A TRANSMISSION SYSTEM FOR DPCM ENCODED MONOCHROME IMAGES OVER A NOISY CHANNEL

by

SINCLAIR CHI

B.A.Sc.(E.E.), University of British Columbia, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

June 1998

© Sinclair Chi, 1998
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical and Computer Engineering

The University of British Columbia
Vancouver, Canada

Date JUNE 29/98
Abstract

Due to the high correlation of neighbouring pixel values of a digital image, DPCM picture transmission has been a well investigated area of image compression. DPCM has found its way into standards such as JPEG and MPEG which have revolutionized the multimedia industry. However, because of the high noise sensitivity of DPCM compressed images, techniques must be developed to reduce the disturbing effects of channel errors. Hence, a DPCM image transmission system over a binary symmetric channel is studied, along with techniques used to improve picture quality. Two concepts which are the foci of the investigation are joint source/channel decoding (JSCD) and post-processing.

The DPCM encoded image usually exhibits some residual redundancy. Thus the encoded image can be modeled as a first-order Markov model, and a bit error rate minimizing decoder takes advantage of the source statistics to correct errors. This is a type of constrained joint source/channel decoding, and does not use added redundancy in the form of FEC to improve picture quality. Usually, side information consisting of source statistics need to be transmitted to the decoder. It is found that an iterative JSCD system which needs no side information has the same error performance as a system which requires error free transmission of the source statistics, but has a larger decoding delay. Furthermore, the JSCD system was generalized to one using a second-order Markov model and the improvement was found to be negligible.

JSCD decoding at a very high probability of error, does not correct all of the channel errors; and there remains some very noticeable errors. Post-processing techniques have been applied to corrupted DPCM images and thus it is proposed that the similar techniques be applied to JSCD DPCM images. A thresholding technique is developed to locate the errors and then
correct the erroneous symbols. Symbol correction methods which make use of source statistics are compared to a correction technique based solely on the analysis of the decoded DPCM image.

Subjective and objective results are presented with the use of the output images and signal-to-noise ratio graphs. These measures demonstrate that the proposed system of a concatenated source/channel decoder followed by a post-processor improves a noise corrupted DPCM image quite effectively.
# Table of Contents

Abstract ii

List of Tables vi

List of Figures vii

Acknowledgment ix

Chapter 1 Introduction 1

1.1 Motivations and Objectives ................................................................. 2

1.2 Thesis Outline .................................................................................... 3

Chapter 2 DPCM Image Compression 5

2.1 DPCM System .................................................................................. 6

2.2 Prediction Coefficient ......................................................................... 7

2.2.1 Chang Donaldson Prediction Coefficient ...................................... 10

2.3 Quantization ..................................................................................... 11

2.3.1 Uniform Quantizer .......................................................................... 11

2.3.2 Nonuniform Pdf Optimized Quantizer ........................................ 12

2.4 Simulation and Numerical Results .................................................... 13

Chapter 3 Joint Source Channel Decoding 18

3.1 Viterbi Decoding Using Residual Redundancy .................................. 19

3.1.1 Training Sequence Sensitivity .................................................... 26

3.1.2 Iterative JSCD ............................................................................... 29

3.1.3 Code Word Mapping ............................................................... 32

3.2 JSCD Using A Second-Order Markov Model ................................... 37

Chapter 4 Post Processing Techniques 42
List of Tables

Table 3.1  Code Word Mapping for 2-Bit DPCM System ..................................................32
Table 3.2  Code Word Mapping for 3-Bit DPCM System ..................................................33
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Histogram of Lenna Image and Difference of Pixel-to-Pixels of Lenna</td>
<td>5</td>
</tr>
<tr>
<td>2.2</td>
<td>Differential Pulse Code Modulation System</td>
<td>6</td>
</tr>
<tr>
<td>2.3</td>
<td>Differential Pulse Code Modulation System with Predictor</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>2-Bit DPCM System Using Lloyd Max Quantization</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>3-Bit DPCM System Using Lloyd Max Quantization</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>Comparison of 2-Bit and 3-Bit DPCM Systems</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Basic Elements of a Digital Communications System</td>
<td>19</td>
</tr>
<tr>
<td>3.2</td>
<td>Trellis Diagram for 2-Bit DPCM + JSCD System</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>2-Bit DPCM + JSCD System for Classical and Reoptimized Prediction Coefficients</td>
<td>23</td>
</tr>
<tr>
<td>3.4</td>
<td>3-Bit DPCM + JSCD System for Classical and Reoptimized Prediction Coefficients</td>
<td>23</td>
</tr>
<tr>
<td>3.5</td>
<td>2-Bit DPCM + JSCD Systems for BER=0.05</td>
<td>24</td>
</tr>
<tr>
<td>3.6</td>
<td>3-Bit DPCM + JSCD Systems for BER=0.05</td>
<td>25</td>
</tr>
<tr>
<td>3.7</td>
<td>Images used for Training Sequence</td>
<td>26</td>
</tr>
<tr>
<td>3.8</td>
<td>Training Sequence Sensitivity for 2-Bit DPCM + JSCD System</td>
<td>27</td>
</tr>
<tr>
<td>3.9</td>
<td>Training Sequence Sensitivity for 3-Bit DPCM + JSCD System</td>
<td>27</td>
</tr>
<tr>
<td>3.10</td>
<td>Performance of Iterative JSCD for a 3-Bit JSCD + JSCD System</td>
<td>30</td>
</tr>
<tr>
<td>3.11</td>
<td>Iterative JSCD for 3-Bit DPCM + JSCD System at BER=0.04</td>
<td>31</td>
</tr>
<tr>
<td>3.12</td>
<td>SNR Curves for Mapping Schemes for 2-Bit DPCM + JSCD Systems</td>
<td>34</td>
</tr>
<tr>
<td>3.13</td>
<td>SNR Curves for Mapping Schemes for 3-Bit DPCM + JSCD Systems</td>
<td>34</td>
</tr>
<tr>
<td>3.14</td>
<td>Mapping Schemes for 2-Bit DPCM + JSCD System</td>
<td>35</td>
</tr>
<tr>
<td>3.15</td>
<td>Mapping Schemes for 3-Bit DPCM + JSCD System</td>
<td>36</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.16</td>
<td>Second-order modeling of Encoded Image</td>
<td>38</td>
</tr>
<tr>
<td>3.17</td>
<td>Second-Order JSCD Performance for 2-Bit DPCM System</td>
<td>39</td>
</tr>
<tr>
<td>3.18</td>
<td>Second-Order JSCD Performance for 3-Bit DPCM System</td>
<td>39</td>
</tr>
<tr>
<td>3.19</td>
<td>Subjective Comparison of First-order and Second-order JSCD Systems</td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>DPCM/JSCD Post Processing System</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Sample DPCM Transmission Error Patterns</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Sliding Window for Lippman's Streak Detection</td>
<td>46</td>
</tr>
<tr>
<td>4.4</td>
<td>Lippman Streak Detection for Probability of Error=0.001</td>
<td>48</td>
</tr>
<tr>
<td>4.5</td>
<td>Flow Chart for Modified Lippman Algorithm</td>
<td>50</td>
</tr>
<tr>
<td>4.6</td>
<td>BER of 2-Bit JSCD + Post Processed Images</td>
<td>52</td>
</tr>
<tr>
<td>4.7</td>
<td>BER of 3-Bit JSCD + Post Processed Images</td>
<td>52</td>
</tr>
<tr>
<td>4.8</td>
<td>Symbol Correction Schemes for 2-Bit DPCM + JSCD Systems</td>
<td>57</td>
</tr>
<tr>
<td>4.9</td>
<td>Subjective Comparisons for Symbol Choice Techniques of a 2-Bit DPCM + JSCD System</td>
<td>58</td>
</tr>
<tr>
<td>4.10</td>
<td>Symbol Correction Schemes for 3-Bit DPCM + JSCD Systems</td>
<td>59</td>
</tr>
<tr>
<td>4.11</td>
<td>Subjective Comparisons for Symbol Choice Techniques of a 3-Bit DPCM + JSCD System</td>
<td>60</td>
</tr>
<tr>
<td>4.12</td>
<td>MSE Error Correction Performance at Low BER for a 3 Bit System</td>
<td>61</td>
</tr>
<tr>
<td>4.13</td>
<td>Streak Detected Images for BER=0.006 for 3 Bit DPCM System</td>
<td>63</td>
</tr>
</tbody>
</table>
Acknowledgment

I would like to express my sincerest gratitude to Dr. Samir Kallel for his invaluable advice, guidance and encouragement during the course of the research.

