SOURCE CODED IMAGE DATA IN THE PRESENCE OF CHANNEL ERRORS

THESIS

BY

TERENCE EDWIN DODGSON

Presented to the Department of Electrical and Electronic Engineering of the UNIVERSITY OF ASTON IN BIRMINGHAM for the degree of DOCTOR OF PHILOSOPHY

BIRMINGHAM

JANUARY 1986

DEDICATED TO : My father Ted Dodgson and mother Inge Dodgson.

ACKNOWLEDGEMENTS

The Author would like to express his appreciation and thanks to his supervisor, Dr.R.L.Brewster, for his guidance, assistance and for his persisting encouragement throughout the research.

The Author is also very grateful to his assistant supervisor, Dr.R.Wilson, for lending his expertise in the field of source image coding.

Finally, my thanks are also due to my colleagues and friends for their helpful discussions.

SOURCE CODED IMAGE DATA IN THE PRESENCE OF CHANNEL ERRORS

by

TERENCE EDWIN DODGSON

submitted for the degree of
Doctor of Philosophy
at
The University of Aston in Birmingham
1986

SUMMARY

The growth and advances made in computer technology have led to the present interest in picture processing techniques. When considering image data compression the tendency is towards transform source coding of the image data. This method of source coding has reached a stage where very high reductions in the number of bits representing the data can be made while still preserving image fidelity. The point has thus been reached where channel errors need to be considered, as these will be inherent in any image communication system.

The thesis first describes general source coding of images with the emphasis almost totally on transform coding. The transform technique adopted is the Discrete Cosine Transform (DCT) which becomes common to both transform coders. Hereafter the techniques of source coding differ substantially i.e. one technique involves zonal coding, the other involves threshold coding.

Having outlined the theory and methods of implementation of the two source coders, their performances are then assessed first in the absence, and then in the presence, of channel errors. These tests provide a foundation on which to base methods of protection against channel errors. Six different protection schemes are then proposed.

Results obtained, from each particular, combined, source and channel error protection scheme, which are described in full are then presented. Comparisons are made between each scheme and indicate the best one to use given a particular channel error rate.

Keywords
CHANNEL, ERROR-PROTECTION, DIGITAL,
IMAGE PROCESSING, RATE-DISTORTION.

CONTENTS

PAGE

Acknowledgements

Summary

Contents (i)

List of Figures (vii)

CHAPTER 1

INTRODUCTION

1.1 General 1

1.2 An Introduction to Source Encoding of Image Data 6

1.3 Aims of Present Research 9

CHAPTER 2

" STATE-OF-THE-ART " REVIEW

2.1 Source Encoding of Image Data 11

2.1.1 Predictive Coding 11

2.1.2 Transform Coding 13

2.1.3 Methods of Quantizing Transformed Coefficients 16

2.1.3.1 Rate-distortion Characteristics and Bit Allocation (Zonal Coding) 16

2.1.3.2 Entropy Coding (Threshold Coding) 18

2.1.3.3 Quantizers 19

2.1.3.4 Coding and Visual Fidelity in the Perceptual Domain 20

2.1.4 Hybrid Coding 22

2.2 Effects of, and Protection Against, Channel Errors 23

2.2.1 Channel Errors in Predictive Coding 24

2.2.2 Error Correction for Channel Errors in Predictive Coding 24

PAGE

2.2.3 Channel Errors in Transform Coding 26

2.2.4 Error Correction for Channel Errors in Transform Coding 27

2.2.5 Channel Coding/Protection 3Ø

CHAPTER 3

SOURCE ENCODING/DECODING

3.1 Introduction - General Transformation Theory 34

3.2 The Discrete Cosine Transform (DCT) from the General Transform Theory 4Ø

3.3 The Discrete Cosine Transform (DCT) Relationship to the Karhunen-Loeve Transform (KLT) 42

3.4 Employment of the DCT when Source Encoding 52

3.5 Zonal Coding 55

3.5.1 A Rate-distortion Function 57

3.5.2 The Human Visual System (HVS) Spatial Frequency to rad./degree 59

3.5.2.1 Spatial Frequency to rad./degree 63

3.5.3 Summary of Zonal Encoding/Decoding Scheme 67

3.6 Threshold Coding 71

3.6.1 Thresholding 73

3.6.2 Huffman, Amplitude and z.r.l. Coding 75

3.6.3 Summary of Threshold Encoding/Decoding Scheme 83

3.7 The Quantizer 88

3.7.1 Relationship for Level Spacing to Give Minimum Distortion 89

3.7.2 An Approximate Method of Obtaining Quantization Levels 94

3.7.3 Level Spacing for the Gaussian Probability Density 96

3.7.4 Use of the Quantizer PAGE 98

CHAPTER 4

RESULTS/PERFORMANCE OF SOURCE ENCODING SCHEMES IN THE ABSENCE OF CHANNEL ERRORS

4.1 Introduction - Assessing Performance of Source Coders 1Ø2

4.2 Fidelity Measures 1Ø5

4.2.1 Experimental Measures 1Ø5

4.2.1.1 Mean Square Error and Signal-to-Noise Measures 1Ø5

4.2.1.2 Error Spectrum 1Ø8

4.2.2 Theoretical Measures 1Ø8

4.2.2.1 Theoretical, Expected, Distortion with the Zonal Encoding Scheme 1Ø9

4.3 Performance in the Absence of Channel Errors 111

4.3.1 Distortion with Zonal Source Encoding 113

4.3.2 Distortion with Threshold Source Encoding 125

4.3.3 Sub-image Energy Considerations 143

4.3.4 Summary/Comparison of Distortion Introduced by Different Schemes 147

4.4 Performance of Unprotected Source Coded Data in the Presence of Channel Errors 15Ø

4.4.1 Random Error Simulation 152

4.4.2 Expected Failure of Source Coding Schemes 154

4.4.3 Errors in Various Components/Preliminary Testing 156

4.4.3.1 Single Errors in Zonal Coded Sub-images 157

4.4.3.2 Single Errors in Threshold Coded Sub-images 167

PAGE

4.4.4 Unprotected Source Coded Data Subjected to Uniform Random Errors 176

4.4.4.1 Zonal Coding - unprotected code subject to various channel error rates 177

4.4.4.2 Threshold Coding - unprotected code subject to various channel error rates 181

4.5 Summary of Overall Performance Including Channel Errors 187

CHAPTER 5

CHANNEL ERROR PROTECTION SCHEMES

5.1 Introduction - Coding Theorems 190

5.1.1 Source Coding Theorem 190

5.1.2 Noisy Coding Theorem 193

5.2 Source Coding Parameter Adjustments 194

5.2.1 Zonal Coder Parameters Adjustment 195

5.2.2 Threshold Coder Parameter Adjustment 196

5.3 Block Codes with Linear Properties 201

5.3.1 General Linear Code Theory 201

5.3.2 Q-ary Symmetric Channels with Syndrome Decoding 206

5.3.3 Hamming Codes 209

5.3.4 Hamming Code Rate and Probable Failure 213

5.4 Convolutional Codes 215

5.4.1 General Convolutional Code Theory 215

5.4.2 Hagelbarger Convolutional Coding 220

5.4.3 Rate and Probable Hagelbarger Code Failure 224

5.5 Maximum Likelihood Decoding of Huffman Code 228

5.6 Channel Error Protection Schemes 232

PAGE

5.6.1 Scheme 1 - Parameter Adjustments Only 233

5.6.2 Scheme 2 - Equal Importance Protection through Hamming Block Coding 234

5.6.3 Scheme 3 - Equal Importance Protection through Hagelbarger Convolutional Coding 235

5.6.4 Scheme 4 - Percentage Sub-image Protection, using Hamming Block Codes 236

5.6.5 Scheme 5 - Percentage Sub-image Protection, using Hagelbarger Convolutional Codes 238

5.6.6 Scheme 6 - The Performance, Maximum Likelihood Decoding of a Huffman Coded Bit-stream 239

CHAPTER 6

RESULTS/PERFORMANCE FOR OVERALL SOURCE CODED AND CHANNEL ERROR PROTECTED IMAGE COMMUNICATION SCHEMES

6.1 Results/Performance with Source Coding Parameter Adjustments Only (Scheme 1) 244

6.2 Results/Performance of Image Coding Systems with Equal Importance Block Coding (Scheme 2) 251

6.3 Results/Performance of Image Coding Systems with Equal Importance Convolutional Coding (Scheme 3) 258

6.4 Results/Performance of Image Coding Systems with Percentage Sub-image Block Coding (Scheme 4) 266

6.5 Results/Performance of Image Coding Systems with Percentage Sub-image Convolutional Coding (Scheme 5) 284

6.6 Theoretical Performance of a Maximum Likelihood Decoder on Huffman Coded Image Data 3Ø2

6.7 Discussion on Results of Combined Source/Channel Error Protection Schemes 3Ø5

CHAPTER 7

CONCLUSION AND DISCUSSION 3Ø8

7.1 Does Source Coder Complexity Make Any Difference? 3Ø9

PAGE

7.2 Are Errors in Certain Coefficients More Likely to Produce More Undesireable Effects than in Others? 313

7.3 Are All Bits of a Word, Corresponding to a Certain Coefficient, As Important As Each Other? 315

7.4 Does Adjustment of Source Coding Parameters Make the Coded Data Less Sensitive to Channel Errors? 316

7.5 What are the Best Combined Source/Error Protection Methods for Transmission of Image Data over a Channel which has a Specified Error Rate? 320

7.6 Suggestions for Further Work 324

APPENDIX 326

REFERENCES 356

LIST OF FIGURES

		PAGE
1.1.1	A Communication System Block Diagram	2
3.3.1	Original Image	44
3.3.2	Ideal and Practical Spatial Correlation Between Adjacent Pixels of Original Image	44
3.3.3	(8x8) Toeplitz Matrices	46
3.3.4	Eigenvalues and Eigenvectors of [Z] for N=8 and ρ=Ø.9	47
3.3.5	Eigenvalues and Eigenvectors of [Z] for N=8 and ρ=Ø.96	47
3.3.7	Comparison of Eigenvectors (ρ=Ø.9) with DCT Basis Vectors	5Ø
3.3.8	Comparison of Eigenvectors (ρ=Ø.96) with DCT Basis Vectors	51
3.4.1	Block Diagram of Source Encoding Schemes	52
3.5.1	Typical "bit-map" for Zonal Quantization	56
3.5.2.1.1	Image Representation	63
3.5.2.1.2	Viewing Distance and Viewing Angle	64
3.5.3.1	Zonal Source Encoder, Block Diagram	68
3.5.3.2	Zonal Source Decoder, Block Diagram	7Ø
3.6.1.1	Method of Scanning Transformed Coefficients	73
3.6.1.2	Representation of Data Stream after Scanning	74
3.6.2.1	Typical Huffman Code Tables	76
3.6.2.2	Flow Diagram for Formation of Huffman Code Tables	84
3.6.3.1	Threshold Source Encoder, Block Diagram	85
3.6.3.2	Threshold Source Decoder, Block Diagram	87
3.7.1.1	Non-uniform Quantization	9Ø
3.7.4.1	Compression Curve of the General Quantizer	1ØØ

PAGE

3.7.4.2 Table "look-up" Form of General Quantizer 1ØØ

3.7.4.3 Compression Curves of the Gaussian Quantizer 1Ø1

4.3.1 Block Diagram for Compression Schemes, Assessment in the Absence of a Channel 112

4.3.1.1 Estimated Variances and Bit-maps 114

4.3.1.2 Image Reconstructions at 1.Ø Bits/pixel (a) 32x32 (b) 16x16, Pixels Square 117

4.3.1.3 Image Reconstructions at 1.Ø Bits/pixel (a) 8x8 (b) 4x4, Pixels Square 118

4.3.1.4(a) Experimental, Distortion versus Bit-rate 119

4.3.1.4(b) Theoretical, Distortion versus Bit-rate 121

4.3.2.1 Amplitude Probability versus Position in Sub-image 127

4.3.2.2 Image Reconstructions at 1.Ø Bits/pixel (a) 32x32 (b) 16x16, pixels Square 131

4.3.2.3 Image Reconstructions at 1.Ø Bits/pixel (a) 8x8 (b) 4x4, Pixels Square 132

4.3.2.4 Image Reconstructions at 1.Ø Bits/pixel (a) 32x32 (b) 16x16 (c) 8x8 (d) 4x4, Pixels Square 133

4.3.2.5 Distortion versus Bit-rate (8 bit Quantization) 134

4.3.2.6 Distortion versus Bit-rate (7 bit Quantization) 135

4.3.2.7 Distortion versus Threshold (7 bit Quantization) 139

4.3.2.8 Distortion versus Threshold (8 bit Quantization) 14Ø

4.3.3.1 (a) Original 512x512 image (b) Ø.4 Bits/pixel Reconstruction (c) Ø.4 Bits/pixel Threshold Reconstruction 146

4.3.4.1 Comparison of Zonal and Threshold Coding 151

4.4.1.1 Probability Line 154

4.4.3.1.1 Zonal Sub-image Terminology Map 157

4.4.3.1.2 A Typical Sub-image 159

PAGE

4.4.3.2.1 'Apparent' E.O.S. Deletion 172

4.4.3.2.2 'Apparent' E.O.S. Insertion 172

4.4.4.1.1 Unprotected, Zonal, Source Coded Data Subject to Various Channel Error Rates (Sub-image 16x16) 178

4.4.4.1.2 Unprotected, Zonal, Source Coded Data Subject to Various Channel Error Rates (Sub-image 8x8) 180

4.4.4.2.1 Unprotected, Threshold, Source Coded Data Subject to Various Channel Error Rates (Sub-image 16x16) 183

4.4.4.2.2 Unprotected, Threshold, Source Coded Data Subject

6.3.3 Zonal Image Coding with Equal Importance Convolutional Coding Subject to Channel Errors at a Rate 1 in 100 264

6.3.4 Threshold Image Coding with Equal Importance Convolutional Coding Subject to Channel Errors at a Rate of 1 in 100 265

6.4.1 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 268

6.4.2 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 269

6.4.3 Zonal, Image Reconstructions at 40% S.I.P., Channel Error Rate 1 in 1000 272

6.4.4 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 273

6.4.5 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 274

6.4.6 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 276

6.4.7 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 277

6.4.8 Threshold, Image Reconstructions at Indicated Percentage S.I.P., Channel Error Rate 1 in 1000 288

6.4.9 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 282

PAGE

6.4.10 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 283

6.5.1 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 286

6.5.2 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 287

6.5.3 Zonal, Image Reconstructions at Indicated Percentage S.I.P., Channel Error Rate 1 in 1000 288

6.5.4 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 291

6.5.5 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 293

6.5.6 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 295

6.5.7 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 1000 296

6.5.8 Threshold, Image Reconstructions at Indicated Percentage S.I.P., Channel Error Rate 1 in 1000 298

6.5.9 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 300

6.5.10 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate 1 in 100 301

6.6.1 A Communication Process Probabilistic Diagram 302

6.6.2 Average Probability of an Incorrect Decoding of Huffman Coded Bit-stream using Maximum Likelihood Decoding 304

CHAPTER 1

INTRODUCTION

1.1 General

Present interest in picture processing techniques has arisen due to recent advances in computer technology. When dealing with image communication the main problem one is faced with is channel capacity and achieving a reduction in the number of bits necessary to represent the image, while maintaining acceptable image accuracy or quality. A typical image requires 256 x 256 x 8 (i.e.524,288) bits when represented simply as pixels, each pixel being a number lying on a grey scale, quantized to 256 levels, requiring 8 bits. This is a large number of bits and it is desireable to reduce this number, before transmission of the data over a channel.

The transmission of images comes under the category of communication, the fundamental problem of which is to reproduce at one point a message selected at another point. In order to analyse the communication problem it is necessary to first

develope a communication system model. The particular model is chosen for two main reasons. First, because experience shows that, it possesses sufficient generality to encompass most of the communication systems met in practice. Second, it is relatively easy to analyse and the results of the analysis can be applied to other systems of a similar nature. The model is shown in Figure 1.1.1, in block diagram form, and is seen to be that representing the classical communication system.

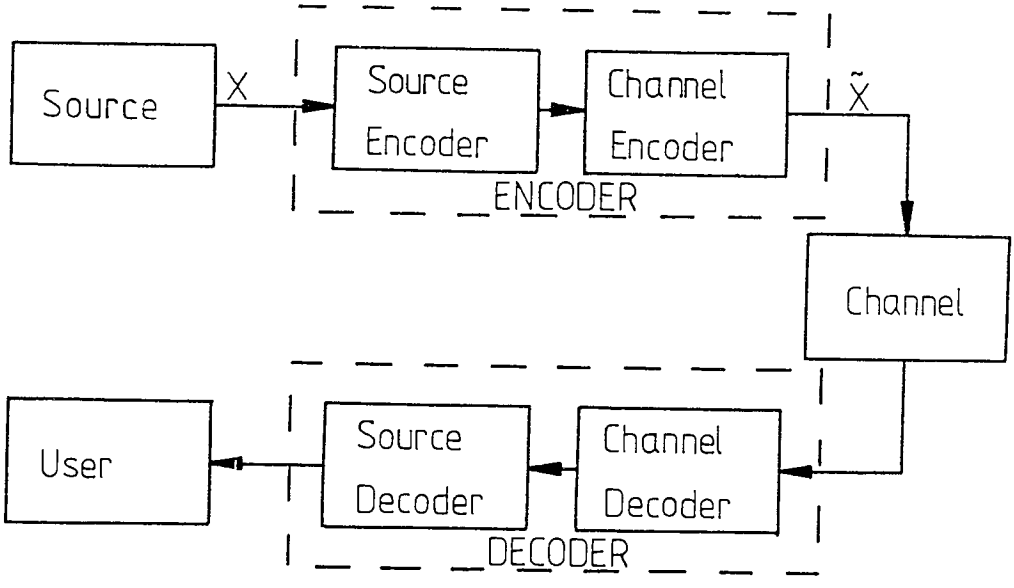


Figure 1.1.1 A Communication System Block Diagram

It will be helpful to run through the general description of each component in the direction of transmission:-

The Source - generates information that the system has to convey to the user.

The Encoder - converts the source output into a form that can be transmitted over the channel.

The Channel - the physical medium, spatially linking the source to the user.

The Decoder - transforms the channel output into a form that can be interpreted by the user.

The User - the recipient, for whom the information was required at the outset.

The encoder and decoder are seen to be divisible into two separate components (refer to Figure 1.1.1). This is a result of being able to divide the fundamental communication problem itself into two component problems:-

The first of these involves deciding from all of the available source data what really should be transmitted and what should not. This results in data compression techniques known as source coding. Inefficient source encoding would involve the use of a further "signal-to-signal" encoder to reduce the data to a form which uses the channel more efficiently. In order to avoid this added complication the source is taken to be the pixels of the image(which are already typically quantized to 256 levels, requiring 8 bits per pixel). Though the main objective of source

encoding is data reduction, the choice of one technique over another is dictated by the requirements of the system in question. The thesis is concerned with image transmission applications for human interpretation and thus, interest lies in techniques which achieve maximum reduction in the amount of data to be transmitted whilst, at the same time, preserving a reasonable fidelity. Reducing the amount of data that must be transmitted is of prime importance and encoding techniques are acceptable as long as the resulting images are acceptable for visual analysis. This implies that some error in the reconstructed image can be tolerated. Source encoding techniques to be considered make use of this fact, which embodies the rate distortion problem. That is to say ; given a defined source, i.e. one whose statistical mechanism generating it's outputs is quantitatively known, and a quantitatively defined distortion measure, one has to determine under what conditions it is possible to design a communication system that reproduces the source output for the user with an average distortion that does not exceed some specified upper limit.

The answer is given by rate-distortion theory [8], [96] and, in simple terms it can be said that, a communication system is realizeable, achieving the specified fidelity, if and only if, the capacity of the channel exceeds the rate-distortion function i.e.the minimum rate at which the source outputs must be produced, in order to achieve a specified average distortion.

Most source encoding strategies and theorems are based on an ideal communication system, in that, the codes are optimum with respect to the given source and fidelity criterion without any regard given to the detailed nature of the channel. The only property of the channel taken into account in these systems is channel capacity, implying that the same source encoder would be used if the channel was replaced by any other one of equal capacity.

Source encoding then, is the process which tries to ascertain what should be transmitted in the image communication system, determining the best form of representing the information so as to be conveyed using the minimum number of bits.

The second of the component problems involves deciding on how to transmit the necessary information accurately, and usually results in an expansion (an increase in redundancy) in the data necessary for transmission. The type of redundancy re-introduced depends on the type of error-protection required. The channel encoder does not require a detailed knowledge of the source. Shannon [96], [97] has produced much material on the encoding of selections from a set of possible messages in such a way that they can be reliably transmitted over a noisy communication channel. The coding theorem for a noisy channel [8] emphasizes the theoretical and practical importance of this coding aspect.

The communication system design, which is to be followed in this work, can thus be summarized as follows. Redundancy is removed from the source by the source encoder, leaving only the most essential information as determined by the fidelity criterion. In order to provide optimum immunity to channel errors, the channel encoder then reinserts redundancy in a special form.

With image coding systems to date (1985) attention has been focused mainly on the source encoding of image data, with little or no consideration given to channel errors. The work in this thesis sets out to rectify this situation, emphasizing the need for channel error protection and how it may be judiciously applied.

1.2 An Introduction to Source Encoding of Image Data.

Due to the many source encoding techniques available, it is necessary to give a brief outline of the main types used for encoding image data. This also serves to introduce some of the terminology and preliminaries. Source encoding techniques can be divided into two main groups:-

First, there is the technique known as predictive coding. The consecutive pixels (picture elements) of an image can be regarded as a data sequence. This data sequence, is often highly correlated in a local region of the image i.e.there is often statistical dependency from one sample to the next. One may, therefore, expect that previously transmitted elements contain information about a currently considered pixel. If accurate prediction of this pixel is possible, then it is not necessary to transmit the pixel but rather a predictor could be placed at the receiver. The more commonly used form of predictive coding is differential pulse code modulation (DPCM). With DPCM only the error signal, i.e.the difference between the predicted and actual pixel values, is transmitted giving a typical compression from 8 bits/pixel to 3 bits/pixel whilst still maintaining reasonable fidelity.

The second group of source encoders rely on, what is known as, a transformation technique. This technique is also known as block quantization. The image is first sub-divided into smaller blocks, sub-images. These sub-images are often square arrays of pixels, for two-dimensional transforming, and are treated independently of each other, as images in their own right. Each sub-image is transformed in order to reduce the correlation between pixels. The idea is then to improve coding efficiency by processing the transformed coefficients independently of each other. If

one uses the same encoder for all sub-images the process is known as non-adaptive tranform coding. With adaptive transform coding a method is used which enables the assignment of a different number of bits to individual sub-images. In general, the sub-images with the most activity/energy, as defined in Section 4.3.3, are assigned a greater number of bits than those of lower activity/energy. This method usually results in lower average bit-rates than the non-adaptive methods, for the same fidelity in reconstruction. A typical, non-adaptive, intraframe, transform coder has little noticeable distortion when coding at around 1.Ø to 1.5 bits/pixel [53].

Due to the large number, and different types, of error-protection codes available an introduction to them is not given at this stage. Rather, it is left until the results on unprotected source coded data in the presence of channel errors have been presented. It is then possible to estimate properties required of the code and how it should be applied to the data - for example, it may not be necessary to provide equal protection to all source encoded bits. Selection of a suitable error-protection code(s) can then be made.

Before embarking on any detailed descriptions of source/channel encoding schemes, the next chapter reviews the more recent work carried out in regions of interest to this thesis.

1.3 Aims of Present Research

The main aim of this thesis is to show the various "trade-offs" one has to consider in the design of a combined source/channel image communication system, answering the question "what would be the best system for transmission of image data over a channel which has a specified error rate?"

The following sections of the literature survey, which are concerned with the areas of interest to this thesis, indicate the work necessary to achieve the main aim. It remains then to draw up ideas and methods into a logical procedure

This involves developing two different, "state-of-the-art", transform, source encoding schemes. The existing problems caused by channel errors are then determined by subjecting each scheme to a noisy channel. From these results, simple, but effective channel coding schemes are devised to reduce, or eliminate, the more undesireable effects of channel errors. The various trade-offs can then be observed by altering the parameters of the image communication system i.e.

a. Type of transform coder

b. Average coding bit-rate (bits/pixel)

c. Transform sub-image size

d. Type of error protection
- Amount of protection
- Particular channel code

e. Channel error rate

An answer can then be provided to the question posed in the first paragraph of this section.

CHAPTER 2

"STATE - OF - THE - ART" REVIEW

In order to arrive at a logical procedure in the design of an image communication system, a review is now given outlining the present state of affairs concerning the areas of interest to this thesis.

2.1 Source Encoding of Image Data

2.1.1 Predictive Coding

Predictive coding makes use of the statistical dependency that exists between adjacent pixels.

In it's simplest form it is known as "linear modulation" or "delta modulation", DM, consisting of a one-step delay function used as the predictor and a one-bit quantizer used to achieve a one-bit representation of the signal. Schindler [94] outlines DM,

drawing reference to Bosworth and Candy [1Ø] who use it in application to video (picture coding). Adaptivity in DM, used to encode video signals, is considered by Lei, Scheinberg, and Schilling [63].

The major drawbacks of DM are slope overload, granularity noise and instability to channel errors. O'Neal [83] looks at two of these limitations (slope overload and granularity noise) and presents an analytical solution for the Signal-to-Noise, S/N, ratio as a function of step size and bit-rate for a signal with arbitrary spectrum and Gaussian amplitude distribution. Computer simulations demonstrate that 375-kc picture-phone signals can be transmitted with peak signal to r.m.s. noise ratios of 2, 4, 8, and 16 times the bandwidth respectively.

The increase in complexity, when compensating for the drawbacks of DM and when using variations of DM, e.g. adaptive DM [55], [63], [98] to increase performance, has to be measured against the more general case of Differential Pulse Code Modulation, DPCM, coding and other techniques for data compression.

The more commonly used form of predictive coding is DPCM

(Culter [25]). The more recent work includes DPCM with entropy coding, DPCMEC. Goyal and O'Neal [35] describe experiments working with a television source encoder consisting of an adaptive DPCMEC system of coding, reported to perform very close to the rate-distortion bound [96]. O'Neal [81] shows that, as the transmission rate gets large DPCMEC forms a source encoding system which performs within 1.53dB of Shannon's rate distortion function [96]. However, O'Neal [81], and O'Neal and Natarajan [82],when comparing DPCM systems with transform coders (excluding any form of entropy coding) conclude that, when coding pictures at low rates (less than, or equal to 1.0 bits/pixel), transform coders produce more subjectively "pleasing" images. Other comparisons [13],[37],[53],[82] of DPCM source coders with corresponding transform coders show that the latter tend to achieve a higher performance, have a smaller sensitivity to fluctuation in data statistics, have errors due to quantization and transmission distributed over the whole of the sub-image and are easier to provide channel error protection for, than for the latter.

2.1.2 Transform Coding

If the distortion in a coding system is defined as the sample mean squared error (see Chapter 3) then, the transformation claimed to keep this distortion minimum is the Karhunen -

Loeve, KL, transform e.g. [53]. The KL transform is often also called the method of Principal Components, eigenvector transform or Hotelling transform due to original work of Hotelling [46]. Although the KL transform is optimal, in the above sense, it is often difficult to compute and has no fast algorithm associated with it. Therefore, in practice, a suboptimal but "fast" unitary transform is used in place of the KL transform. A "fast" transform is simply an algorithm that can compute transform coefficients much more rapidly than other available algorithms, usually due to the inherent properties of the transform itself.

