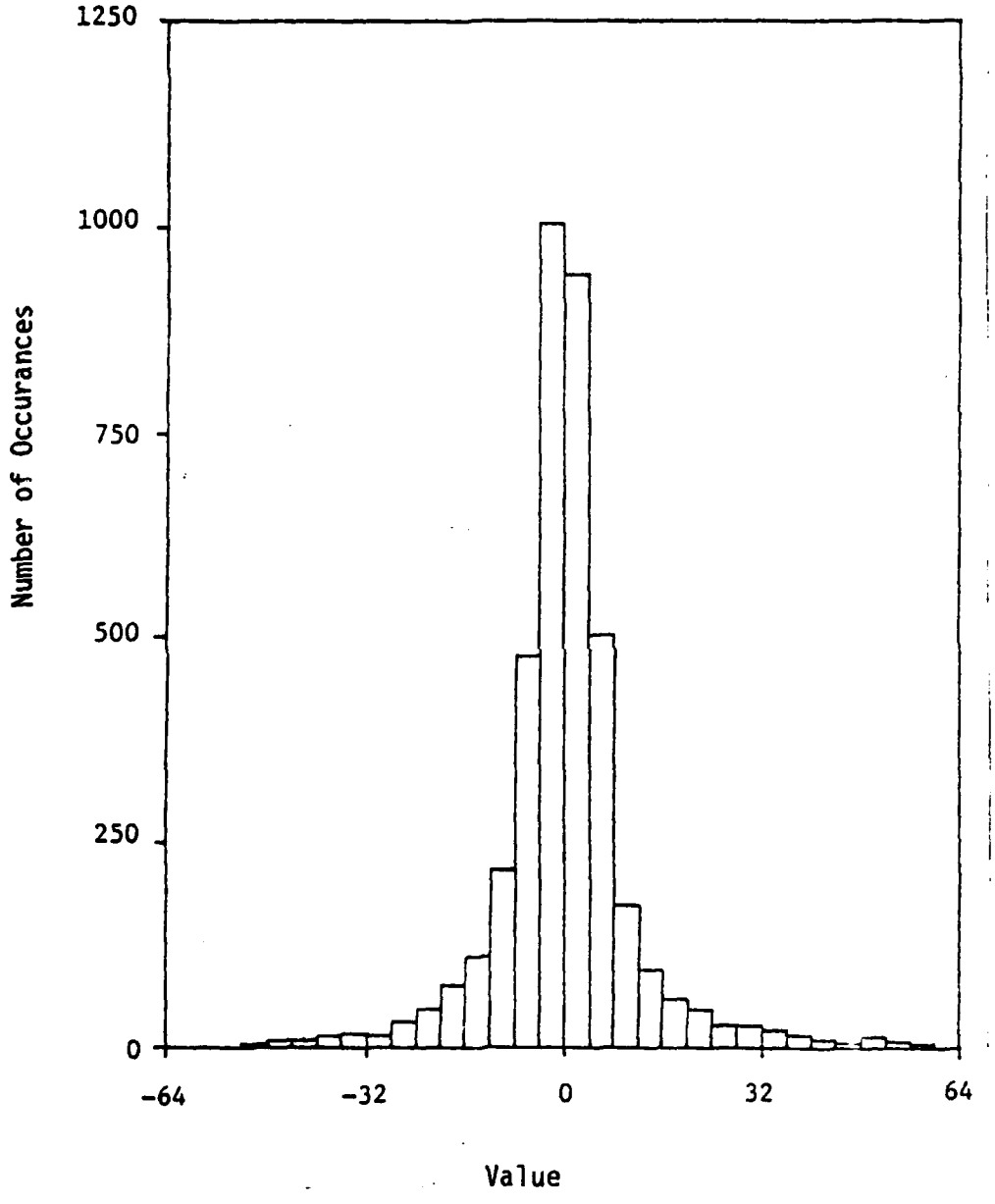


Figure 7.11. Histogram of Differences of Block Means, Moon Image

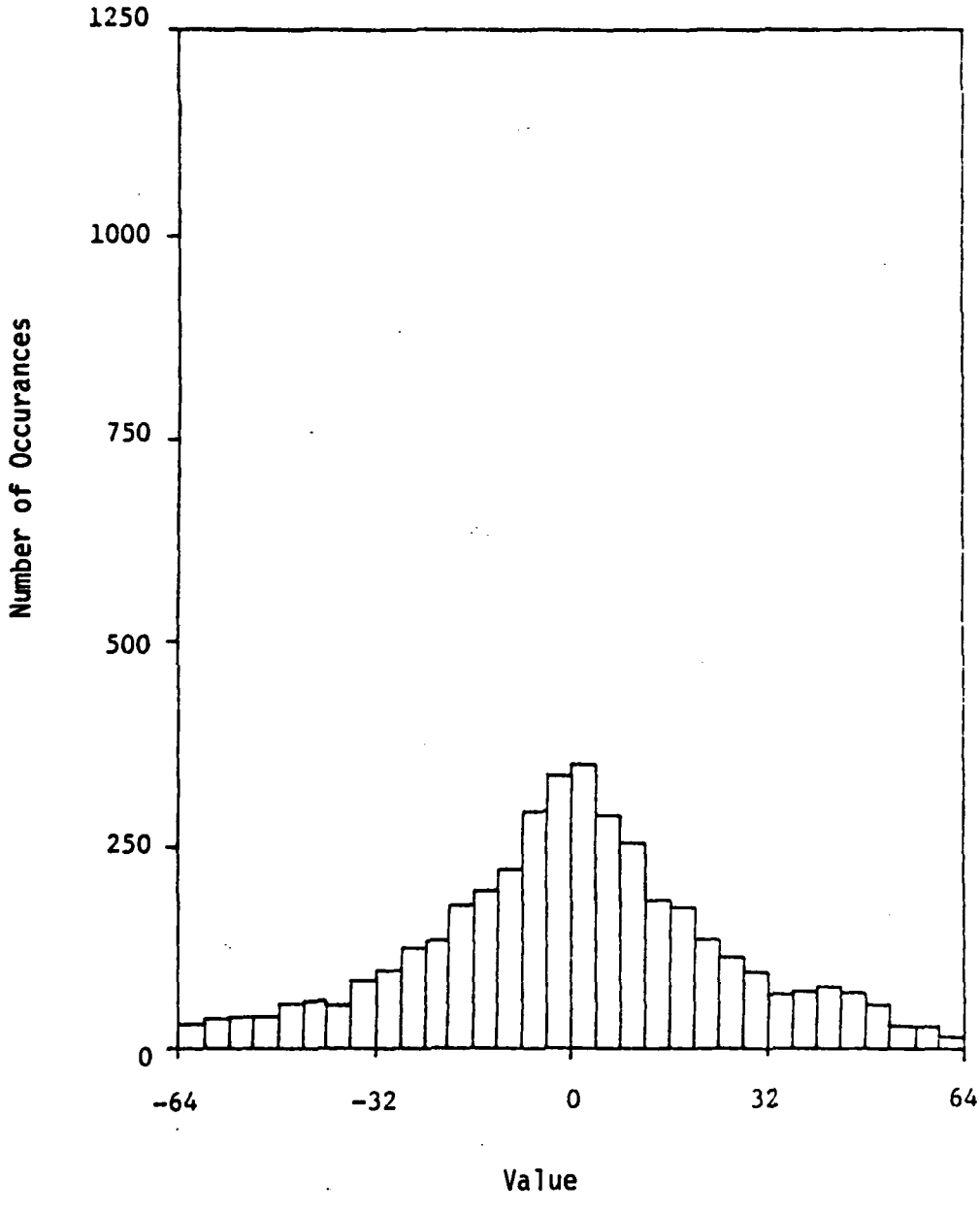


Figure 7.12. Histogram of Differences of Block Means, Aerial Image

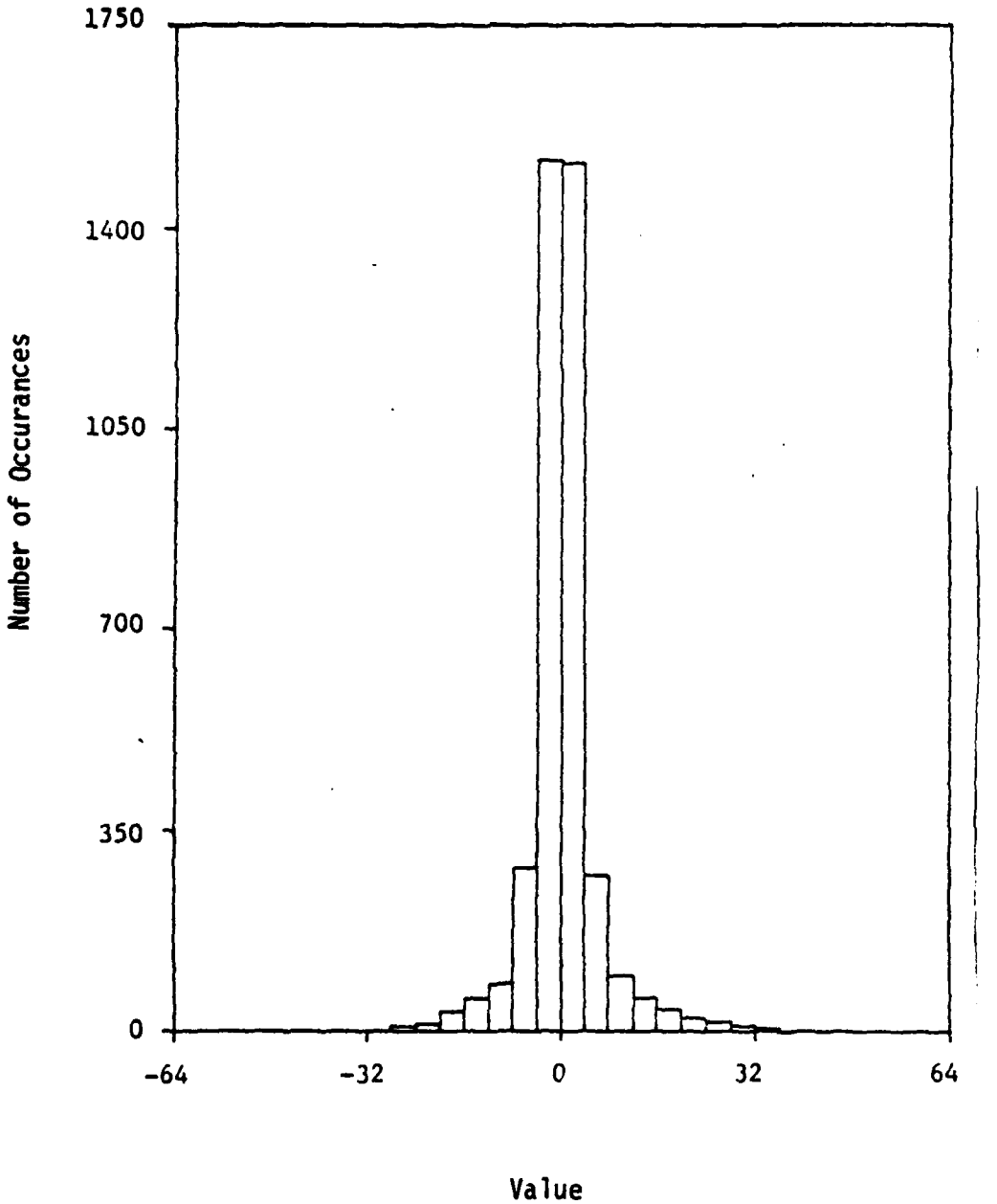


Figure 7.13. Histogram of Differences of Block Standard Deviations, Girl Image

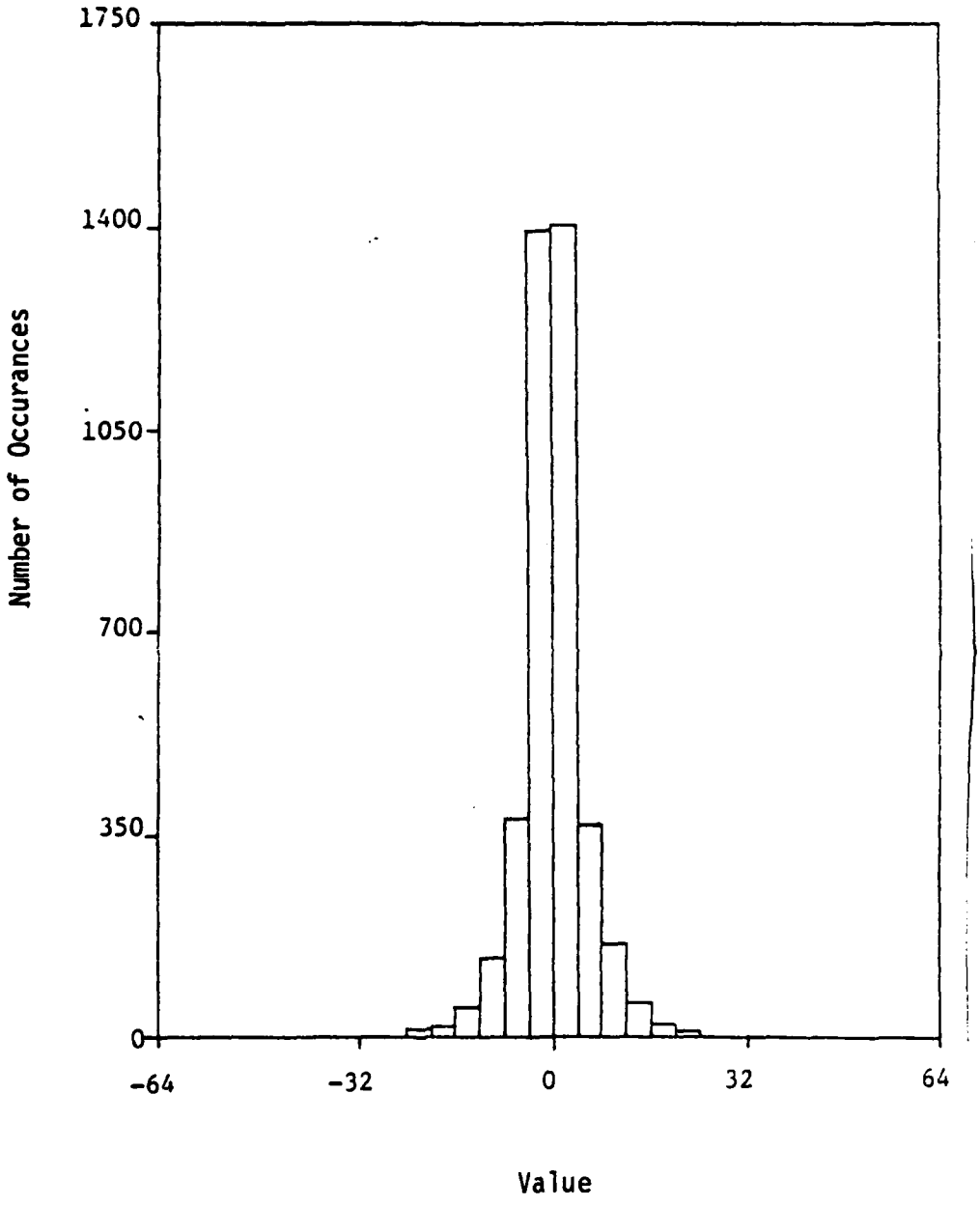


Figure 7.14. Histogram of Differences of Block Standard Deviations, Moon Image

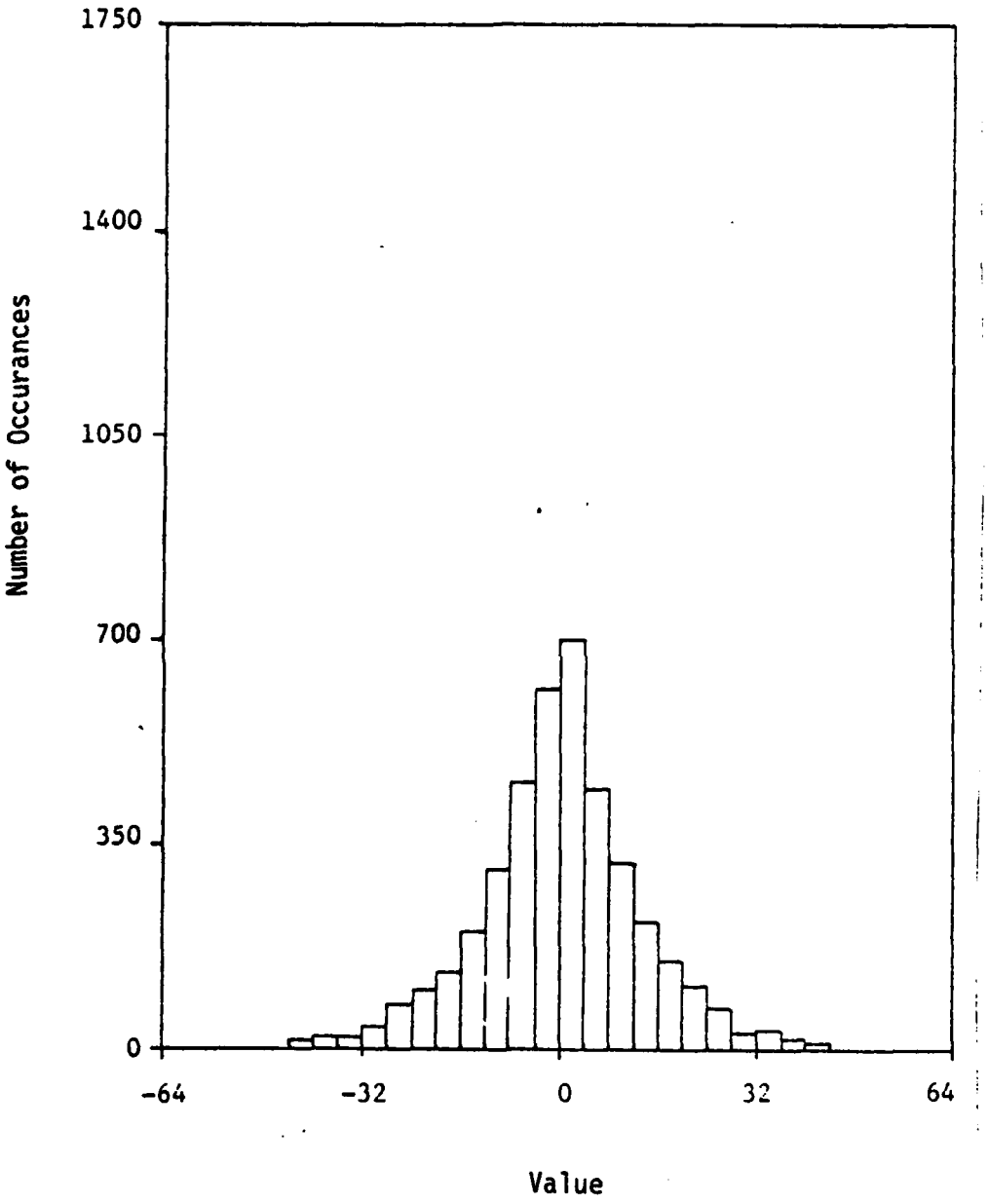


Figure 7.15. Histogram of Differences of Block Standard Deviation Aerial Image

type (differences of means vs. differences of standard deviations), and also on the image from which the data was derived.

Based on the reasoning and results presented above, the DPCM system was implemented as follows: two DPCM loops were used, one each for the block means and the block standard deviations. The n X n bit map for each block remained unaffected by these DPCM loops. First order linear predictors were used for both loops, and the quantizers were of the form optimized for Laplacian distributions, from Adams and Giesler [23]. The specific quantizer values used in a particular simulation were determined from this general Laplacian form and the pre-calculated value of the variance of the data entering the DPCM loop.