I would also like to thank Dr. Robert Link for his careful proofreading and for his helpful insights.

Finally, I am grateful for the constant love and support of my family and friends.

This work was funded by NSERC, BC ASI, and Mobile Data Solutions Inc, of Richmond, BC.
Chapter 1 Introduction

The area of wireless communications has exploded at an astonishing rate, and today wireless networks offer more services than conventional voice telephony. Utility companies now can provide reliable voice and data services via networks such as cellular digital packet data (CDPD) networks. However, the cost of image transmission over digital wireless networks is still quite high due to the large amount of data contained in digital images. Reducing the size of the data file representing an image is known as image compression. A compression technique known as differential pulse code modulation (DPCM) is one means of source coding images. It is a successful technique that has found its way into standards such as JPEG and MPEG [1]. Unfortunately, DPCM systems are very sensitive to noise, and most communication channels can never be considered to be error free. Errors in DPCM encoded data propagate when decoding and cause streaks in the images which are visually very disturbing. Therefore channel errors must be mitigated in DPCM pictures.

Several techniques have been developed to increase the picture quality of corrupted DPCM images. Conventional error correction techniques, such as BCH codes, have been studied [2],[3]. However, forward error correction codes (FEC) add as much as 25 per cent of redundancy to give acceptable performance even over a BSC of channel error rate less than 0.01.

One alternate technique developed by Ngan and Steele [4] uses PCM updates to locate transmission errors. PCM updates allow the decoder to calculate the difference of the PCM update and the DPCM output. If this difference exceeds a certain threshold then the pixels are replaced by neighbouring pixels which are assumed to be error free. This method also requires side information and increases the bit rate by a significant amount.
Another thresholding technique, developed by Lippmann [5], uses the identification of error patterns. Once an erroneous pixel is located, its effect is mitigated by replacing the erroneous pixel by the corresponding pixel in the previous line. This method depends on the fact that adjacent scan lines have highly correlated corresponding pixels.

Other techniques utilize the source statistics of the image to help correct transmission errors. When the compression and error correction of data are treated together, the approach is generally known as joint source/channel coding (JSCD). Sayood and Borkenhagen [6] have developed a method which corrects errors without the aid of an explicit channel coder. It is based on a Viterbi like decoding procedure where the code word combinations which are more likely than others are favoured by a relative amount which depends on the amount of channel noise. Source statistics are obtained by modeling the image as a first order Markov model.

Emani and Miller [7] have also developed techniques which are similar to that of [6]. They introduce symbol-by-symbol JSCD decoders for first order and second order Markov models. They show that significant improvement can be gained by the use of second order models over first order. However, we have found that although the method of [6] can be modified to use a second order model, and that this leads to better results by objective measure, by subjective measure this generalization of the method fails. Furthermore, for the post processing techniques that we developed, the first order Markov model system, besides being simpler, is actually superior to the second order model.

1.1 Motivations and Objectives

The goal is to design the best possible DPCM image system where robustness to noise is the primary design criterion. Several techniques have been introduced which combat the effects of
transmission errors, and this thesis will introduce a design which is a hybrid of some of these techniques. Having reproduced the results obtained in [6], we found that the JSCD decoded images at high bit error rates can be further improved. The first order JSCD systems leave many errors which go uncorrected. These uncorrected errors still cause enough of a disturbance to the eye in the way of streaks across the image. Hence, we propose a cascaded system which uses a JSCD decoder directly followed by a post-processor. The post-processor has two objectives: The first is to locate the DPCM data samples which have been received erroneously, by examining the decoded image for streaks. The second objective is to choose the best possible candidate to replace the erroneous sample. The goal, once again, is to achieve the cleanest image possible, not only by objective measure but more importantly by subjective measure as well.

1.2 Thesis Outline

In Chapter 2, background knowledge of the DPCM system is introduced. Key components such as the predictor coefficient and quantization are explained and their effects on the overall system are introduced. The well known classical design procedure for a DPCM encoder needs to be modified when the encoded data is to be transmitted over a noisy channel.

In Chapter 3, the design of JSCD systems are introduced. This chapter consists primarily of a review of known results; however, we do investigate several generalizations or modifications of the known system, and choose the one most appropriate for our overall system. The use of first and second-order Markov models are explored. Furthermore, code word mapping and its effects on JSCD systems are investigated.

Chapter 4 introduces post processing techniques which attempt to correct errors which were not corrected by the JSCD methods of Chapter 3. We introduce a post-processing algorithm
which uses techniques built upon those found in [5] to locate errors, and uses hereby developed
techniques to correct these errors. A final chapter provides a summarizing discussion and conclud-
sions.
Chapter 2  DPCM Image Compression

Digital images contain much redundancy from sample to sample which is more prevalent in lower detailed images than in higher detailed ones. This redundancy allows samples to be predicted from previous samples received and hence compression can be achieved by the removal of such repetition. Most raw grayscale digital images are represented by samples of 8 bits per pixel covering a range of levels from 0 to 255. Figure 2.1 shows that the standard monochrome Lenna image contains levels with a considerably broad range. However, if the difference of each pixel to pixel is taken, a large percentage will fall within a smaller range which can then be encoded with fewer bits. Figure 2.1 also shows the histogram of the pixel-to-pixel differences of the Lenna image. Encoding the difference of sample values rather than the actual samples has led to a method of image compression known as Differential Pulse Code Modulation (DPCM), which is a component of the JPEG standard.

Figure 2.1 Histogram of Lenna image and Difference of Pixel-to-Pixels of Lenna
2.1 DPCM System

DPCM is a form of lossy compression, or in other words there is some loss of information and the data cannot be reconstructed exactly. Both lossiness and compression stems from quantization which is a process by which a set of source output values is represented by a much smaller set of code words. Figure 2.2 is a block diagram depicting the first order differential encoding system.

Consider a sequence of pixel values \( \{x_n\} \) then the difference sequence \( \{d_n\} \) is obtained by taking the difference \( \{x_n - x_{n-1}\} \). The difference sequence is then quantized to obtain the sequence \( \{\hat{d}_n\} = \{Q[d_n]\} = \{d_n + q_n\} \) where \( q_n \) is the quantization error. At the receiver, the reconstructed sequence \( \{\hat{x}_n\} \) is obtained by adding the quantized difference to the previous reconstructed value. So

\[
\hat{x} = \hat{x}_{n-1} + \hat{d}_n.
\]
For better DPCM performance, the difference value $d_n$ should be as small as possible which would give less quantization noise. This can be achieved if the sample $x_n$ is predicted by some function of the past values of the reconstructed sequence. Figure 2.3 shows the delay block being replaced by a predictor block. It is obvious that the two major components of DPCM are the predictor and the quantizer. Therefore the following sections will look at how each effect the performance of DPCM.

![Diagram](image.png)

Figure 2.3 Differential Pulse Code Modulation System with Predictor

### 2.2 Prediction Coefficient

The performance gain of DPCM systems are attributed to the reduction of the variance and dynamic range of the difference sequence. This difference can be minimized if predictive techniques are used to remove mutual redundancy between neighbouring pixels which is the effect of minimizing the variance of the difference sequence defined as

$$
\sigma_d^2 = E[(x_n - p_n)^2]
$$

(2.2)
where $E[\cdot]$ is the expectation operator. The predictor outputs $p_n$ are formulated from a function $f(\cdot)$ which gives

$$p_n = f(\hat{x}_{n-1}, \hat{x}_{n-2}, \ldots, \hat{x}_0).$$

The best selection of $f(\cdot)$ which minimizes $\sigma_d^2$ is desired, but an explicit solution is difficult to achieve due to a coupling effect [8]. The choice of $f(\cdot)$ will effect $\sigma_d^2$ from (2.2) and (2.3) which will then have an effect on the reconstruction values $\hat{x}_n$ which in turn will have an effect on the selection of $f(\cdot)$. However, in most cases an assumption is used known as the fine quantization assumption. That is the quantizer step sizes are assumed to be so small that $x_n$ and $\hat{x}_n$ are indistinguishable. Therefore (2.3) becomes

$$p_n = f(x_{n-1}, x_{n-2}, \ldots, x_{n-N}).$$

We now make another assumption that the source is a stationary process and the function which will minimize $\sigma_d^2$ is the expectation $E[x_n|x_{n-1}, x_{n-2}, \ldots, x_{n-N}]$. The problem is further simplified by requiring the prediction function to be linear which gives

$$p_n = \sum_{i=1}^{N} a_i x_{n-1};$$

where $N$ is the order of the predictor. With the help of the fine quantization assumption, the predictor design problem is to find $\{a_i\}$ so as to minimize $\sigma_d^2$ which can be denoted as follows
$$\sigma_d^2 = E \left[ \left( x_n - \sum_{i=1}^{N} a_i x_{n-i} \right)^2 \right]. \quad (2.6)$$