For stationary random sequences there are many fast, unitary transforms [53], [106] which approach the optimization property of the KL transform as the size of the data vector goes to infinity. The Fourier [5], [21], [24], Sine [52], [107] and Cosine [2], [19], [54] are all members of this larger family of sinusoidal transforms [54].

Jain [54] shows that, for first-order stationary Markov processes, the cosine transform matrix performs the closest to the KL transform when the correlation parameter, ρ, lies in the interval (0.5, 1.0), even when the number of samples in the sub-image is small. The Discrete Cosine Transform, DCT, is thus the more commonly used unitary transform when source encoding [16], [18],[79] giving good reconstructions at 0.5 to 1.0 bits/pixel for

a monochrome image and 2 bits/pixel for a colour image. One particular two-dimensional DCT scheme proposed by Chen [16], using entropy coding is claimed to achieve an average bit-rate of Ø.2 bits/pixel for monochrome images and Ø.4 bits/pixel for colour images. It is referred to again at a later stage.

Nonsinusoidal unitary transforms are also often used in source coding systems, due to their fast computational implementation e.g.Hadamard [39],[9Ø], Haar [91], and Slant [91] transforms (square-wave transforms). In comparison schemes, e.g.Ngan [79], Habibi and Wintz [39], Pratt, Chen and Welch [91], the general order of best performance is:- KLT, DCT, FFT, Hadamard, and Haar transform schemes - with the Slant transformation giving a performance either between the DCT and FFT or between the FFT and Hadamard transformation. This resulting order of performance indicates a line of preference for the DCT. This then is the transformation technique to be adopted in this thesis. The next stage is to review, briefly, the more recent work on methods of quantizing transform coefficients.

2.1.3 Methods of Quantizing Transformed Coefficients

2.1.3.1 Rate-Distortion Characteristics and Bit Allocation (Zonal Coding)

(a) The problem of zonal coding can be treated as one of deciding what information, in the image communication system, must be transmitted. From this point of view the problem can be tackled using rate-distortion theory - the concept of which was introduced by Shannon [96]. Berger [8] shows that, by considering the picture vector source components to be independent Guassian random variables of arbitrary mean and specified variance, then a relatively simple rate-distortion function (given in Section 3.5.1), with respect to the mean squared error criterion can be found.

The rate is the average number of bits per pixel required for transmission, hence the number of quantization levels for each transformed coefficient can be determined. This method for bit allocation is known as minimum mean squared error bit allocation for independent Gaussian random variables.

(b) A procedure for quantizing variables with unequal variances was first considered by Huang and Schultheiss [48]. They found

that the number of bits to be used to code a particular coefficient should be proportional to the log-to-base-two of that coefficient's variance. This result is said [39] to include the fact that Huang and Schultheiss use the optimum non-uniform quantizer developed by Max [70]. Kurtenbach and Wintz [61] use the optimum uniform quantizer and this is reported to be the reason for some discrepancy in the development of their bit assignment algorithm. The algorithm developed by Kurtenbach and Wintz is successfully used by Habibi and Wintz [39], for image coding with several linear transformations i.e. FFT, KLT, and Hadamard transformation.

(c) Segall [95] considers the more general problem of optimal allocation of the total number of bits to the components of a memoryless stationary vector source with independent components. This allocation is then applied to various encoding schemes. When applied to the minimum mean squared error encoding scheme, the bit allocation is seen to decompose into the allocation discussed previously in part (a). When applied to the sample-by-sample quantization scheme, Segall's bit allocation decomposes to the result discussed in part (b).

The zonal scheme to be developed is to be based on the minimum mean squared error, hence, using Segall's result,

preference is for the procedure in part (a) (which is to be developed in more detail in Chapter 3).

2.1.3.2 Entropy Coding

(Threshold Coding)

Threshold coding has been recently employed by Chen [16] in his scene adaptive coder. Chen reports that, at the time of writing (1981), the performance of his coder is superior to any existing coding scheme available in the literature. He backs this up by presenting "nearly perfect" reconstruction of a Ø.2 bits/pixel monochrome image and a Ø.4 bits/pixel colour image. The coder has also been hardware implemented. The basic outline of the coder is given at a later stage as it is adopted in this thesis (since it shows such good promise of good reconstructions at low bit-rates). It seems a natural "next step" to observe it's performance in the presence of channel errors and to compare it with zonal coding.

2.1.3.3 Quantizers

The quantization problem, of analogue signals in general, has been studied fairly extensively and a theoretical solution with certain assumptions has been developed - Max [70]. Panter and Dite [84] show how non-uniform distribution of quantization levels can lead to minimum mean squared error distortion when quantizing. The method involves taking into account the statistical properties of the signal, which has a known probability density. At a later date, Max [70] independently derives the equations of the parameters of a quantizer with minimum distortion. As with Panter an Dite, the equations are not soluble without resorting to numerical methods. However, Panter and Dite put forward an approximate method for a quicker derivation of levels and this is the method adopted in this thesis.

Other, more recent, work on quantizers includes that by Kurtenbach and Wintz [61], solving the problem for a noisy channel and showing how the solution reduces to that derived by Max, and Panter and Dite, for a noiseless channel. However, as Kurtenbach and Wintz themselves point out, the solution gains little over the noiseless condition. Also, since assumptions, which may not be strictly valid e.g.the assumption of Gausssian signal statistics

are made anyway, it is thought best to adhere to the noiseless channel solution.

2.1.3.4 Coding and Visual Fidelity in the Perceptual Domain

In the previous sections, apart from mention of the mean squared error criterion, no special consideration has been given to the visual evaluation of images. Shannon's rate-distortion function [96] provides a bound which is not applicable unless a quantitative measure of distortion can be found which corresponds reasonably with the qualitative evaluation of the observer. This particular problem is still in it's very first stages of development.

By picking a class of distortion measures for which the average rate-distortion function can be calculated, and for which optimum coding can be simulated, Mannos and Sakrison [68] compare different distortion measures in the class by simulating the encoding of a fixed image at a fixed rate (bits/pixel) under different distortion measures and subjectively judging the quality of the encoded images. The result is the proposition, by Mannos and Sakrison, of a weighted mean squared error criterion. By weighting the quantization distortion with the proposed weighting

factor, a transform coding technique can be developed producing distortion which, subjectively, corresponds to the eye's visual perception. A review of this technique can be found in Jain [53]. It is only possible to be critical of this weighted distortion function in a positive sense, as it is (from the author's point of view) the best attempt to date at proposing a mathematical model, fairly consistent with results obtained in various psychophysical experiments (e.g.[12], [14], [57], [85], [100],etc.). The result is thus adopted in this thesis for use with the zonal coder to be developed. The coder itself will not be that outlined by Jain [53] or Mannos and Sakrison [68], but is developed along a different approach.

A coding technique that includes a visual system model has been developed by Hall [42]. The method uses the visual spatial frequency model as a preprocessor before the quantization step. The zonal coder bit-assignment mapping procedure is not based on a variance map, but rather on the power spectral density in the transform domain. The power spectrum equation is derived by Hall and Hall [43] and incorporates Mannos and Sakrison's distortion function [68]. Ngan [79] uses the results of Hall and Mannos and Sakrison to implement a perceptual domain zonal coder. This coder is compared, by Ngan, to four other zonal coding techniques. Of the five coding techniques, Ngan concludes that the perceptual domain zonal coder is preferable to the other four schemes and

thus demonstrates the importance of incorporating the human visual characteristics into the coding scheme. As this comparison is one of the more recent studies, it is a natural selection of the perceptual domain coder as one of the schemes selected for further study in this thesis.

2.1.4 Hybrid Coding

As the name implies, a hybrid coder is one which utilizes a cascade of unitary transformation and DPCM.

Habibi [38] proposes two hybrid coding systems, presenting experimental results for a typical picture when using the KLT, DCT, FFT, Slant and Hadamard transformation. He also examines the visual effects of channel errors and compares system performance with two-dimensional DPCM and conventional two-dimensional transform coders. The conclusions show that, in general, hybrid performance lies somewhere between transform coding and DPCM. Jain [53] arrives at the same conclusion, although it is possible to produce hybrid source coding systems that achieve very respectable reductions in average bit-rate, for example Ø.5 bit/pixel - Wilson, Knutsson, and Granlund [1Ø5]. As efficient coding, in terms of lowest acceptable average bit-rates, is a concern of this

thesis, hybrid coding is not to be considered - but was introduced for completeness.

2.2 Effects of, and Protection Against, Channel Errors

The effects of, and protection against, channel errors is one of the major concerns of this thesis. In many data compression designs the channel effects are often ignored, by assuming noiseless channels. However, in a practical system this cannot be done i.e. channel effects must be taken into account. The usual method of catering for channel errors is to use an error protecting/correcting code, designed to reduce the probability of errors. For simplicity, equal protection is usually provided to all samples. Before deciding on which code to use, it is usual to observe channel effects on unprotected source coded data, from which the characteristics required of the channel code can be determined. Results differ for different source coding techniques.

2.2.1 Channel Errors in Predictive Coding

The effects of channel errors when using predictive source coding, in relation to image data compression, have been more extensively covered than the alternative of using transform coding. Various techniques have been used to try to optimize DPCM coder performance to cope with noisy channels e.g.choice of prediction coefficient (Essman and Wintz [29], Huang and Rao [47]), addition of a predictor output leak and a prediction function leak (Maxemchuk and Stuller [71]), use of periodic PCM updates (Arguello, Sellner and Stuller [6], Ngan and Steel [80]). However, all of these techniques, although they lead to improvement in received picture quality indicate a need for error-correcting codes in order to acheive the high picture quality required.

2.2.2 Error Correction for Channel Errors in Predictive Coding

More recent, and typical, work on error correction with DPCM has been undertaken by Lippman [65] and Modestino and Daut [75], all of who propose protection of the most significant bits in the output data stream. Results lead to designs which are reasonably

insensitive to channel errors and that can provide a performance approaching the rate-distortion bound. However, the approach becomes less, unless some degree of protection is provided to all bits of the quantizer output.

It has already been established (Section 2.1.1) that transform coding can achieve higher performance, in terms of lower average bit-rate, than the corresponding DPCM coder. It has now been found that a DPCM system would require channel protection (i.e. an increase in bit-rate) probably extending to all transmission bits and this gives another reason for turning attention to transform coding. Even if it turns out that all bits in a transform coding scheme require protection, the overall bit-rate would probably still be less than the corresponding DPCM system with channel protection. It may also turn out that, the types of distortion caused by channel errors when using transform coding may be easier to protect against than those of a DPCM system. The converse of this would still be an important result. Effects of channel errors on transform coders are less reported than in DPCM. The foregoing sections thus provide several reasons for leaving DPCM coders and considering channel errors in connection with transform coding. DPCM is thus no longer considered in this thesis.

2.2.3 Channel Errors in Transform Coding

Channel errors in transform coding can manifest themselves, visually, in different ways depending on the type of source coding scheme used. In general, for non-adaptive schemes, i.e. schemes where each sub-image is coded to the same number of bits, the channel errors are distributed over the entire sub-image [53]. For schemes that are adaptive in nature, e.g. adaptive zonal schemes or threshold schemes, the channel errors are not necessarily restricted to a particular sub-image and "slicing" or truncation of sub-images can occur. Also, in such schemes channel errors can result in reconstructed images which have hardly any resemblance to the original.

To try to compensate for channel errors, one can design the quantizer of the system to reduce the mean squared error between input and output signals under noisy channel conditions, as apposed to the noiseless case e.g. [61]. This has already been dealt with in Section 2.1.3.3

Jain's review [53] indicates a method for optimizing transform coding with regard to channel errors. Basically, it involves the formation of two bit-maps for the sub-image - one indicating bit-allocations for the sub-image when source coding, and the

other indicating bit-allocations when channel coding. As both allocations are based on the variances of the transformed coefficients, more channel protection is provided to samples which have larger variances (more important for transmission). Results are shown for images coded at 1.Ø bits/pixel and for the high error rate of Ø.Ø1, where the overhead due to channel protection is said to be only 15 percent. The improvement in signal-to-noise ratio is 1ØdB and is shown to be visually significant. The actual method employed for channel coding is not mentioned. In order that the channel coding conforms to the channel bit-map allocation, it is suspected that the method may be relatively complex. It is hoped that, simple channel coding techniques can be used against errors, which can be applied to either a zonal scheme or a threshold scheme, without having to conform to channel coding bit-maps.

2.2.4 Error Correction for Channel Errors in Transform Coding

The effects of channel errors when using transform coding have not been so extensively covered to date. The need exists to examine these effects and to study methods to reduce those more undesireable when reconstructing the image. As previously mentioned, this is one of the major concerns of this thesis.

Modestino and Daut [74] show that block transform coding is very sensitive to channel errors, as are many other encoding schemes. By providing selective error control protection to those bits which, they say, "contribute most significantly to image reconstruction" it is reported that, a design can be achieved which improves the reconstructed quality without sacrificing transmission bandwidth. The same results and conclusions are presented a year later by Modestino, Daut and Vickers [76]. Good reconstructions however are seen to be achieved at relatively low channel error rates or with protection applied to all bits. These results give an incentive to try a different coding approach, to be developed later in the thesis, which takes the idea of coding selective bits one step further, in the hope that the resulting design can then cope with relatively high channel error rates.

Kronander [60] also looks at channel errors and their influence on transform coded images, paying attention to two different schemes i.e.zonal and threshold coding. Again the idea of protecting most significant bits is proposed and, for very important code words (e.g.the End-Of-Block, EOB, code word) the (23|12) Golay code is chosen as the channel code. The code is said to be insufficient if burst errors occur and the recommendation of using a convolutional code is made - to cope with both random and burst errors. This design suggestion is taken in this thesis. Kronander does not present any numerical results.

A paper presented by Mitchell and Tabatabai [73] on the subject of channel error recovery for transform image coding is worth noting at this point. A method is presented to inspect the sub-image boundaries of a reconstructed two-dimensional transform coded image, to locate sub-images which are most likely to contain errors, to approximate the size and type of error in the sub-image, and to eliminate this estimated error from the picture. The results do show a marked improvement when the suggested algorithm is applied, though there are cases where the algorithm itself introduces other sub-images in error. Apart from this one method, there is no other (to the author's knowledge) that attempts channel error recovery in this fashion. As the algorithm is not "fool-proof" and as it extends the image reconstruction time by a factor of two, such attempts to restore sub-images in error are not considered in this thesis. It was thought to be worthwhile mentioning the idea at this stage as such post-processing methods (provided they are reliable) would be beneficial, not only from the reconstructed fidelity point of view but also, because no additional channel error protection bits, or changes to the transmitter, are required when implementing the algorithm.

2.2.5 Channel Coding/Protection

In order to provide protection of source coded image data, by means of a channel code, it becomes necessary to determine the properties required of the channel code.

Speaking in very general terms, there are two classes of codes and the methods used to implement them differ substantially. One can consider a channel-coded block of data bits to consist of k information bits and n-k parity bits (i.e. n is the total number of bits in the block). If the n-k parity bits check only the k information bits immediately preceding them in time, the code is called a block code [97] and the n-bit block is called a code word. If the n-k parity checks in a block also check information bits which appeared in preceding blocks the code is called a convolutional [27], or recurrent code.

The problem of correcting errors with a block code is a topic which has been the subject of numerous technical journal articles. The best course of action is thus to restrict attention to block codes which readily lend themselves to analysis and whose implementation is relatively simple. Golay [33] and Hamming [45] devised the first non-trivial error-correcting codes and defined several of the concepts basic to coding theory.

The binary Golay code [32],[33] is a (23|12) cyclic code (a subclass of the block codes) and is the only nontrivial binary perfect three-error-correcting code [66], [72]. Kronander [6Ø] implements this code for protection of "more significant" source coded image data with the conclusion that, provided errors do not occur in bursts and provided "protection of the synchronization suffices" the encoding schemes considered perform satisfactorily.

The most powerful random-error-correcting codes known are the Bose-Chaudhuri-Hocquenghem, BCH, codes e.g.[9], which again are a sub-class of the cyclic codes. Also, a decoding algorithm that is not too complex can be used to implement such codes [66]. The BCH codes are based on a mathematical theory which provides a systematic procedure for constructing words of length 2^p-1 in which t errors can be corrected by use of no more than pt check digits. They may thus be regarded as a generalization of the Hamming code, which achieves the same result when t=1.

Hence, if the assumption is to be made - that errors do not occur in bursts - then it would seem reasonable that, instead of employing the Golay (23|12) code (which is almost 5Ø% redundant) one could perhaps use a Hamming code [45] (e.g.

(15,11) or (7,4) code). In the case of the (15,11) code one can achieve less redundancy. It remains to be seen if there is a drop in performance when using the Hamming codes, and if so whether this drop is significant. Considering only single errors in a block, and this could be a reasonable assumption if the block considered is small enough, then, the single error correction given by Hamming in terms of parity checks is perhaps one of the simplest methods to implement. These reasons give the incentive for employing Hamming coding schemes.

The investigation of convolutional codes has proceeded along two separate paths. One path involves the use of algebraic decoding, which is fairly mechanical in nature. The first practical, algebraically decoded, convolutional codes were Hagelbarger's recurrent burst-correcting codes [40]. These codes are said to be best for small burst-lengths, b, and a relatively high efficiency (or code rate), R [66]. Since these codes are also very simple to implement they are adopted for use in this thesis even though they are suboptimal.

The other path involves the use of probabilistic decoding, one of the first such methods being investigated by Fano [30]. More recent work on probabilistic nonsequencial decoding of convolutional codes includes that presented by Viterbi [102] whose

algorithm provides a very powerful method of decoding convolutional codes. The probabilistic approach is not adopted in this thesis, as emphasis is placed on simplicity. It is to be kept in mind that results obtained when using the algebraic decoding method of Hagelbarger may indicate that, even better results could be expected when implementing such an algorithm as Viterbi's (at the cost of increased decoder complexity).

This section indicates that, if error protection is to be used to protect source-coded image data the choice is to be|-

(a) Hamming Codes - for random-error-correction

(b) Hagelbarger's convolutional coding/decoding technique - for burst-error-correction and for comparison with Hamming Codes for random-error-correction.

CHAPTER 3

SOURCE ENCODING/DECODING

3.1 Introduction - General Transformation Theory.

Image representation can be undertaken through the use of theory related to orthogonal functions and matrices. Naturally enough, source encoding of image data uses this as a mathematical basis from which to work. It is first necessary to define some of the properties of matrices and, where it is beneficial, to show how certain matrix expressions can be interpreted in a series summation form.

To give the theory an underlying purpose, it is necessary to keep in mind the reason for performing a transformation on the image data. Essentially, the reason for doing this is so that the image data in it's transformed state possesses more desireable properties than the non-transformed, i.e. original, data. The properties required of the transformed data, and reasons for wanting them, are :-

(i) decorrelation - to ensure the image data is then in a highly non - redundant form, which will give a minimum encoding rate (bits/ pixel)

(ii) concentrated signal energy (energy packing) - to achieve minimum mean square error.

The sampled image f(m,n) is a finite, two-dimensional array and can be naturally represented by a matrix of sampled values:-

$$[f] = \{f(m,n)\} = \{f_{mn}\} \qquad (3.1.1)$$

where, { } denotes the set of sampled values.

and, m,n are the coordinates in the spatial domain.

A general vector transformation is achieved by forming

$$\bar{F} = [C].\bar{f} \qquad (3.1.2)$$

where, $\bar{f}$ is the image vector, obtained by the column stacking of each matrix [f],

$\bar{F}$ is the vector of transformed data values,

and, [C] is the transformation matrix.

An important simplification results if [C] is the product of a square M x M matrix [A] with a square N x N matrix [B], as shown below :-

$$[C] = [A] \times [B]$$

$$= \begin{bmatrix} a_{11}[B] & \cdots & a_{1M}[B] \\ \vdots & & \vdots \\ a_{M1}[B] & \cdots & a_{MM}[B] \end{bmatrix} \qquad (3.1.3)$$

Equation (3.1.3) indicates that [C] is a Kronecker product matrix, and the important simplification introduced by the Kronecker product matrix is that, the general vector transformation of equation (3.1.2) may be expressed in the equivalent matrix form :-

$$[F] = [A].[f].[B] \qquad (3.1.4)$$

[A] and [B] must also be non-singular, implying

$$|A| \neq \emptyset \ , \ |B| \neq \emptyset \qquad (3.1.5)$$

The operation in equation (3.1.4) is called a separable transform since [A] operates on the columns and [B] operates on the rows of the picture matrix [f]. The transform [F] can also be written as :-

$$F(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A(u,m).f(m,n).B(n,v) \qquad (3.1.6)$$

where, $u=0,1,\ldots\ldots,M-1$, $v=0,1,\ldots\ldots,N-1$

and are the coordinates in the transform domain.

With the non-singular condition of equation (3.1.5) holding, the inverses of [A] and [B] are uniquely defined. Multiplication of both sides of equation (3.1.4), first by the inverse $[A]^{-1}$ from the left and then by the inverse $[B]^{-1}$ from the right yields

$$[f] = [A]^{-1}.[F].[B]^{-1} \qquad (3.1.7)$$

[f] is thus defined as the inverse transform of [F].

A real square matrix [T] is symmetric if

$$[T] = [T]^{t} \qquad (3.1.8)$$

where, $[T]^t$ is the transpose of [T]

[T] is also <u>orthonormal</u> if

$$[T]^t.[T] = [I] \qquad (3.1.9)$$

where, [I] is the identity matrix

and, it also follows that if equations (3.1.8) and (3.1.9) hold

$$[T]^t = [T]^{-1} = [T] \qquad (3.1.10)$$

Hence if equations (3.1.8) and (3.1.9) hold for [A] and [B], real, then equations (3.1.4) and (3.1.7) become

$$[F] = [A].[f].[B] \quad , \quad [f] = [A].[F].[B] \qquad (3.1.11)$$

A complex square matrix [C] is <u>Hermitian</u> if

$$[C]^{*t} = [C] \qquad (3.1.12)$$

where, $[C]^*$ is obtained by taking the complex conjugate of every element of [C].

[C] is unitary if

$$[C]^{*t}.[C] = [I] \qquad (3.1.13)$$

For a complex square matrix that is both Hermitian and unitary,

$$[C]^{-1} = [C] \qquad (3.1.14)$$

Provided equations (3.1.12) and (3.1.13) hold, then equation (3.1.11) is still valid for complex matrices.

The general expressions developed, in particular, expressions (3.1.6) and (3.1.11) can be used to derive the specific expressions of various transforms. As an example, and for later use, the derivation of the Discrete Cosine Transform (DCT) is considered:-

3.2 The Discrete Cosine Transform (DCT) from the General Transform Theory.

The two matrices [A] and [B] are defined to be N x N dimensionally, conforming to :-

$$[A] = [C_{mu}] \qquad (3.1.15)$$

where,

$$C_{mu} = \begin{cases} \sqrt{\frac{2}{N}} \cdot \cos\left[\frac{(2m+1)u\pi}{2N}\right] & , u = 0 \\ \frac{2}{N} \cdot \cos\left[\frac{(2m+1)u\pi}{2N}\right] & , u = 1,2,\ldots\ldots,N-1 \end{cases}$$

$$[B] = [C_{nv}] \qquad (3.1.16)$$

where,

$$C_{nv} = \begin{cases} \sqrt{\frac{2}{N}} \cos\left[\frac{(2n+1)v\pi}{2N}\right] & , v = 0 \\ \frac{2}{N} \cos\left[\frac{(2n+1)v\pi}{2N}\right] & , v = 1,2,\ldots\ldots,N-1 \end{cases}$$

Using equation (3.1.6) with (3.1.11), (3.1.15) and (3.1.16) gives the DCT , F of an array f as :-

$$[F] = [A].[f].[B]$$

or

$$F(u,v) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} A(u,m).f(m,n).B(n,v)$$

i.e.

$$F(u,v) = \frac{4}{N^2} . C(u,v) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n).\cos\left[\frac{(2m+1)u\pi}{2N}\right].\cos\left[\frac{(2n+1)v\pi}{2N}\right] \qquad (3.1.17)$$

where,

$C(u,v) = 1/2, \quad u = v = 0$

$C(u,v) = 1, \quad u,v = 1,2,\ldots\ldots,N-1$

The general expressions can also be used to find the inverse of F(u,v) as :-

$$f(m,n) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u,v).F(u,v).\cos\left[\frac{(2m+1)u\pi}{2N}\right].\cos\left[\frac{(2n+1)v\pi}{2N}\right]$$

$$m, n = 0,1,\ldots\ldots,N-1 \qquad (3.1.18)$$

3.3 The Discrete Cosine Transform (DCT) Relationship to the Karhunen-Loève Transform (KLT).

The optimum transform, in a mean square error sense, was introduced in Chapter 1 as the KLT, or eigenvector transform. It was hinted that there is a relationship between the DCT and KLT, which indicates a preference for the selection of the DCT over any other fast, unitary transform under specific conditions. Since the DCT is used throughout this thesis, this important relationship is now considered in more depth.

A certain mathematical model of the image is first assumed. The assumption is that the input image has a correlation of first-order Markov form :-

$$\text{correlation, } \rho_x(T) = \exp[\,-r|T|\,] \qquad (3.3.1)$$

where,

$\rho_x(T)$ = correlation between pixels

T = pixel separation

r = a correlation parameter

This provides a suitable starting point, provided the assumption is reasonably valid. In order to determine this, a particular image is considered. The image is shown in Figure 3.3.1.

It's spatial correlation function is determined and compared with the ideal curve obtained from equation (3.3.1). The two curves are shown in Figure 3.3.2, from which the assumption of a first-order Markov model (with correlation parameter, ρ = Ø.96 in this case) can be seen to be reasonably valid.