For example, the variance of the differences of the block means for the Girl image was found from prior calculations to be 306.426, while the variance of the standard deviations was determined to be 51.879. These values would then be used to scale the Laplacian quantizers within the respective DPCM loops. This approach has the obvious disadvantage of requiring the variances of the data to be determined in advance; a method of avoiding this severe limitation while improving the DPCM system performance will be presented in the next section.

Simulations of this BTC/DPCM system were performed using both 4 bit and 5 bit Laplacian quantizers within the DPCM loop. An ideal channel was assumed in all cases. These simulations correspond to average data rates of 1.50 and 1.625 bits/pixel respectively. The results of these simulations for the three images under consideration are listed in Table 7.5.

TABLE 7.5. RESULTS OF THE BTC/DPCM SIMULATIONS

(MSE)

Image	1.50 bits/pixel	1.625 bits/pixel
Girl	44.79852	37.60272
Moon	49.79671	42.31354
Aerial	146.05792	135.07271

The results given in Table 7.5 were obtained with the predictor coefficients for both DPCM loops set to unity. Further simulations suggested that the performance decreases slightly for coefficients of magnitude less than 1.

The MSE figures listed in Table 7.5 for the average bit rate of 1.625 bits/pixel may be compared to the figures given in Table 7.4 (page 141) for the simultaneous two dimensional and independent uniform quantizers. It can be seen that the DPCM scheme provides performance comparable to that obtained using the simultaneous or independent quantizer techniques. The images produced by the DPCM scheme are comparable in quality to those rendered by Mitchell's two-dimensional quantizer in the respect that the problem of false contouring which often results from coarse quantization of the block means is not apparent to the eye. In addition, the problem of excessive graininess, a symptom of coarse quantization of the block standard deviations, is not apparent in the DPCM images.

The results described above provided an indication that the application of DPCM to BTC was a feasible and productive modification. Unlike Mitchell's two-dimensional quantization scheme, which was derived on a strict empirical basis, the DPCM scheme is easily modified to provide higher or lower bit rates simply by varying the number of bits assigned to the Laplacian quantizers within the DPCM loops. A further advantage of the DPCM scheme is that its performance may be enhanced through the application of a learning algorithm which operates on the DPCM quantizers. The development of this learning algorithm and its application to the BTC/DPCM scheme is the subject of the following section.

Development and Application of an Unsupervised Learning Algorithm

The DPCM scheme described above involved the use of two Laplacian quantizers with variances which were fixed for each image. Improved performance may be realized by allowing the quantizers to adapt to changes in the distribution of the data within the image. In general adaptivity may be realized through midification of either the predictor or the quantizer within the DPCM loop, but Mussman [24] has shown that any adaptive predictor may be interpreted as an adaptive quantizer. The following discussion will therefore deal exclusively with the adaptive quantization problem.

Various investigators [25 - 28] have considered adaptive quantization schemes. Most techniques developed to date may be considered to be members of either the forward adaptive class or the backward adaptive class. Forward adaptive systems derive variable quantizer step size

based on the quantization error. Both forward and backward adaptive systems typically seek to modify the parameters of a single general quantizer in order to reduce quantization error.

A more general scheme has been developed by Griswold and Sayood [29] based on the estimation theory work of Patrick [30]. This method involves classifying input values as belonging to one of several possible probability distributions. Estimation techniques are employed to choose the most probable distribution from which the current data values are being drawn; this chosen distribution is referred to as the "active" distribution. In the unsupervised learning approach, the technique of modifying the dynamic range of a single quantizer is replaced by the process of selecting a single distribution from a group of possible distributions and utilizing optimal quantization.

An outline of the development of the unsupervised learning algorighm is now presented.

Given a set of data values denoted by an L-dimensional column vector $\underline{X}$,

$$\underline{X} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_L \end{bmatrix}. \tag{7.33}$$

It is assumed that a feature extraction or processing operation had made the conversion to an L-tuple. A sequence of n vectors is denoted by

$$\dot{\underline{X}}_n = [\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_n,] \tag{7.34}$$

where the vectors $\underline{X}_i$, i = 1,2,...,n are understood to be members of some observation space 0 of column vectors.

Suppose that associated with each of the samples $\underline{X}_1$, $\underline{X}_2$,...,$\underline{X}_n$ is a probability distribution with the possibility of some of the samples being drawn from, for example, $F(\underline{X}\P\underline{a}_1)$, some from $F(\underline{X}\P\underline{a}_2)$, etc., where $\underline{a}_1$ and $\underline{a}_2$ are parameter vectors which characterize the probability distributions $F(\underline{X}\P\underline{a}_1)$ and $F(\underline{X}\P\underline{a}_2)$. A family of distributions may be defined as

$$\beta = F(\underline{X}\P\underline{a});\ \underline{a} \in A,\ \underline{X} \in V_L \tag{7.35}$$

where A is the space of parameter vectors characterizing the probability distributions of interest, and V_L is the vector space containing the observation elements $\underline{X}_i$, i = 1,2,...,n. For values of $\underline{a}_i$ and $\underline{a}_2$ in A, any sample of $\underline{X}$ ($\underline{X} \in V_L$) could be from any one of the member distributions in the family β.

A finite mixture distribution (see [30]) may be defined as

$$H(\underline{X}) = \sum_{i=1}^{N} F(\underline{X}\P\underline{a}_i)P(\underline{a}_i) \tag{7.36}$$

where $\underline{a}_i$, i = 1,2,...,N form the parameter space A, and $P(\underline{a}_i)$, i = 1,2,...,N are the unknown mixing parameters.

Given the mixture of $H(\underline{X})$, the samples $\underline{X}_1$, $\underline{X}_2$, $\underline{X}_n$, some of which are from $F(\underline{X}\P\underline{a}_1)$, some from $F(\underline{X}\P\underline{a}_2)$, etc., and the distribution family β, unsupervised learning is concerned with the attempt to decide which samples are from which distribution, for the purposes of estimating the parameters characterizing that distribution.

The problem in the application of unsupervised learning to adaptive DPCM (ADPCM) is to resolve the mixture (unknown) into the "active" distribution which defines the current source event [29]. With respect to DPCM, the samples $\underline{X}_i$, i = 1,2,...,n are vectors whose elements are the differences to be coded, and we have the condition that any sample difference input to the quantizer in the DPCM loop could be from any member distribution. The task is then to assign the input value to one of a finite number of quantizers, and to estimate the parameters for the particular "active" quantizer.

If the number of distinct quantizers is m (finite) with corresponding parameter points $\underline{b}_1,\underline{b}_2,\ldots,\underline{b}_m$ and associated class probabilities $P_1,P_2,\ldots,P_m$, then following Patrick [30] we may define

$$\underline{b} = b_1,b_2,\ldots,b_m,P_1,P_2,P_m \tag{7.37}$$

where $\underline{b} \in A^M \times P^M$ (Cartesian product) and $m < M$; A^M is the vector space of distribution parameters, and P^M is the mixing parabeters. The mixing distribution conditioned on $\underline{b}$ is

$$H(\underline{X}\P\underline{b}) = \sum_{i=1}^{M} F(\underline{X}\P i,\underline{b}_i)P_i \tag{7.38}$$

where

$$\sum_{i=1}^{M} P_i = 1 \text{ and } P_i > 0 \text{ for all } i. \tag{7.39}$$

Equivalently, the mixing density conditioned on $\underline{b}$ is given by

$$h(\underline{X}\P\underline{b}) = \sum_{i=1}^{M} f(\underline{X}\P\underline{b}_i)P_i \quad . \tag{7.40}$$

assuming the densities $f(\underline{X}\ \underline{b}_i)$ exist.

The application of unsupervised learning involves resolving this finite mixture density by estimating or searching for a solution vector $\underline{b}$. Estimating $\underline{b}$ identifies which probability density function is active at a given time. Bayes or Maximum Likelihood estimation proceedures may be used for this purpose just as if there were no mixture density [29], [30]. Patrick [30] has developed an expression for $f(\underline{B}\P\underline{X}_n)$ based on the "decision directed" estimator of Bayes. The decision directed approach uses decisions made on past samples to approximate the classificaiton of the current sample. The use of Bayes Theorem allows the problem to be converted from one requiring "a priori" information to one requiring "posteriori" information. From [30], the expression for $f(\underline{b}_j\P\underline{X}_n)$ in terms of posteriori information is

$$f(\underline{b}_j\P\underline{X}_n) = \begin{cases} f(\underline{b}_j\P\underline{X}_{n-1}), \text{ if } \sum_{i\neq j} (W_i)_n > (W_j)_n \\ \dfrac{f(\underline{X}_n\P\underline{b}_j)f(\underline{b}_j\P\underline{X}_{n-1})}{f(\underline{X}_n\P j)}, \text{ otherwise} \end{cases} \tag{7.41}$$

where

$$(W_i)_n = \frac{(P_i)_{n-1} f(\underline{X}_n\P i,(\underline{b}_i)_{n-1})}{\sum_{k=1}^{m} (P_k)_{n-1} f(\underline{X}\P k,(b_k)_{n-1})}$$

$\underline{X}_n$ is a sequence of vectors $\underline{X}$ at stage n

n is the stage of iteration

i is the ith distribution which defines a quantizer through parameter $\underline{b}_i$

j is the jth distribution which defines a quantizer through parameter $\underline{b}_j$.

The mixing parameters $(P_k)_n$ remain to be defined.

The term $\sum_{i \neq k} (W_i)_n$ above is the probability that the kth quantizer is not "active" given a data sequence $\underline{X}_n$. $(W_k)_n$ is the probability that the kth distribution, which is associated with quantizer k, did cause the data sequence at stage n. Note that equation (7.39) above presumes the initial distribution estimates to be reasonably good approximations of the actual distributions within the image and therefore the decisions of the classifier will be generally correct. Therefore, the distribution defining parameters are updated only when the data indicates that another class of distribution is "active". If only one distribution is allowed to be active at any one time, the unsupervised learning problem may be converted into a supervised learning situation in which the decision directed estimates defining the active distribution are based on the last n-1 decisions as training sets.