To solve for minimization we take the derivative of $\sigma_d^2$ with respect to each $a_i$ and equate it to zero which gives us the following $N$ equations

$$\sum_{i=1}^{N} a_i R_{xx}(i-j) = R_x(x(j)) \quad j = 1, \ldots, N \quad (2.7)$$

where $R_{xx}(x)$ is the autocorrelation function of $x_n$:

$$R_{xx}(k) = E[x_n x_{n+k}]. \quad (2.8)$$

If there are $M$ points of data, the autocorrelation can be estimated by the following average

$$R_{xx}(k) = \frac{1}{M-k} \sum_{i=1}^{M-k} x_i x_{i+k} \quad (2.9)$$

In our simulations we used a single tap predictor therefore only one prediction coefficient needs to be solved for and it simplifies to

$$a_1 = \frac{R_{xx}(1)}{R_{xx}(0)}. \quad (2.10)$$

For the standard 512 x 512 monochrome Lenna image, a prediction coefficient $a_1=0.99573$ was obtained. Of course these prediction coefficients are image dependent and would have to be recalculated for each different image, which in the first order case is not too computationally intensive. The choice of the prediction coefficient has some interesting effects. A very
good predictor will minimize the variance of the difference sequence which will allow for the
greatest compression gain and reconstruction quality. However, under noisy conditions where
some symbols may be received in error, the same predictor may not be ideal. Due to the error
propagating nature of DPCM images, the "classical" prediction coefficient is not best suited for
transmission of DPCM images under noisy channels.

2.2.1 Chang Donaldson Prediction Coefficient

The output signal distortion is found to be caused by the sum of three factors. These are:
quantization errors, channel transmission errors, and mutual errors arising from the coupling of
quantization and channel errors. Thus there was a need to study the effects of the quantizer and
predictor under noisy digital channels. Chang and Donaldson have studied the effects of channel
transmission errors and have shown that the signal-to-noise ratio (SNR) of a well-designed
DPCM system outperforms a PCM system operating on the same noisy digital channel [9]. Their
work led to the understanding of the SNR dependence on predictor coefficient values, number of
quantization levels, and channel error probability. When the channel is very noisy, the optimal
prediction coefficient becomes

\[ a_{opt} = \frac{1 - (1 - a_1^2)^{\frac{1}{2}}}{a_1} \]  \hspace{1cm} (2.11)

where \( a_1 \) is the normalized first autocorrelation coefficient (2.10). For the Lenna image we get a
reoptimized predictor coefficient of 0.91158 which is less than \( a_1 \). Due to the use of a smaller
predictor coefficient, the transmission errors will not propagate as far as they would in the case of
the classical prediction case, because the error continuously undergoes attenuation by the predic-
tion coefficient. The smaller the prediction coefficient the greater the attenuation.

2.3 Quantization

It was mentioned that the quantizer is a key function of the DPCM system, and rightfully so because quantization is associated with lossy compression. Quantization is the means of representing a large or possibly an infinite amount of data by a much smaller set. The process is simple. At the encoder the range of the source values is divided into a given number of intervals, and each interval is associated with one code word. Therefore, the decoder upon receiving the code word, will map it to a reconstruction value which is normally a midpoint of the interval used at the encoder. This leads to the premier design problem of quantizers, the choice of the intervals and/or reconstruction levels. Choosing the correct reconstruction levels is important because with every quantization there is an associated quantization error which is the difference between the quantizer input and output. This is also referred to as quantizer distortion or quantization noise. Therefore, a poorly designed quantizer will result in a poorly reconstructed image. There are two main types of scalar quantizers: uniform quantizers and non-uniform quantizers.

2.3.1 Uniform Quantizer

The uniform quantizer is the simplest and most commonly used quantizer. As the name suggests, the reconstruction levels are uniformly distributed across a range, and the intervals are all the same size, except possibly for the two outer intervals. For an \( n \) bit quantizer there would be \( 2^n \) intervals and reconstruction levels. For a source which produces a uniform distribution between the interval \([-X, X]\) the step size of each level would be
\[ \Delta = \frac{2X}{2^n}. \]  

(2.12)

For sources which do not produce uniform distributions, a uniform quantizer would not be a good choice. From the histogram of Figure 2.1, it is obvious that to quantize the image efficiently, the quantizer of choice would be a non-uniform one.

### 2.3.2 Nonuniform Pdf Optimized Quantizer

For sources which have a nonuniform distribution, but a high density within a narrow range, small step sizes are needed to quantize the values to reduce quantizer distortion. Smaller intervals in those regions cause larger intervals in other regions.

For an \( M \) level quantizer let the set of decision boundaries be \( \{t_i\}_{i=0}^{M} \) and the reconstruction levels be \( \{r_i\}_{i=1}^{M} \). Then the quantizer, \( Q \), maps data values to quantizer levels by the rule

\[ Q(x) = r_i \text{ iff } t_{i-1} < x \leq t_i. \]  

(2.13)

The mean squared quantization error is denoted as

\[ \sigma_q^2 = \sum_{i=1}^{M} \int_{t_{i-1}}^{t_i} (x-r_i)^2 f_X(x)dx; \]  

(2.14)

where \( f_X(x) \) is the pdf of random variable \( X \). Equation (2.14) is minimized by taking the derivative with respect to \( r_j \), equating to zero, and solving for \( r_j \) to give
\[ r_j = \frac{\int_{t_{j-1}}^{t_j} x f_x(x)dx}{\int_{t_{j-1}}^{t_j} f_x(s)dx} \] (2.15)

Now taking the derivative of Equation (2.15) with respect to \( t_j \) gives

\[ t_j = \frac{r_{j+1} + r_j}{2} \] (2.16)

which turns out to be the midpoint of the two adjacent reconstruction levels. Solving for equations (2.15) and (2.16) will give the reconstruction levels and decision boundaries which will minimize the mean squared quantization error. However, these are nonlinear equations which must be solved simultaneously. The Lloyd-Max algorithm is a method which will solve the two equations iteratively [10]. The resulting quantizer has some interesting properties.

- The mean values of the input and output are equal.
- The variance of the Lloyd-Max quantizer output is always less than or equal to the variance of the input.

### 2.4 Simulation and Numerical Results

The quality of reconstructed digital images can be measured by both subjective and objective means, and the most commonly used objective performance measure is the signal-to-noise ratio (SNR) which is defined as follows

\[ \text{SNR} = 10 \log_{10} \left( \frac{\text{Power of the signal}}{\text{Power of the noise}} \right) \]
\[ SNR = 10\log_{10} \frac{\sum x_n^2}{\sum (x_n - \hat{x}_n)^2} \] (2.17)

where \( x_n \) is the original source pixel value and \( \hat{x}_n \) is the reconstructed pixel value.

The following results are obtained for the Lenna image using a row by row scan. We reinitialized the DPCM system after each row because it helps to combat the effects of propagating errors and achieves a higher quality reconstructed picture. The source of the channel errors was simulated by the use of a binary symmetric channel with probability of error rate \( p_e \). Two systems are investigated: the two bit system and the three bit system.

![Figure 2.4 2-Bit DPCM System Using Lloyd Max Quantization](image-url)

Figure 2.4 2-Bit DPCM System Using Lloyd Max Quantization
The graphs of Figures 2.4 and 2.5 will provide a basis upon which the performance of the JSCD and post-processing system will be evaluated. The top curves are the systems utilizing Chang and Donaldson prediction coefficients or renamed as Reoptimized DPCM (RDPCM).
c) 2-Bit RDPCM
Prob. of Error=0.05, SNR=15.99 dB

d) 3-Bit RDPCM
Prob. of Error=0.05, SNR=14.00 dB

e) 2-Bit DPCM
Prob. of Error=0.05, SNR=7.68 dB

f) 3-Bit DPCM
Prob. of Error=0.05, SNR=6.16 dB

Figure 2.6 Comparison of 2-Bit and 3-Bit DPCM Systems
The images corresponding to the graphs are shown in Figures 2.6. It is evident that the reoptimized DPCM system using the Chang-Donaldson prediction coefficient in noisy conditions is beneficial. The streaks are much shorter than the classical case and the images are far more recognizable. At a binary symmetric channel bit error rate of 0.05, there is a SNR improvement of as much as 8 dB for the two bit case and approximately a 7 dB gain for the three bit case.
Chapter 3  Joint Source Channel Decoding

DPCM systems are known to be very sensitive to noise which is almost unavoidable in most communication channels. The previous chapter showed the effects of an error which results in a streak possibly propagating all the way to the end of a scan line. Even at very low channel bit error rates, the affects are unpleasing to the eye. We will be considering the use of joint source channel coding techniques to improve the error performance.