Figure 3.3.1 Original Image

Pixel Separation	Correlation Between Pixels
1	0·961
2	0·904
3	0·853
4	0·812
5	0·770
6	0·747
7	0·721
8	0·690
9	0·675
10	0·652
11	0·631
12	0·613
13	0·595
14	0·577
15	0·559

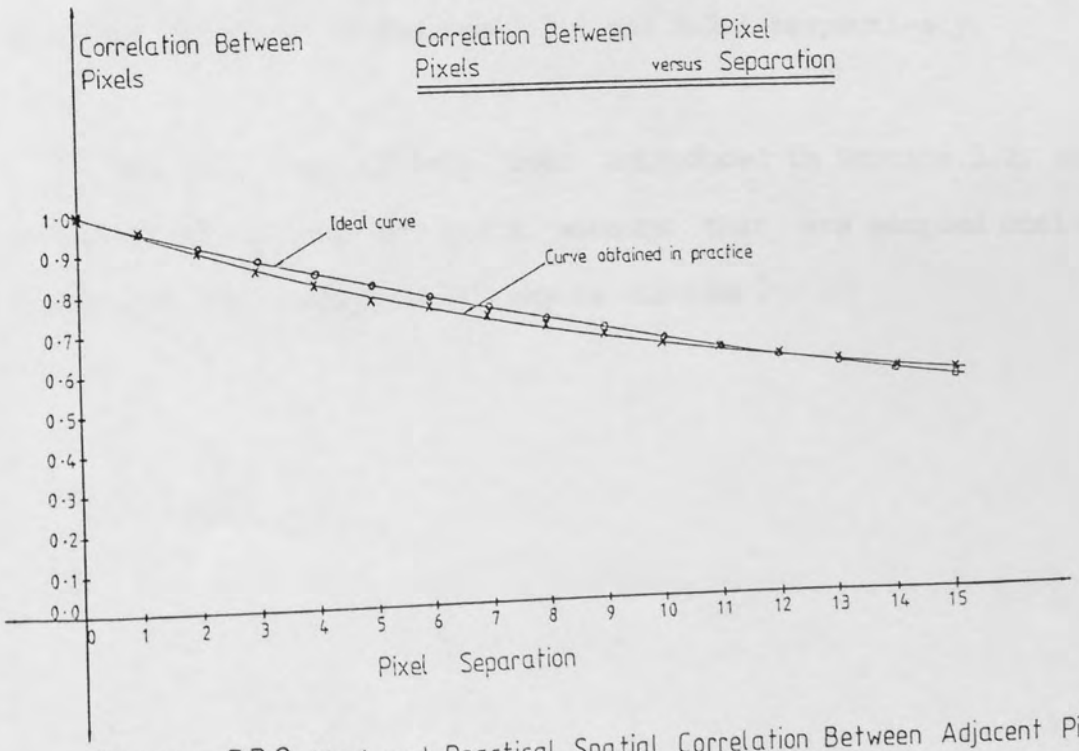


Figure 3.3.2 Ideal and Practical Spatial Correlation Between Adjacent Pixels of Original Image.

The image data correlation matrix can thus be modelled using the class of Toeplitz matrices defined as :-

$$[Z] = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 & \cdots & \cdots & \rho^{N-1} \\ \rho & 1 & \rho & \rho^2 & \rho^3 & \cdots & \rho^{N-2} \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & & & & & \vdots \\ \vdots & & & & & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \cdots & & & \cdots & 1 \end{bmatrix} \qquad (3.3.2)$$

Toeplitz matrices for $\rho = 0.9$ and $\rho = 0.96$ are shown in Figure 3.3.3

The eigenvectors of [Z] for N=8 and $\rho = 0.9$, and for N=8 and $\rho = 0.96$ are shown in Figures 3.3.4 and 3.3.5 respectively.

The DCT has already been introduced in Section 3.2, and consists of a set of basis vectors that are sampled cosine functions. The transform [C] may be written :-

1.000	0.900	0.810	0.729	0.656	0.590	0.531	0.478
0.900	1.000	0.900	0.810	0.729	0.656	0.590	0.531
0.810	0.900	1.000	0.900	0.810	0.729	0.656	0.590
0.729	0.810	0.900	1.000	0.900	0.810	0.729	0.656
0.656	0.729	0.810	0.900	1.000	0.900	0.810	0.729
0.590	0.656	0.729	0.810	0.900	1.000	0.900	0.810
0.531	0.590	0.656	0.729	0.810	0.900	1.000	0.900
0.478	0.531	0.590	0.656	0.729	0.810	0.900	1.000

(a)

1.000	0.960	0.922	0.885	0.849	0.815	0.783	0.751
0.960	1.000	0.960	0.922	0.885	0.849	0.815	0.783
0.922	0.960	1.000	0.960	0.922	0.885	0.849	0.815
0.885	0.922	0.960	1.000	0.960	0.922	0.885	0.849
0.849	0.885	0.922	0.960	1.000	0.960	0.922	0.885
0.815	0.849	0.885	0.922	0.960	1.000	0.960	0.922
0.783	0.815	0.849	0.885	0.922	0.960	1.000	0.960
0.751	0.783	0.815	0.849	0.885	0.922	0.960	1.000

(b)

Figure 3.3.3 (8×8) Toeplitz Matrices for (a) $P = 0.9$, and, (b) $P = 0.96$

E-VALUE MATRIX

0.62029E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.10083E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.32957E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.16437E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.10383E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.75719E-01	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.61350E-01	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.54734E-01

E-VECTOR MATRIX

0.32366E+00	-0.47125E+00	0.47051E+00	-0.42870E+00	-0.36592E+00	0.29009E+00	-0.20000E+00	-0.10017E+00
0.34863E+00	-0.42484E+00	0.22084E+00	0.72511E-01	0.33906E+00	-0.48725E+00	0.46342E+00	0.27727E+00
0.36562E+00	-0.29406E+00	-0.16712E+00	0.48247E+00	0.35878E+00	0.91961E-01	-0.45793E+00	-0.41513E+00
0.37423E+00	-0.10491E+00	-0.44940E+00	0.27968E+00	-0.34992E+00	0.41234E+00	0.18864E+00	0.49071E+00
0.37423E+00	0.10491E+00	-0.44940E+00	-0.27968E+00	-0.34992E+00	-0.41234E+00	0.18864E+00	-0.49071E+00
0.36562E+00	0.29406E+00	-0.16712E+00	-0.48247E+00	0.35878E+00	-0.91962E-01	-0.45793E+00	0.41513E+00
0.34863E+00	0.42484E+00	0.22084E+00	-0.72511E-01	0.33906E+00	0.48725E+00	0.46342E+00	-0.27727E+00
0.32366E+00	0.47125E+00	0.47051E+00	0.42870E+00	-0.36592E+00	-0.29009E+00	-0.20000E+00	0.10017E+00

Figure 3.3.4 Eigenvalues and Eigenvectors of [z] for N= 8 and $P = 0.9$

E-VALUE MATRIX

0.72130E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.47329E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.13495E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.64343E-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.39590E-01	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.29948E-01	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.24125E-01	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.21462E-01

E-VECTOR MATRIX

0.34132E+00	-0.48284E+00	0.46654E+00	-0.42677E+00	-0.35339E+00	0.27446E+00	-0.20247E+00	-0.97182E-01
0.35170E+00	-0.41909E+00	0.20312E+00	0.97238E-01	0.33902E+00	-0.48789E+00	0.46920E+00	0.27736E+00
0.35867E+00	-0.28476E+00	-0.18297E+00	0.48404E+00	0.36437E+00	0.10406E+00	-0.45293E+00	-0.41692E+00
0.36220E+00	-0.10089E+00	-0.45569E+00	0.27232E+00	-0.35700E+00	0.41936E+00	0.18372E+00	0.48974E+00
0.36220E+00	0.10089E+00	-0.45569E+00	-0.27232E+00	-0.35700E+00	-0.41936E+00	0.18371E+00	-0.48974E+00
0.35867E+00	0.28476E+00	-0.18297E+00	-0.48404E+00	0.36437E+00	-0.10406E+00	-0.45293E+00	0.41693E+00
0.35170E+00	0.41909E+00	0.20312E+00	-0.97238E-01	0.33902E+00	0.48787E+00	0.46920E+00	-0.27736E+00
0.34132E+00	0.48284E+00	0.46654E+00	0.42677E+00	-0.35339E+00	-0.27446E+00	-0.20247E+00	0.97185E-01

Figure 3.3.5 Eigenvalues and Eigenvectors of [z] for N= 8 and $P = 0.96$

$$[C] = N^{-1/2}.[C_{mk}] \qquad (3.3.3)$$

where,

$$C_{mk} = \begin{cases} 1, & k = \emptyset \\ \sqrt{2}.\cos\left[\dfrac{(2m+1)k\pi}{2N}\right], & m = \emptyset,1,\ldots\ldots,N-1 \text{ and } k = 1,2,\ldots\ldots,N-1 \end{cases}$$

The basis vectors are given by

$$\left\{ N^{-1/2}, N^{-1/2}\sqrt{2}.\cos\left[\frac{(2m+1)k\pi}{2N}\right], k=1,2,\ldots\ldots N-1,\ m=\emptyset,1\ldots\ldots,N-1 \right\} \qquad (3.3.4)$$

and, for comparison, for N = 8 are given by

$$\left\{ 8^{-1/2}, \frac{1}{2}.\cos\left[\frac{(2m+1)k\pi}{16}\right],\ k = 1,2,\ldots\ldots,7,\ m = \emptyset,1,\ldots\ldots,7 \right\}$$

With this value of N the [C] matrix is calculated as shown in Figure 3.3.6

Ø.354	Ø.49Ø	Ø.462	Ø.416	Ø.354	Ø.278	Ø.191	Ø.Ø98
Ø.354	Ø.416	Ø.191	-Ø.Ø98	-Ø.354	-Ø.49Ø	-Ø.462	-Ø.278
Ø.354	Ø.278	-Ø.191	-Ø.49Ø	-Ø.354	Ø.Ø98	Ø.462	Ø.416
Ø.354	Ø.Ø98	-Ø.462	-Ø.278	Ø.354	Ø.416	-Ø.191	-Ø.49Ø
Ø.354	-Ø.Ø98	-Ø.462	Ø.278	Ø.354	-Ø.416	-Ø.191	Ø.49Ø
Ø.354	-Ø.278	-Ø.191	Ø.49Ø	-Ø.354	-Ø.Ø98	Ø.462	-Ø.416
Ø.354	-Ø.416	Ø.191	Ø.Ø98	-Ø.354	Ø.49Ø	-Ø.462	Ø.278
Ø.354	-Ø.49Ø	Ø.462	-Ø.416	Ø.354	-Ø.278	Ø.191	-Ø.Ø98

Figure 3.3.6 Discrete Cosine Transform Matrix for N = 8

The basis vectors of the [C] matrix are plotted against the eigenvectors of [Z] for N = 8. The results are shown in Figures 3.3.7 and 3.3.8 for ρ = Ø.9 and Ø.96 respectively. The close resemblance, apart from 18Ø degrees phase shift, between the eigenvectors and the basis vectors as defined by equation (3.3.3) is apparent.

This then, is the relationship that favours the DCT as the fast unitary transform which approaches the optimality of the KLT for specific values of ρ.

Falling in line with this is the observation [93] that the DCT also comes closest to the KLT in it's energy compaction properties.

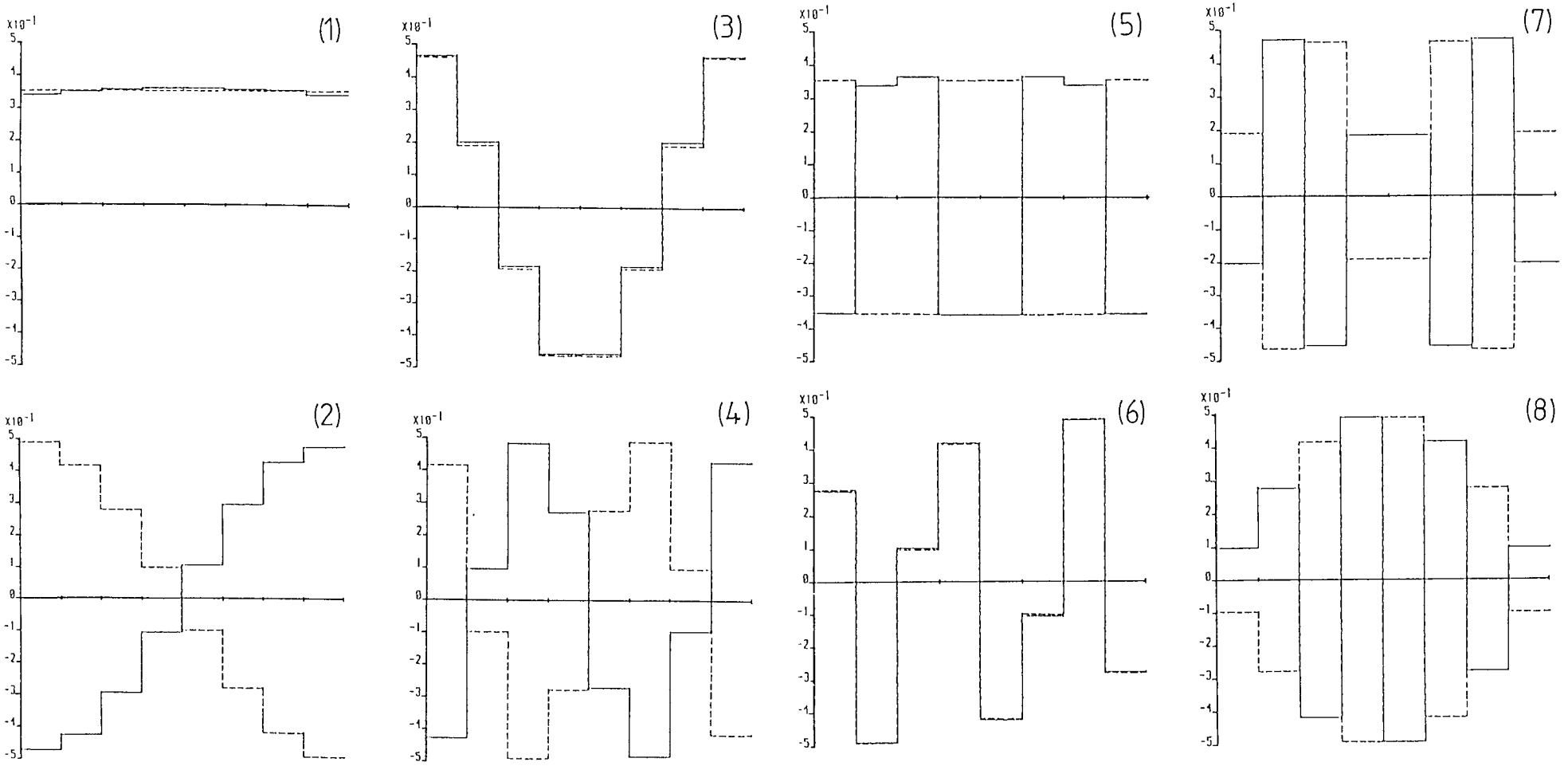


Figure 3.3.7 Comparison of Eigenvectors of (8×8) Toeplitz Matrix ($P = 0{\cdot}9$) with the D.C.T. Basis Vectors.

-----D.C.T.
——— Eigenvectors

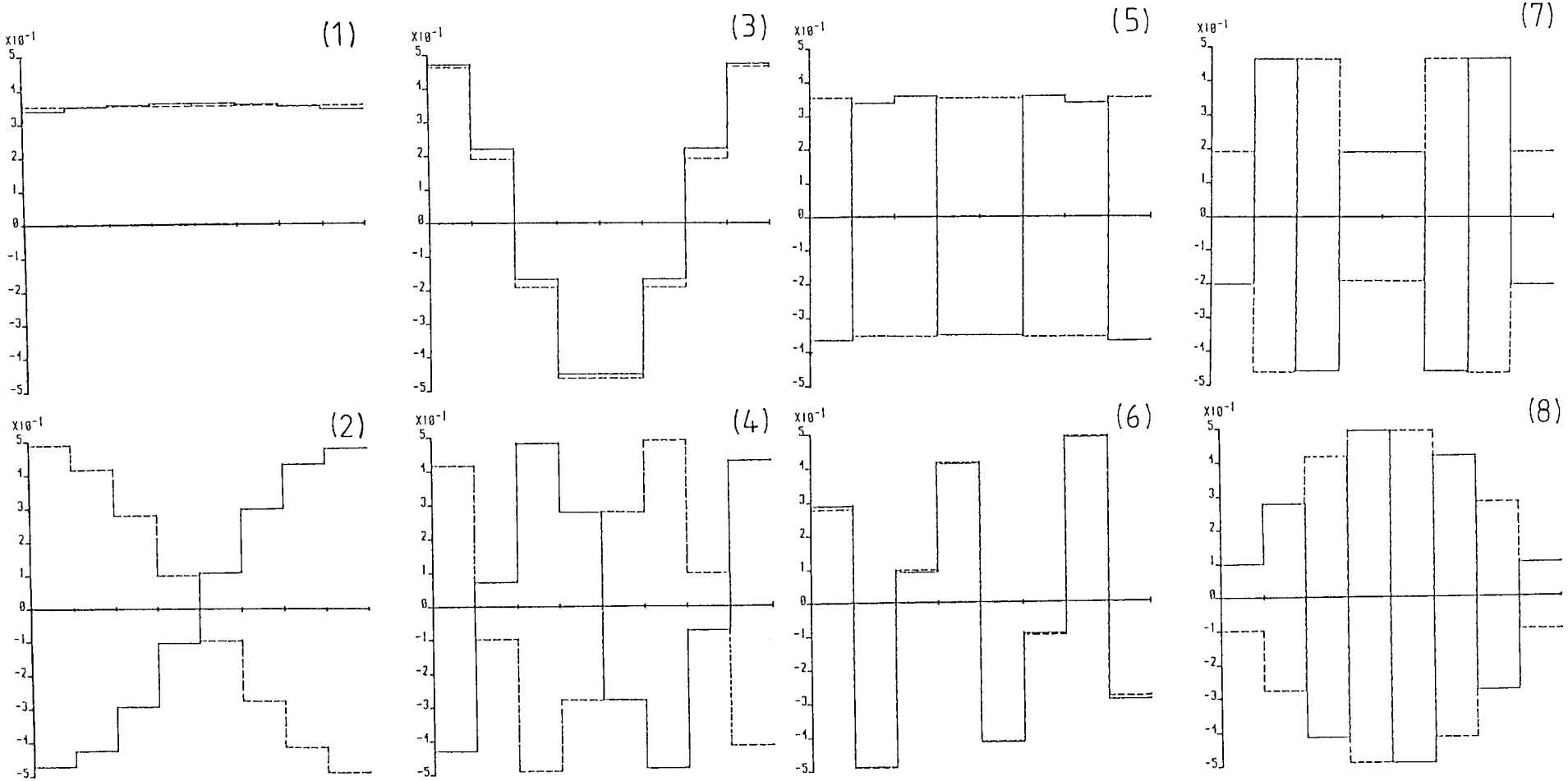


Figure 3.3.8 Comparison of Eigenvectors of (8×8) Toeplitz Matrix ($P = 0{\cdot}96$) with the D.C.T. Basis Vectors.

----- D.C.T.

——— Eigenvectors

3.4 Employment of the DCT when Source Encoding.

The classical communication system in Figure 1.1.1 indicates the stage under observation when talking of source encoding. It is intended to develope two state-of-the-art, source encoding/decoding schemes. The initial stages of both schemes are identical in that they employ the same transformation technique. However, they then differ fairly substantially.

To clarify where the two schemes differ, and where they do not, the source encoder shown as a component in Figure 1.1.1 can be subdivided and redrawn as shown in Figure 3.4.1 :-

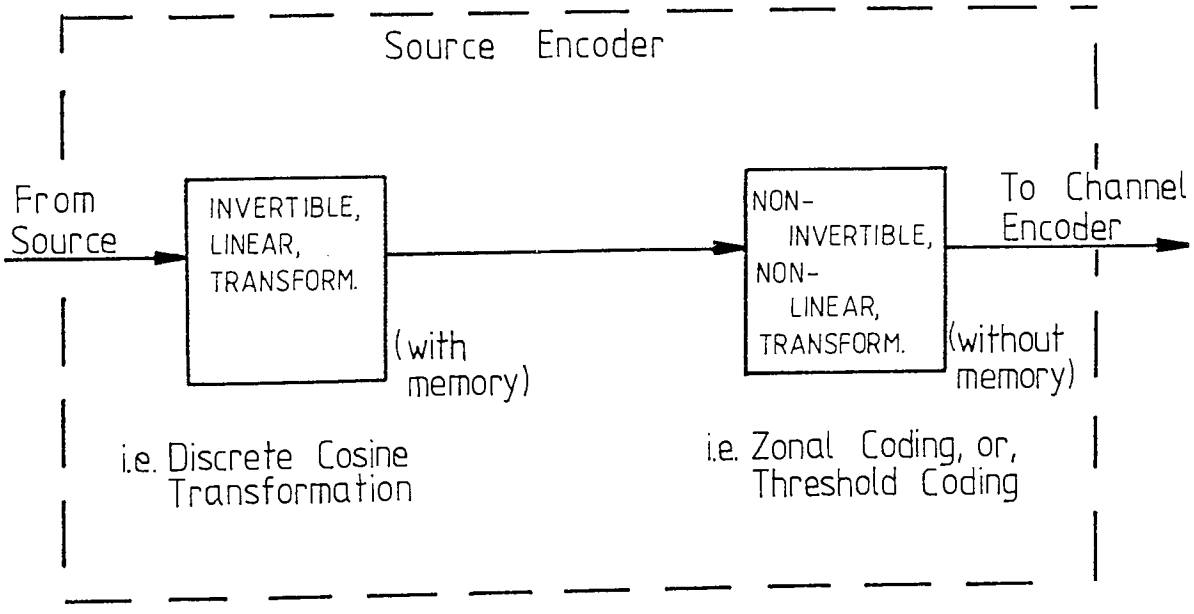


Figure 3.4.1 Block Diagram of Source Encoding Schemes.

Since the two-dimensional DCT's, as given by equations (3.1.17) and (3.1.18), are separable, they can be computed by successive applications of a one-dimensional DCT.

The forward, one-dimensional, DCT is given by :-

$$F(u) = \frac{2c(u)}{N} \sum_{m=0}^{N-1} f(m).\cos\left[\frac{(2m+1)u\pi}{2N}\right] \quad , \; u = 0,1,\ldots\ldots,N-1$$

(3.4.1)

where,

$$c(u) = \frac{1}{\sqrt{2}} \quad , \text{ for } u = 0$$

$$= 1 \quad , \text{ for } u = 1,2,\ldots\ldots N-1$$

(3.4.2)

The inverse transform is given by

$$f(m) = \sum_{u=0}^{N-1} c(u).F(u).\cos\left[\frac{(2m+1)u\pi}{2N}\right] \quad , \; m = 0,1,\ldots\ldots,N-1$$

(3.4.3)

These equations were implicitly indicated in equations (3.1.15) and (3.1.16). This together with the fact that the DCT can be obtained from manipulation of the real part of the Fast Fourier Transform (FFT), is the method adopted in this thesis.

It is not the intention to examine any more of the properties of the DCT, but rather, having shown the reasons for it's choice, to give the method of implementation and to indicate what it achieves.

The original data is divided into sub-images (of size 4x4, 8x8, 16x16, or 32x32 pixels square) and each sub-image is then treated as an image in it's own right. The application of the forward, two-dimensional DCT (through use of successive applications of the forward, one-dimensional DCT) results in the transformation of data from the spatial domain to the transform domain. In the transform domain, due to the properties of the DCT already given e.g. unitary, orthogonality etc., the sub-image data is now regarded as decorrelated, concentrated in signal energy and in addition is regarded as Gaussian in nature.

In order to make the thesis more readable, the algorithms developed to implement the two-dimensional DCT are put in Appendix 1.

Having dealt with the transformation technique common to both source encoders, the next two sections (Section 3.5 and Section 3.6) deal with the processes which make the two schemes substantially different.

3.5 Zonal Coding

Zonal coding involves the assignment of a certain number of bits to a particular coefficient in a transformed sub-image (resulting in the formation of a zonal mask). A typical "bit-map" can be seen in Figure 3.5.1 which shows the bit allocation for a transformed sub-image 16x16 pixels square. When used for all sub-images, this bit-map indicates that an average of 1.1 bits/pixel can be achieved.

9 6 6 5 5 4 3 3 2 1 1 0 0 0 0 0
6 6 6 5 5 4 3 3 2 1 1 0 0 0 0 0
6 6 6 5 5 4 3 3 2 1 1 0 0 0 0 0
5 5 5 5 4 4 3 2 2 1 0 0 0 0 0 0
5 5 5 4 4 3 3 2 1 1 0 0 0 0 0 0
4 4 4 4 3 3 2 2 1 0 0 0 0 0 0 0
3 3 3 3 3 2 2 1 1 0 0 0 0 0 0 0
3 3 3 2 2 2 1 1 0 0 0 0 0 0 0 0
2 2 2 2 1 1 1 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Figure 3.5.1 Typical "bit-map" for Zonal Quantization

In order to obtain Figure 3.5.1 and other similar, bit-maps (for different compression rates) it is necessary to identify :-

(a) The source-user pair used to model the transformed data - the source-user pair is completely described once both the statistical mechanism generating the source outputs and the distortion measure that presents the fidelity of reproduction to the user are quantitatively specified.

(b) The rate-distortion function applicable to the source-user pair.

(c) The method of estimation of the variances of each transformed coefficient.

The next sections deal with the above three aspects (a), (b), (c), culminating with the over-all summary of the zonal source encoding/decoding scheme.

3.5.1 A Rate-Distortion Function

Source encoding by means of a data compression algorithm involves the ordering of source data, in accordance with it's relative importance at the receiver, and then condensing or deleting the less significant information before actual transmission. This results in a certain amount of distortion on reconstruction and the idea then is to keep this distortion at an acceptable level.

Bit-allocation, when zonal coding, is an attempt to realize the relative importance of transformed coefficients (i.e. more bits are assigned the more important a coefficient is). Hence the zonal coding algorithm is based on a theory that tries to resolve the problem of what information should be transmitted. To date there is only one mathematical theory (to the author's

knowledge) that provides a fairly rigorous basis for this type of problem and it is that known as Rate-Distortion theory. This theory is based on Shannon's [96] original concept of the rate-distortion function of an information source, and uses as it's foundations the ideas belonging to information theory. The following definition summarizes the use of a rate-distortion function¦-

The rate-distortion function, R(D), of a source-user pair is the effective rate at which the source produces information subject to the constraint that the user can tolerate an average distortion of D.

This definition implies that, should the source produce information at any rate less than R(D), then the resulting distortion would be unacceptable. For image compression the rate R(D) would be in bits/pixel. If the rate-distortion function of the source-user pair (of the image coder) is found, the number of bits to assign to a particular coefficient can be deduced.

By minimizing the mutual information between the source and user the rate-distortion function for a memoryless, Gaussian source of arbitrary mean and variance σ^2, with respect to the mean square error criterion is shown by Berger [8] to be ¦-

$$R(D) = \begin{cases} \frac{1}{2}.\log_2(\sigma^2/D) & , \emptyset \leq D \leq \sigma^2 \\ \\ \emptyset & , D > \sigma^2 \end{cases} \qquad (3.5.1.1)$$

where,

R(D) is the rate-distortion function

σ^2 is the variance of the transformed coefficient

D is the maximum average distortion permissible.

This simple, but important, result can be incorporated into the zonal scheme provided the source data is of the correct form and provided the mean square error criterion is applicable. Subsequent sections show how various manipulations enforce the conditions required.

3.5.2 The Human Visual System (HVS) Spatial Frequency Model

It is not the intention of this thesis to prove any of the formulae used to model the characteristics of the Human Visual System (HVS). Proofs can be found in the papers mentioned in Section 2.1.3.4 and which are listed in the "References" section.

The formulae are simply accepted, whereas their application is studied in more detail, this being considered more important as far as the encoder is concerned.

Ideally, when judging reconstructed images, it would be very helpful to have a measure of distortion which, subjectively, corresponds to human evaluation. That is to say, the distortion measure is to classify a set of reconstructed images in the same order that an independent human observer would. At the same time, the distortion measure must not be so complex that it becomes non-analytical. There thus appears to be two methods of approach to the problem :- First, one could do many psychophysical and physiological experiments to try to come up with an acceptable distortion measure or, secondly, one could take a known, analytical, class of distortion measures and, by undertaking subjective human tests, deduce which of these measures provide the best answer to the problem. By undertaking the second procedure a weighting function can be derived [68] :-

$$H(w_r) = 2.6(Ø.Ø192 + Ø.114w_r).\exp[-(Ø.114w_r)^{1.1}]$$

(3.5.2.1)

where,

$$w_r = w_s \sqrt{i^2 + j^2}$$

(i,j) are the coordinates of the transformed coefficient

w_s is the scale factor to account for conversion from spatial frequency to radians per degree (rad./deg.)