For the BTC/DPCM case of interest here, the family of distributions to be considered is the zero mean Laplacian, defined by

$$\beta_{\text{Laplacian}} = \frac{1}{\sqrt{2}\sigma_i} \exp\left(-\frac{\sqrt{2}}{\sigma_i} \left|\underline{X}_k\right|\right), \tag{7.42}$$

where

$$\sigma_i \in A, \ \underline{X}_k \in V_L \ .$$

For this case, the parameter defining a particular distribution is based on the scaler quantity σ_i, the standard deviation of that distribution. In the supervised learning framework, the decision directed

estimate for the parameter of distribution k at stage n is given by (see [29]):

$$(\sigma_k)_n = \begin{cases} (\sigma_k)_{n-1} & \text{if} \quad \sum_{i \neq k} (W_i)_n > (W_k)_n \\ \sqrt{2}\,\dfrac{|x_k|}{n} + \dfrac{n-1}{n\sqrt{2}}\,(\sigma_k)_{n-1} & \text{otherwise} \end{cases} \tag{7.43}$$

The terms $\sum_{i \neq k} (W_i)_n$ and $(W_k)_n$ are defined as for (7.39). Since the probability that the current data value x_k came from the kth distribution is related to the posteriori parameter estimate through the mixing parameter P_i, an estimate of these paremeters is required. These parameters must satisfy two conditions, namely

$$\sum_{i=1}^{n} (P_i)_n = 1, \qquad (P_k)_n \geq 0\ , \tag{7.44}$$

where m is the number of distinct quantizer classes. If the range of the quantizer spans the range of observations, then for any data vector

$$\underline{X} \in \begin{cases} Q_1 \text{ with probability } P_1 \\ Q_2 \text{ with probability } P_2 \\ \cdot \\ \cdot \\ \cdot \\ Q_m \text{ with probability } P_m\ , \end{cases} \tag{7.45}$$

where Q_i, $i = 1,2,\ldots,n$ denotes the ith quantizer. For any specific sequence $\underline{X}$ we have a multinomial distribution for which we are required to estimate $\underline{P}$, where $\underline{P} = [P_1, P_2, \ldots, P_m]$. Following [30], it is

assumed that the density function $f(\underline{P})$ is Diriclet and reproducing, leading to the expression (see [29]):

$$(P_k)_n = \frac{(\underline{X}_1)_n}{n + n_o + v} + \frac{n + n_o + (v-1)}{n + n_o + v} (P_k)_{n-1} \qquad (7.46)$$

where

$$(\underline{X}_1)_n = \begin{cases} 1 \text{ if } (\underline{X}_1) \in k \text{ at iteration } n \\ 0 \text{ if } (\underline{X}_1) \notin k \text{ at iteration } n \end{cases}$$

$$\sum_{i=1}^{v} (P_i)_n = 1,$$

n is the current iteration or stage

n_o is the total number of data points

v is the number of possible "active" distributions.

Equations (7.41) and (7.44) above provide expressions for both the posteriori parameter vector and the mixture parameters at any stage n based on a previous sequence of samples. These equations were directly applicable to the BTC/DPCM loops. For the purposes of this research, four different quantizers were allocated for each of the DPCM loops. The quantizer parameters (variances) and mixture parameters (class probabilities) were updated within the two DPCM loops according to equations (7.41) and (7.44). This scheme, comprised of BTC with DPCM and unsupervised learning, represents the end result of the investigation presented in this section. The next section will briefly discuss the application of bit weighting to this BTC/DPCM unsupervised learning scheme in an

attempt to provide a degree of error protection within the channel without sacrificing bandwidth. Finally, the performance of the coding scheme developed above will be discussed in Section V.

4. APPLICATION OF BIT WEIGHTING

As discussed in Section II, the application of bit weighting to a modulation method provides a degree of error protection without requiring the addition of channel coding bits to the source data. For the case in which the number of different energy levels J is less than the number of bits per word n, the relative energy levels for the binary antipodal case were given by (7.17).

Bit weighting may also be applied to multilevel signalling techniques such as QAM. From a channel efficiency standpoint, QAM, with 4 bits/symbol, has a much higher performance potential than binary signalling techniques. Assuming ideal Nyquist pulse shaping and a channel bandwidth of B Hertz; QAM attains a bit rate of 4B bits/s, while binary antipodal modulation is limited to 2B bits/s under the same ideal conditions [31], [6]. It is this theoretical factor of two increase in bit rate which makes QAM attractive.

The signal space diagram for a Grey-coded QAM scheme is presented in Figure 7.16.

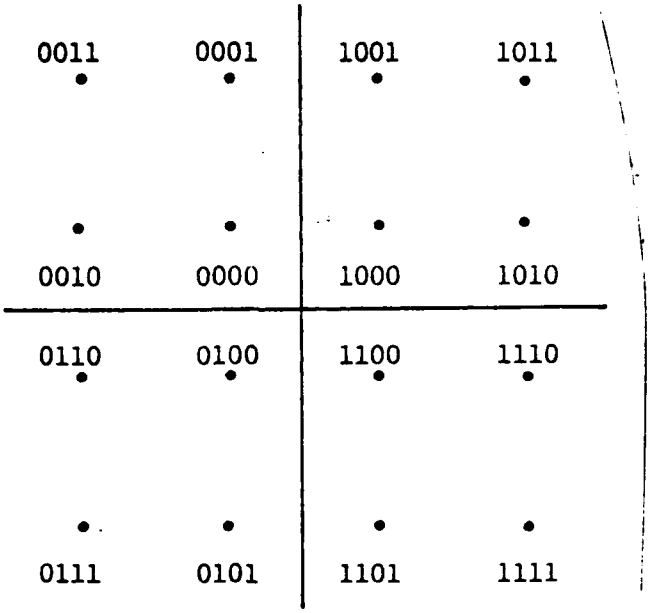


Figure 7.16. Signal Space for Grey-Coded QAM

Assuming additive white Gaussian noise with variance $N_o/2$ (see [20]), with an average channel symbol energy of E, the probability of a symbol error for the QAM scheme is given by [7.31], [7.32]:

$$P_s \cong 3 \cdot Q\left(\sqrt{\frac{E}{5N_o}}\right) = 3 \cdot Q\left(\frac{d}{\sqrt{2N_o}}\right) \qquad (7.45)$$

where d is the signal spacing as denoted in Figure 7.16, and it is assumed that all signals are equiprobable. The individual bit error probabilities are not all equal, however, due to the nature of the Grey coding. Assuming bit 0 is defined to be the least significant bit within a word, and bit 3 the most significant, Sundberg [7] finds the individual bit error probabilities to be

$$P_0 = P_1 = Q\left(\sqrt{\frac{E}{5N_o}}\right)$$

$$P_2 = P_3 = \frac{1}{2} Q\left(\sqrt{\frac{E}{5N_o}}\right) \qquad (7.46)$$

where P_i is the bit error probability in QAM bit i. Thus, the signal set shown in Figure 7.16 is already "weighted" due to the fact that the error probability for bits 2 and 3 is half of the error probability for bits 0 and 1. This signal set is thus naturally used in such a way that the most significant bits within an input digital word are placed in the bit positions 2 or 3.

The signal set described above may be modified in order to accomodate situations in which bit probability of error ratios of other than 2:1 are required. The development of a scheme in which two distinct energy levels may be specified is described below.

Considering one quadrant of the signal space, as shown in Figure 7.17, two distances d_1 and d_2 and a related angle θ may be defined as shown. For the unweighted signal set, $d_1=d_2=d$ and $\tan \theta=1/3$.

The relative bit error probabilities may now be difined as a function of the angle θ, and the relative energy levels e_1 and e_2 may be optimized as a funciton of the A-factors of the bits of interest and the average channel signal to noise ratio E/N_o. Appendix A details the deriviation of the equation for the optimum angle θ as a function of the A-factors and E/N_o. From Appendix A, it is found that the optimum θ is

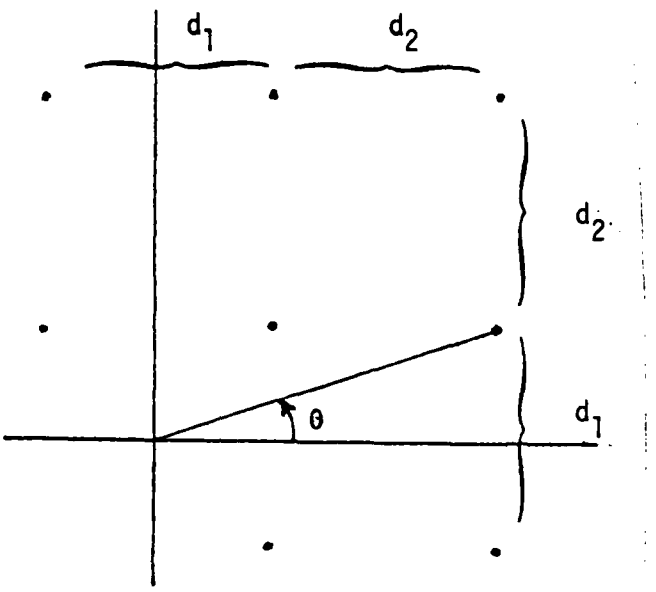


Figure 7.17. Distance Definitions for Two Level Weighted QAM

determined by solving

$$\frac{\partial \epsilon_a^2}{\partial \Theta} = 0 = \sum_{i=0}^{1} A_i \left[\phi\left(\sqrt{\frac{E}{N_o}} \sin\left(\frac{\pi}{4} - \phi\right)\right) \sqrt{\frac{E}{N_o}} \cos\left(\frac{\pi}{4} - \phi\right) \right] \qquad (7.47)$$

$$- \frac{1}{2} \sum_{i=2}^{3} A_i \left[\phi\left(\sqrt{2}\sqrt{\frac{E}{N_o}} \sin\phi\right) \cdot \sqrt{2}\sqrt{\frac{E}{N_o}} \cos\phi \right],$$

where

$$\phi(x) = \frac{1}{2} \exp\left(-x^2/2\right) \qquad (7.48)$$

and A_i is the A-faccor for bit i.