A simple digital communication system consists of nine basic elements (see Figure 3.1) [11]. In order they are: an information source, source encoder, channel encoder, digital modulator, channel, digital demodulator, channel decoder, source decoder and finally the information sink. The source encoder has the responsibility of removing redundant information and hence compressing the data. The channel encoder's duty is to protect the data, usually by introducing some redundancy, from the effects of noise encountered during the transmission through a channel. Though conventional source encoder and channel encoders are separate entities, similar performance and reduced complexity can be achieved by use of joint source channel (de)coders (JSCD)[12].

Sayood and Borkenhagen [13] have categorized JSCD systems into three classes: joint source channel coders, a true integration of both the source and channel coders; concatenated source channel coders, where a fixed bit rate is allocated between a source coder followed by a channel coder; and constrained joint source channel coders, which involves a modification of a given source coder and/or decoder according to noise conditions. This chapter will primarily focus on the latter set of JSCD systems.
3.1 Viterbi Decoding Using Residual Redundancy

A source coder ideally removes all redundancy from the source information and outputs an uncorrelated sequence. However, in reality, a source coder can not remove all redundant information due to the source encoder's inability to precisely model a source. What remains is known as residual redundancy. In our case, the DPCM encoder assumes that the image is an autoregressive (AR) process which may be near to but not a true perfect match. This leads to a source encoder output sequence containing some structure which can be utilized to combat channel errors. The DPCM encoder outputs a given set of code words, and if there is any redundancy/structure to the source output at all, some code word combinations may occur more frequently than others. The decoder can then use knowledge of code word-to-code word transition probabilities to correct some errors, inhibiting very highly improbable transitions and encouraging more probable code word combinations.

For the source output, or channel input sequence, let the symbol alphabet be denoted by $A = \{a_0, a_1, \ldots, a_{Q-1}\}$. Let the channel input and output sequences be $s = \{s_0, s_1, \ldots, s_{K-1}\}$.
and \( \hat{s} = \{ \hat{s}_0, \hat{s}_1, \ldots, \hat{s}_{K-1} \} \) respectively. The optimum receiver must minimize the probability of making an error, which is

\[
P(E) = \sum_{\hat{s}} P(E|\hat{s})P(\hat{s}),
\]

where \( P(\hat{s}) \) is independent of the receiver’s operation. Therefore to minimize (3.1) one must minimize

\[
P(E|\hat{s}) = P(\hat{s} \neq s|\hat{s})
\]

which is the same as maximizing \( P(\hat{s} = s|\hat{s}) \), where \( \hat{s} \) is the decoder’s choice of output sequence. From Baye’s rule, maximizing the previous expression is the equivalent to maximizing

\[
\frac{P(\hat{s}|s)P(s)}{P(\hat{s})}.
\]

Furthermore, if the channel is assumed to be memoryless, then

\[
P(\hat{s}|s) = \prod_i P(\hat{s}_i|s_i).
\]

and if the source is modeled as an \( M \)th-order Markov sequence

\[
P(s) = \prod_i P(s_i|s_{i-1}, s_{i-2}, \ldots, s_{i-M}).
\]

If (3.4) and (3.5) are substituted into (3.3), the expression to be maximized becomes
\[ P(s|\bar{s}) = \frac{\prod_i P(\hat{s}_i|s_i)P(s_i|s_{i-1}, s_{i-2}, \ldots, s_{i-M})}{P(\bar{s})}, \quad (3.6) \]

and by taking the logarithm of both sides, the expression becomes

\[ \log P(s|\bar{s}) = \sum_i \log(P(\hat{s}_i|s_i)P(s_i|s_{i-1}, s_{i-2}, \ldots, s_{i-M})) - \log P(\bar{s}). \quad (3.7) \]

Maximizing the summation term in (3.7) will give a minimum error rate decoder. This is similar to the cumulative metric used in Viterbi decoders for convolutional codes [14]. Viterbi decoding requires the use of a trellis at the decoder which has \( Q \) states. For our case, the number of states depend on the number of DPCM code words used. For each state, there are \( Q \) branches leaving and entering with an associated branch metric

\[ \log P(\hat{s}_i|s_i) + \log P(s_i|s_{i-1}, s_{i-2}, \ldots, s_{i-M}). \quad (3.8) \]

At a given state in the trellis, only the path with the largest cumulative metric is kept, and is known as the survivor. Thus at any given time, there will be \( Q \) surviving paths and the one with the largest cumulative metric corresponds to the most likely transmitted sequence. Figure 3.2 depicts the trellis diagram for a two bit DPCM + JSCD system. The branch metric consists of two terms. The first term \( \log P(\hat{s}_i|s_i) \) is the term which depends only on the channel characteristics.

For our case, we assume the channel to be a binary symmetric channel and that the cross-over probability of error of the channel is known. The second term \( \log P(s_i|s_{i-1}, s_{i-2}, \ldots, s_{i-M}) \) is based entirely on the statistics of the information and can be acquired from a training sequence, or measured at the source and transmitted as FEC (Forward Error Correction) protected side information to the receiver. For our case, we consider the first order Markov model so (3.8)
becomes \( \log P(s_i | s_t) + \log P(s_t | s_{t-1}) \).

Figure 3.2 Trellis Diagram for 2-Bit DPCM + JSCD System

Using standard Lenna, modeled as first order Markov we obtained the SNR curves as shown in Figure 3.3 and 3.4. For the two bit system, JSCD gives a small improvement for both the classical and reoptimized cases. At high bit error rates of around 0.1 the JSCD gain is less than 1 dB. This gain is not large enough to make any significant impact on the image and hence subjectively the corrupted and JSCD decoded images are not that different from each other. However, for the three bit DPCM system there is a larger gain of nearly 3 dB at high error rates and Figure 3.6 show that JSCD does improve the picture quality. The three bit system obviously performs better than the two bit system in terms of JSCD gains. For the three bit system there are double the amount of symbols than the two bit system and thus it is more probable for an error to create an unlikely transition in the three bit system than in the two bit system. Thus JSCD is able to correct many more "large step" errors which degrade the image more so than other errors. Consequently different code word mappings will give different performances which will be explained further in section 3.1.3. In the above, Gray code was used to map quantizer levels to code words.
Chapter 3 Joint Source Channel Decoding

Figure 3.3 2-Bit DPCM + JSCD System for Classical and Reoptimized Prediction Coefficients

Figure 3.4 3-Bit DPCM + JSCD System for Classical and Reoptimized Prediction Coefficients
a) 2 Bit DPCM  
SNR = 7.68 dB

b) 2 Bit DPCM + JSCD  
SNR = 8.56 dB

c) 2 Bit RDPCM  
SNR = 15.99 dB

d) 2 Bit RDPCM + JSCD  
SNR = 16.56 dB

Figure 3.5 2-Bit DPCM + JSCD Systems for BER=0.05
Chapter 3 Joint Source Channel Decoding

Figure 3.6 3-Bit DPCM + JSCD Systems for BER=0.05

a) 3 Bit DPCM
SNR = 6.16 dB

b) 3 Bit DPCM + JSCD
SNR = 7.65 dB

c) 3 Bit RDPCM
SNR = 14.00 dB

d) 3 Bit RDPCM + JSCD
SNR = 16.72 dB
3.1.1 Training Sequence Sensitivity

In our work the source statistics are assumed to be acquired at the source and transmitted as side information to the receiver. However, another possibility is to train the receiver on an image whose source statistics will then be used to decode all other images. In [6] it is reported that the JSCD system is quite robust with respect to the training sequence. Even if the training image and the transmitted image are quite different, the similar statistics allow for some correction of erroneous symbols. We investigated the training sequence sensitivity with different standard images and find contrasting results. We considered training the receiver on the Mandrill image (Figure 3.7) and the transmitting and decoding the Lenna image. We also tried training the receiver on the Peppers image (Figure 3.7) and decoding the Lenna image.