The weighting factor has the relevance that, provided the transform domain coefficients are multiplied by it, then the mean square error criterion can be adopted as the distortion (or fidelity) measure. The use of the weighting function also ties in well with the previous section where the mean square error criterion is required for use with the stated rate-distortion function.

As indicated in the previous section (3.5.1), the bit-allocation procedure often involves determining the variances of the transformed coefficients. An estimate of these variances can be made by calculating their spectral density (the formula of which comes from accounting for the eye's response to different spatial requencies). The power spectrum equation [42] is :-

$$S(w_r) = \frac{2a\sigma^2}{a^2 + w_r^2} \mid H(w_r) \mid^2 \qquad (3.5.2.2)$$

where,

σ = variance of the image

a = correlation parameter

$H(w_r)$ is as given in equation (3.5.2.1)

By putting

$$\sigma^2_{i,j} = S(w_r) \qquad (3.5.2.3)$$

where, $\sigma^2_{i,j}$ is the variance of the (i,j)th element, then using equation (3.5.2.2) with the rate-distortion function given in equation (3.5.1.1) gives the number of bits to assign to the (i,j)th coefficient (since the rate-distortion function is in bits/pixel). This procedure is not carried out when considering the d.c. component in the top left hand corner of the transformed sub-image. In order to cope with the d.c. component, equation (3.5.2.2) would need modifying. However, it suffices to simply allocate a fixed number of bits (typically 9) to the d.c. component.

It is helpful at this point to show how the scale factor w_s, and the correlation parameter, a, are determined. This provides a more pictorial insight into observation of spatial frequencies.

3.5.2.1 Spatial Frequency to rad./deg.

The image used throughout most of this thesis consists of a square of pixels of dimension 256x256 (i.e. 119mmx119mm). For the following analysis, it is convenient to think of this as a square made up of 256 horizontal lines, 119mm high, as shown in Figure 3.5.2.1.1

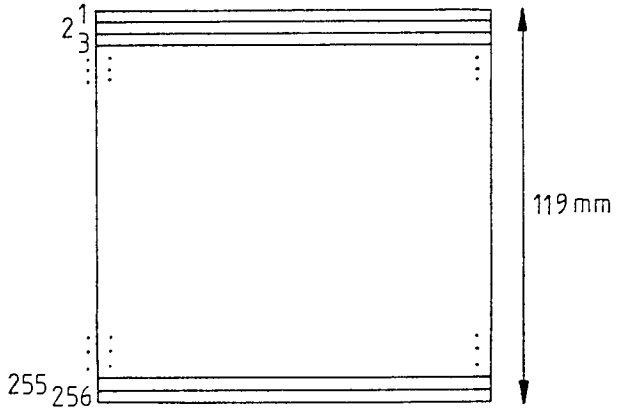


Figure 3.5.2.1.1 Image Representation

The maximum spatial frequency is thus given by considering two adjacent lines whose separation, d, is :-

$$d = \frac{119}{256} \times 2 = 0.9296\text{mm}$$

The normal viewing distance is taken to be four times the image height, thus this maximum spatial frequency can be considered viewed as in Figure 3.5.2.1.2 :-

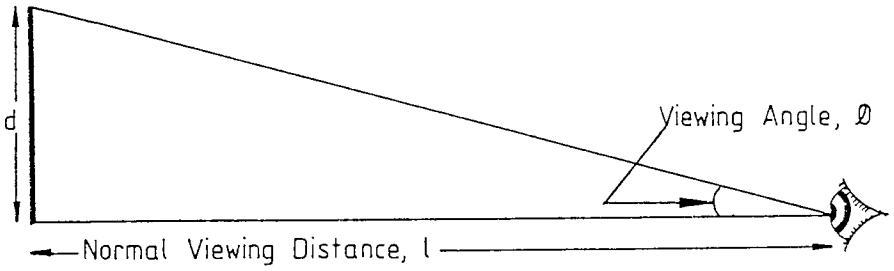


Figure 3.5.2.1.2 Viewing Distance and Viewing Angle

The normal viewing distance, l = 119x4 = 476mm

Thus the viewing angle for one cycle is

$$Ø = \tan^{-1}(d/l) \text{ degrees} \qquad (3.5.2.1.1)$$

$1/Ø$ then is the number of cycles per degree i.e.

$$1/Ø = (\tan^{-1}(d/l))^{-1} \text{ cycles/degree} \qquad (3.5.2.1.2)$$

Taking 1 cycle = 2π radians

$$1/\emptyset = 2\pi.(\tan^{-1}(d/l))^{-1} \text{ rad./deg.} \qquad (3.5.2.1.3)$$

The value $1/\emptyset$ is the maximum that can occur for a given sub-image. w_s has to account for possible modifications required if sub-images of different sizes are used i.e. the above expression for $1/\emptyset$ becomes equal to w_s (the scale factor) if normalization to the sub-image size is carried out. Hence :-

$$w_s = \frac{1}{\emptyset.B} \qquad (3.5.2.1.4)$$

where,

B = the number of vertical (or horizontal) lines in a square sub-image

For example, using the previous values and a sub-image size of 32x32 pixels square, the scale factor becomes :-

$$w_s = \frac{2\pi}{32}.(\tan^{-1}(\emptyset.9296/476))^{-1}$$

$$= \underline{1.755 \text{ rad./deg.}}$$

To recall (equation (3.3.1)) if the data model assumed is the first-order Markov one, then the correlation constant given by

$$\rho_x(T) = \exp[\, -r\, T \,]$$

is converted to the parameter, a, by using

$$a = \frac{2\pi.\rho_x(T)}{\emptyset\ .\ B} \qquad (\ 3.5.2.1.5\ )$$

This is the correlation parameter of equation (3.5.2.2). If the values used previously are taken as an example, and $\rho_x(T)$ is $\emptyset.96$ (from Figure 3.3.2), the correlation parameter is found to be :-

$$a = 2\pi.(\ \tan^{-1}(\ o.9296/476\ )\ )^{-1}.\frac{\emptyset.96}{32}$$

$$= \underline{1.685}$$

3.5.3 Summary of the Zonal Encoding/Decoding Scheme.

The procedure involved when undertaking zonal source encoding is best seen from the block diagram in Figure 3.5.3.1. Once the sub-image size has been set, the estimated variances of the transformed coefficients are found using the power spectrum equation (equation (3.5.2.2)). These variances, together with the rate-distortion function (equation (3.5.1.1)), are used to obtain the zonal bit-map. Once this has been done one can source code in accordance with the flow of the block diagram.

Thus the appropriate sub-image is selected and the data is translated to the centroid by subtraction of the mean of the image from each data value. A two dimensional cosine transform is then applied to the sub-image, decorrelating the data. The transformed coefficients are normalized to their estimated variances before being quantized in accordance with the pre-determined zonal bit-map. The actual bit assignment can be done using the rate-distortion function since the various manipulations result in the ability to classify the data as memoryless and Gaussian in nature, and to which the mean square error criterion apply. After quantization, the source encoding of the sub-image is complete and

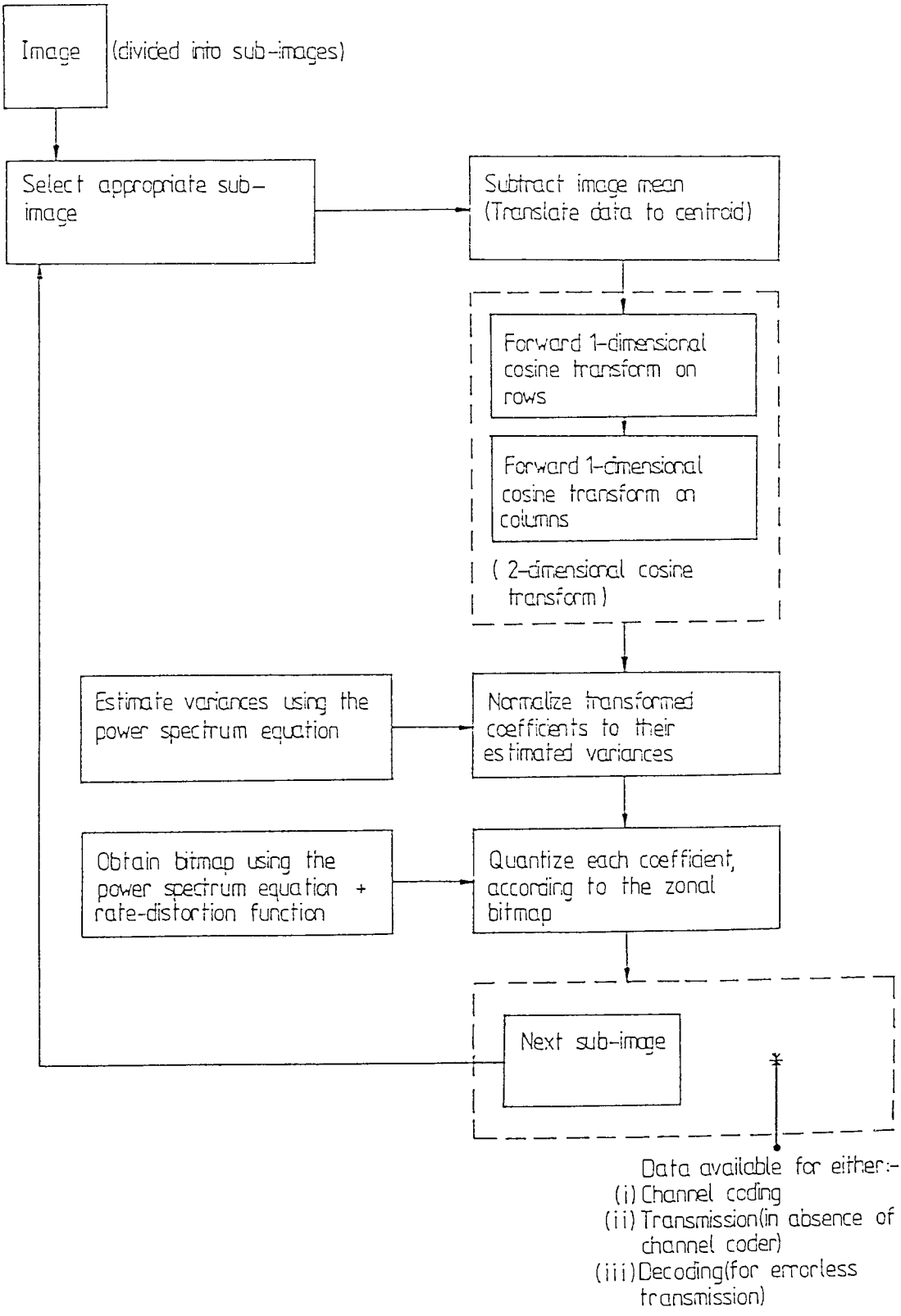


Figure 3.5.3.1 Zonal Source Encoder, Block Diagram

another sub-image is selected to undergo the same procedure. Once all sub-images have been processed the source encoding of the image itself is complete.

The zonal source coder block diagram also indicates three possibilities of what is to be done with the data after source encoding. Option (iii), decoding for errorless transmission, simply involves inverse operations to those described previously. These "inverse operations" are shown in Figure 3.5.3.2 and are seen to form the zonal source decoder. The encoder and decoder of Figure 3.5.3.1 and Figure 3.5.3.2 respectively form the zonal source data compression scheme to be analysed in the presence of a channel.

The advantages and disadvantages of the zonal scheme, without channel coding, are to be deduced from preliminary tests (Chapter 4).

The scheme is seen to be relatively simple and is kept non-adaptive (the same bit-map is used for all sub-images). This is intentional as it is to be later compared with the threshold scheme (which is adaptive in nature) and the merits, or otherwise, of each scheme are to be examined when channel errors are inherent in the overall communication system.

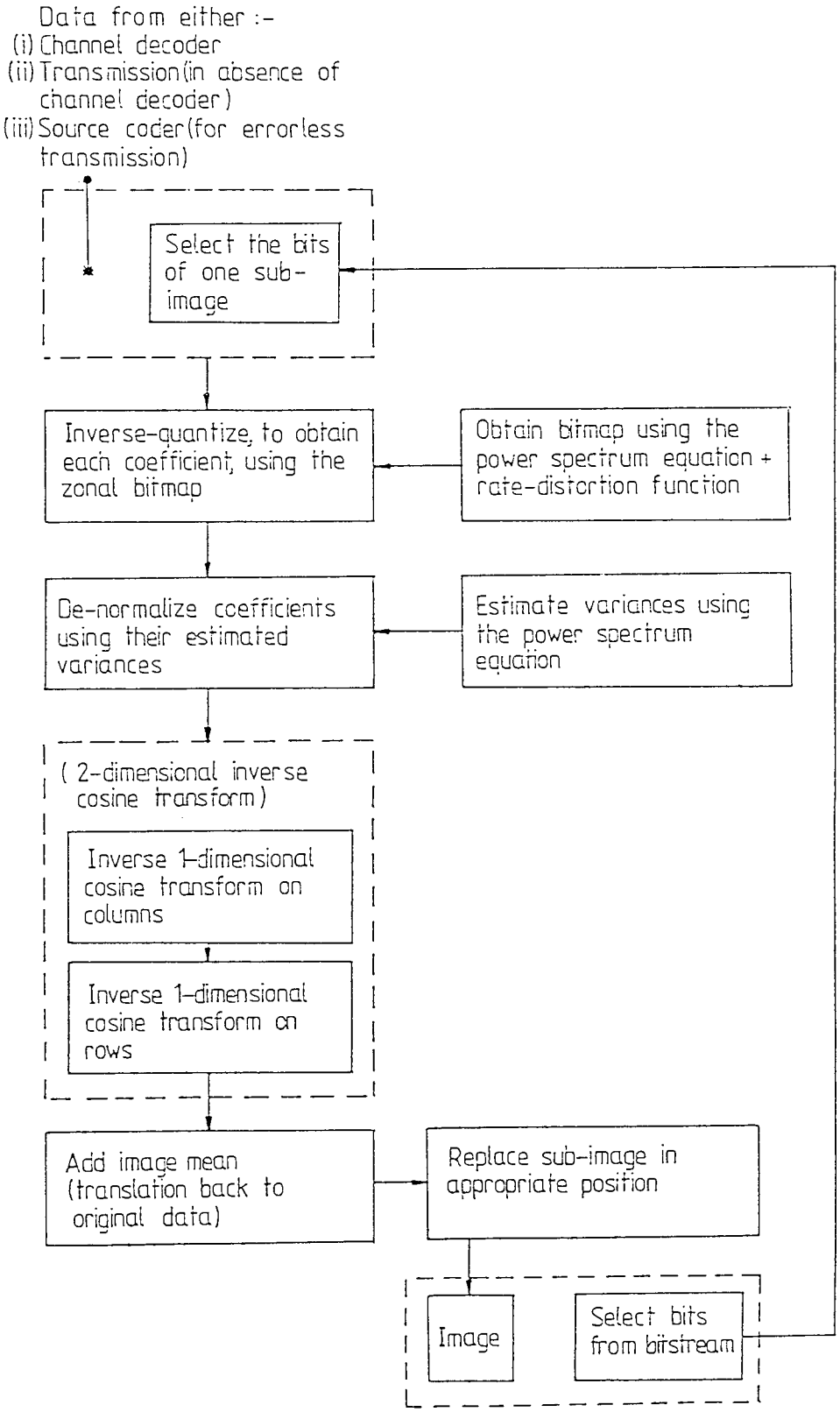


Figure 3.5.3.2 Zonal Source Decoder, Block Diagram

3.6 Threshold Coding.

Although threshold coding is based primarily on intuitive considerations, it results in compression rates which are as good as any achieved to date. The ideas behind threshold coding are simple. The main aim is to exploit statistical redundancy in the data stream. In order to do this a "pre-run" through the image data is necessary to accumulate information about the source. Examples of what information about the source is required, assuming the method of scanning the data is fixed, are given below :-

(a) Maximum and minimum possible amplitudes of transformed coefficients, for both the d.c. and all a.c. coefficients. These values are required in order to set the normalization constants for the data. The "fixed" quantizer has values which lie between -1 and +1, hence the need for normalization.

(b) Probabilities of occurrence of quantized, transformed coefficient amplitudes. These are required since the methods of coding the amplitudes involves statistical coding.

(c) Probabilities of occurrence of strings of zeros, of various lengths, here-after referred to as zero-run-lengths,

abbreviated as z.r.l.. These are required for a similar reason to that given in (b).

As indicated in (a), the method of quantizing involves a "fixed" quantizer i.e. "fixed" in the sense that each transformed coefficient is quantized to the same number of bits, or levels (typically seven or eight bit quantizing).

The pre-run will only extract information for the particular image it is carried out on. Should the coder be required to handle more than one sub-image the pre-run could be made to cover all images and the resulting information treated as though coming from one source. Alternatively, coding tables for a particular class of images could be deduced and the coder modified to identify which class is in the process of being encoded. Then the coder could select the correct table for that particular class. This would be useful in situations where classes of images differ substantially.

The next sections deal with the mechanics of individual parts of the threshold coding scheme. A summary is than provided to give an overall view of the complete scheme.

3.6.1 Thresholding

With the threshold scheme, a method of scanning coefficients of each sub-image is established. The accepted scanning method is from the top left-hand corner of the sub-image, moving in a zig-zag pattern down towards the bottom right-hand corner, as shown in Figure 3.6.1.1 :-

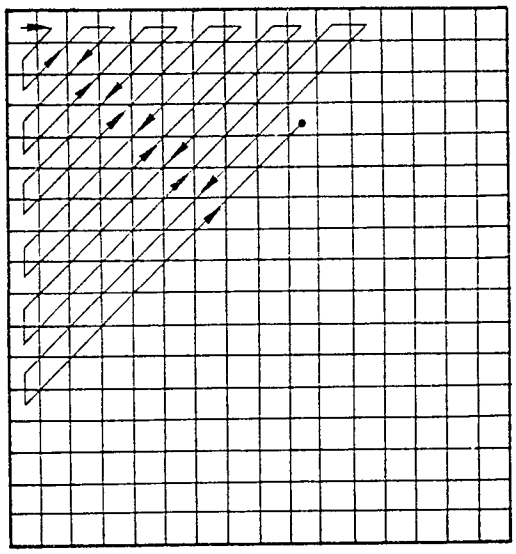


Figure 3.6.1.1 Method of Scanning Transformed Coefficients

When no more coefficients are encountered along the scanning path (i.e. only zeros exist to the bottom right-hand corner) then a special code word is used to indicate the end of the sub-image data - hereafter referred to as the end-of-sub-image code word, abbreviated as the E.O.S. codeword.

The scanning procedure effectively turns the data into a one-dimensional form, or rather a string of coefficients of varying amplitude. This can be represented as shown in Figure 3.6.1.2.

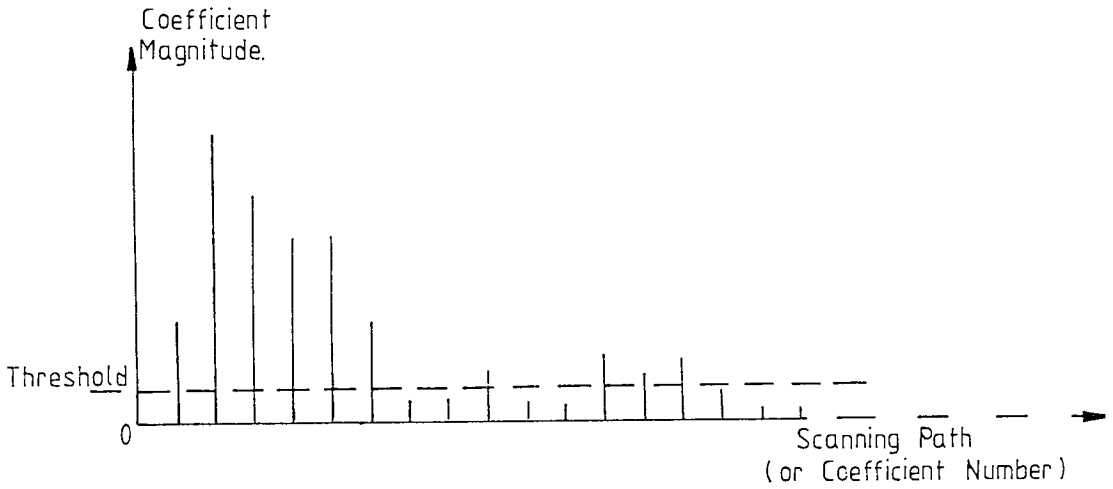


Figure 3.6.1.2 Representation of Data Stream after Scanning

Figure 3.6.1.2 also indicates an arbitrary setting of the threshold. Any transformed coefficient magnitude that occurs below the threshold results in that particular coefficient being replaced by a zero value. Any transformed coefficient magnitude that lies above the threshold results in that coefficient having the threshold subtracted from it, and this new value replaces the old. The result is a string of data with many zero values. It is also the case that many zero values occur successively, giving rise to the zero-run-lengths (z.r.l.) mentioned earlier.

After normalization (to ensure the data lies between -1 and +1, required for quantization) and quantization the data is ready for coding. The normalization and quantization processes also result in more values taking on the zero value (which, in turn, creates or lengthens the z.r.l.). The method of coding adopted is to take account of the quantized coefficients and to also code their addresses (i.e. position in the bit stream). This is a form of run-length coding.

The statistical properties of the quantized coefficients and the z.r.l. are accounted for by use of the Huffman coding algorithm [-

3.6.2 Huffman, Amplitude and z.r.l., Coding

The d.c. coefficients (in the upper left-hand corner of each sub-image) are treated separately from the a.c.coefficients and, when coding, are typically truncated to 9 bits. The amplitudes of all remaining non-zero coefficients are coded by use of an amplitude look-up table. This table consists simply of Huffman code-words formed in the "pre-run" mentioned in Section 3.6. The table is thus derived from the probability of occurrence of typical transform coefficients. The z.r.l.'s are coded in a

similar manner, by the use of a z.r.l. look-up table. Typical Huffman code tables can be seen in Figure 3.6.2.1 .

AMPLITUDE CODE WORDS.	Z.R.L. CODE WORDS
01101	00
10000	010
10011	111
10101	1001
11010	01110
11101	10100
11111	10101
010011	10110
010100	011010
Z.R.L. CALL = 00	
E.O.S. = 010101	

Figure 3.6.2.1 Typical Huffman Code Tables for Non-zero Amplitudes and z.r.l.'s

The amplitude codes also include two relatively important code words, these being :-

(i) The end-of-sub-image (E.O.S.) code word, and,

(ii) The z.r.l. prefix code word.

As already mentioned, the E.O.S. code word indicates when the last significant coefficient of a sub-image nas been coded,and thus terminates coding of this particular sub-image. The z.r.l. prefix code word is required to differentiate between the amplitude code words and the z.r.l. code words.

As the construction of the Huffman codes leads to bit-reduction and adaptivity of the threshold coder it is worthwhile at this point to see how these optimal, uniquely decodeable codes are formed. This also includes the setting down of a rigorous mathematical structure of formation of the codes, applicable, not only to binary coding but also, to coding using any radix. The procedure to be outlined is the one adopted in this thesis :-

Let, r - be the number of terms, or elements, to be coded

s - be the base/radix of the code words to be generated

$\overline{p}$ - be the probability vector of all elements

Then, a uniquely decodeable (UD) s-radix code for vector $\overline{p}$ whose average length is $n_s(\overline{p})$ is referred to as an optimal code for $\overline{p}$.

It is easier to explain the Huffman coding algorithm through the use of a numerical example, rather than to rely totally on algebra. To this end

Let, radix s=4 (i.e. the only possible code characters will thus be Ø,1,2, or 3)

Also, let the probability of all elements be the vector

$$\bar{p} = (\text{Ø.24}, \text{Ø.21}, \text{Ø.17}, \text{Ø.13}, \text{Ø.1Ø}, \text{Ø.Ø7}, \text{Ø.Ø4}, \text{Ø.Ø3}, \text{Ø.Ø1})$$

The first step is to replace the vector $\bar{p}$ with a simpler one $\bar{p}'$, which is obtained by combining the <u>three</u> smallest probabilities in $\bar{p}$:-

$$\bar{p}' = (\text{Ø.24}, \text{Ø.21}, \text{Ø.17}, \text{Ø.13}, \text{Ø.1Ø}, \text{Ø.Ø8}, \text{Ø.Ø7})$$

note:the elements are rearranged to stay in decreasing order

$\bar{p}'$ is further reduced by combining the <u>four</u> smallest probabilities in $\bar{p}'$ to give

$$\bar{p}'' = (\text{Ø.38}, \text{Ø.24}, \text{Ø.21}, \text{Ø.17})$$

again, rearranging elements to stay in decreasing order.

The reason why three element probabilities were combined when going from $\bar{p}$ to $\bar{p}'$ and four when going from $\bar{p}'$ to $\bar{p}''$ is due to the following conditions that must be satisfied for encoding :-

Let, s' - denote the number of probabilities that must be combined in the reduction of $\bar{p}$ to $\bar{p}'$.

Then, it turns out that s' is uniquely determined by

$$s' \in \{ 2,3,\ldots \ldots,s\} \qquad (3.6.2.1)$$

$$s' \equiv r \ (\text{mod } s-1) \qquad (3.6.2.2)$$

So that in the numerical example considered ;

Reduction from $\bar{p}$ to $\bar{p}'$;

$s = 4, \ r = 9$

$\therefore \ s' \equiv 9 \ (\text{mod } 3), \ 9/3 = 3$ remainder $\emptyset$

But from equation (3.6.2.1)

$s \in \{ 2,3,\ldots \ldots,s \}$ and $\emptyset \notin \{ 2,3,\ldots \ldots,s \}$

$\therefore \ s' \equiv 9 \ (\text{mod } 3) \ , \ 9/3 = 2$ remainder 3

and, $3 \in \{ 2,3,\ldots \ldots,s\}$

i.e. the three smallest probabilities are required when going from $\bar{p}$ to $\bar{p}'$.

Reduction from $\bar{p}'$ to $\bar{p}''$

$$s = 4 \ , \ r = 7$$

$$\therefore \ s' \equiv 7 \bmod 3 \ , \ 7/3 = 1 \text{ remainder } 4$$

and,

$$4 \in \{ 2,3,\ldots \ldots,s\}$$

i.e. the four smallest probabilities are required going from $\bar{p}'$ to $\bar{p}''$.

It can be seen that, after reduction the smallest number of terms is given by

$$r' = r - s' + 1$$

and the number of probabilities to be taken for further reduction

$$\equiv r - s' + 1 \bmod s\text{-}1$$

this is congruent to 1 mod s-1, which implies

$$(s')' = s \qquad (3.6.2.3)$$

i.e after the first reduction the number of terms to be taken for any further reduction is $s' = s$.