The two level weighted QAM scheme may be applied to the BTC/DPCM source coder output on the following manner. The output of the coder consists of two classes of data: 1) Laplacian quantizer code data from the DPCM loop, and 2) pixel code (bit map) data from the quantization of the actual pixel values. The signal spaces required to implement bit weighting must be derived through application of the A-factors of these two data classes to equation (7.47).

The A-factors for the 4 bit Laplacian quantizer code words were derived according to the procedure outlined by Rydbeck and Sundberg [7.19]; these A-factors are listed in Appendix B. In the case of the pixel code bits, all bits are of equal importance regardless of position since each identifies the correct quantizer level for a given pixel. Assuming that all pixels within a block are of equal importance, it is required that the A-factors for all the bits in the pixel code be equal in order to force equal probability of error for all bits. Since it is the relative magnitude of the A-factors within a word which influences θ in (7.47), rather than their absolute value, all A-factors for pixel code bits were assigned a nominal value of 1.

The application of two level weighted QAM to the BTC/DPCM source code thus involved solving (7.47) based on the two sets of A-factors and the particular channel signal to noise ratio (SNR) of interest. For each value of channel SNR, two QAM signal spaces were required: one "weighted" scheme for the Laplacian quantizer code bits, and an "unweighted" scheme for the pixel code bits.

An example of each of the two signal spaces may be found on pages and . Both plots were generated for the case E/N_0 = 13.9 dB, which

corresponds to an average bit error rate of 10^{-2} for this modulation method. Note that the signal space for the Laplacian quantizer code has been distorted in such a manner that the most significant bits (2 and 3) are protected at the expense of a higher probability of error for bits 0 and 1.

Equation (7.47) was solved numerically for each E/N_0 value of interest, resulting in a value Θ_0. The relative energy levels were then found from (see [7]):

$$\begin{aligned} e_1 &= 10\sin^2 {}_0 \\ e_2 &= 5 \sin^2(\tfrac{\pi}{4} - \Theta_0) , \end{aligned} \tag{7.49}$$

where e_1 is the relative energy assigned to bits 0 and 1, and e_2 is the relative energy assigned to bits 2 and 3. The individual bit error probabilities are then (see Appendix A)

$$\begin{aligned} P e_{0,1} &= Q\left(\sqrt{e_1 \cdot \frac{E}{5N_0}}\right) \\ P e_{2,3} &= \frac{1}{2} Q\left(\sqrt{e_2 \cdot \frac{E}{5N_0}}\right) \end{aligned} \tag{7.50}$$

where $Q(\cdot)$ is the standard Q-function [20], and E/N_0 is the channel SNR.

The error probabilities defined in (7.50) were used in a channel simulation routine which simulated bit errors for the complete forward and inverse BTC/DPCM coding operation. These simulations were performed for bit error rates of 10^{-6}, 10^{-4}, and 10^{-2} for the BTC/DPCM coder

described in Section 3. As a basis of comparison, standard cosine transform coding at the same bit rate (1.5 bits/pixel) was also simulated at these error rates. The results of these simulations are presented in the following section.

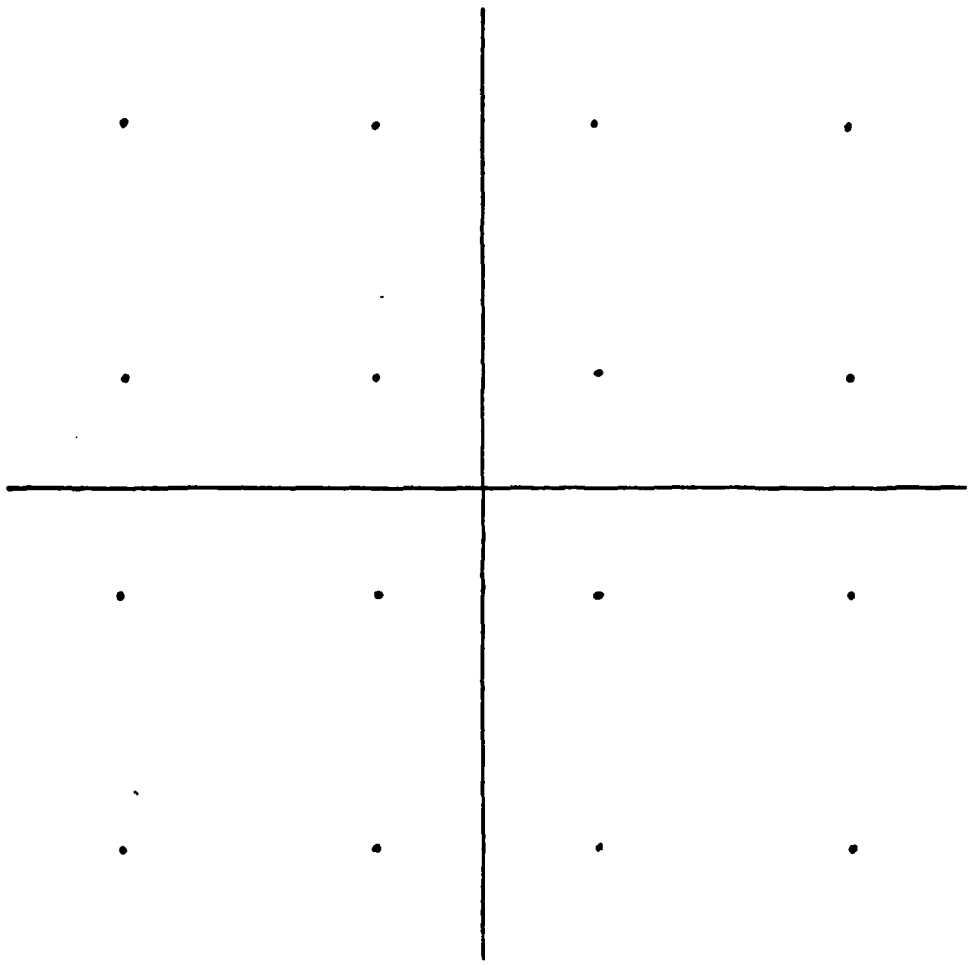


Figure 7.18. Signal Space for Pixel Code at 10^{-2} BER

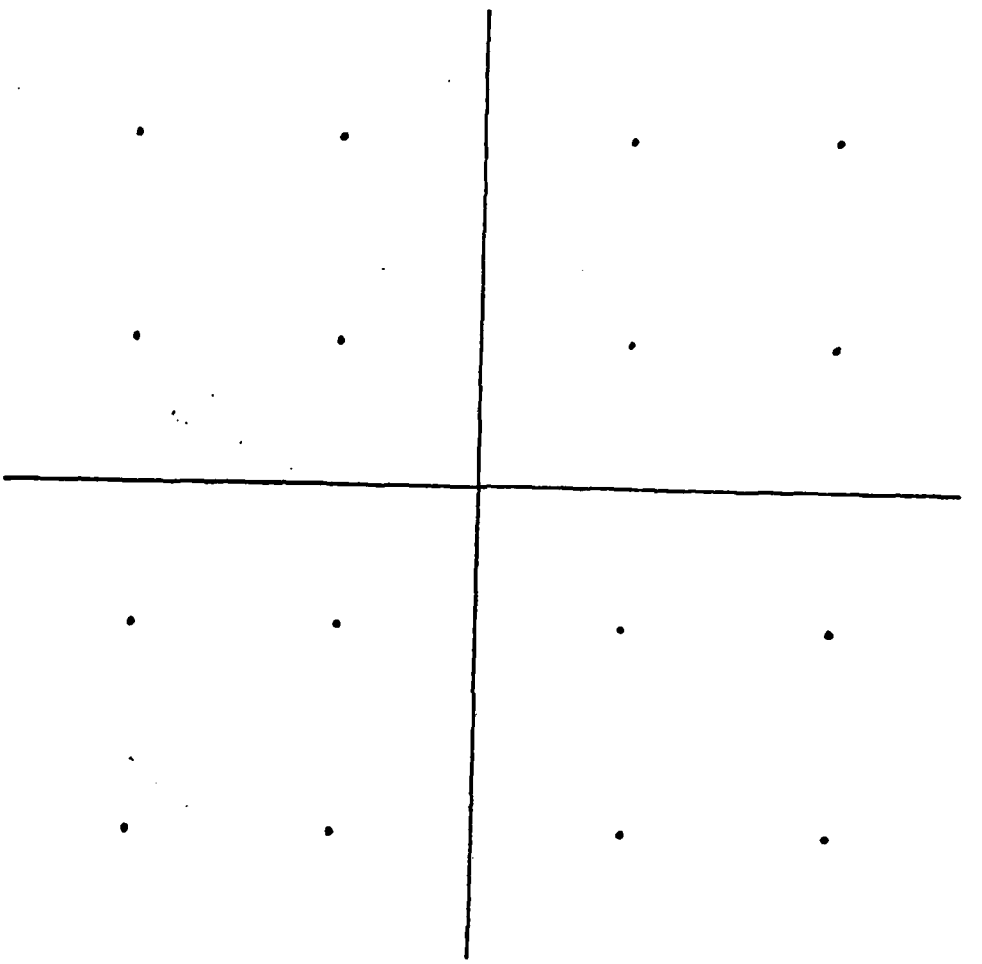


Figure 7.19. Signal Spcae for Quantizer Code at 10^{-2} BER

5. CHANNEL SIMULATION RESULTS

The goal of this research was to develop an image transmission scheme which would yield images comparable in quality to transform coding schemes under noisy channel conditions while requiring only a relatively simple hardware implementation. As directed in Sections 1 and 2 , the BTC spatial coder was chosen as the basis for this image transmission system because of its implementation simplicity and its good performance in terms of image quality.

The performance of the basic BTC coder at 2 bits/pixel was discussed in terms of MSE in Section 3 (see Table 7.1, page 136). The performance of this coder in visual terms may be evaluated by comparing Figure 7.20 to Figure 7.21 on page 172. Figure 7.20 is the original Girl image, coded at 8 bits/pixel. Figure 7.21 is coded using BTC at 2 bits/pixel with uniform quantizers for the block means and standard deviations. Figure 7.22 shows the result of attempting to reduce the bit rate of this basic BTC scheme to 1.5 bits/pixel. Severe false contouring is apparent in this image.