![Mandrill](image1.png) ![Peppers](image2.png)

Figure 3.7 Images used for Training Sequence
Figure 3.8 Training Sequence Sensitivity for 2-Bit DPCM+JSCD System

Figure 3.9 Training Sequence Sensitivity for 3-Bit DPCM+JSCD System
A third possibility we considered was to receive Lenna over the noisy channel without JSCD error correction. The statistics were measure on the noise corrupted encoded image, and the image was redecoded using JSCD with the noise corrupted statistics.

From Figure 3.8 for two bit DPCM systems, it is evident that obtaining source statistics from other images give little or no improvement at all. JSCD with statistics of the uncorrupted Lenna image give very little improvement, so it is not surprising that training on other images give virtually no improvement. However, for the case of the three bit system we get quite different results. One can see that the JSCD system trained by the Peppers image give very close results to that of the system which receives Lenna statistics uncorrupted. This is quite surprising, because the Peppers image and the Lenna image are quite different in appearance. Even training on the Mandrill image, gives a slight improvement. The Peppers image gives better performance than the Mandrill image, because the source statistics of the DPCM encoded image are more similar to the Lenna statistics. It is interesting to note that if the JSCD receiver is trained on the corrupted Lenna image itself, there is also some improvement in SNR shown in Figure 3.9. However, the JSCD gain appears to breakdown as the probability of error increases.

We find that the three bit system is more robust than the two bit system in respect to JSCD training sequences. However, we recommend that the source statistics be transmitted as side information with FEC, for the amount is insignificant, only amounting to a 0.5% increase in bit rate [7].
3.1.2 Iterative JSCD

Training the receiver with the corrupted statistics taken from the received Lenna image still gives an SNR improvement (Figure 3.9). The corrupted image retained enough correct information that redundancy was still inherent. Hence, the first-order Markov model taken from the corrupted image still allowed for the correction of some highly improbable state transitions. These results led us to attempt an iterative JSCD system where there is no need to transmit source statistics as FEC side information.

The JSCD decoder receives the corrupt encoded image with knowledge of channel characteristics. It then models the image as a first-order Markov model and performs JSCD as described previously in section 3.1. The receiver once again will take first-order Markov statistics of the JSCD output and perform JSCD again using the new model. The procedure is continued until the output no longer changes or until a reasonable result is obtained.

This iterative approach was applied to the Lenna image for the three bit system and from Figure 3.10 one can see that each iteration gives an improvement in SNR. Within three iterations, the result is nearly the same as the performance of JSCD with reception of error free source statistics. The curves do converge, but we find that more iterations do not necessarily give better results. For the Lenna image, a total of three iterations give the highest SNR curve and any iterations beyond that tend to drop the SNR curve slightly.
Figure 3.10 Performance of Iterative JSCD for a 3-Bit DPCM + JSCD System

The subjective quality of the iterative JSCD images shown in Figure 3.11 agrees to the curves of Figure 3.10. It is found that the third JSCD iteration of the Lenna image is quite similar to the quality of the JSCD image trained on error free statistics.
Chapter 3 Joint Source Channel Decoding

Figure 3.11 Iterative JSCD for 3-Bit DPCM + JSCD at BER 0.04

a) DPCM
   SNR = 14.91 dB

a) 1st Iteration
   SNR = 16.12 dB

c) 3rd Iteration
   SNR = 17.48 dB

d) JSCD (error free stats)
   SNR = 17.68 dB
3.1.3 Code Word Mapping

Code word mapping is the means by which different quantizer levels are assigned different binary code words. We consider fixed length codes where quantization levels are assigned binary code words of a specific number of bits. There are several different code word mapping schemes, and each perform differently when channel errors are present. The following is a list of some mapping schemes:

(i) **Natural Binary** - The assignment follows a natural counting pattern. One signal is assigned to a code word and the next adjacent signal is assigned a code word incremented by one and so on.

(ii) **Gray Code** - Adjacent signals are assigned a code word with a maximum Hamming difference of one. The magnitude of one bit error patterns are thus minimized.

(iii) **MAP Rule Code** [7] - Mapping is assigned according to symbol occurrence probabilities. The most probable symbol is assigned to the code word 000. The least probable symbols are then assigned to a code word of one Hamming distance away. Correction of single error patterns are then possible by using a priori data. The second most probable symbol is assigned a code word which is a maximum Hamming distance from the most probable symbol code word. The same rule is then applied to the remaining code words.

<table>
<thead>
<tr>
<th>Quantizer Level</th>
<th>Probability</th>
<th>Natural Binary</th>
<th>Gray</th>
<th>MAP Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.043</td>
<td>00</td>
<td>00</td>
<td>01</td>
</tr>
<tr>
<td>1</td>
<td>0.365</td>
<td>01</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>0.527</td>
<td>10</td>
<td>11</td>
<td>00</td>
</tr>
<tr>
<td>3</td>
<td>0.065</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>
Table 3.2 Code Word Mapping for 3 Bit DPCM System

<table>
<thead>
<tr>
<th>Quantizer Level</th>
<th>Probability</th>
<th>Natural Binary</th>
<th>Gray</th>
<th>MAP Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.005</td>
<td>000</td>
<td>000</td>
<td>001</td>
</tr>
<tr>
<td>1</td>
<td>0.009</td>
<td>001</td>
<td>010</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>0.025</td>
<td>010</td>
<td>110</td>
<td>010</td>
</tr>
<tr>
<td>3</td>
<td>0.046</td>
<td>011</td>
<td>100</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>0.226</td>
<td>100</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>5</td>
<td>0.337</td>
<td>101</td>
<td>111</td>
<td>000</td>
</tr>
<tr>
<td>6</td>
<td>0.093</td>
<td>110</td>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>7</td>
<td>0.255</td>
<td>111</td>
<td>001</td>
<td>111</td>
</tr>
</tbody>
</table>

Tables 3.1 and 3.2 show the code word mappings used for the Lenna image in the simulations. For the two bit system the natural binary and MAP rule are equivalent by the fact that the two most probable symbols are given code words which are a maximum Hamming distance apart. Figure 3.12 and 3.13 shows the JSCD performance for the different mapping schemes where only the reoptimized Chang - Donaldson prediction coefficients were used.

For the two and three bit systems the Gray coding consistently outperforms all other mapping schemes. Again, this occurs because the effects of one bit error patterns are minimized by adjacent symbols having code words which are one Hamming distance apart. From the simulations it is evident that the MAP rule coding consistently has the largest JSCD gain. For the three bit system, at a BER of 0.05 a gain of 4.01 dB is observed using the MAP rule. At a BER 0.1 the gain is as much as 4.25 dB. The MAP rule gives the largest gain because channel errors cause more unlikely state transitions to occur. Then the JSCD using the source statistics is able to correct many of these unlikely transitions.
Figure 3.12 SNR Curves for Mapping Schemes for 2-Bit DPCM + JSCD Systems

Figure 3.13 SNR Curves for Mapping Schemes for 3-Bit DPCM + JSCD Systems
Chapter 3  Joint Source Channel Decoding

Figure 3.14  Mapping Schemes for 2-Bit DPCM + JSCD System at Probability of Error = 0.05

a) Natural Binary DPCM + JSCD
SNR = 15.03 dB

b) MAP Rule DPCM + JSCD
SNR = 14.97 dB

c) Gray Code DPCM + JSCD
SNR = 16.56 dB
Chapter 3 Joint Source Channel Decoding

Figure 3.15 Mapping Schemes for 3-Bit DPCM + JSCD System at Probability of Error = 0.05

a) Natural Binary DPCM + JSCD
SNR = 15.97 dB

b) MAP Rule DPCM + JSCD
SNR = 14.62 dB

c) Gray Code DPCM + JSCD
SNR = 16.72 dB
From Figures 3.14 and 3.15, one can see that the Gray code is far superior to any of the other mapping schemes both objectively and subjectively. There are less prominent streaks due to errors associated with widely separated quantizer levels in the Gray coded images. Hence, for our system we choose to use the Gray coded JSCD, giving us the highest reconstructed signal to noise ratios.