Collecting the results for clarity, the procedure being followed for reduction of the initial vector $\overline{p}$ is :-

For the first reduction :

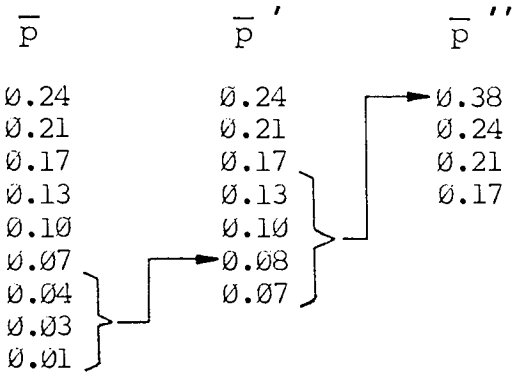
$$s' \in \{ 2,3,\ldots \ldots,s \} \qquad (3.6.2.1)$$

$$s' = r \ (\text{mod } s-1) \qquad (3.6.2.2)$$

For subsequent reductions

$$(s')' = s \qquad (3.6.2.3)$$

and the example under consideration can be written as :-

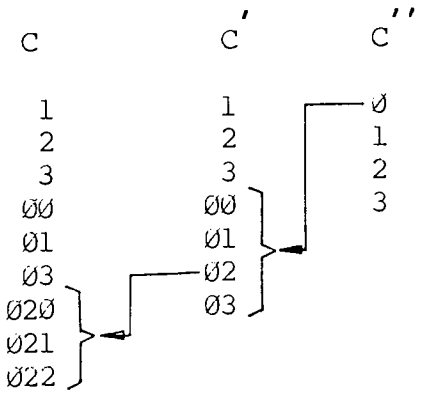


Since s = 4, the optimal UD code for $\bar{p}''$ is

$$C'' = \{ Ø, 1, 2, 3 \}$$

achieving $n_4(\bar{p}'') = 1$ (average word length)

Starting from this trivial code, and working backwords the optimal codes C' and C for $\bar{p}'$ and $\bar{p}''$ respectively can be achieved by expanding the code in a simple way :-



The code in C is that required for the elements in vector $\bar{p}$.

It can be noted that the binary codes are less complicated to construct, since the radix is s = 2 and for any number of terms $r \geq 2$, $s' = 2$ i.e. conditions (3.6.2.1) and (3.6.2.2) reduce to (3.6.2.3) implying that at any stage the number of terms to take for reduction is the smallest two. The algorithm used in this thesis, to construct Huffman tables is outlined in the flow diagram of Figure 3.6.2.2

3.6.3 Summary of Threshold Encoding/Decoding Scheme

The block diagram given in Figure 3.6.3.1, shows the procedure involved when undertaking threshold coding. Once the sub-image size has been set, the "pre-run" mentioned in Section 3.6 can be carried out. This proceeds as follows :-

The appropriate sub-image is selected and the data is translated to it's centroid by subtraction of the mean of the image from every data value. A two-dimensional cosine transform is then applied to the sub-image, decorrelating the data. The pre-set threshold is then used to modify the data as outlined in Section 3.6.1. After normalization and quantization the

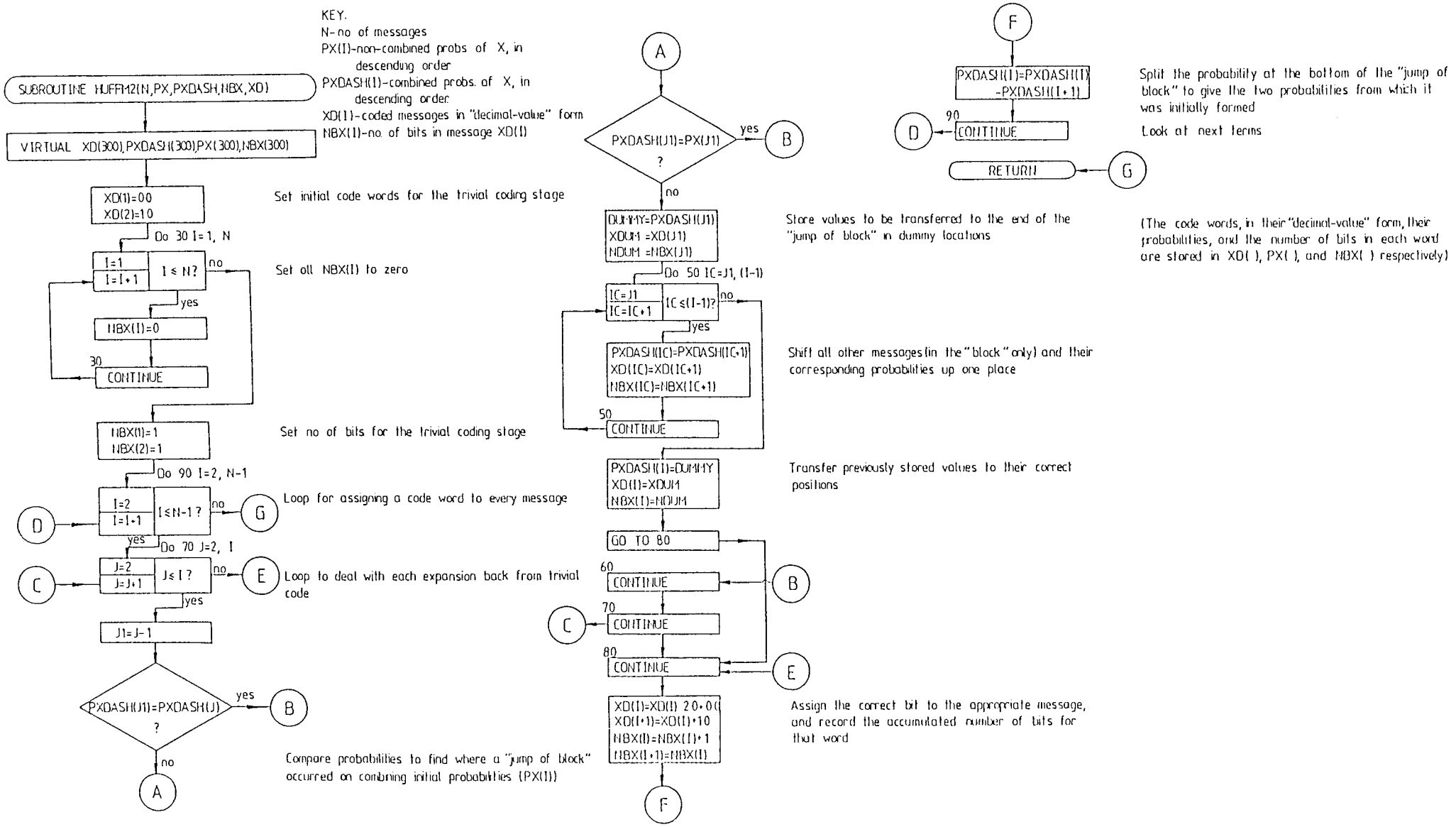


Figure 3.6.2.2 Flow Diagram for Formation of Huffman Code Tables

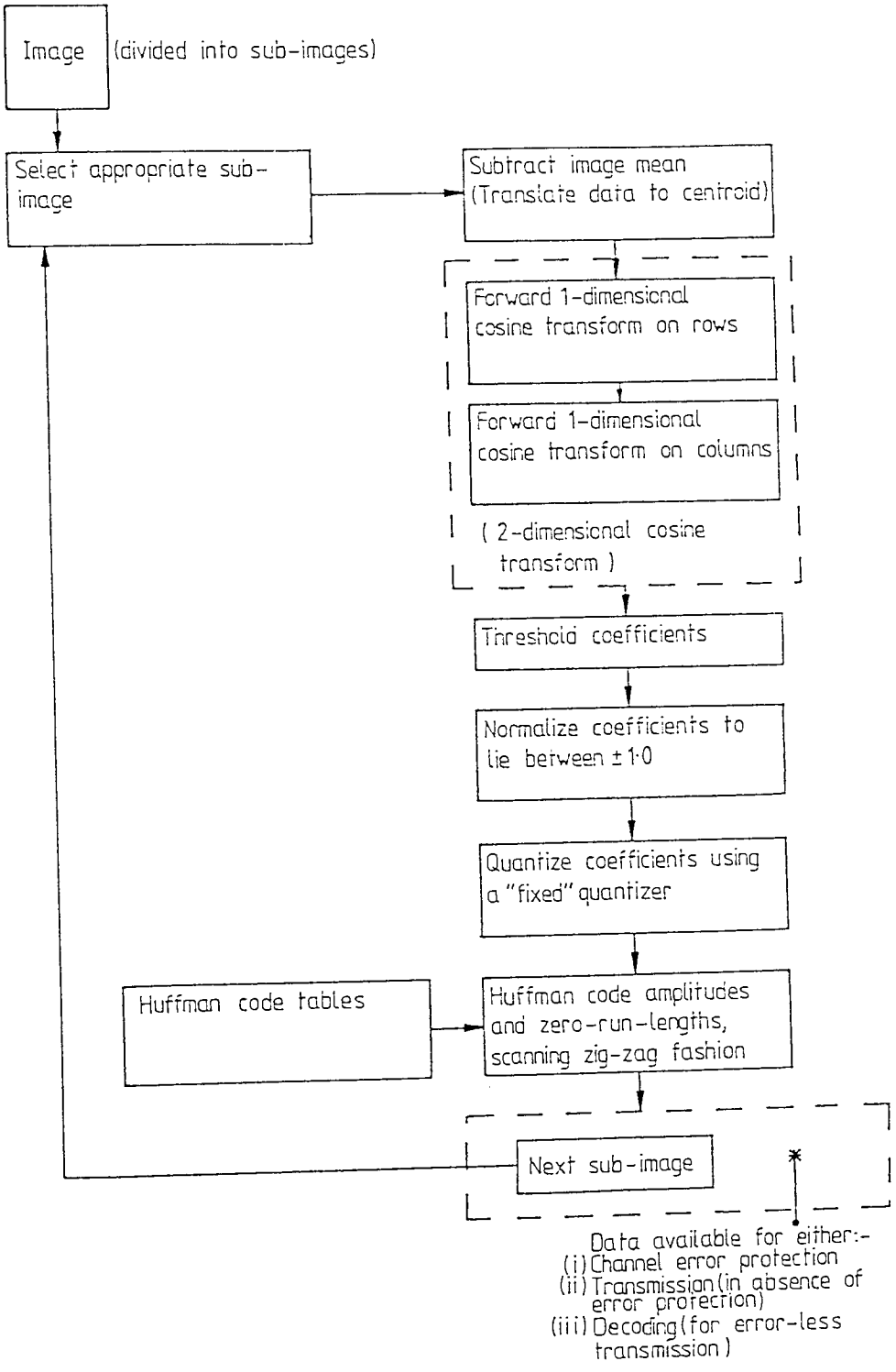


Figure 3.6.3.1 Threshold Source Encoder, Block Diagram

probabilities of occurrence of the amplitudes and z.r.l. are determined for that particular sub-image. The results are stored and further sub-images selected and subjected to the same procedure. Once all sub-images have been processed in this manner tables can be formed which contain the probabilities of the amplitudes and z.r.l. averaged over the whole image. From this table the Huffman algorithm developed in Section 3.6.2 can be used to create the required Huffman code tables. The actual coding of the image for compression can now be undertaken.

In order to code the image the procedure outlined above is repeated for each sub-image, up to and including the quantization stage. Then the coefficients are coded using the Huffman code tables developed in the "pre-run", as shown in Figure 3.6.3.1. The bit stream obtained after Huffman coding is then available for one of three options, these again being :-

(i) Channel coding

(ii) Transmission (errorless)

(iii) Decoding (for errorless transmission)

Option (iii) involves only the inverse operations to those undertaken when threshold source encoding. These "inverse operations" are shown in Figure 3.6.3.2, and form the threshold source decoder. The encoder and decoder of Figure 3.6.3.1 and

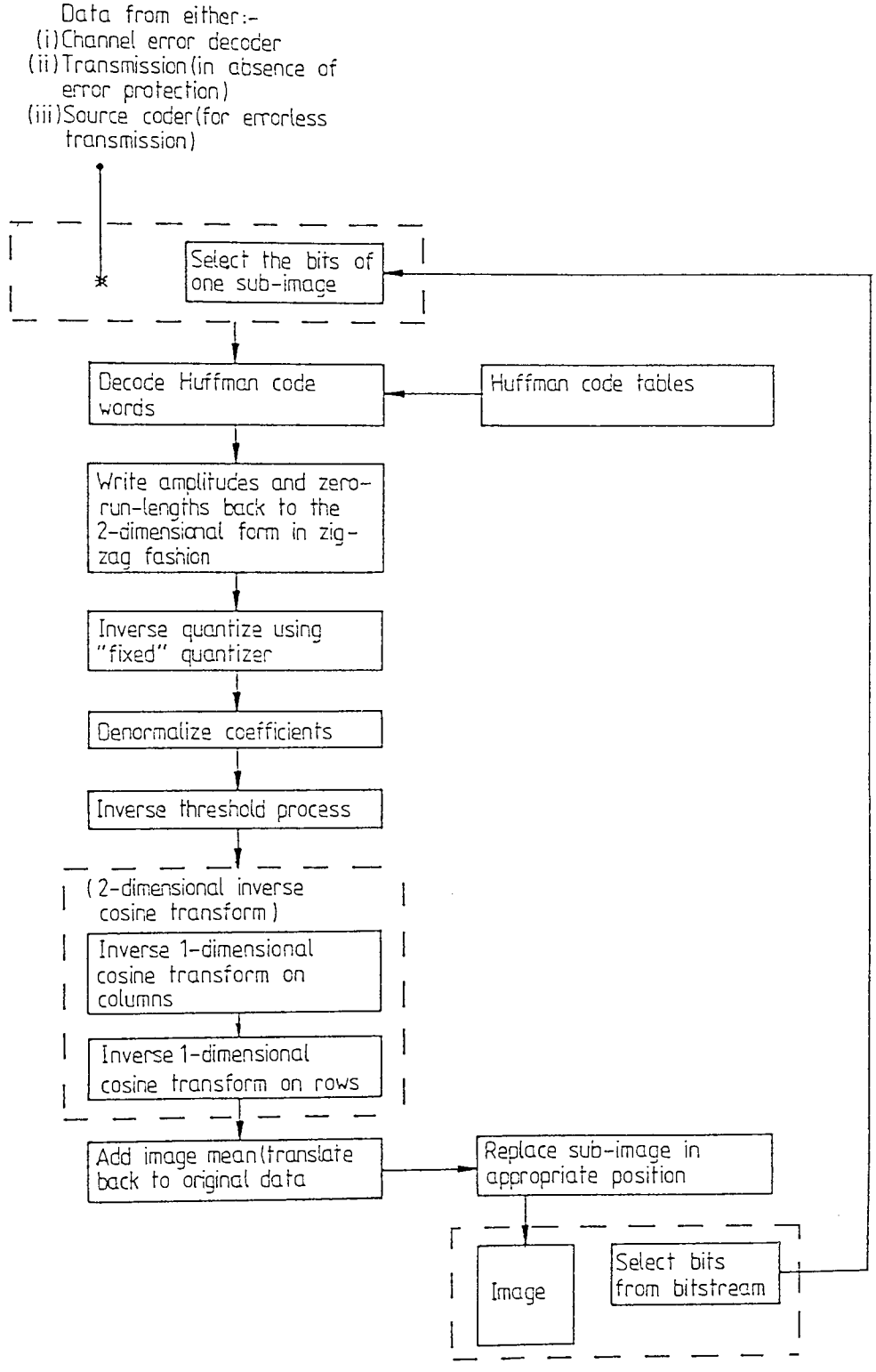


Figure 3.6.3.2 Threshold Source Decoder, Block Diagram

3.6.3.2 respectively form the threshold data compression scheme to be analysed in the presence of a channel.

The advantages and disadvantages of the threshold scheme, without channel error protection, are to be deduced from preliminary tests (Chapter 4). The scheme is seen to be more complex than the zonal source encoder previously described. The adaptivity of the scheme is seen to arise from the nature of the scheme itself i.e. more bits are automatically assigned to sub-images of higher activity/energy (see Chapter 4). The merits, or otherwise, of the scheme are to be deduced at a later stage.

3.7 The Quantizer

The previous sections outline the methods of quantizing transformed coefficients, but do not pay attention to the actual quantizer itself. It is also an important step in the compression process, and so it is necessary to determine the type of quantizer to use.

The process of quantization introduces distortion. In order to reduce the effect of this distortion, non-uniform level distribution is used i.e. in regions where very accurate definition

of the signal is required a high density of quantization levels is desired. Conversely, where high accuracy is not so important, the level spacing can be relatively sparse. By considering the statistical properties of the signal, a non-uniform level distribution is found which is a function of the probability density of the source. The assumption of Gaussian data statistics has already been made. Hence, the probability density to be used is the Gaussian probability density.

The quantizer is to take on the "table look-up" form, which is probably the simplest to implement. The table, normalized to a value of ±1, consists simply of the output level numbers that are assigned to a particular range of input coefficients. In practice, the signal would first undergo compression followed by uniform quantization, which has the same effect as considering non-uniform spacing. Thus, in order to visualize the effect of non-uniform quantizing, compression curves can be plotted (see Section 3.7.4)

3.7.1 Relationship for Level Spacing to Give Minimum Distortion

The quantized signal y(t) is taken as shown in Figure 3.7.1.1. The quantization levels are symmetrical about the zero

level but, apart from this, are placed arbitrarily in the interval (-1, +1). The levels are denoted by y_{-n}, y_{-n+1}, ..., $y_{\emptyset}$,

y_1, ..., y_{n-1}, y_n

where,

$$y_k = y_{-k}$$

$$y_{\emptyset} = \emptyset$$

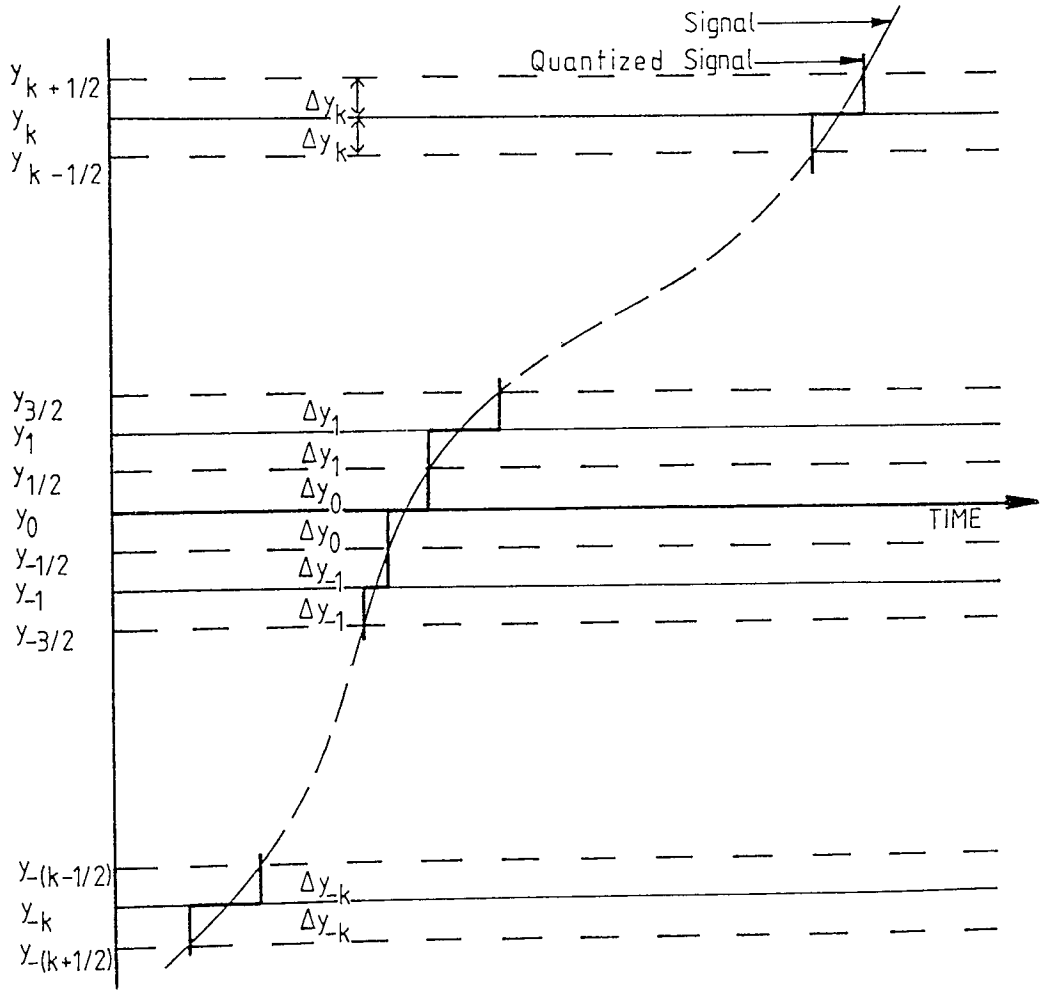


Figure 3.7.1.1 Non-uniform Quantization.

Considering the k^{th} level, choose

$$y_{k-1/2} < y < y_{k+1/2} \qquad (3.7.1.1)$$

This value would be transmitted as level y_k.

The crossover values are the y's with fractional subscripts.

The mean square distortion is defined as :-

$$\sigma_k = \int_{y_{k-1/2}}^{y_{k+1/2}} (y - y_k)^2 . P(y) . dy \qquad (3.7.1.2)$$

where,

σ_k = distortion in the k^{th} level

$y - y_k$ = error of transmitted signal

$P(y)$ = probability density of signal y

$P(y)$ can be assumed constant over the region of integration, for close level spacing, so that

$$P(y) = P(y_{av.})$$

where,

$$y_{av.} = \frac{y_{k+1/2} + y_{k-1/2}}{2}$$

Now from equation (3.7.1.2)

$$\sigma_k = \int_{y_{k-1/2}}^{y_{k+1/2}} (y - y_k)^2 . P(y) . dy$$

$$= P(y_{av.}) \int_{y_{k-1/2}}^{y_{k+1/2}} (y - y_k)^2 . dy$$

$$= \frac{P(y_{av.}) . [(y_{k+1/2} - y_k)^3 + (y_k - y_{k-1/2})^3]}{3}$$

(3.7.1.3)

Differentiating σ_k w.r.t. y_k gives equation (3.7.1.3) as |-

$$\frac{d(\sigma_k)}{dy_k} = \frac{3 . P (y_{av.})[(y_{k+1/2} - y_k)^2 .(-1) + (y_k - y_{k-1/2})^2]}{3}$$

$$= P(y_{av.})[-(y_{k+1/2} - y_k)^2 + (y_k - y_{k-1/2})^2]$$

(3.7.1.4)

For a minimum

$$\frac{d(\sigma_k)}{dy_k} = \emptyset$$

which, using equation (3.7.1.4) gives

$$y_{k+1/2} - y_k = y_k - y_{k-1/2}$$

or

$$y_k = \frac{y_{k+1/2} + y_{k-1/2}}{2} \qquad (3.7.1.5)$$

Equation 3.7.1.5 implies that the condition for making σ_k a minimum is for y_k to lie half-way between $y_{k+1/2}$ and $y_{k-1/2}$:-

$$y_{k+1/2} = y_k + \Delta y_k$$

$$y_{k-1/2} = y_k - \Delta y_k \qquad (3.7.1.6)$$

where,

Δy_k is as shown in Figure 3.7.1.1

Substituting equation (3.7.1.6) into (3.7.1.3) gives :-

$$\sigma_k = \frac{2 \cdot P(y_k) \cdot \Delta y_k^3}{3} \qquad (3.7.1.7)$$

Equations (3.7.1.5) and (3.7.1.7) are the required expressions showing the level spacing required to give minimum distortion σ_k in the y_k level.

3.7.2 An Approximate Method of Obtaining Quantization Levels

Full use is made of the fact that the quantization levels are symmetrically situated about the zero axis. This means only positive values need be considered - the negative values can be found by simply negating the positive ones. An approximate method of obtaining levels can be acheived by writing :-

$$y_k = \Delta y_0 + 2\Delta y_1 + \ldots \ldots + 2\Delta y_{k-1} + \Delta y_k$$

$$= \frac{K}{N}\left[\frac{1}{P^{1/3}(y_0)} + \frac{2}{P^{1/3}(y_1)} + \ldots \ldots + \frac{2}{P^{1/3}(y_{k-1})} + \frac{1}{P^{1/3}(y_k)}\right]$$

$$= \frac{K}{2}\left[\frac{1}{P_{\emptyset}^{1/3}(y)} + \frac{2}{P_{1}^{1/3}(y)} + \ldots\ldots + \frac{2}{P_{k-1}^{1/3}(y)} + \frac{1}{P_{k}^{1/3}(y)}\right]\cdot\frac{2}{N}$$

(3.7.2.1)

where,

K is a constant

N is the total number of levels

The series given in equation (3.7.2.1) can be approximated by the integral

$$y = A\int_{\emptyset}^{z}\frac{1}{P^{1/3}(z)}\cdot dz \qquad (3.7.3.2)$$

where, the variable on the right is changed to z to avoid confusion. 'A' is a constant of proportionality chosen so that when z=1, y=1. Hence

$$y = \frac{\int_{\emptyset}^{z}\frac{1}{P^{1/3}(z)}\cdot dz}{\int_{\emptyset}^{1}\frac{1}{P^{1/3}(z)}\cdot dz} \qquad (3.7.3.3)$$

Using equation (3.7.3.3) the quantization levels can be found. The input values are given by the non-uniform spacing y, and the output uniform spacing is represented by the variable z. Examples can be seen in Section 3.7.4

3.7.3 Level Spacing for the Gaussian Probability Density

The source data is manipulated into a form that is assumed to take on a Gaussian distribution function.

The Gaussian probability density function is given by :-

$$P(z) = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sigma_a} \cdot \exp\left[-(z-\mu)^2 / (2.\sigma_a^2) \right] \qquad (3.7.3.1)$$

where,

μ = mean

σ_a = standard deviation

Substituting equation (3.7.3.1) into (3.7.3.3) gives :-

$$y = \frac{(2\pi)^{1/6} \cdot \sigma_a^{1/3} \cdot \int_{\emptyset}^{z} \exp[-(z-\mu)^2/(6\sigma_a^2)] \cdot dz}{(2\pi)^{1/6} \cdot \sigma_a^{1/3} \cdot \int_{\emptyset}^{1} \exp[-(z-\mu)^2/(6\sigma_a^2)] \cdot dz} \qquad (3.7.3.2)$$

Putting $\mu = \emptyset$, the above integral can be found to be :-

$$y = \frac{\left[z + \frac{z^3}{\sigma_a^2 3.6} + \frac{z^5}{\sigma_a^4 5.6^2 .2!} + \frac{z^7}{\sigma_a^6 7.6^3 .3!} + \ldots \right]_0^Z}{\left[z + \frac{z^3}{\sigma_a^2 3.6} + \frac{z^5}{\sigma_a^4 5.6^2 .2!} + \frac{z^7}{\sigma_a^6 7.6^3 .3!} + \ldots \right]_0^1} \qquad (3.7.3.3)$$

N.B. The evaluation of equation (3.7.3.2) to give equation (3.7.3.3) can be found in Appendix 1. This evaluation was left out to enhance readibility.