Figure 7.23 demonstrates the result of the BTC/DPCM coding with unsupervised learning at the rate of 1.5 bits/pixel. Note the improvement obtained in relation to the image of Figure 7.22. One deficiency apparent in Figure 7.23 is that of the verticle stripes, which are most apparent in the left-hand "background" section of this image. These stripes occur at the beginning of the DPCM stripes, which run horizontally, and are due to the fact that 4 bit uniform quantizers were used to code the block means and standard deviations at the beginning of the stripes. The 4 bit uniform quantizers

Figure 7.20
Original Girl Image

Figure 7.21
Unmodified BTC (Uniform Quantizers) at 2.0 bits/pixel;
10^{-6} BER

Figure 7.22
Unmodified BTC (Uniform Quantizers) at 1.5 bits/pixel;
10^{-6} BER

Figure 7.23
BTC/DPCM at 1.5 bits/pixel;
10^{-6} BER

were used in order to realize a bit rate of 1.500 bits/pixel. If a small increase in bit rate could be tolerated, these initial blocks could be coded with 6 bit uniform quantizers; Figure 7.24 (page 174) was coded in this manner. The bit rate for this image is 1.531 bits/pixel.

All of the images described above were uncorrupted by channel errors. However, as noted previously, a realistic image transmission system must consider the effect of channel errors on the output images. Figure 7.25 demonstrates the effect of a 10^{-2} bit error rate (BER) on a 1.5 bit/pixel cosine transform coded image. Transform coding typically distributes the effect of channel errors throughout the transform blocks, which in this case were 16 X 16 pixels each. Figure 7.26 illustrates the effect of a 10^{-2} BER on an image coded at 1.516 bits/pixel with the BTC/DPCM unsupervised learning scheme without bit weighting. Figure 7.27 is also the result of the BTC/DPCM coder, at 10^{-2} BER, but bit weighting was implemented in this case. Though the visual improvement is not dramatic in this particular image, the general effect of bit weighting was to decrease the number of serious errors at a cost of increasing the number of minor errors.

The MSE results for all of the images discussed above are tabulated in Table 7.6.

Figure 7.24
BTC/DPCM at 1.53 bits/pixel;
10^{-6} BER

Figure 7.25
Cosine Transform Coding at
1.5 bits/pixel;
10^{-2} BER

Figure 7.26
BTC/DPCM at 1.51 bits/pixel;
10^{-2} BER

Figure 7.27
BTC/DPCM at 1.51 bits/pixel;
10^{-2} BER

TABLE 7.6. MSE RESULTS FOR THE IMAGES OF FIGURES 7.20-7.27

Figure No.	Bits/Pixel	BER	MSE
20	8.000	0	0.0
21	2.000	0	35.07
22	1.500	0	60.350
23	1.500	0	42.298
24	1.532	0	39.443
25	1.500	10^{-2}	336.787
26	1.516	10^{-2}	178.015
27	1.516	10^{-2}	86.874

The figures listed above for the BTC/DPCM coder (Figures 7.26 and 7.27) are the result of a slight modification of the scheme described in Section III. During the initial channel simulations of this coder at high error rates, it was discovered that image quality decreased rapidly at error rates in the neighborhood of 10^{-2}. This effect was due to the distruptive effect of errors on the unsupervised learning algorithm. At high error rates, the insupervised learning loop in the receiver was unable to track the decisions made by the loop in the transmitter, with the result that the unsupervised learning increased the degradation effect of the channel errors.

It was determined that this negative effect of the learning could be eliminated by inhibiting the updating process whenever an error occured. It was not necessary for the receiver to know the precise bit in error, or

have the capability of correcting the error; the inhibit could be performed by simply knowing which word contained an error. This level of error detection could be realized through the use of a two-dimensional parity code. This code naturally adds overhead bits, but the increase in data rate is slight; if data blocks of 32 words by 32 words (16 X 16 BTC image blocks) were coded using a two-dimensional parity code, the increase in bit rate would be 0.0156 bits/pixel. All results for the BTC/DPCM coder presented here have been generated on the assumption that such an error detection scheme is operating.

The MSE results of the BTC/DPCM channel simulation for all three images are summarized in Table 7.7 (page 179).

As can be seen from the results given in Table 7.7, the relatively small error rate of 10^{-4} had little effect on system performance, while the high error rate of 10^{-2} did result in noticeable degradation. Bit weighting in general improved the system performance at this high error rate, however. The results of the BTC/DPCM coding used in conjunction with weighted QAM are shown in Figures 7.28 through 7.30 (page 177) for the Moon image, and in Figures 7.31 through 7.33 for the Aerial image.

The MSE results for cosine transform coding channel simulation of the three images are provided in Table 7.8 as a basis for comparison with the BTC/DPCM coded images.

Figure 7.28
Original Moon Image

Figure 7.29
BTC/DPCM at 1.51 bits/pixel
with bit weighting
10^{-6} BER

Figure 7.30
BTC/DPCM at 1.51 bits/pixel
10^{-2} BER

Figure 7.31
Original Aerial Image

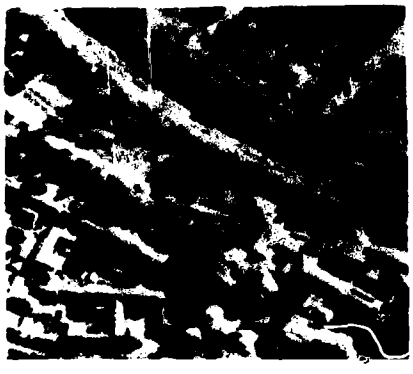


Figure 7.32
BTC/DPCM at 1.51 bits/pixel
10^{-6} BER

Figure 7.33
BTC/DPCM at 1.51 bits/pixel
10^{-2} BER

TABLE 7.7. RESULTS OF THE BTC/DPCM CHANNEL SIMULATIONS

(MSE)

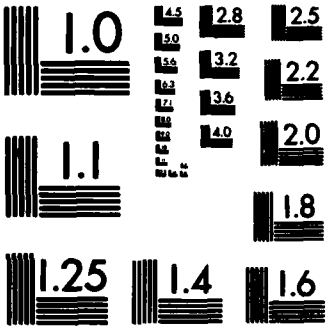
Image	BER	Without Bit Weighting	With Bit Weighting
Girl	10^{-6}	42.298	42.298
	10^{-4}	42.381	52.426
	10^{-2}	178.015	86.874
Moon	10^{-6}	46.109	46.109
	10^{-4}	46.277	46.351
	10^{-2}	141.149	101.601
Aerial	10^{-6}	147.358	147.358
	10^{-4}	149.149	149.204
	10^{-2}	450.546	277.933

AD-A134 187 BIT WEIGHTING AND SOFT DECISION DEMODULATION FOR IMAGE COMPRESSION & ERRO.. (U) TEXAS A AND M UNIV COLLEGE STATION DEPT OF ELECTRICAL ENGINEE.. J D GIBSON ET AL. 3/3

UNCLASSIFIED AUG 83 AFWAL-TR-83-1099 F33615-80-C-1002 F/G 9/2 NL

END

FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TABLE 7.8. RESULTS OF COSINE TRANSFORM CODING CHANNEL SIMULATIONS

Image	BER	Image MSE
Girl	10^{-6}	32.407
	10^{-4}	37.516
	10^{-2}	336.787
Moon	10^{-6}	31.024
	10^{-4}	33.084
	10^{-2}	186.112
Aerial	10^{-6}	50.385
	10^{-4}	53.074
	10^{-2}	287.647

Inspection of Tables 7.7 and 7.8 reveals that although the BTC/DPCM scheme with unsupervised learning was unable to match the very good performance of the cosine transform coder under ideal (error free) conditions, the transform coder suffered greater degradation at an error rate of 10^{-2}. With the application of bit weighting, the BTC/DPCM scheme provided noticeably improved performance for both the Girl and Moon images at 10^{-2} BER. When operating on the Aerial image, the BTC/DPCM system with weighting provided only a marginal improvement over the transform coder at this error rate.

The conclusions drawn from these simulation results are summarized in the following section.

6. CONCLUSIONS

As stated in the introduction, the goal of the research presented in this task was to develop the basis for a workable image transmission system which would provide good quality images, low bandwidth requirements, and error protection for non-ideal channels.

A spatial coding method (Block Truncation Coding) was chosen as the basis for the system source coder due to the excellent efficiency of BTC in terms of image quality relative to computational requirements. As demonstrated in Section III, the basic BTC coder proposed by Delp and Mitchell in [4] performed well at a bit rate of 2.0 bits/pixel, but its performance deteriorated seriously as the bit rate was decreased to 1.5 bits/pixel. Based on this observation, and noting the need to achieve as low a bit rate as possible in order to satisfy the bandwidth constraint, a modified form of BTC was developed. This modified form incorporated a DPCM loop into the BTC coder, and utilized a novel adaptive quantization scheme based of an unsupervised learning algorithm developed by Griswold and Sayood [29]. This source coder operates at 1.51 bits/pixel without the need for complex, high-speed implementation hardware.

The modified source coder was shown to be compatible with a weighted QAM modulation scheme. The weighted energy levels were calculated for the source code produced by the modified BTC coder, and channel simulations were performed at bit error rates of 10^{-6}, 10^{-4}, and 10^{-2}. To aid in evaluating the performance of the proposed system, these simulations were also performed on a cosine transform coder operating at 1.5 bitx/pixel.

As indicated by the numerical (MSE) and visual results presented in Section V, the source coder developed as a result of this research performs well at high error rates in comparison to the cosine transform coder. Under error-free conditions, The performance of the modified BTC coder does not match that provided by the transform coder, though for two of the three images tested (the Girl and Moon images) the difference in performance was not substantial.

A fair evaluation of the modified BTC coder must consider the decrease in performance which occurs when images such as the Aerial image are being processed. This degradation appears to be a result of the two-level quantizer inherent in the BTC technique; in images such as the Aerial in which there exists a high degree of variability in the pixel values within a block (4 X 4 pixels), the two-level quantizations scheme naturally leads to a relatively large error between the coded and original images. It should be noted, however, that the cosine transform coder also suffered a degradation in performance when operating on the Aerial image as compared to the Girl and Moon images.