3.2 JSCD Using A Second-Order Markov Model

For the previous sections we have only considered first-order Markov models. These models are based on statistics of the transitions of one symbol to another symbol along the horizontal direction of a scan line. However, we know that image redundancy is not directional. Generally, the surrounding pixels at any given point are alike and hence we can model the image as a second-order or even third-order Markov model. Previously, we introduced the branch metric as the following $\log P(\hat{s}_i|s_i) + \log P(s_i|s_{i-1}, s_{i-2}, \ldots, s_{i-M})$ for an $M$th order Markov model. However, instead of looking two symbols behind the current symbol we choose a symbol which is closer in distance. We choose the symbol directly before the current symbol and the symbol directly above on the previous scan line. Then the branch metric actually becomes

$$\log P(\hat{s}_{r,c}|s_{r,c}) + \log P(s_{r,c}|s_{r,c-1}, s_{r-1,c})$$  \hspace{1cm} (3.9)

where $r$ is the row number, $c$ is the column number, and $\hat{s}$ is a previously decoded symbol. Figure 3.16 depicts the second-order modeling arrangement used. Unfortunately, there is a cost associated with increasing the order. As the order increases, the amount of possible state transitions increase exponentially, thus the computations and complexity also increase.
Figure 3.16 Second-order modeling of Encoded Image

For the two bit case the gain of the second-order model is minimal over the first-order case. As seen in Figure 3.17 there is about a 0.5 dB increase over the JSCD O(1) case at high BERs and virtually no improvement at other BERs. In the three bit case (Figure 3.18), however, there is a continuous improvement for the entire range. The second-order JSCD system achieves a gain of 2.95 dB at a BER of 0.02 and as much as 3.61 dB at 0.10 over the DPCM corrupted image. However, this is not a drastic improvement over the first-order model. It only amounts to less than a 1 dB improvement throughout which hardly makes a subjective impact to the image at all.

From a quick glance of Figure 3.19 one can see that there is barely a difference between the first-order and second-order systems. However, if the images are compared carefully, there seems to be a slight difference. It appears that the first-order images leave a sharper image. There is more clarity around highly detailed areas such as around the eyes. On the other hand, the second-order system does seem to have fewer streaks which would account for the slightly higher SNR. In the area of Lenna’s hat, it is evident that the second-order model creates a blur of streaks, or multiple width streaks. This occurs, because unlikely transitions in the vertical direction are also corrected by JSCD O(2). If a streak causes this abrupt transition, then the second-order system might possibly introduce another streak which has happened in the cases shown.
Chapter 3 Joint Source Channel Decoding

Figure 3.17 Second-Order JSCD Performance for 2-Bit DPCM System

Figure 3.18 Second-Order JSCD Performance for 3-Bit DPCM System
c) JSCD O(1), BER = 0.06
SNR = 16.05 dB

b) JSCD O(2), BER = 0.06
SNR = 16.94 dB

Figure 3.19 Subjective Comparison of First-order and Second-order JSCD Systems
Although the second-order Markov model approach does not give a great improvement, there are other approaches which outperform the first-order system. The Modified MAP (MMAP) and Maximal SNR (MSNR) in [7] showed significant gains. The MSNR is a minimum cost decoder where the cost is simply defined as the squared difference of the quantizer values. Link and Kallel [15] have also developed a system which used second order statistics, by increasing the trellis by one dimension to make optimal use of the image redundancy in both vertical and horizontal directions. The two dimensional trellis used in a technique referred to as “sheet decoding” outperforms the MMAP and MSNR techniques by objective and subjective measure.

We find that it best to post-process the first-order JSCD images as opposed to higher order modeled images because the streaks left by the first-order system are uncorrelated single width streaks. They are isolated and more easily detectable than the correlated streaks found in second-order Markov modeled JSCD images.
Chapter 4 Post Processing Techniques

Image processing techniques are frequently used to improve the appearance of a digital image. Sometimes an image may have to be brightened or perhaps the contrast increased. For our case we would like to improve the clarity of the image by further removing the streaks which were unable to be corrected by JSCD. As can be seen from the Figures 3.5 and 3.6, there are many residual streaks, dark and light which still are disturbing to the eye. If these streaks can be accurately located, methods can be applied to correct the data causing the streaks. This chapter investigates such streak detection algorithms, and the methods for the removal of detected streaks.

First the characteristics of the streaks will be investigated. Upon understanding how streaks occur and appear, detection algorithms will be studies and designed. The latter sections deal with symbol correction techniques. The entire system is shown in Figure 4.1.

Figure 4.1 DPCM/JSCD Post Processing System
4.1 Error Patterns

In Chapter 2 the effects of transmission errors on DPCM images were described and two types of error patterns were identified. These error patterns depend on the sign of the transmission error magnitude. If \( \bar{x}_i \) is the pixel value of the streak and \( x_i \) is the correct value, the transmission error can be described by the error \( \Delta x_i = \bar{x}_i - x_i \). The streak ends at some value \( m \) at which point \( \Delta x_m \) is below some threshold, or the end of a scan line has been reached. An error pattern with a positive magnitude will cause a brighter streak than the surrounding pixels and similarly an error pattern with a negative magnitude will cause a darker streak. Figure 4.2 demonstrates the two types of streaks caused by transmission errors. However, the two cases do not need to treated separately. Generally all streaks fade and the length of the error streaks are dependent on the magnitude of the error. The error causes the symbol to represent a different quantizer level. As the separation between the correct quantizer level and the incorrect quantizer become wider, the magnitude of \( \Delta x_i \) becomes greater. Prediction coefficients also have an effect on the length of the streaks. For first order DPCM systems, the value which gets quantized is \( x_i - a_1 x_{i-1} \) where \( a_1 \) is the single tap prediction coefficient. Therefore \( \Delta x_i \) gets attenuated by the prediction coefficient, and after \( n \) pixels the error caused only by the channel error and not quantization error becomes

\[
\Delta x_{i+n} = a^n \Delta x_i.
\]  

Prediction coefficients closer to the value one cause longer less attenuated streaks.

When the probability of error gets high enough that several errors may occur within one scan line, abruptly terminated short streaks may occur. This occurs due to cancellation of errors.
On the other hand, adjacent errors may occur which cause double width streaks or possibly longer than normal streaks.

So the difficulty of the problem lies in detecting the described error patterns and there are many different approaches to this problem. The following sections will introduce the detection algorithms.

\[
\Delta x_i = \tilde{x}_i - x_i > 0
\]

\[
\Delta x_i = \tilde{x}_i - x_i < 0
\]

**Figure 4.2 Sample DPCM Transmission Error Patterns**

### 4.2 Streak Detection Techniques

#### 4.2.1 Clipped Reconstructed Sample Values

Because pixel values are usually only between the values of 0 and 255, it makes sense for the decoder to have a decoding boundary of the same range. Thus, if a transmission error causes some reconstructed values to fall outside of the range then the values would be clipped to the nearest boundary and the transmission error would be curtailed [5]. Not only can this approach
be utilized to mitigate the effects of transmission errors but in a similar fashion be used to detect streaks. If an image, prior to transmission, is known not to have any extremity pixel values of 0 or 255, then a reconstructed image which contains such values undoubtedly contains errors at or near those locations. For our chosen image Lenna there are no pixel values which take on the value of 0 or 255 (Figure 2.1). Thus, before any post processing begins a search for pixel values of 0 and 255 within the DPCM decoded image can take place. This could be a first pass before any other post-processing takes place. A thresholding method which will be described in the following section catches most of these errors and hence to save computations, our system will simply bypass the search for extreme values.

4.2.2 Thresholding Techniques

From the error profiles shown in Figure 4.2 it makes sense to use an algorithm encompassing some sort of thresholding. If adjacent pixels differ by a large amount, much larger than a given threshold, then an error could have occurred. Lippman [5] has developed a technique for channel error correction in DPCM picture transmission. Based on the premise that most images have high correlations of pixels from line to line, a thresholding technique can be used to detect errors.

For a given decoded DPCM scan line, a window of pixels of size \( W \) is considered. Referring to Figure 4.3, the pixels within the window of the current line II are compared to the adjacent pixels of the previous line I and the next line III. This window of three lines slides pixel by pixel along each row computing two sets of gradients. The first set are the differences of the pixels of the current line II and line I. The second set are the differences of the pixels between line II and line III. Figure 4.3 illustrates the window of pixels and gradients calculated.
This algorithm is analogous to searching for edges along both sides of the current line. The gradients are constantly compared to a threshold constant $K$. Lippman proposes a constant $K_1$ for the very first pixel gradient $g_{p1}$ and $K_2$ for the remaining gradients. If all gradient magnitudes exceed the corresponding thresholds, $g_{p1} > K_1$ and $g_{pi} > K_2$ for all $i$ such that $2 \leq i \leq W$, then a streak is declared to be detected and the initial sample value of the current line is deemed as the beginning of the streak and the corresponding symbol is declared to be received in error.