The q^{th} term of the numerator of equation (3.7.3.3) can be seen to be :-

$$q^{th} \text{ term} = \frac{z^{2q-1}}{(2q-1).6^{q-1}.(q-1)!.\sigma_a^{2q-2}} \qquad (3.7.3.4)$$

The q^{th} term of the denominator of equation (3.7.3.3) is of the same form as equation (3.7.3.4) but with z=1. For computing purposes the term can be written as

$$q^{th}\ \text{term} = \frac{q.(z/\sigma_a)^{2q-1}.\sigma_a}{(2q-1).6^{q-1}.q!} \qquad (3.7.3.5)$$

The level spacing, for the Gaussian Probability Density, then can be found by using equations (3.7.3.4), (3.7.3.5) and (3.7.3.3).

3.7.4 Use of Quantizer

For both methods of source encoding/decoding (zonal and threshold) the quantizers used are of the form outlined in Sections 3.7.1 to 3.7.3.

As already seen, with the zonal compression scheme quantization is carried out according to the bit-map. Each zone of the bit-map uses a different quantizer, although the only real difference is the number of levels the quantizer has. The quantizers themselves are of the same form, the "look-up" tables generated in the same way, through use of the same equations, outlined in the

previous sections. In this way, a "variable" quantizer is formed and utilized as required.

With the threshold quantizer a "fixed" quantizer is used - "fixed" in the sense that only one "look-up" table is used which contains, typically, 256 or 128 levels (i.e. an 8 or 7 bit quantizer). In the next chapter it will become obvious that a trade-off occurs between the number of bits used for quantization and the setting of the threshold when trying to acheive a certain bit-rate for a given amount of distortion. The results of this trade-off are used to determine the number of quantizer levels to be used.

Figure 3.7.4.1. shows the general compression curve, indicating the non-uniform spacing of the General Quantizer. This shows the type of results to expect when determining the level spacing dictated by the Gaussian probability density. Actual compression curves obtained are shown in Figure 3.7.4.3 for standard deviations of σ_a = 1.Ø, Ø.5, Ø.25 and Ø.15.

The general form of the quantizer, in table "look-up" form is shown in Figure 3.7.4.2. The third column (DIVY(N)) indicates the range of input values assigned to a particular level.

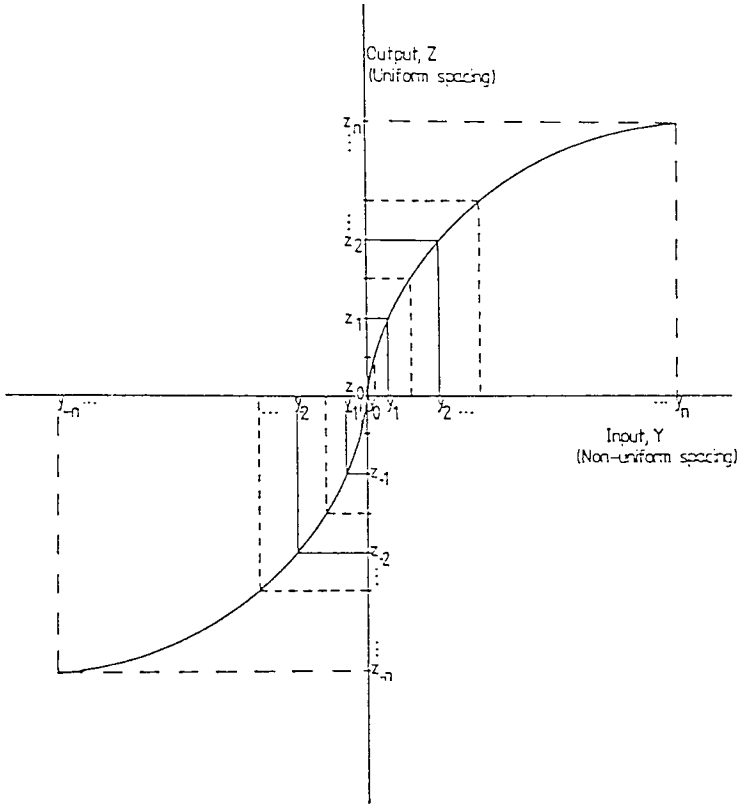


Figure 3.7.4.1 Compression Curve for Non-uniform Spacing of the General Quantizer

Output, Z(n) Uniform spacing	Input, Y(n) Non-uniform spacing	Range, DIVY(n)
Z(127)	Y(127)	DIVY(127)
Z(126)	Y(126)	DIVY(126)
Z(125)	Y(125)	
Z(1)	Y(1)	DIVY(2)
Z(128)	Y(128)	DIVY(1)
Z(129)	Y(129)	DIVY(129)
		DIVY(130)
Z(253)	Y(253)	DIVY(254)
Z(254)	Y(254)	DIVY(255)
Z(255)	Y(255)	

Figure 3.7.4.2 Table "look-up" Form of General Quantizer (for 8-bit Quantization)

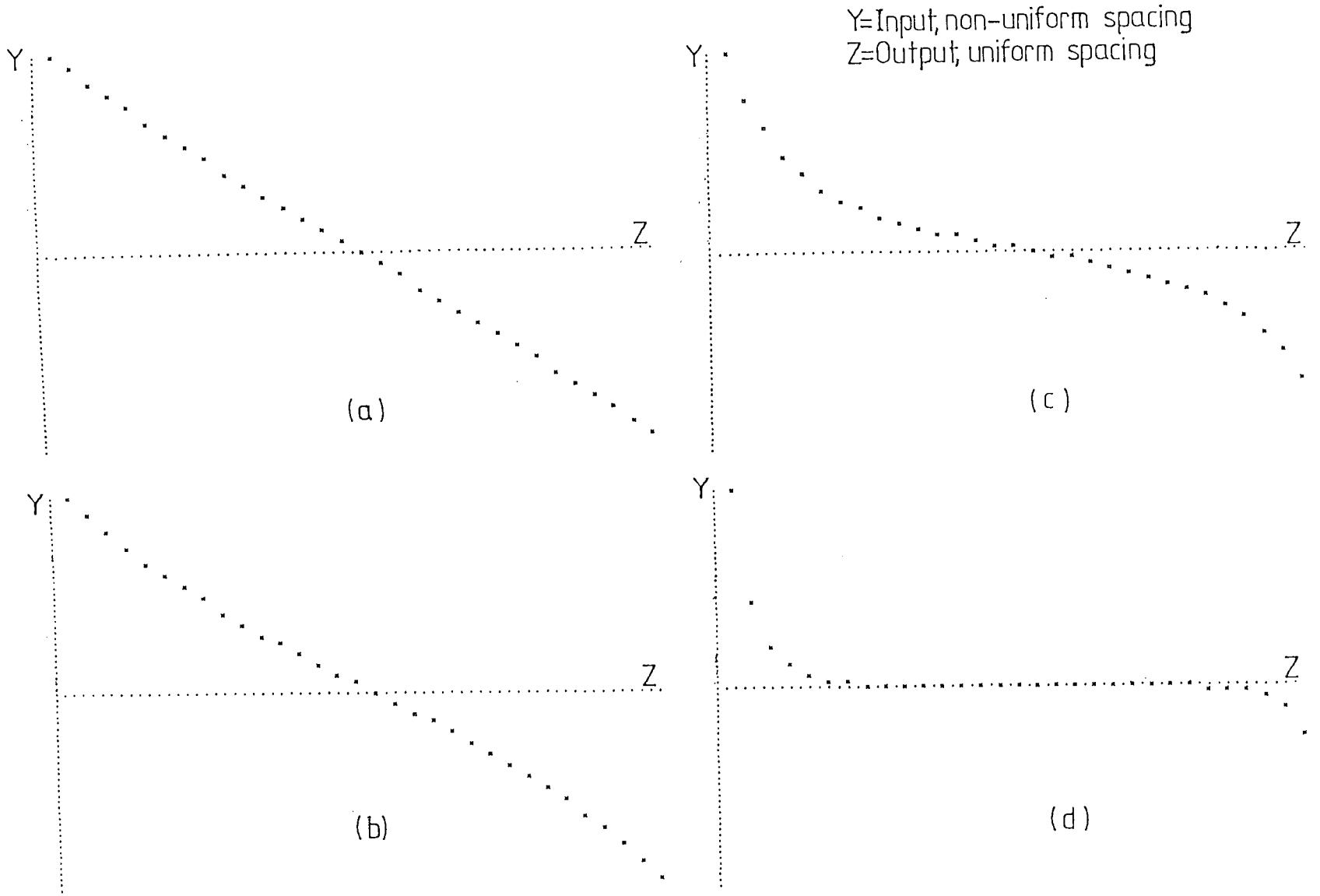


Figure 3.7.4.3 Compression Curves for Non-uniform Spacing of the Gaussian Quantizer, using Standard Deviations of :- (a) 1·0, (b) 0·5, (c) 0·25, (d) 0·15

CHAPTER 4

RESULTS/PERFORMANCE OF SOURCE ENCODING SCHEMES IN THE ABSENCE OF CHANNEL ERROR PROTECTION

4.1 Introduction - Assessing Performance of Source Coders

The two methods of assessing the performance of the source encoding schemes are:-

(a) Subjective assessment of reproduced images

(b) Quantitative assessment of reproduced images

Ideally the two methods of judging reproduced images, given above, should correlate exactly. Image processing, however, can still be regarded as in it's infancy and so it may be expected that the ideal condition, where the two different methods of assessment produce the same end result, is yet to be found. Until it is, the performance can be estimated by a combination of the two types of assessments. This is the method adopted in this thesis.

The quantitative measure used in conjunction with zonal coding has already been described when considering the initial design of this coder in Chapter 3. This measure is the mean square error fidelity criterion. Based on this measure, the distortion, first in the absence of channel errors, can be plotted against bit-rate and the effect of altering sub-image size observed. The subjective assessment has it's basis in the visual observation of the reproduced image, for which photographs of reconstructed images are used. This assessment is enhanced by the use of error-spectrum photographs where it is considered beneficial to the observer. The error-spectrum is a comparison on a pixel-by-pixel basis between the original image and it's reconstruction for a given compression ratio.

As the threshold source coder design is based primarily on intuitive considerations, no specification has, as yet, been given to the quantitative (or subjective) method of assessment. However, since one of the aims of this thesis is to observe if there is some kind of trade-off between source coder complexity and error-protection coding, when considering channel errors, it makes sense to use the same methods of assessment as those used with the zonal coder.

The fore-mentioned methods, then, are the main methods of assessment used throughout the rest of the work. However, when

considering channel errors, further preliminary tests are carried out on the coders when no channel error protection is present.

The first of these tests carried out is to try to find the most sensitive/important bit positions for a particular sub-image. Single bit-errors are thus deliberately "forced" into a particular position and the subjective result determined. In this way, information is gathered about the relative importance of bit position in a sub-image. Further preliminary tests involve subjecting the two encoding schemes, working at a particular bit rate, to various error rates to see how unprotected source coded data stands up to uniform random channel errors.

Finally, from the information acquired by use of these tests, a summary and comparison table can be drawn up. From this table methods of channel error protection and modifications, if necessary, to the source coders can be deduced, resulting in source/channel coding schemes that can tolerate higher channel error rates.

4.2 Fidelity Measures.

4.2.1 Experimental Measures.

4.2.1.1 Mean Square Error and Signal-to-Noise Measures.

The techniques used for image data compression result in degradation of the reconstructed image. A widely used measure of reconstructed image fidelity for an NxM size image is the average mean square error, defined as :-

$$e_{ms}^2 = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} E(u_{i,j} - \dot{u}_{i,j})^2 \qquad (4.2.1.1.1)$$

where i,j are the horizontal and vertical pixel coordinates

the set, $\{u_{i,j}\}$ represents the NxM original image.

the set, $\{\dot{u}_{i,j}\}$ represents the NxM reproduced image.

and, E indicates estimated value.

Experimentally, the average mean square error is estimated by the average sample mean square error, defined by :-

$$e_{ms}^2 \simeq \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (u_{i,j} - \dot{u}_{i,j})^2 \qquad (4.2.1.1.2)$$

It is useful to normalize equation (4.2.1.1.2) to give results lying in the interval (Ø, 1). For this purpose the normalizing constant is determined by :-

$$K_n = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} (u_{i,j})^2 \qquad (4.2.1.1.3)$$

Giving the normalized average sample mean square error in the given image as :-

$$e_{nms}^2 \simeq \frac{1}{NMK_n} \sum_{i=1}^{N} \sum_{j=1}^{M} (u_{i,j} - \dot{u}_{i,j})^2 \qquad (4.2.1.1.4)$$

where, K_n is as given by equation (4.2.1.1.3).

In this thesis, results are also to be presented using the signal-to-noise ratio (SNR). There are two definitions of SNR that are used, corresponding to the above error. These are defined as :-

$$SNR = 1Ø \log_{1Ø} \frac{(\text{peak to peak value of original image data})^2}{e_{ms}^2} \text{ dB}$$

(4.2.1.1.5)

and,

$$SNR' = 10\log_{10} \frac{\sigma_u^2}{e_{ms}^2} \quad dB \qquad (4.2.1.1.6)$$

where,

σ_u^2 is the variance of the original image.

Although SNR' is more widely used as a measure of signal-to-noise ratio in the signal processing literature, since it is related to the signal power and noise power, and is perhaps more meaningful because it gives 0dB for equal signal and noise power, SNR is used more commonly when image coding. The original image takes the form of discrete samples quantized to a relatively large number of grey levels. Typically the number is 256 levels or 8 bits. This is the case in this work, with exceptions clearly indicated. With this number of levels the peak-to-peak value is 255. Hence equation (4.2.1.1.5) becomes :-

$$SNR = 10\log_{10} \frac{(255)^2}{e_{ms}^2} \qquad (4.2.1.1.7)$$

Thus, equations (4.2.1.1.4) and (4.2.1.1.7) are the ones used for quantitative assessment in this thesis, as they are seen to be related to the required mean square error fidelity criterion.

4.2.1.2 Error Spectrum

The error-spectrum used to enhance the subjective assessment of the reproduced image is defined simply as the scaled magnitude error :-

$$s_{i,j} = K_s (| u_{i,j} - \dot{u}_{i,j} |) \qquad (4.2.1.2.1)$$

where,

$\{u_{i,j}\}, \{\dot{u}_{i,j}\}$ are as defined previously.

$\{s_{i,j}\}$ is the NxM error spectrum

K_s is a constant over the whole of the error spectrum, controlling it's brightness.

4.2.2 Theoretical Measures

It is convenient to have some type of theoretical estimate of the quantitative distortion that occurs over the whole of the reconstructed image.

4.2.2.1 Theoretical, Expected, Distortion with the Zonal Encoding Scheme.

The theoretical estimation has it's foundation in rate-distortion thoery. When considering the mean square error criterion, along with a Gaussian source, the rate-distortion measure is given by :-

$$\text{rate-distortion function, } R(D) = \frac{1}{2N} \sum_{j=1}^{N} \max \left[\emptyset, \log_2 \frac{\sigma_j^2}{K_r D} \right] \qquad (4.2.2.1.1)$$

$$\text{distortion in sub-image of size NxN, } D = \frac{1}{N} \sum_{j=1}^{N} \emptyset . \min (K_r D, \sigma_j^2) \qquad (4.2.2.1.2)$$

where,

$$\left.\begin{array}{ll} \emptyset = H_{w_r} & \sigma_j^2 \leq K_r D \\ \emptyset = 1.\emptyset & K_r D < \sigma_j^2 \end{array}\right\} \qquad (4.2.2.1.3)$$

$$\sigma_j^2 = S_{w_r}, \quad i=j \qquad (4.2.2.1.4)$$

$K_r D$ is a constant

σ_j^2 are the leading diagonal terms of the transform domain estimated covariance matrix.

S_{w_r} is the power spectrum estimate of σ_j^2, as given by equation (3.5.2.2)

H_{w_r} is the weighting factor of equation (3.5.2.1)

These equations are in-keeping with similar, but not the same, equations found in refs. [2] and [86]. Equation (4.2.2.1.2) implies that, for each coefficient position, $K_r D$ and σ_j^2 must be compared and whichever is the smaller is used in the distortion summation, the greater value being discarded.

The function $\emptyset$, as defined by equation (4.2.2.1.3), takes into account the weighting coefficients that were used to account for the subjective assessment. Use of (4.2.2.1.1) through (4.2.2.1.4) give an estimate of the rate-distortion theoretical performance on a "per sub-image basis".

To normalize equation (4.2.2.1.2), ensuring results lie in the interval ($\emptyset$, 1), use is made of the normalizing constant :-

$$K_{tn} = \frac{1}{N} \sum_{j=1}^{N} \sigma_j^2 \qquad (4.2.2.1.5)$$

Giving the normalized, theoretical distortion as :-

$$D_n = \frac{1}{N.K_{tn}} \sum_{j=1}^{N} \emptyset.\min(K_r D, \sigma_j^2) \qquad (4.2.2.1.6)$$

where,

K_{tn} is as given in equation (4.2.2.1.5)

4.3 Performance in the Absence of Channel Errors

As the title suggests, this section deals with assessing the performance of the source encoding schemes without transmission over a channel. The output from the source encoder is fed directly into the input of the source decoder as indicated in Figure 4.3.1, following the notation of Section 4.2 :-

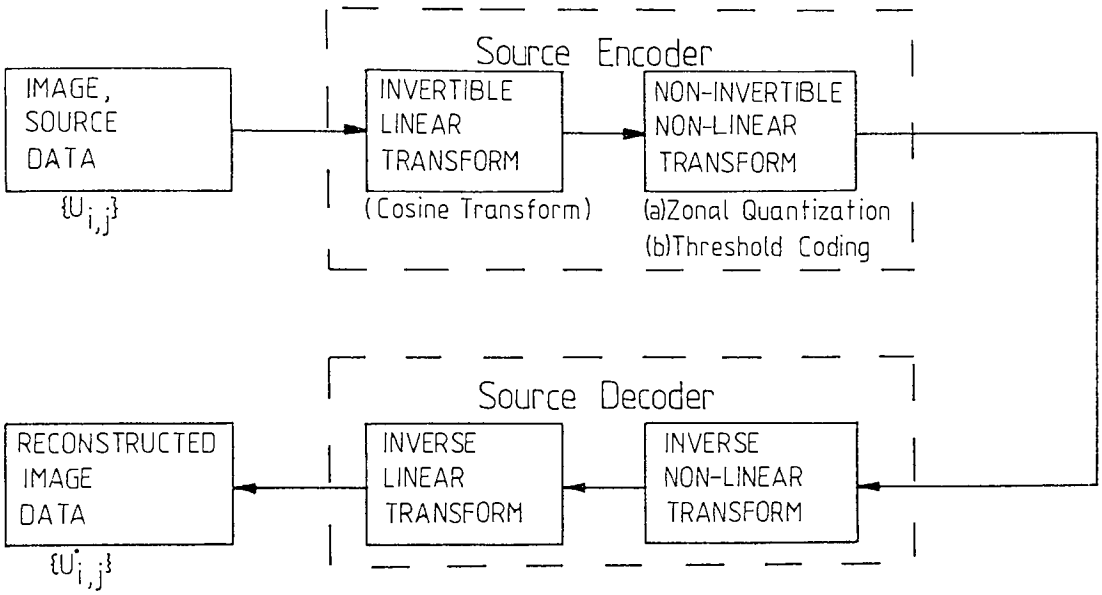


Figure 4.3.1 Block Diagram for Data Compression Schemes, Assessment in the Absence of a Channel.

The method of presentation of the assessment is to follow through each coding scheme, in Sections 4.3.1 and 4.3.2, indicating various aspects of the data input and output at various stages, culminating in the use of the fidelity measures mentioned in Section 4.2 to plot graphs and present visual images (photographs) which, in turn, lead to comparative deductions.

4.3.1 Distortion with Zonal Source Encoding.

The continuous 'flow' of the zonal source encoder is shown in Figure 3.5.3.1. This section indicates the relevant data input necessary, and various data output/results acquired at various stages.

Setting image parameters ¦-

Image total horizontal and vertical resolution¦ 256x256 (pixels square)

Sub-image resolution is set either at ¦ 4x4, 8x8, 16x16, or 32x32 pixels square

The Bit-map ¦-

As the zonal coding system is non-adaptive, one particular bit-map is deduced for a fixed sub-image size. Recalling that the bit-map is based on estimated variances, typical examples of estimated variances and resulting bit maps for an average encoding rate of 1.Ø bits/pixel are shown in Figure 4.3.1.1. Other similar results can be obtained to yield different average encoding rates.

```
9 6 6 5 4 4 3 2 2 1 0 0 0 0 0 0
6 6 6 5 4 4 3 2 2 1 0 0 0 0 0 0
6 6 5 5 4 4 3 2 2 1 0 0 0 0 0 0
5 5 5 4 4 3 3 2 1 1 0 0 0 0 0 0
4 4 4 4 3 3 2 2 1 0 0 0 0 0 0 0
4 4 4 3 3 2 2 1 1 0 0 0 0 0 0 0
3 3 3 3 2 2 1 1 0 0 0 0 0 0 0 0
2 2 2 2 2 1 1 0 0 0 0 0 0 0 0 0
2 2 2 1 1 1 0 0 0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1610.00 427.01 276.28 125.96  52.43  20.93   8.14   3.10   1.17   0.43   0.16   0.06   0.02   0.01   0.00   0.00
 427.01 389.62 232.83 109.70  46.91  19.08   7.52   2.90   1.10   0.41   0.15   0.05   0.02   0.01   0.00   0.00
 276.28 232.83 145.42  74.56  34.12  14.59   5.96   2.36   0.91   0.35   0.13   0.05   0.02   0.01   0.00   0.00
 125.96 109.70  74.56  42.08  20.93   9.56   4.12   1.70   0.68   0.26   0.10   0.04   0.01   0.01   0.00   0.00
  52.43  46.91  34.12  20.93  11.28   5.53   2.53   1.10   0.46   0.18   0.07   0.03   0.01   0.00   0.00   0.00
  20.93  19.08  14.59   9.56   5.53   2.90   1.40   0.64   0.28   0.12   0.05   0.02   0.01   0.00   0.00   0.00
   8.14   7.52   5.96   4.12   2.53   1.40   0.72   0.35   0.16   0.07   0.03   0.01   0.00   0.00   0.00   0.00
   3.10   2.90   2.36   1.70   1.10   0.64   0.35   0.17   0.08   0.04   0.02   0.01   0.00   0.00   0.00   0.00
   1.17   1.10   0.91   0.68   0.46   0.28   0.16   0.08   0.04   0.02   0.01   0.00   0.00   0.00   0.00   0.00
   0.43   0.41   0.35   0.26   0.18   0.12   0.07   0.04   0.02   0.01   0.00   0.00   0.00   0.00   0.00   0.00
   0.16   0.15   0.13   0.10   0.07   0.05   0.03   0.02   0.01   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.06   0.05   0.05   0.04   0.03   0.02   0.01   0.01   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.02   0.02   0.02   0.01   0.01   0.01   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.01   0.01   0.01   0.01   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
```

```
9 5 4 3 1 0 0 0
5 5 4 2 1 0 0 0
4 4 3 2 1 0 0 0
3 2 2 1 0 0 0 0
1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
825.00 353.78  90.13  15.14   2.24   0.31   0.04   0.01
353.78 222.02  60.27  11.17   1.76   0.25   0.03   0.00
 90.13  60.27  20.82   4.80   0.88   0.14   0.02   0.00
 15.14  11.17   4.80   1.39   0.31   0.06   0.01   0.00
  2.24   1.76   0.88   0.31   0.08   0.02   0.00   0.00
  0.31   0.25   0.14   0.06   0.02   0.00   0.00   0.00
  0.04   0.03   0.02   0.01   0.00   0.00   0.00   0.00
  0.01   0.00   0.00   0.00   0.00   0.00   0.00   0.00
```

```
7 3 1 0
3 2 0 0
1 0 0 0
0 0 0 0
470.00 115.41  3.84  0.08
115.41  31.78  1.56  0.04
  3.84   1.56  0.15  0.01
  0.08   0.04  0.01  0.00
```

Figure 4.3.1.1 Estimated Variances and Bit-maps (giving approximately 1·0 bits/pixel average) for Different Sub-image Sizes

Image Mean ¦-

This is calculated before-hand by simply adding all pixel values and dividing the result by the total number of pixels. The mean for the original image of Figure 3.2.1 is found to be 98.346. This value is subtracted from every pixel of the sub-image before proceeding with transforming.

Cosine Transformation

No further input required, the number of values to be handled being determined by the sub-image resolution. The output is simply numerical transformed values.

Zonal Quantization

Uses results from the bit-map determination (Figure 4.3.1.1) and pre-formed quantizers. The formation of the quantizers has been seen to rely on the standard deviation of the signal input to the quantizer (for a fixed probability distribution). This standard deviation varies for each, fixed, sub-image size. A typical example is ¦-

For sub-image size 16x16 pixels square,

estimated average standard deviation = Ø.146

The bit-map determines the zonal mask, hence no further input is required.

Decoding/Visual Assessment :-

Also requires no further input (all relevant data being given during encoding). The output is the reconstructed image. Typical examples, again at 1.Ø bits/pixel, are shown in Figures 4.3.1.2 and 4.3.1.3. The error spectra between the reconstructed images and the original image of Figure 3.2.1 is shown on the right of each reconstructed image, in order to enhance assessment i.e. the error spectrum helps to locate the regions where errors of higher magnitude occur - indicating which particular features or textures the coder has more difficulty coping with.

Quantitative Assessment :-

The quantitative assessment comes from resulting graphs that can be obtained from the reconstructed images. The graphs show how the average sample mean square error varies with the average bit rate, for a particular sub-image size. By displaying all the graphs on one set of axes, the effect of varying sub-image size can be clearly seen. This is undertaken, and the result is as shown in Figure 4.3.1.4(a).

(a) SNR = 3Ø.13 dB

(b) SNR = 29.2 dB

Figure 4.3.1.2 Image Reconstructions, and Error Spectra, at Approximately 1.Ø Bits/pixel for Sub-image Size (a) 32x32 (b) 16x16, Pixels Square.

(a) SNR = 27.4 dB

(b) SNR = 24.7 dB

Figure 4.3.1.3 Image Reconstruction, and Error Spectra, at Approximately 1.Ø Bits/pixel for Sub-image Size (a) 8x8 (b) 4x4, Pixels Square.