In general, the image coding and transmission scheme developed in the course of this research satisfies the original goals as outlined in the introduction of this paper. The modified source coder required only relatively simple computations to be performed for its implementation, thus leading to a straightforward hardware realization. The weighted QAM technique provided a degree of error protection while attaining efficient use of available channel bandwidth. This scheme shows promise in applications in which a relatively simple hardware realizaion is required, Bandwidth is constrained, and a high channel error rate is unavoidable.

APPENDIX A

It is required to optimize the two level signal set defined in Figure 7.17 (page 165) for minimum digital noise as a function of the signal set parameters d_1 and d_2, or alternatively, as a function of the single parameter θ. Following the procedure given in [20] for determining the probability of bit errors for an arbitrary rectangular signal set (see page 254 of this reference), it is found that for the signal set defined in Figures 7.16 and 7.17 in the presence of additive white Gaussian noise with variance $N_0/2$, the individual bit error probabilities are given by

$$P[E]_0 = P[E]_1 = Q\left[\frac{d_2}{\sqrt{2N_0}}\right] + 1/2\ Q\left[\frac{2d_1 + d_2}{\sqrt{2N_0}}\right] \qquad (1)$$

$$P[E]_2 = P[E]_3 = 1/2\ Q\left[\frac{2d_2 + d_1}{\sqrt{2N_0}}\right] + 1/2\ Q\left[\frac{d_1}{\sqrt{2N_0}}\right] \qquad (2)$$

where $Q(d/\sqrt{2N_0})$ is the probability of error for two signals separated by a distance d. Based on these bit error probabilities, the digital noise as defined by Sundberg [7.7] is given by

$$= \sum_{i=0}^{1} A_i \left[Q(d_2/\sqrt{2N_0}) + 1/2\ Q(2d_1 + d_2/\sqrt{2N_0})\right]$$

$$+ \sum_{i=3}^{4} A_i \left[(1/2)Q(2d_2 + d_1/\sqrt{2N_0})\right.$$

$$\left. + (1/2)Q(d_1/\sqrt{2N_0})\right] . \qquad (3)$$

Solving for the digital noise in terms of the angle θ and simplifying yields

$$= \sum_{i=0}^{1} A_i \, Q(\sqrt{E/N_0} \sin((\pi/4) - \theta))$$

$$+ \sum_{i=2}^{3} A_i \, (1/2) Q(\sqrt{2E/N_0} \sin \theta) \, .$$

This equation may be optimized by taking the partial derivative with respect to θ and setting the resulting equation to zero. This operation, accomplished through the application of Leibnitz' rule (considering the Q-function in its integral form), leads to equation (7.47) of the text.

APPENDIX B

The A-factors for the 4-bit Laplacian quantizers discussed in the text were determined to be:

Bit #	A-factor
0	0.646091
1	2.003751
2	5.378554
3	18.955492

SECTION VIII

COMBINED BIT WEIGHTING AND SOFT DECISION DEMODULATION

1.0. INTRODUCTION

Bit weighting (BW) and soft decision demodulation (SDD) as described in the preceding sections are obviously quite different approaches to the problem of reducing channel error effects in data compression systems. Bit weighting is used at the transmitter to increase the transmitted energy of significant bits, while soft decision is implemented at the receiver to protect against catastrophic errors in the most significant bits. These two techniques are complementary and can be employed simultaneously in a data compression system.

2.0. COMBINED BW/SDD SIMULATIONS

The combined BW/SDD simulations were performed for an additive white Gaussian noise channel with a channel SNR = 7.35 dB for an unprotected independent bit error probability of 10^{-2}. Laplacian quantizers were used for the coefficients, and the bit weighting energies for one through eight bits are listed in Table 8.1.

For SDD, the four most significant bits (MSBs) of coefficient 1 (dc coefficient), the three MSBs of coefficient 2, and the three MSBs of coefficient 17 were monitored. The soft decision thresholds for these bits are given in Table 8.2.

Monte Carlo simulation runs were performed to compare bit weighting alone, soft decision demodulation alone, and combined bit weighting and soft decision demodulation. For these experiments, images were coded at 1 bit/pel using the 2D-DCT with 16 by 16 blocks.

Table 8.3 lists the average SNR of SDD alone, BW alone, and combined BW/SDD for the Monte Carlo runs performed at a BER = 10^{-2}. In terms of SNR, BW and SDD achieve almost identical performance, while joint BW/SDD provides a 1.6 - 1.7 dB

TABLE 8.1. BIT WEIGHTING ENERGIES

No. of Bits	Energy Allocation Bit Number 1	2	3	4	5	6	7	8
1	0.100E+01							
2	0.110E+01	0.898E+00						
3	0.130E+01	0.120E+01	0.508E+00					
4	0.141E+01	0.143E+01	0.821E+00	0.341E+00				
5	0.142E+01	0.142E+01	0.105E+01	0.708E+00	0.405E+00			
6	0.142E+01	0.142E+01	0.108E+01	0.108E+01	0.622E+00	0.365E+00		
7	0.126E+01	0.142E+01	0.126E+01	0.956E+01	0.956E+00	0.573E+00	0.573E+00	
8	0.120E+01	0.135E+01	0.120E+01	0.120E+01	0.896E+00	0.896E+00	0.629E+00	0.629E+00

TABLE 8.2. SOFT DECISION THRESHOLDS

Coefficient	Bit No.	Threshold
c_1	1	1.92922994E-01
	2	2.50516981E-01
	3	1.69860646E-01
	4	3.28001715E-02
c_2	1	1.25782982E-01
	2	1.66228101E-01
	3	7.80232921E-02
c_{17}	1	1.25782982E-01
	2	1.66228101E-01
	3	7.80232921E-02

TABLE 8.3. COMBINED BIT WEIGHTING/SOFT DECISION PERFORMANCE (BER=10^{-2})

	SDD	BW	Combined BW/SDD
SNR(dB)	10.66	10.56	12.26

improvement over either technique alone.

Figures 8.1 - 8.3 shown typical reconstructed images corresponding to the comparison in Table 8.3. Figure 8.1 shows a reconstructed image at a BER = 10^{-2} for SDD alone, Figure 8.2 is BW alone, and Figure 8.3 shows combined BW/SDD. The improvement in Figure 8.3 is obvious, and using BW and SDD together always has produced a better image.

3.0. CONCLUSIONS

Joint BW/SDD is recommended whenever the specific application admits their implementation.

Figure 8.1. SDD Alone at BER = 10^{-2}

Figure 8.2. BW Alone at BER = 10^{-2}

Figure 8.3. Combined BW/SDD at BER = 10^{-2}

REFERENCES

Section II

[1] N. Rydbeck and C. E. Sundberg, "Analysis of digital errors in nonlinear PCM systems," IEEE Trans. Comm., vol. COM-24, pp. 59-65, Jan. 1976.

[2] C. E. Sundberg, "Soft decision demodulation for PCM encoded speech signals," IEEE Trans. Comm., vol. COM-26, pp. 854-859, June 1978.

Section III

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, pp. 90-93, Jan. 1974.

[2] W. K. Pratt, Digital Image Processing, New York: Wiley-Interscience, 1978, Ch. 10.

[3] A. G. Tescher, "Transform image coding," Advances in Electronics and Electron Physics, Suppl. 12, New York: Academic Press, pp. 113-115.

[4] H. Murakami, Y. Hatori, and H. Yamamoto, "Comparison between DPCM and Hadamard transform coding in the composite coding of the NTSC color TV signal," IEEE Trans. Commun., vol. COM-30, pp. 469-479, March 1982.

[5] A. N. Netravali and J. O. Limb, "Picture coding: A review," Proc. IEEE, vol. 68, no. 13, pp. 366-406, March 1980.

[6] J. Max, "Quantizing for minimum distortion," IEEE Trans. Inform. Theory, vol. IT-13, pp. 336-338, April 1967.

[7] P. A. Wintz and A. J. Kurtenbach, "Waveform error control in PCM telemetry," IEEE Trans. Inform. Theory, vol. IT-14, pp. 7-12, March 1960.

[8] L. A. Marascuilo and M. McSweeny, Nonparametric and Distribution-Free Methods for the Social Sciences, Monterrey, California: Brooks-Cole, 1977.

[9] S. D. Silvey, Statistical Inference, London: Chapman Hall, 1975, Ch. 9.

Section IV

[1] C. E. W. Sundberg, "Soft decision demodulation for PCM encoded speech signals," IEEE Trans. Commun., vol. COM-26, pp. 845-859, June 1978.

[2] K. N. Ngan and R. Steele, "Enhancement by simple statistical methods of PCM and DPCM images corrupted by transmission errors," Conf. Rec., IEEE 1980 Nat. Telecommun. Conf., Houston, Texas, Nov. 30 - Dec. 4, pp. 50.5.1 - 50.5.5.

[3] P. A. Wintz and A. J. Kurtenbach, "Waveform error control in PCM telemetry," IEEE Trans. Inform. Theory, vol. IT-14, pp. 650-661, September 1968.

[4] J. Max, "Quantizing for minimum distortion," IEEE Trans. Inform. Theory, vol. IT-6, pp. 7-12, March 1960.

[5] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, New York: Wiley, 1965.

[6] N. C. Griswold and J. D. Gibson, "Bit Weighting and Soft Decision Demodulation for Image Compression Error Control," Phase I Final Report, AF Contract F33615-80-C-1002, September 1980.

[7] J. W. Modestino and D. G. Daut, "Combined source-channel coding of images," IEEE Trans. Commun., vol. COM-27, pp. 1644-1659, November 1979.

[8] J. W. Modestino, D. G. Daut, and A. L. Vickers, "Combined source-channel coding of images using the block cosine transform," IEEE Trans. Commun., vol. COM-29, pp. 1261-1274, September 1981.

[9] W. K. Pratt, Digital Image Processing, New York: Wiley, 1978, p. 675.