For Lippman’s simulations a value of $K_1=12$ and $K_2=10$ with a window size of $W=10$ were used. These values are purely experimental and may prove to work well for some images and not so well for others. A rule of thumb for choosing the threshold values would be to choose low values so most steaks including faint ones have a chance of being detected but not too low otherwise too many false alarms occur when dealing with highly detailed images. Thus when dealing with the transmission of many different images it is difficult to arrive at a set of parameters which will be suitable for all cases.
So with Lippman’s algorithm we tried it on the Lenna image. However, a window size of 8 as opposed to 10 was found to give better performance because of the Chang Donaldson prediction coefficient causing the streaks to be much shorter. For these particular simulations, it is assumed that once a streak has been detected the symbol is automatically corrected correctly. This would not be feasible in real cases, and so in the latter sections of this chapter correction techniques will be discussed and evaluated.

From Figures 4.4 one can see that the algorithm works well with a 3 bit system. The streaks in the 2 bit system are very faint and difficult to detect even to the eye. However, the streaks in the 3 bit system are far more pronounced and easily distinguishable. Thus the effect of the Lippman algorithm is far more impressive. Figure 4.4d) shows a fairly clean image compared to that of the streak filled image of Figure 4.4c).

Not all the streaks were detected. The algorithm fails when the streaks are too faint and the gradients fall below the threshold values. It will also fail if the streaks are much smaller than the window size; and furthermore in areas of high detail a streak may go undetected. Lippman reports good results at a probability of error rate of $10^{-4}$ and we have shown reasonable results for a higher rate of $10^{-3}$, but we would like to implement this for even high error rates of up to $10^{-1}$. At these higher rates far more streaks will overlap, perhaps some cancelling one another. Thus there will far more streaks shorter than the window size, meaning far more undetected streaks. As well there will be many more faint streaks adding to the distortion of the image and causing the detection of sharper streaks to be that much harder. The Lippman algorithm also requires a set of input threshold parameters. Hence, a modified Lippman algorithm which addresses these problems is presented in the next section.
a) 2 Bit DPCM

b) 2 Bit DPCM + Lippman Streak Detection

c) 3 Bit DPCM

d) 3 Bit DPCM + Lippman Streak Detection

Figure 4.4 Lippman Streak Detection for Probability of Error = 0.001
4.2.3 Modified Iterative Lippman Algorithm

The problem with the Lippman algorithm is that there were not enough streaks detected and at high error rates we would like to correct far more streaks than the original Lippman algorithm is capable of. Also, there is a need to choose proper thresholding parameters, as the detection performance is very sensitive to thresholds chosen. A modified version of the Lippman algorithm has been developed to tackle these problems. To correct a greater portion of errors, it uses the channel statistics to predict the approximate number of errors per scan line. Thus a fixed number of symbols can be attempted to be corrected per line. Because the JSCD system uses knowledge of the channel, the same information could be passed on the post-processing system.

The algorithm uses the following steps:

- From a start of the line reset the target counter and use the Lippman window technique to locate all possible streaks with a starting set of thresholds.
- If a streak or streaks are detected then attempt to correct the streak which has the largest gradient and redecode the row.
- If no streaks are detected lower the thresholds by one unit.
- If thresholds are below the lower bound then use a smaller window size.
- If the counter meets the target value then go to next line and redo process.

The flow chart in Figure 4.5 shows the order and logic by which these steps are executed.
Figure 4.5 Flow Chart for Modified Lippman Algorithm
For the simulations we took the Gray coded JSCD images and applied the modified Lippman algorithm to the images. A window size of 8 was used for $W_1$ and for the second pass of each scan line a size of 4 pixels was used for $W_2$. Only starting threshold values are needed and as long as they are relatively high, small differences in initial values make little or no difference at all in performance of streak detection. The only other parameter needed is the target value of symbols to fix per row.

We chose a percentage of the possible errors per scan line. Assuming the JSCD decoder corrects some errors, not all errors will have remained and the actual BER arriving at the post-processor will be less. Furthermore, due to the nature of Gray codes causing faint streaks more often than high transition streaks, the thresholding algorithm will miss a majority of streaks. Hence the target value must be set at a value which will not be too high where non errors will be misappropriately get corrected; but must not be set too low such that the number of errors corrected will be insignificant. We chose a percentage of 50 per cent of possible errors per scan line. So for a scan line of 512 pixels at a BER of 0.01, the approximate number of errors would be 10. Then the modified algorithm attempts to fix at most 5 symbol errors per scan line, but does not need to always correct at total of 5 symbols every line.

Figures 4.6 and 4.7 gives an idea of how many streaks get detected at high bit error rates. Once again all streaks detected are corrected with the proper known symbols and hence the curves are optimistic upper limits.
At a high probability of error $= 0.10$ the two bit system reduces the amount of errors by 13.8% whereas the 3 Bit system reduces the BER by 9.3%. Furthermore, looking closely at the lower error rates of the 2 Bit system, one can see that the streak detection algorithm improves the image at points where JSCD does not contribute to any error correction at all. At high error rates it can be seen that much of the gain in the three bit system comes from JSCD whereas at low error
rates of the gain is due to streak detection and correction.

4.3 Symbol Correction Techniques

Once a streak has been detected, a technique which eliminates or reduces the streak must be implemented. One such technique is to paint over the streak with neighbouring pixel values having assumed that the picture has relatively high pixel to pixel correlations. At high error rates, this procedure becomes undesirable, because streaks which fall above or below other streaks occur more frequently. Hence, we propose to attack the problem by correcting the DPCM code word by redecoding as opposed to estimating the decoded pixel values. The JSCD technique that we used was based on decoding sequences of symbols, but now we are redecoding with an approach which looks at a single symbol at a time. The following sections describe different methods of attempting to choose the correct symbol. They are the MAPRI Symbol, MAPRI Transition and Minimum Squared Error (MSE) correction schemes.

4.3.1 MAPRI Symbol

In this case, the symbol in error at position \( i \) is replaced with the symbol \( s_i \) which maximizes the a priori probability of being correct. Therefore the term which must be maximized is

\[
P(s_i).
\]  

(4.2)

The first probability term is again related to the channel conditions and the second term is simply the probability of occurrence of each symbol. If the symbol which maximizes (4.2) is the same as the deemed erroneous symbol, we choose the symbol which gives the second highest value of (4.2). This method has greater potential with the two bit system, simply because there are less
symbols and the probability of choosing the correct symbol is much higher than would be for the three bit system. For the Lenna image, our two bit system consists of a codeword which amounts to 53 per cent of all code words. The next most frequent symbol occurs 36 per cent of the time for the Lenna image.

4.3.2 MAPRI Transition

This procedure also relies on a priori information about the source. However, instead of choosing the most likely codeword, it selects the most likely codeword transitions. Hence, the metric which must be maximized is

\[ P(s_i|s_{i-1}) . \] (4.3)

Therefore, once a streak has been detected and the starting symbol in error has been determined, it requires knowledge of the previous symbol before the start. Assuming the previous symbol is correct, it chooses the most likely state transition and replaces the symbol in error. It is evident that the method relies heavily on the performance of streak detection. If a streak has been detected but the symbol in error has not been pinpointed, then a correct symbol may be changed which may cause worse effects. Hence, we developed two schemes to compare.

**Scheme I:** Force the symbol which is assumed to be in error to be changed. So if the most likely transition happens to terminate with the same symbol received then choose the second most likely transition. Consequently, the corrected symbol will always differ from the received symbol.

**Scheme II:** If the previous symbol and the symbol in question happens to be the most likely transition then do not attempt to make any correction on the latter symbol. Keep the
received symbol and carry on looking for other streaks.

### 4.3.3 Minimum Squared Error

The previous methods utilize a priori information about the source. However, in this method the decoded DPCM image is used to decide the best symbol. Using the window mentioned in section 4.2.2 the best candidate is chosen by selecting the codeword which gives the minimum squared error. So in the two bit case, the codeword in error is replaced by one of the four code words and the row is redecoded. Of course the true error can not be calculated, but as in the streak detection method a window of pixels are taken and the error is assumed to be the difference of the pixel in the current row and the corresponding pixel in the previous row within the same column. So the squared error calculation is as follows

\[
\sum_{i=0}^{W} (x_{r,c+i} - x_{r-1,c+i})^2
\]  

(4.1)

where \( r \) is the row, \( c \) is the column and \( W \) is the length of the window. As in the MAPRI Transition Scheme II the original symbol received is allowed to be the best candidate as long as it minimizes the squared error. Because of the dependence of the pixels from the previous row, this method works well when streaks are not lying on top of one another. Another drawback to this algorithm is that it requires the most amount of computations and hence takes the longest time to perform. However, this method appears to perform the best in terms of objective signal to noise ratios which is revealed in the following section.
4.4 Simulation and Discussion of Results

The four different symbol correction techniques were tested on the JSCD decoded Lenna image using the modified Lippman algorithm described in section 4.2.3. Window sizes of 8 and 4 were used and initial values for the thresholds were $K_1 = 30$ and $K_2 = 28$.