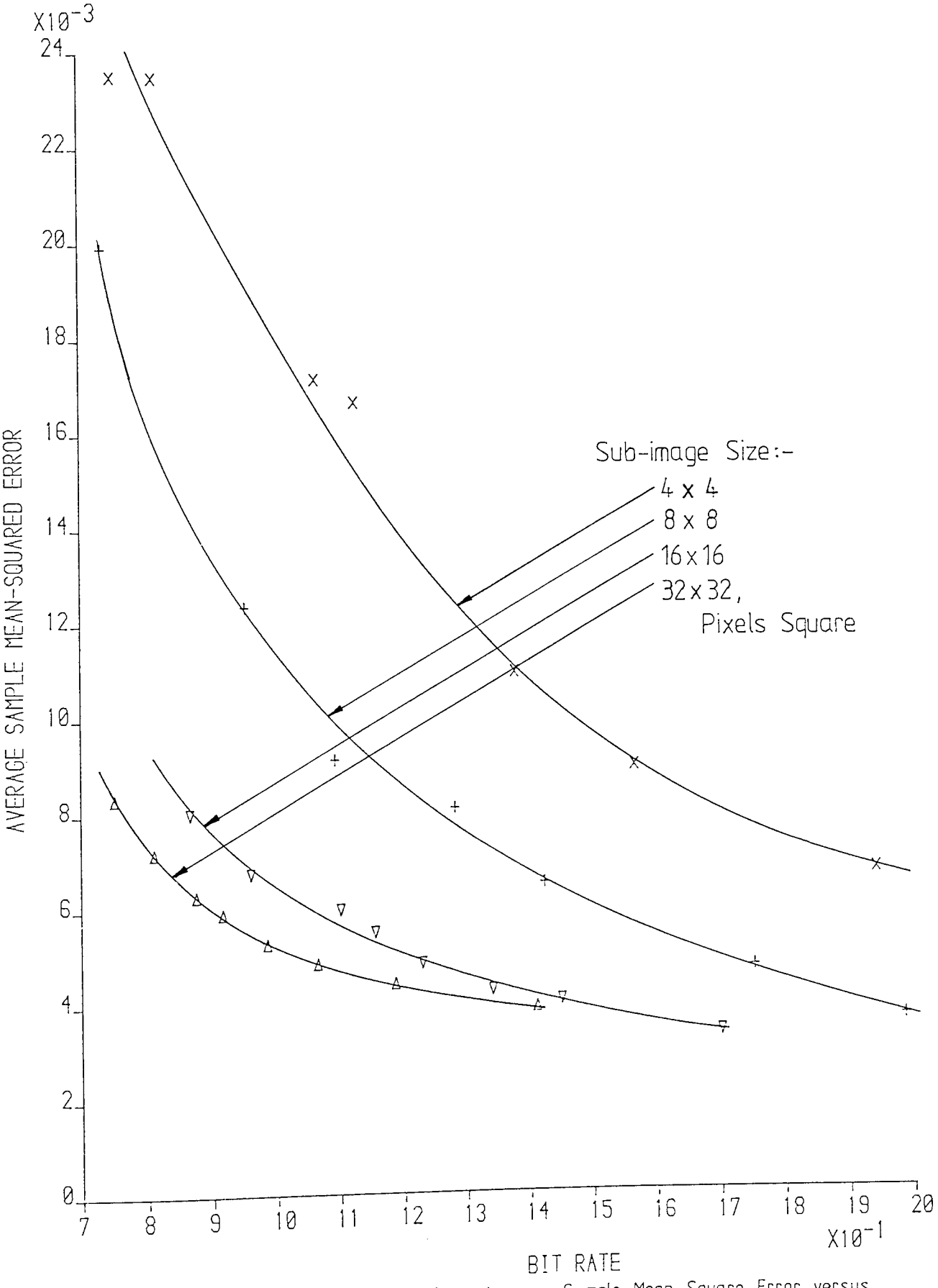


Figure 4.3.1.4(a) Experimental, Normalized, Average Sample Mean Square Error versus Bit-rate.

The graph of Figure 4.3.1.4(b) indicates the theoretical performance of the coder. The SNR for reconstructed images at 1.Ø bits/pixel, using different sub-image sizes, are indicated in Figures 4.3.1.2 and 4.3.1.3.

Conclusions Based on Subjective and Quantitative Assessments :-

Both the subjective and quantitative results show that varying sub-image size can cause a marked difference in the distortion achieved at a given bit-rate. The photographs of Figures 4.3.1.2 and 4.3.1.3 show this in that, for the larger sub-image sizes (32x32, 16x16 pixels square) the reconstructed images could be termed acceptable, whereas for the smaller sub-image sizes (8x8, 4x4 pixels square) the reconstructions would be termed unacceptable due to the distortion having it's manifestation as visible sub-image boundaries and blurring (or in some cases complete loss) of features, textures and patterns. The graph of Figure 4.3.1.4 shows the effect, since the distortion/bit-rate curves get closer to the origin of the graph the larger the sub-image is.

What can be seen more obviously from the subjective assessment is exactly how the distortion manifests itself in the reconstructed images. Looking at the reconstructed images that are termed as acceptable (Figure 4.3.1.2), it is seen that the

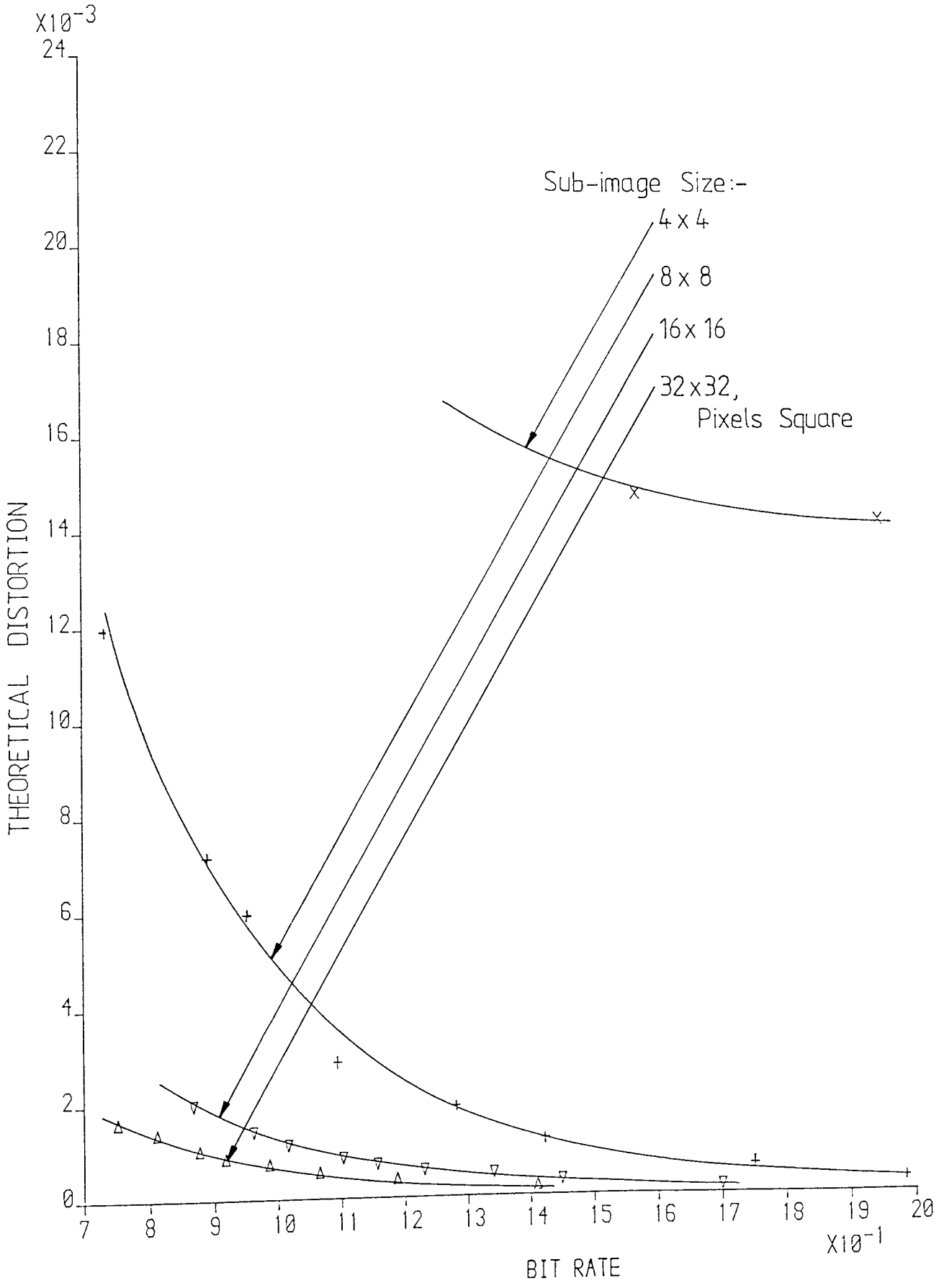


Figure 4.3.1.4(b) Theoretical, Distortion versus Bit-rate

distortion introduced by this method of zonal coding is, relatively, uniformly spread over the entire image. This distortion is also relatively more pleasing to the eye since it has a "snow-like" distribution appearance. The effect would be the same for the smaller sub-image sizes provided the bit-rate is taken high enough to get rid of the appearance of sub-image edges (known as "blocking") which is another form of distortion that arises with the zonal coder when either :-

(a) the number of transformed coefficients has been too severely reduced so that an unfaithful reproduction of most sub-images results, or

(b) the quantization, determined by the bit-map, is too coarse.

The reason for why this "sub-image boundary effect" becomes noticeable with the smaller sub-image sizes of Figure 4.3.1.3 (compared with those of Figure 4.3.1.2) can thus be deduced :- When transforming the larger sub-image sizes, more coefficients are obtained that are (or are close to) the zero value. As this is not the case with the smaller sub-image sizes, effect (a) or (b) inevitably results when trying to achieve the same, relatively low bit-rate. This is particularly true for the smallest sub-image size (4x4 pixels square), where even at relatively

high bit-rates, around 2.Ø bits/pixel, the reconstructed image is obviously degraded. A typical bit-map, in this case to obtain a bit-rate of 1.94 bits/pixel is shown below:-

7 5 3 Ø

5 4 2 Ø

3 2 Ø Ø

Ø Ø Ø Ø

For fairly accurate quantization, 7 bits can be considered sufficient, from which the coarse quantization when using the above bit-map is apparent, although the probable loss of an important coefficient is not. The effect shown in Figure 4.3.1.3 for the 4x4 sub-image size is that of each 4x4 group of pixels effectively displaying the same value - hence the "step-like" nature of this reconstruction. The effect starts to take place around 1.5 bits/pixel, for the 4x4 pixel square sub-image size.

The quantitative results, in the form of the experimentally determined distortion versus average bit-rate curve shown in Figure 4.3.1.4(a), obtained by using the distortion measure of Section 4.2.1.1, show, more clearly, the amount of change in resulting distortion that can be expected when altering sub-image size. Tables A1.4.3.1.1, A1.4.3.1.2, A1.4.3.1.3, and A1.4.3.1.4 in the Appendix show the experimental results and the appropriate

parameter settings required to achieve these results. The curves obtained in theory, shown in Figure 4.3.1.4(b), obtained using the theory in Section 4.2.2, show the same trend as those achieved in practise, with the selection of sub-image sizes 32x32 and 16x16 giving distortion/bit-rate curves relatively close together and with the selection of smaller sub-image sizes resulting in curves indicating the very marked increases in distortion for a given average bit-rate. Tables A1.4.3.1.5, A1.4.3.1.6, A1.4.3.1.7, and A1.4.3.1.8 also in the Appendix indicate the results for Figure 4.3.1.4(b).

Use of Combined Assessments for Setting Encoding Parameters :-

From the foregoing results, various parameters can be set to achieve a desired result. Through the use of subjective assessments, an acceptable bit-rate of 1.Ø bits/pixel can be obtained using sub-images of size either 16x16 or 32x32 pixels square. From the graph of Figure 4.3.1.4(a), this is seen to correspond to respective distortion values of approximately Ø.ØØ488, Ø.ØØ678 (30.1, 28.7 dB). The respective thresholds of acceptability can thus be set at these values. For the 8x8 pixels square sub-image size an acceptable average bit-rate of 1.4 bits/pixel is deduced, corresponding to an acceptability threshold of Ø.ØØ656 (28.8 dB), and for the 4x4 pixels square sub-image the acceptable bit-rate is found to be over 1.96 bits/pixel.

When considering the zonal encoding scheme, without transmission over a channel, the decision would thus be to opt for the largest sub-image size as this would result in the lowest achievable compression ratio for a set distortion threshold-level-of-acceptability. However, it remains to be seen whether there are advantages in having a smaller sub-image size when the source coded data is subject to channel errors.

4.3.2 Distortion with Threshold Source Encoding

The continuous 'flow' of the threshold source encoder is given in Figure 3.6.3.1. As with the zonal scheme, this section indicates the relevant data input necessary, and the various data output/results acquired at various stages.

Setting Image Parameters :-

Image total horizontal and vertical resolution : 256x256 (pixels square)

Sub-image resolution is set at either 4x4, 8x8, 16x16, or 32x32 pixels square

Huffman Code Tables :-

The form of the Huffman code tables has already been seen in Figure 3.6.2.1. However, a more visual insight into how the probability of a certain coefficient amplitude varies in relation to it's position in the sub-image, when scanning in a zig-zag nature, can be seen from the photograph of Figure 4.3.2.1. The vertical axis is the average probability associated with each coefficient, and the horizontal axis is the coefficient position or number. These have been scaled accordingly, in order to ensure that the probability curve obtained for each sub-image covers most of the horizontal axis. Figure 4.3.2.1, then, shows the average histogram for amplitudes over all sub-images. The curve is discussed at a later stage.

Image Mean

Cosine Transformation

} :-

These are handled in the same manner as when considering the zonal scheme (see Section 4.3.1).

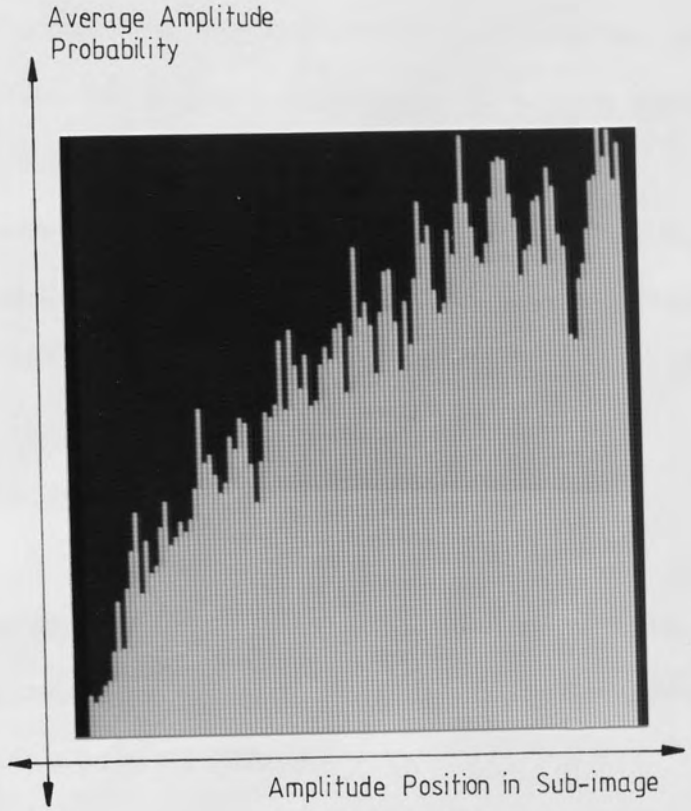


Figure 4.3.2.1 Average Amplitude Probability versus Position in Sub-image, when scanning in a Zig-zag Fashion, for Sub-image Size 16x16.

Thresholding ¦-

Set to attain the desired average bit-rate for a given sub-image size and a given quantizer. A typical example is ¦-

sub-image size = 16x16 pixels square

number of quantization levels = 127 (7 bits)

desired average bit-rate = 1.Ø bits/pixel

∴ threshold, t = 18.Ø

Further examples can be seen from the tables and corresponding graphs presented under the quantitative assessment part of this section.

Normalization ¦-

Data is subject to normalization to ensure it is in the correct range when quantizing. The normalization values are found during the 'pre-run' involved when forming the Huffman code tables. Typical examples are ¦-

sub-image size = 16x16 pixels square

D.C.normalization value $= \frac{1}{\text{max. d.c. value}} = \frac{1}{1600.0}$

$= \underline{\underline{0.000625}}$

threshold, t = 18.0

A.C.normalization value $= \frac{1}{(\text{max. a.c.value} - t)} = \frac{1}{1010.0}$

$= \underline{\underline{0.00099}}$

Again, further examples can be obtained from the tables given later in this thesis.

Quantization :-

Once the average standard deviation of the input data is determined, which will vary depending on sub-image size, either 127 level (7 bit) or 255 level (8 bit) quantization is carried out. The trade-off between the number of levels and the value of threshold, to obtain a certain bit-rate with lowest distortion, is considered later in this section. Since a "fixed" quantizer is used, no further input is required.

Decoding/Visual Assessment :-

All the relevant data is given during encoding and the decoder utilizes the same information. The output is the reconstructed image. Typical examples at 1.Ø bits/pixel are shown in Figures 4.3.2.2, 4.4.2.3, and 4.3.2.4. To enhance assessment the corresponding error spectra are shown on the right of each reconstructed image, for Figures 4.3.2.2 and 4.3.2.3 only.

Quantitative Assessment :-

Using the same distortion measure as for the zonal coder, the average sample mean square error is plotted against average bit-rate, for a particular sub-image size and "fixed" number of quantization levels. The effect of varying sub-image size is seen, once again, by displaying all the graphs on one set of axes, resulting in Figures 4.3.2.5 and 4.3.2.6. The SNR for the reconstructed images at 1.Ø bits/pixel, using different sub-image sizes, are indicated in Figures 4.3.2.2 through 4.3.2.4.

(a) SNR = 32.1 dB

(b) SNR = 31.8 dB

Figure 4.3.2.2 Image Reconstructions, and Error Spectra, at 1.Ø Bits/pixel for Sub-image Size (a) 32x32 (b) 16x16 for a 'fixed' Quantization to 255 Levels (8 Bits)

(a) SNR = 31.5 dB

(b) SNR = 29.1 dB

Figure 4.3.2.3 Image Reconstructions, and Error Spectra, at 1.Ø Bits/pixel for Sub-image Size (a) 8x8 (b) 4x4, for 'fixed' Quantization to 255 Levels (8 Bits)

(a) SNR = 32.3 dB

(b) SNR = 32.1 dB

(c) SNR = 31.7 dB

(d) SNR = 29.7 dB

Figure 4.3.2.4 Image Reconstructions at Approximately 1.Ø Bits/pixel for Sub-image Size (a) 32x32 (b) 16x16 (c) 8x8 (d) 4x4, and 'fixed' Quantization to 127 Levels (7 Bits).

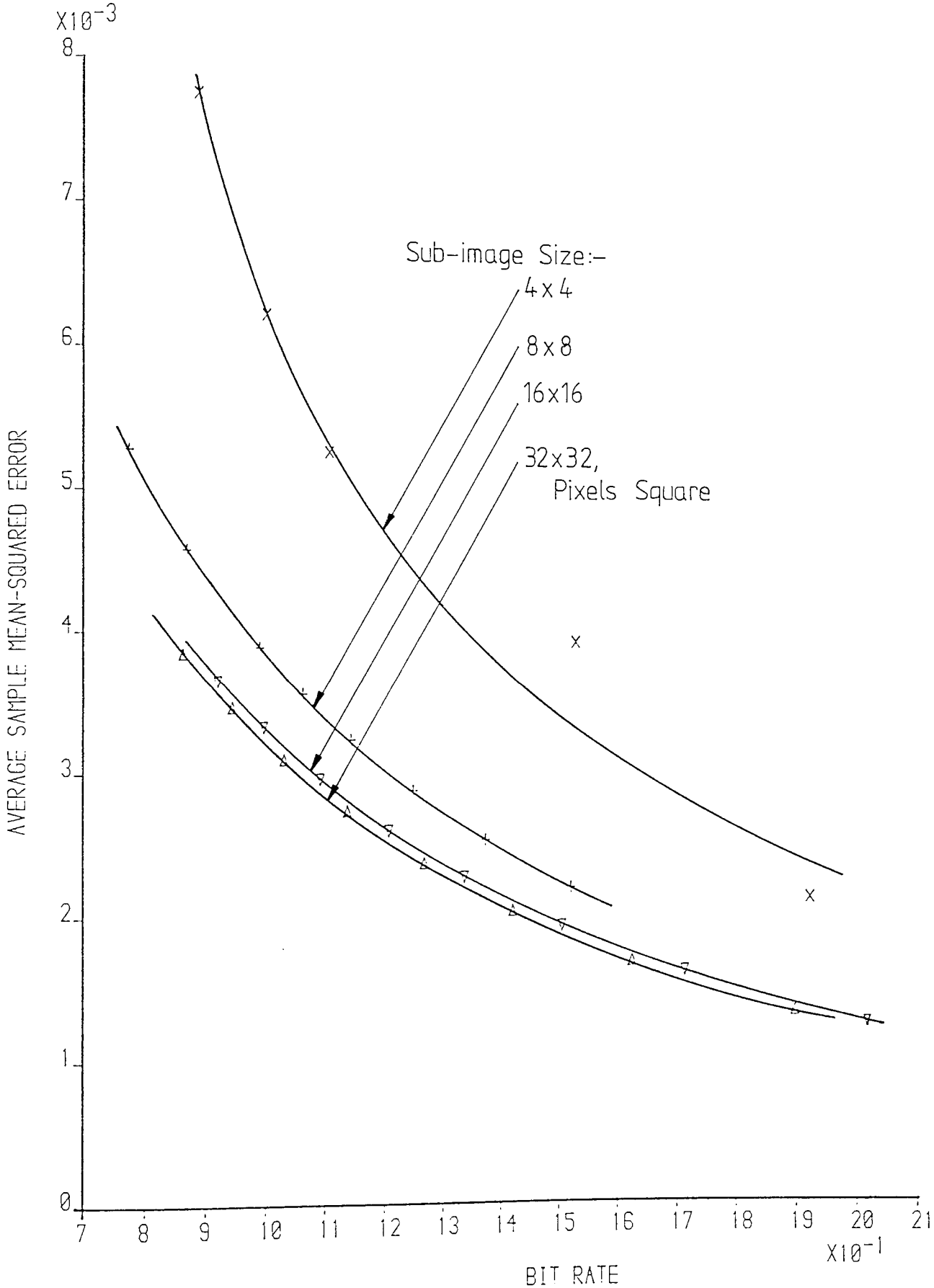


Figure 4.3.2.5 Normalized, Average Sample Mean Square Error versus Bit-rate, for 'fixed' 255 Level (8 Bit) Quantization.

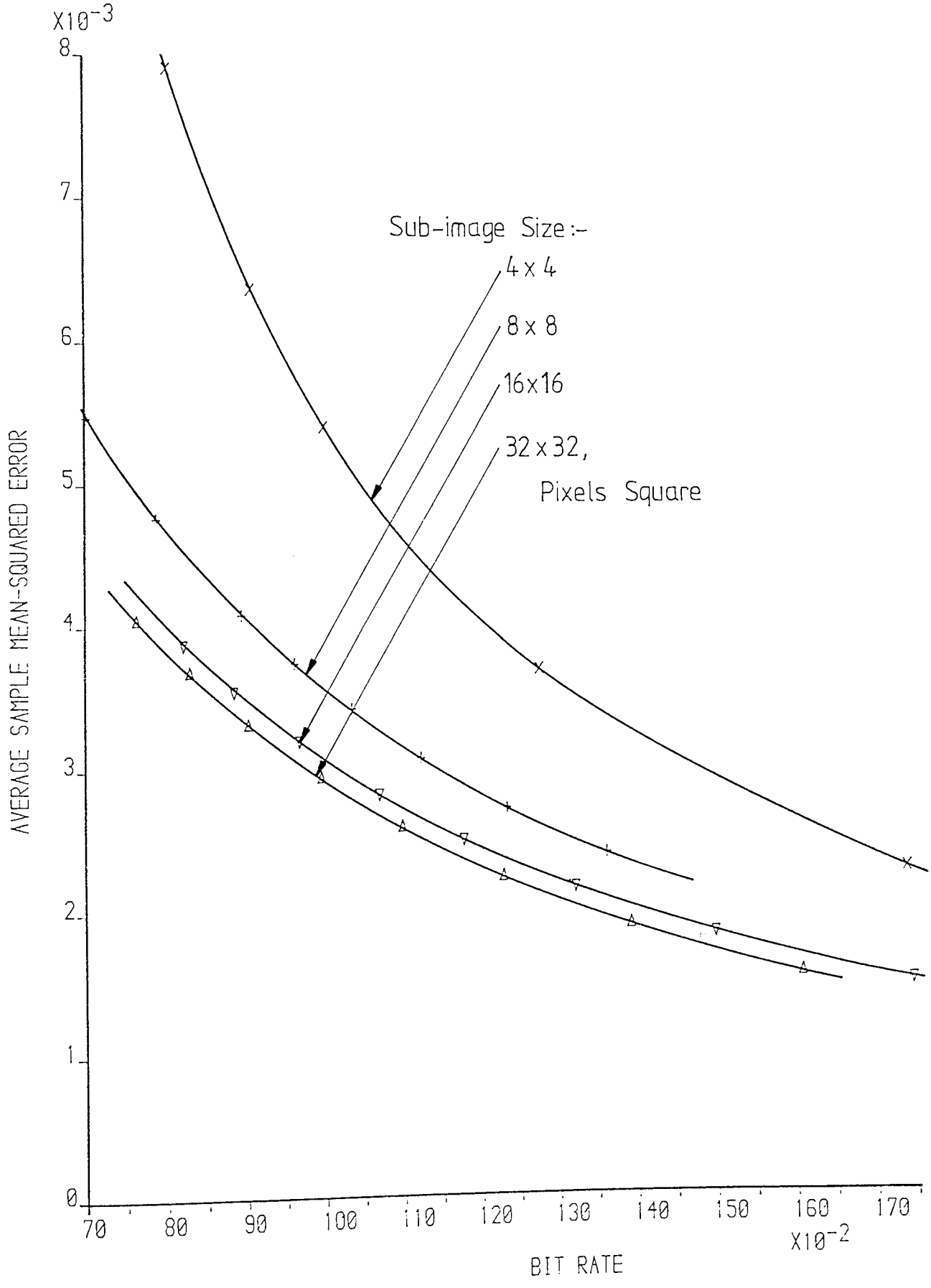


Figure 4.3.2.6 Normalized Average Sample Mean Square Error versus Bit-rate, for 'fixed' 127 Level (7 Bit) Quantization

Conclusions Based on Subjective and Quantitative Assessments :-

As was the case with the zonal coding scheme, both the subjective and the quantitative assessments for the threshold coding scheme show that varying the sub-image size does have an effect on the distortion achieved at a given bit-rate. However, the effect is not as great as it was with the zonal coding scheme. The photographs of the relevant Figures show the effect of varying sub-image size in the fact that the distortion can be more easily seen with the smaller sub-image sizes. The graphs show this fact as the curves become closer to the origin, the larger the sub-image is.

What can be seen more clearly from the subjective assessment, again, is the form with which the distortion manifests itself in the reconstructed images. The "snow-like" effect, that was apparent with the zonal scheme, is not particularly evident with the threshold scheme. However, the limiting effect that took place with the zonal coder due to the appearance of sub-image edges ("blocking") is found to occur also with the threshold coder. Even though a comparison between the two subjective results shows that the threshold coder has a superior performance on a pixel-by-pixel basis, this limiting effect prevents coding at an average rate below 1.Ø bits/pixel, this time, however, for the 8x8 as well as the 16x16 and 32x32 pixels square sub-image sizes.

With the 4x4 pixels square sub-image size, acceptability of reproduction occurs at higher average rates of 1.9 bits/pixel or above. The appearance of sub-image edges is due to the fact that the threshold is, at this stage, becoming quite high and thus starts to result in the more important coefficients being set to zero. This, in turn, results in unfaithful sub-image reproductions, causing discontinuities at the sub-image boundaries and hence the "blocking" effect. On the "sub-image level" the threshold coding scheme is seen to retain textures, patterns and features better than the zonal scheme at low average bit-rates. It is not so evident, subjectively, what the effect of using a different number of quantization levels has had on the reconstructed images, i.e. differences between the appropriate images of Figures 4.3.2.2 and 4.3.2.3 with Figure 4.3.2.4 are not apparent, indeed they appear almost identical. To determine the effect, the quantitative assessment is referred to.