[10] J. A. Roese, W. K. Pratt, and G. S. Robinson, "Interframe cosine transform image coding," IEEE Trans. Commun., vol. COM-25, pp. 1329-1339, November 1977.

Section V

[1] K. N. Ngan and R. Steele, "Enhancement of PCM and DPCM images corrupted by transmission errors," IEEE Trans. Commun., vol. COM-30, pp. 257-265, Jan. 1982.

[2] O. R. Mitchell and A. J. Tabatabai, "Channel error recovery for transform image coding," IEEE Trans. Commun., vol. COM-29, pp. 1754-1762, Dec. 1981.

[3] R. C. Reininger and J. D. Gibson, "Soft decision demodulation and transform coding of images," Conf. Rec., 1982 Int. Conf. Commun., Phila., PA, June 13-17, pp. 4H.3.1-4H.3.6.

[4] R. A. Duryea, "Performance of a Source/Channel Encoded Imagery Transmission System," M. S. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Dec. 1979.

[5] J. W. Modestino, D. G. Daut, and A. L. Vickers, "Combined source-channel coding of images using the block cosine transform," IEEE Trans. Commun., vol. COM-29, pp. 1261-1274, Sept. 1981.

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

[7] P. A. Wintz and A. J. Kurtenbach, "Waveform error control in PCM telemetry," IEEE Trans. Inform. Theory, vol. IT-14, pp. 650-661, September 1968.

[8] J. Max, "Quantizing for minimum distortion," IEEE Trans. Inform. Theory, vol. IT-6. IT-6, pp. 7-12, March 1960.

[9] R. C. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," submitted for publication, IEEE Trans. Commun., April 1982.

[10] N. Rydbeck and C. E. Sundberg, "Analysis of digital errors in non-linear PCM systems," IEEE Trans. Commun., vol. COM-24, pp. 59-65, Jan. 1976.

Section VI

[1] E. Bedrosian, "Weighted PCM", IRE Trans. on Info. Theory, Vol. IT-4, pp. 45-49, March 1958.

[2] C.E. Sundberg, "Optimum Weighted PCM for Speech Signals", IEEE Transactions on Communications, Vol. COM-26, No. 6, pp. 872-881, June 1978.

[3] N. Rybeck, C.E. Sundberg, "Analysis of Digital Errors in Nonlinear PCM Systems", IEEE Transactions on Communications", Vol. COM-24, pp. 59-65, Jan. 1976.

[4] C.E. Sundberg, N. Rydbeck, "Pulse code modulation with error correcting codes for TDMA Satellite Communication Systems", Ericsson Technics, Vol. 32, (1976), No. 1, pp. 3-56.

[5] N. Rydbeck and C.E. Sundberg, "PCM/TDMA Satellite Communication System with Error Correcting and Error Detecting Codes", Ericsson Technics, Vol. 32, No. 3, pp. 195-247, 1976.

[6] J.M. Wozencraft and M.I, Jacobs, Principles of Communication Engineering, Wiley, 1965.

[7] A. Habibi, "Hybrid Coding of Pictorial Data", IEEE Trans. Commun., Vol. COM-22, pp. 614-624, May 1974.

[8] W.K. Pratt, ed., Image Transmission Techniques, New York, Academic Press, 1979.

[9] R.C. Gonzalez and P. Wintz, Digital Image Processing, Addison-Wesley Publishing Co., Reading Mass., 1977.

[10] W.K. Pratt, Digital Image Processing, New York: Wiley, 1978.

[11] R.C. Reininger, J.D. Gibson, "Distributions of the two Dimensional DCT Coefficients for Images", Departmental tech note, Dept. EE, Texas A&M University, College Station, Texas 77843.

Section VII

[1] T.W. Goeddel and S.C. Bass, "A Two-Dimensional Quantizer for Coding of Digital Imagery", IEEE Trans. Commun., Vol. COM-29, pp. 60-67, January 1981.

[2] A. Habibi, "Hybrid Coding of Pictorial Data", IEEE Trans. Commun., Vol. Com-22, pp. 614-624, May 1974.

[3] O.R. Mitchell, S.C. Bass, E.J. Delp, T.W. Goeddel, and T.S. Huang, "Image Coding for Photoanalysis", Proc. Soc. Inf. Disp., Vol. 21/3, pp. 279-292, July 1981.

[4] E.J. Delp and O.R. Mitchell, "Image Compression Using Block Trucation Coding", IEEE Trans. Commun., Vol. COM-27, pp. 1335-1342.

[5] D.H. Morais and K. Feher, "NLA-QAM: A New Method for Generating High Power QAM Signals through Nonlinear Amplifications", Proc. Int. Conf. Commun., Denver, CO, June 14-18, Vol. 1, ppl 3.3.1-3.3.6.

[6] G.J. Foschini, R.D. Gitlin and S.B. Weinstein, "Optimization of Two-Dimensional Signal Constellations in the Presence of Gaussian Noise", IEEE Trans. Commun., Vol. COM-22, pp. 28-38, January 1974.

[7] C.E. Sundberg, "Optimum Weighted PCM for Speech Signals", IEEE Trans. Commun., Bol. COM-26, pp. 872-881, June 1978.

[8] W.K. Pratt, ed., Image Transmission Techniques, New York, Academic Press, 1979.

[9] J. Max, "Quantizing for Minumum Distortion", IRE Trans. Info. Theory, Vol. IT-6, pp. 7-12, March 1960.

[10] S.A. Kassam, "Quantization Based on the Mean-Absolute-Error Criterion", IEEE Trans. Commun., Vol. COM-26, pp. 267-270, February 1978.

[11] O.R. Mitchell, E.J. Delp, and S.G. Carlton, "Block Truncation Coding: A New Approach to Image Compression", Conference Record, 1978 IEEE International Conference on Communications (ICC '78), Vol. 1, June 4-7, 1978, pp. 12B.1.1-12B.1.4.

[12] E.J. Delp and O.R. Michell, "Some Aspects of Moment Preserving Quanitzers", Conference Record, 1979 IEEE International Conference on Communications (ICC '79), Vol. 1, June 19-14, 1979, pp. 7.2.1-7.2.5.

[13] D.R. Halverson, N.C. Griswold, and G.L. Wise, "On Generalized Block Truncation Coding Quantizers for Image Compression", Proc. of the 1982 Conference of Information Sciences and Systems, March 19-19, 1981; to be published.

[14] W.H. Chen and C.H. Smith, "Adaptive Coding of Monochrome and Color Images", IEEE Trans. Commun. Vol. COM-25, pp. 1285-1292, November. 1977.

[15] D. Jameson and L.M. Hurvich, editors, Handbook of Sensory Physiology: Visual Psychophysics, New York, Springer-Verlag, 1972.

[16] E. Bedrosian, "Weighted PCM", IRE Trans. Info. Theory, Vol. IT-4, pp. 45-49, March 1958.

[17] I.T. Young and J.C. Mott-Smith, "On Weighted PCM", IEEE Trans. Info. Theory, Vol. IT-11, pp. 596-597, Oct. 1965.

[18] A.G.J. Holt and J. Yates, "Comments on Weighted PCM", IEEE Trans. Info. Theory, Vol. IT-18, pp. 817, November 1972.

[19] N.Rydbeck and C.E. Sundberg, "Analysis of Digital Errors in Non-linear PCM Systems", IEEE Trans. Commun., Vol. COM-24, pp. 59-65, January 1976.

[20] J.M. Wozencraft and M.I. Jacobs, Principles of Communication Engineering, Wiley, 1965.

[21] D.J. Healy and O.R. Mitchell, Digital Video Bandwidth Compression Using Block Truncation Coding", IEEE Trans. Commun., Vol. COM-29, pp. 1809-1817, December 1981.

[22] H.G. Mussmann, "Predictive Image Coding", Image Transmission Techniques, W.K. Pratt, ed., Academic Press, 1979.

[23] W.C. Adams and C.E. Giesler, "Quantizing Characteristics for Signals Having Laplacian Amplitude Density Function", IEEE Trans. Commun., Vol. COM-26, pp. 1295-1297, August 1978.

[24] H.G. Mussmann, "The General Model of a Switched Quantizer", Proc. Int. Zurich Seminar Digital Communication, pp. C1(1)-C1(7), 1974.

[25] J.D. Gibson, "Adaptive Prediction in Speech Differential Encoding Systems", Proc. of IEEE, Vol. 68, No. 4, pp. 488-525, April 1980.

[26] P. Cummiskey, N.S. Jayant, and J.L. Flanagan, "Adaptive Quantization in Differential PCM Coding of Speech", Bell System Technical Journal, Vol. 52, pp. 1105-1118, September 1973.

[27] L.S. Goldong and P.M. Schultheis, "Study of an Adaptive Quantizer", Proc. IEEE, Vol. 55, pp. 293-297, March 1967.

[28] D.J. Goodman and A. Gersho, "Theory of an Adaptive Quantizer", IEEE Trans. Commun., Vol. COM-22, No. 8, pp. 1037-1045, August 1975.

[29] N.C. Griswold and K. Sayood, "Unsupervised Learning Approach to Adaptive Differential Pulse Code Modulation", IEEE Trans. on Pattern Analysis and Machine Intelligence, to be published.

[30] E.A. Patrick, Fundamentals of Pattern Recognition, Prentice-Hall, 1972.

[31] M. Schwartz, Information Transmission, Modulation, and Noise, McGraw Hill, 1980.

[32] M.K. Simon and J.G. Smith, "Hexagonal Multiple Phase-and-Amplitude Shift Keyed Signal Sets", IEEE Trans. Commun., Vol. COM-21, pp. 1108-1115, October 1973.

Section VIII

[1] C.-E. Sundberg, "Soft decision demodulation for PCM encoded speech signals," IEEE Trans. Comm., vol. COM-26, pp. 854-859, June 1978.

[2] N. Rydbeck and C.-E. Sundberg, "Analysis of digital errors in nonlinear PCM systems," IEEE Trans. Comm., vol. COM-24, pp. 59-65, Jan. 1976.

*U.S.Government Printing Office: 1983 — 759-062/506

END

FILMED

11-83

DTIC