The window sizes were chosen so that long streaks are first detected and then shorter streaks which are cancelled by a following error are detected. The window size can also be related to the prediction coefficient since the predictor has an effect on the length of the streaks. We find that the higher the prediction coefficient is the longer the window should be. A simple relation which can be used as a rule of thumb, but does not necessarily give the optimal window size for any given image, would be to choose the window size of integer value $W_1$ such that

$$a_1^{W_1} = 0.5$$

(4.1)

where $a_1$ is the single tap prediction coefficient. The rule for the second window size is to simply choose a window which is half the size of $W_1$ rounded to the nearest integer.

For the simulations, Gray codes were used on the Reoptimized DPCM systems for the two bit and three bit cases. Furthermore, the same random number seed used for the binary symmetric channel was used for all four techniques so each could be compared fairly against one another. Figures 4.8 and 4.10 depict the SNR curves for the two bit system and three bit system respectively.
For the two bit system, we can see that the MSE technique performs the best achieving a gain of 2.3 dB at a BER of 0.02. It is also evident that the MAPRI Symbol and both MAPRI Transitions techniques perform nearly the same. At very high BERs of 0.07 and over, all schemes seem to breakdown and have very poor gains of only 0.8 dB. This can be explained by that fact that there are too many overlapping and overlying streaks and streak detection becomes much more difficult. From Figure 4.9 one can see subjectively the images are in agreement with the SNR curves. Figure 4.9b) depicting the MSE image is much cleaner than the other images. What remains are very faint streaks which are less annoying that the darker streaks which have been removed.
Figure 4.9 Subjective Comparisons for Symbol Choice Techniques of a 2-Bit DPCM + JSCD System

a) DPCM + JSCD, Prob of Error = 0.01
   SNR = 19.31 dB

b) MSE
   SNR = 23.17 dB

c) MAPRI Symbol
   SNR = 22.17 dB

d) MAPRI Transitions Scheme II
   SNR = 22.11 dB
For the three bit case, the MSE also outperforms all other schemes. It achieves a gain over the JSCD decoded image of 2.69 dB at a BER of 0.01. This translates to a gain of 4.56 dB over the DPCM decoded image. Even at very high BER's of up to 0.10 the gain is 1.61 dB. From the curves we can see that opposite to that of the JSCD gain over DPCM, the MSE gain attenuates as it approaches higher probability of errors. This simply can be explained by that fact that there are too many streaks which overlap one another thereby making error correction much more difficult. From Figure 4.11d) it is evident that the algorithm cleans the corrupted image quite nicely in the areas of very low detail. However, within the areas of high detail and sharp gray level transitions, there remains some artifacts. As in the case of the 2-bit system the MAPRI schemes do not differ
Figure 4.11 Subjective Comparisons for Symbol Choice Techniques of a 3-Bit DPCM + JSCD System
by a great deal and the images of Figure 4.11 clearly shows little difference at all. These schemes also appear to work well in areas of low detail and have trouble within areas of sharper transitions.

For both the two and three bit systems the MSE correction method is the best choice subjectively and objectively.

4.4.1 Performance at Low BER

The curves of Figure 4.8 and 4.10 show that the system works well for very low BER's of less than 0.02. In fact, the gains of the streak detection schemes from the MSE scheme are greater than the gains achieved by JSCD. Therefore, a system designed for low BERs is better off without the JSCD decoder because it gives only a small improvement at the price of a significant increase in complexity.

![Figure 4.12 MSE Error Correction Performance at Low BER for a 3-Bit System](image-url)
In Figure 4.12 we have replotted the SNR curves for the three bit system at a low BER of less than 0.02. Also, we have added a curve which represents the modified Lippman algorithm applied only to a corrupted DPCM image. Surprisingly, the streak detection performed on the corrupted DPCM image performs better than the streak detection performed on the JSCD image. The gain is quite small, but the amount of signal processing time that is reduced is quite significant. Although the SNR curve is higher for the case without JSCD, from Figure 4.13 the subjective quality is better for the case where JSCD has taken place. This occurs because JSCD removes many of the errors including overlapping streaks. So the detection algorithm doesn’t have to deal with as many double or triple width streaks. It is for these special cases, the modified Lippman detection algorithm fails. Otherwise, most single width streaks are detected and corrected leaving a reasonably clear picture.
Chapter 4 Post Processing Techniques

Figure 4.13 Streak Detected Images for BER=0.006 for 3-Bit DPCM system

a) DPCM
SNR = 22.70 dB

b) JSCD
SNR = 24.22 dB

c) DPCM + Mod Lippman
SNR = 27.40 dB

d) JSCD + Mod Lippman
SNR = 26.59 dB
Chapter 5 Conclusion and Future Work

The design of a DPCM system which combats against channel errors without the addition of redundancy has been investigated. Our proposed system incorporates three components. They are the DPCM encoder/decoder, joint source channel decoder, and the post-processor. Using the standard monochrome image, Lenna, a DPCM encoder was implemented with a Lloyd Max quantizer. Then a binary symmetric channel injects errors into the encoded image and once the corrupted image has been decoded, the classical DPCM system was compared to the reoptimized DPCM system using the Chang Donaldson prediction coefficient. At a bit error rate of 0.1 we find the reoptimized DPCM system to outperform the classical system by about 7dB for the two bit system and by over 6 dB for the three bit system.

The joint source channel decoders were also studied. Assuming the system has knowledge of the channel characteristics and knowledge of the source statistics, the JSCD decoder can correct a significant portion of the channel errors. These are bit error rate minimizing decoders which use Markov statistics within a trellis to choose the most likely sequence of quantizer values. It is found from previous studies that JSCD can improve a corrupted image significantly. At very high error rates, an almost indistinguishable image is turned into something that is recognizable. We find that JSCD works far better for a three bit system than the two bit, giving a gain of as much as 3 dB for the three bit system. We also find that the system is also quite sensitive to the training sequence used. However, an iterative approach gives similar error performance without the cost of having to transmit side information. The trade-off is a longer decoding delay for less amount of transmitted data. Furthermore, having studied and implemented different code word mapping schemes, we find that the Gray code works best for both bit systems by
achieving the highest SNR and highest subjective quality. The JSCD system was also modified to incorporate second-order Markov statistics, but the gain achieved over the first-order systems is not worth the increase in complexity. Instead, we propose the use of post-processing techniques to correct the errors which were unable to be corrected by JSCD.

Our post-processing system incorporates an iterative post-processing technique. It is a modification of a simple thresholding technique developed by Lippman [5]. It attempts to locate a streak using a sliding window and instead of painting the corrupted pixels within the window, our system goes back to the DPCM encoded image and attempts to correct the symbol. Different symbol correction techniques were investigated and it was evident that the MSE scheme outperformed the MAP Symbol and MAP Transition schemes. The MSE scheme relies heavily on image redundancy at the DPCM decoded level, and the MAP schemes rely heavily on redundancy at the DPCM encoded level. For all cases, the post processing gives a continuous improvement for both the two bit and three bit systems. A gain of over 4 dB is seen for the three bit system with the use of the MSE correction scheme. More importantly, the images are much cleaner with very disturbing dark and bright streaks removed.

In conclusion, the DPCM/JSCD post-processed system achieves its goal of obtaining an image which is far cleaner and appealing than the noise corrupted image. No channel coding was used; only the residual redundancy encapsulated by a first-order Markov model was used. By objective and subjective measure, our system has proven to mitigate the effects the channel errors.

Further research in this area could be conducted. It includes a system for not only monochrome images but colour as well. Also, the techniques described herein could be applied to JPEG images and MPEG video sequences. Furthermore, similar techniques could be developed
for images that have been source coded by methods other than DPCM. The JSCD techniques herein described required only that the source code words have a fixed length. Examples of such source coders include trellis coded quantization and vector quantization. However, the post-processing techniques that we have developed are very specific to DPCM.
Bibliography