From the graphs (Figures 4.3.2.5 and 4.3.2.6) it can be seen that a trade off occurs between :-

(a) having a relatively higher threshold in conjunction with a 255 level quantizer, and,

(b) having a lower threshold in conjunction with a 127 level quantizer,

to achieve the same average bit-rate. As the subjective assessment does not show clearly any preference, the choice of (a) or (b) is made through the use of the quantitative assessment i.e. which of the two specifications results in the lowest values of distortion. Comparison of the graphs gives the decision in favour of (b). This seems intuitively right as the higher threshold may tend to set to zero more coefficients than coarser quantization would. However, this would not be the case should the quantization be extremely coarse - as the results when using zonal coding would seem to suggest. The graphs also verify the subjective assessment in that, when using sub-image sizes of 32x32 or 16x16 pixels square, the difference in the amount of distortion that occurs at a given average bit-rate is not very great. The graphs consist of smooth curves with points lying either on, or very close to, the curves, indicating a fairly rigorous relationship between the coding technique and the amount of distortion produced by it. Further graphs (Figures 4.3.2.7 and 4.3.2.8) show that an approximately linear relationship can be assumed between the amount of distortion introduced into the reconstructed image and the threshold level, over the range indicated. When written in the standard straight line equation form, this relationship can be written as :-

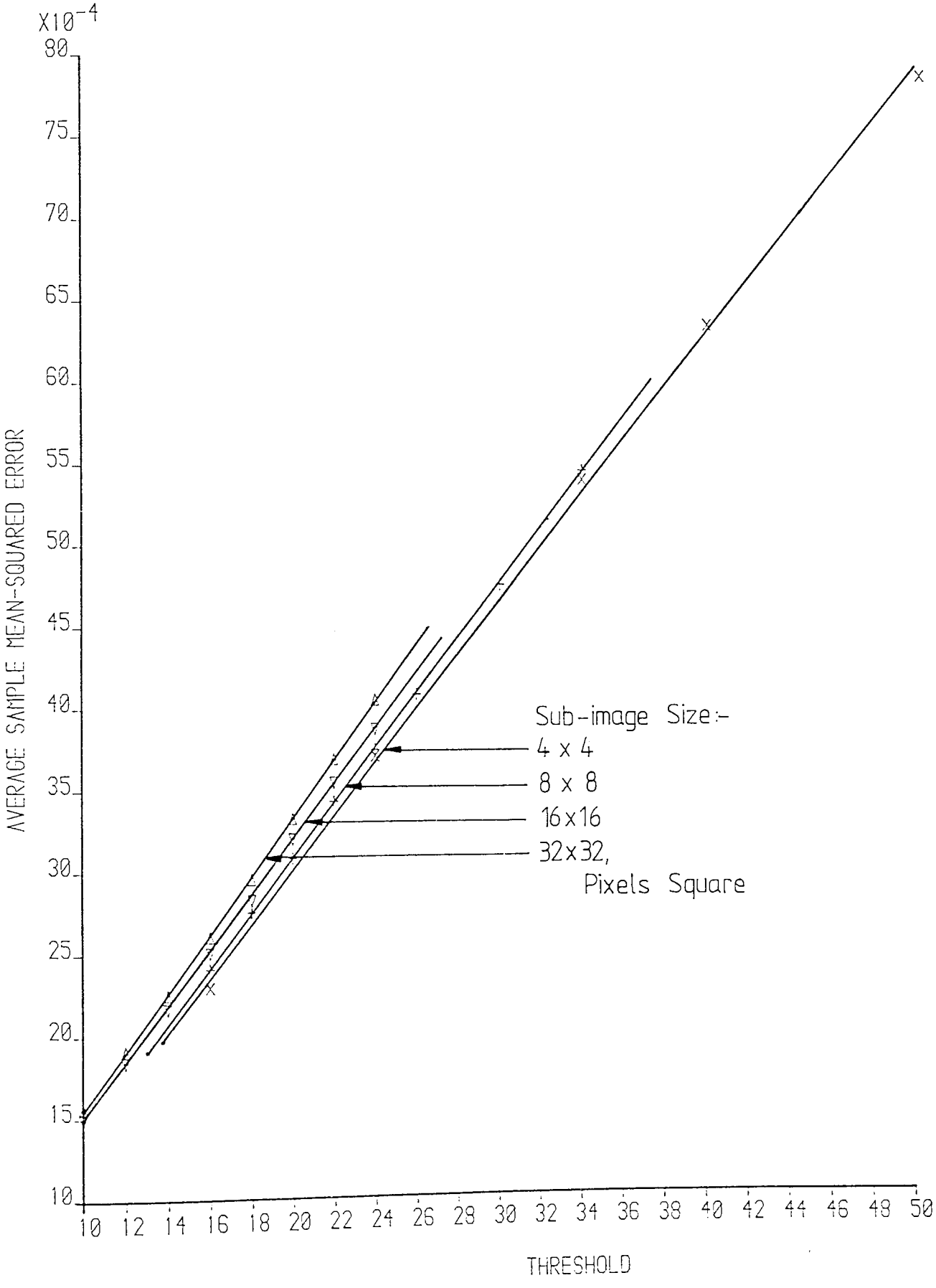


Figure 4.3.2.7 Distortion in Reconstructed Image versus Threshold, for 127 Quantization Levels

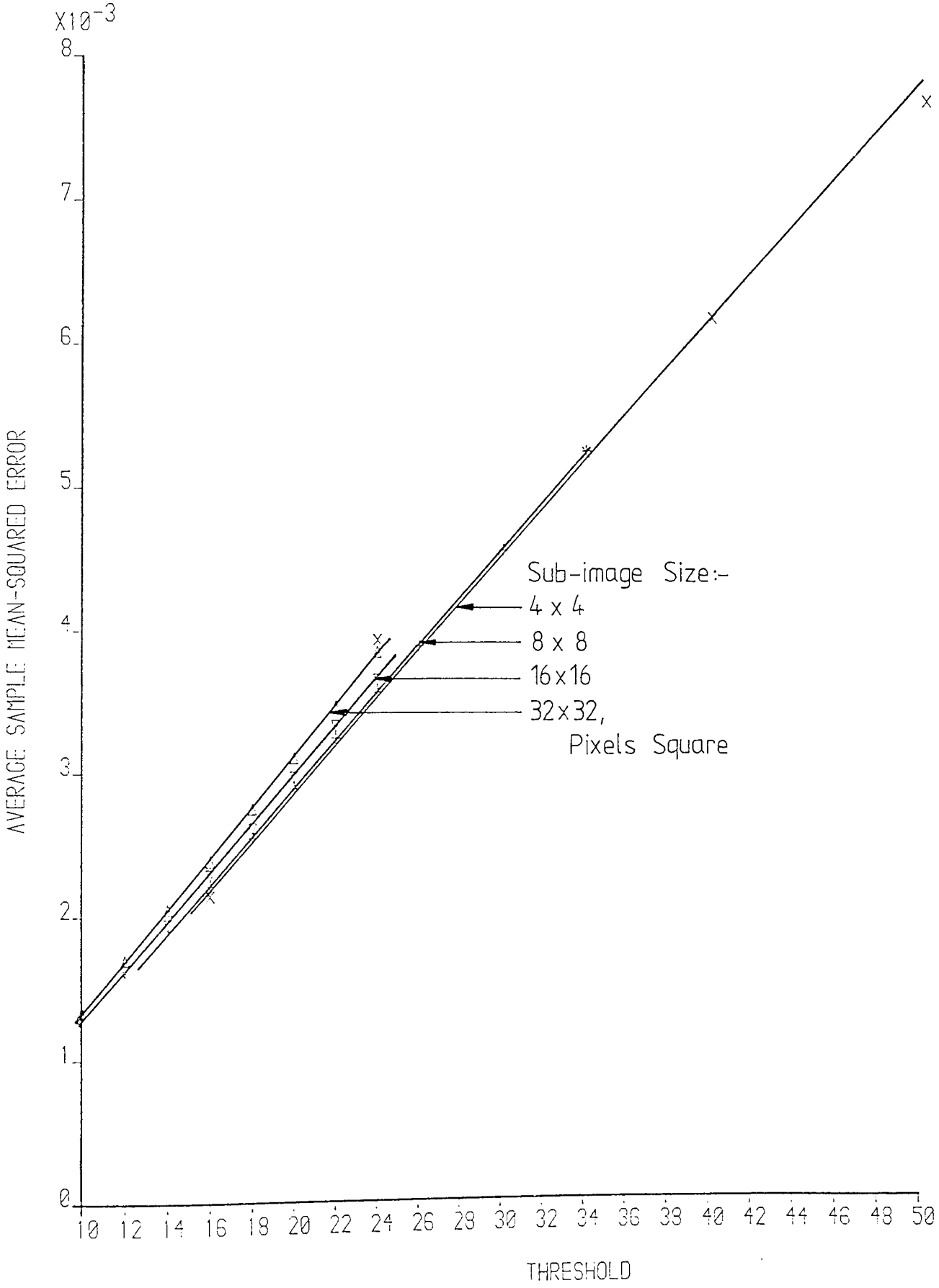


Figure 4.3.2.8 Distortion in Reconstructed Image versus Threshold, for 255 Quantization Levels

$$D = mt + c \qquad (4.3.2.1)$$

where,

D is the normalized, average sample mean square distortion

t is the threshold level

m and c are the constants corresponding to the gradient and D axis intercept respectively.

Values of m and c, as obtained from Figures 4.3.2.7 and 4.3.2.8, indicated the ranges :-

$$7.006 * 10^{-5} \leq m \leq 1.703 * 10^{-4}, \text{ and,}$$

$$-5.806 * 10^{-4} \leq c \leq 8.269 * 10^{-4}$$

The tables for all the graphs plotted are given in the Appendix (Tables A1.4.3.2.1 through A1.4.3.2.8)

Use of Combined Assessments for Setting Encoding Parameters :-

The results show that, when threshold coding, various parameters can be set for coding at a given rate. Through the use of the subjective assessments, an acceptable bit-rate of 1.0 bits/pixel can be set when using the sub-images of size either 16x16 or 32x32 pixels square. The sub-image size of 8x8 pixels square would, perhaps, be classified as just unacceptable. From

the graphs, this is seen to correspond to approximate distortion values of Ø.ØØ3 (32.3 dB), Ø.ØØ3 (32.3 dB) and Ø.ØØ4 (31.Ø dB) which can be termed as the acceptable levels for sub-image sizes 32x32, 16x16, and 8x8 (with reservations) pixels square respectively. For the sub-image size 4x4 pixels square, the acceptable average bit-rate is found to be 1.91 bits/pixel, with an acceptable distortion of Ø.ØØ2 (33.7 dB).

Again, the deduction can be made that, when encoding, without transmission over a channel, larger sub-image sizes are preferable, with differences in distortion/bit-rate becoming negligible for sizes over 16x16 pixels square.

It may have been noticed that a slight controversy has arisen during the assessment of the different types of source encoder. That is that, although the threshold encoding method is reputed [16] to achieve the best compression rates with the least amount of distortion, this does not seem to have occured, in terms of compression, with the schemes here. Although the threshold scheme does achieve lower overall distortion it is seen that the lowest acceptable average bit-rate is identical to that obtained when zonal coding (1.Ø bits/pixel). This was seen to be due to the limiting distortion effect of "blocking". In order to resolve the slight controversy it is necessary to consider what can be termed as "sub-image energy considerations".

4.3.3 Sub-image Energy Considerations.

Sub-image energy can be defined as :-

$$E_s \triangleq \sum_{u=\emptyset}^{N-1} \sum_{v=\emptyset}^{N-1} [\ C(u, v)\ ]^2 - [\ C(\emptyset, \emptyset)\ ]^2 \qquad (\ 4.3.3.1\ )$$

where,

NxN is the sub-image size

(u, v) are the cosine transform domain coordinates

C(u, v) are the transformed coefficients.

This energy measure could have been used to make the zonal encoding scheme adaptive. This was not carried out however, as it would have defeated one of the objectives of this thesis. First, the energy of each sub-image would be measured using equation (4.3.3.1) and this energy could then be plotted against the cumulative probability associated with sub-images over the entire original image. Boundaries could then be drawn, and regions defined, corresponding to different bit-maps. It makes sense to assign the region with the highest energy the largest number of bits and the region with the lowest energy, the smallest number of bits.

Although the threshold coding scheme does not use the above method, as it is already adaptive in nature, the end result is the same i.e. more bits are automatically assigned to sub-images of higher energy. The above, however, serves to show more clearly how the energy of a sub-image can be defined and the part it plays towards making the system adaptive.

A certain conclusion can now be drawn. It is that, should an image be such that when divided into sub-images each one has roughly the same energy then there would not be much advantage gained in making the system adaptive. This is relatively true of the image used throughout most of this thesis. It is actually only the central part, and the part that contains the most information at that, of an image that was originally 512x512 pixels square. Many of the regions which would correspond to low energy regions, and thus more redundant, have been lost and hence the lower compression ratio, usually associated with threshold coding is not expected in this case. This answers the controversy that arose in the previous section, but just to prove the point a different image, consisting of obvious regions of high energy and low energy, is taken and subject to both encoding schemes (zonal - non-adaptive and threshold - adaptive in nature) in order to achieve a low average bit-rate of Ø.4 bits/pixel. The 512x512 pixels square original image, zonal coded reconstruction and threshold coded reconstruction, using sub-image size 16x16 pixels

square, are shown in Figure 4.3.3.1 (a), (b), and (c) respectively. The inability of the non-adaptive zonal scheme to cope at such a low rate is apparent.

Before introducing the topic of channel errors a summary/comparison can be drawn on the performance of the source encoding schemes in the absence of transmission.

(a)

(b)

(c)

Figure 4.3.3.1 (a) Original 512x512 Image (b) Ø.4 Bits/pixel, Non-adaptive Zonal Coded Reconstruction (c) Ø.4 Bits/pixel, Threshold Coded Reconstruction.

4.3.4 Summary/Comparison of Distortion Introduced by Different Schemes.

The summary/comparison is, perhaps, better given by a table listing the various properties and results :-

Zonal Scheme(non-adaptive)	Threshold Scheme(adaptive by nature)
(a) Less complex to implement than the threshold scheme, with less "overhead" transmission bits e.g. when quantizing only the bit-map formula is required (the bit map itself need not be transmitted).	More complex to implement than the zonal scheme, with a "pre-run" necessary and more "overhead" information e.g.Huffman code tables.
(b)The distortion introduced is related to the bit map (and in this way the variable quantizer).	The distortion introduced is related to the threshold and the accuracy of quantization (and the trade-off between the two should be considered -in this case a relatively lower threshold, with 127 level quantization is opted for).

Zonal Scheme(non-adaptive)	Threshold Scheme(adaptive by nature)
(c)The source coding errors are distributed over the entire sub-image, giving the overall error-spectrum of the image itself of relatively uniform distribution i.e. as apposed to tight clusters of pixels in error, for acceptable compression ratios	As for the zonal scheme.
(d)At acceptable bit-rates, the main manifestation of the distortion is "snow-like" in nature. Although this form of distortion is numerically of a high level (compared to the threshold scheme distortion) it's appearance is more pleasing than the other form of distortion(see (e)) which occurs at lower bit-rates.	At acceptable bit-rates, the main manifestation of the distortion is a slight "smoothing" of the image.

Zonal Scheme(non-adaptive)	Threshold Scheme(adaptive by nature)
(e)At unacceptable, relatively low, bit-rates the limiting distortion effect is "blocking".	"Blocking" is also found to be the limiting factor.
(f)Varying the sub-image size, particularly less than 16x16 pixels square has a marked effect on distortion at a given bit-rate.	As zonal scheme, but less pronounced.
(g)Pixel-by-pixel, worse than threshold scheme.	Pixel-by-pixel, better than the zonal scheme.
(h)In the absence of transmission, larger sub-image sizes e.g.16x16 or 32x32 are opted for	As zonal scheme.
(i)Generally results in higher compression ratios and distortion than the threshold scheme.	Generally lower compression ratios and distortion than the zonal scheme, but is dependent on sub-image energy considerations.

The general, overall, subjective/quantitative result is that, as expected, the threshold scheme has a better performance, in the absence of channel errors, than the zonal scheme considered. Figure 4.3.4.1, a graph of normalized distortion versus sub-image size at a fixed bit-rate (in this case 1.Ø bits/pixel), shows this clearly in a quantitative comparison between the zonal and the threshold schemes.

4.4 Performance of Unprotected Source Coded Data in the Presence of Channel Errors.

In order to determine the amount of error protection, and the type of error protection, that is necessary if the data bit-stream is to be subjected to channel errors, it is helpful to observe the performance when no protection is provided. Intuition indicates that, if the source coding schemes are optimum/efficient (in a redundancy sense) then, the information to be transmitted would be that most vital and, should it be corrupted in any way, reconstructions may result in the total loss of the image. However, it is known that some redundancy still exists when using the schemes outlined previously, for example, with the zonal scheme, the dependency between sub-images is not used. This

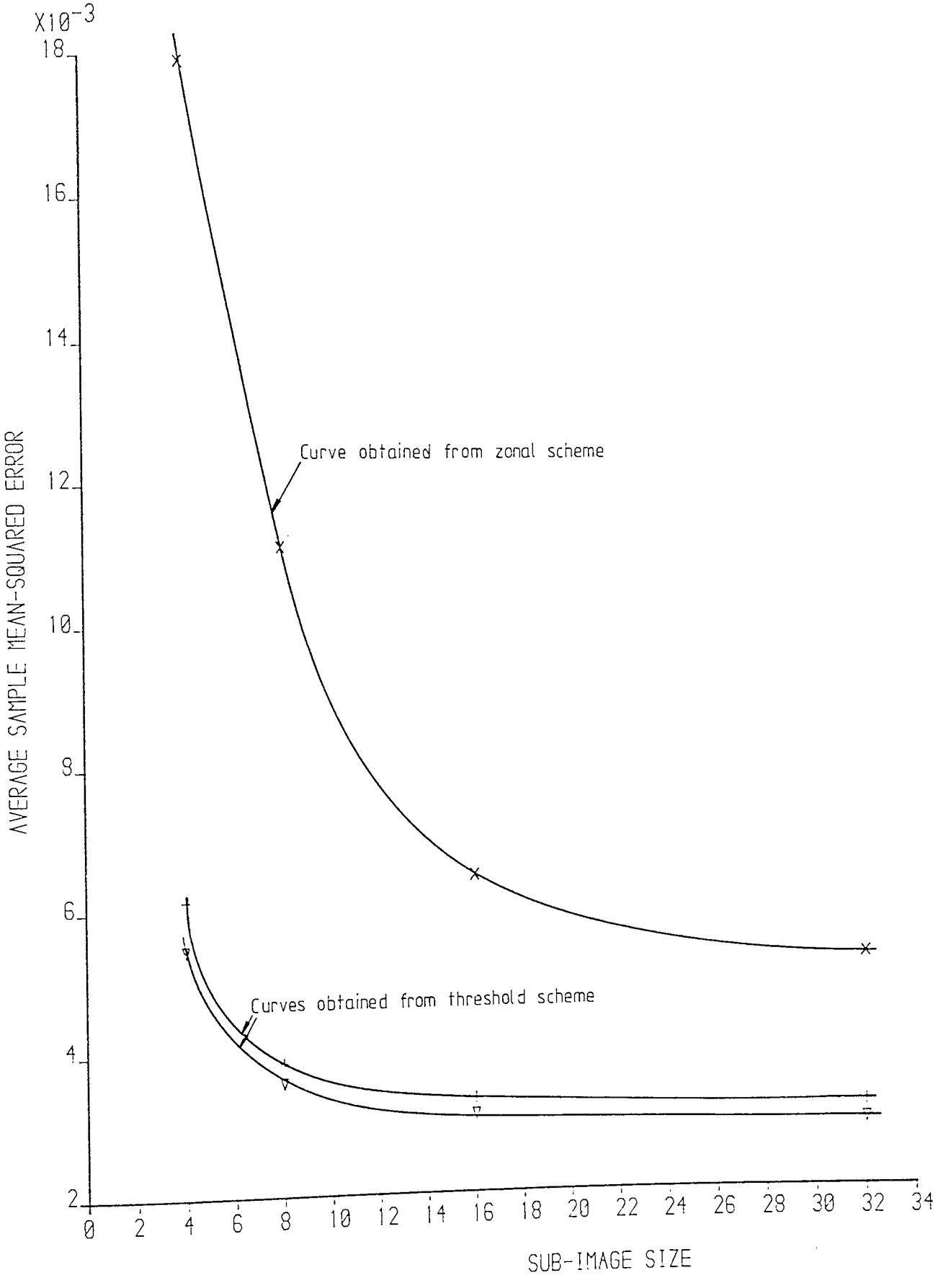


Figure 4.3.4.1 Comparison of Zonal and Threshold Coding at 1·0 Bits/pixel

redundancy may prove useful when considering channel errors, perhaps reducing the amount of extra redundancy that may have to be introduced to ensure the system can cope with various channel error rates. It remains to be seen if this is the case.

4.4.1 Random Error Simulation

As there are many types of channels, it is not the intention to simulate any one of them in particular. Instead, a method is used which introduces errors into the bit-stream in a uniform, independent and random fashion. This simply involves the use of a random number generator algorithm, which adheres to the following conditions :-

$$\text{(a) } P(e_i, e_j) = P(e_i).P(e_j) \qquad (4.4.1.1)$$

where,

e_i, e_j are errors in locations i, j

$P(e_i)$, $P(e_j)$ are the probabilities with which these errors occur.

$$\text{(b) } P(e_i) = P(e_j) = \ldots \quad \ldots = P(e) \qquad (4.4.1.2)$$

(P(e) is thus the error rate)

(c) The random number sequence, which produces the error simulation does not repeat itself.

Should these three conditions hold then it is reasonable to say that errors can be introduced which are independent, random and have a uniform probability distribution. The random number generator, used for the work in this thesis, satisfies the three conditions with reasonable accuracy. It's actual method of producing the random numbers, which, in turn, are used to create the channel errors, is not considered important in this work. Suffice it to say that any generator that adheres reasonably well to the three given conditions should produce similar results to those obtained in this thesis.

The method of simulating channel errors from the random numbers is now outlined :-

All random numbers produced lie between the values zero and one. These values form the extremes of the probability line shown in Figure 4.4.1.1

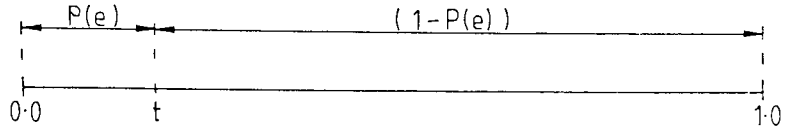


Figure 4.4.1.1 Probability Line

The error rate required (P(e)) is set at the required value, forming a threshold, t, as shown in this Figure. The random number generator is set running, and produces one random number for each data bit of the coded bit-stream. Should the random number produced lie above t (i.e. to the right in the diagram), the corresponding bit is left unchanged - if, however, the number lies below t (i.e. to the left) the corresponding data bit is changed (and, in this way, an error is created). The channel error simulation is complete once all the data bits in the bit-stream have been subjected to this procedure.

4.4.2 Expected Failure of Source Coding Schemes.

Due to the processes involved when source coding/decoding, and the high dependency of the schemes on subjective evaluation of the reconstructed images, it is not really possible to come up with a meaningful value depicting the probable failure of the schemes due to channel errors. The channel error-rate will be the probability associated with any particular bit being in error, but it is not yet known which bits are the most important and how

sensitive the reconstructions are, with regard to the various transformed components.

How difficult the problem, of coming up with a value of probable failure is, can be seen from the outline of defining in what way the schemes can be considered as "having failed". i.e. Failure of the source coding schemes, due to channel errors, can be said to occur when an error in the bit-stream makes a particular sub-image :-

(a) change overall brightness
(b) change it's pattern and/or features } to such an extent that the sub-image stands out disturbingly, from a subjective point of view.

Difficulty arises when trying to assign a value on what is meant by "stands out disturbingly" and the above two considerations suggest that changes can occur in the sub-image, due to channel errors, altering the brightness, patterns or features completely - but due to the nature of the surrounding sub-images (e.g. perhaps they are of a high energy content) the changed sub-image may not stand out at all.

As mentioned in Chapter 2, some work has been done assuming either :-

(a) the most significant bits of all remaining transformed components ought to be protected, or,

(b) only the d.c.and the first few a.c. components need be protected

However, the suggestions are not conclusive and it is one of the aims of this thesis to produce more conclusive results. This is initiated in the next section :-

4.4.3 Errors in Various Components/Preliminary Testing.

By deliberate "forcing" of errors into certain positions, it is proposed to determine which transformed components, and which bits of the corresponding words, are more sensitive to channel errors. Rigorous rules are not expected, but it is hoped that generalized areas of sensitivity are found. Due to the nature of the testing, the results are all obtained by subjective viewing.

4.4.3.1 Single Errors in Zonal Coded Sub-images.

A fixed sub-image size of 16x16 pixels square is used for all these results. In order to make explanation easier, Figure 4.4.3.1.1 - the sub-image terminology map, is used, when considering the zonal coding scheme.

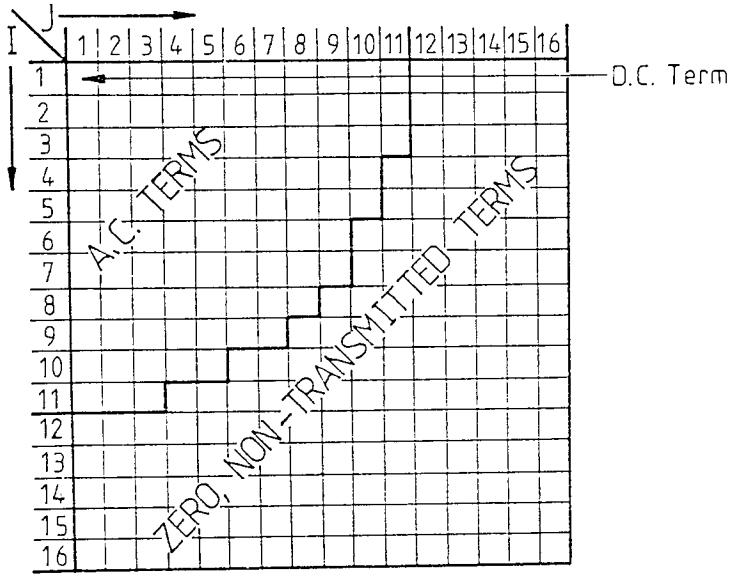


Figure 4.4.3.1.1 Zonal Sub-image Terminology Map.

The positions/terms assigned the same number of bits are assumed to carry the same amount of importance. Similar error effects are thus expected from the terms within each zone i.e. terms having the same number of assigned bits. Hence only the terms in the positions given below are cosidered, covering the term with the maximum number of assigned bits, to the term with only 1 bit assigned to it :-

(1, 1) - position of the d.c. term

(1, 2) - " " " first a.c. term

(1, 3) - " " " second a.c. term

(1, 4) - " " " third a.c. term

.

.

.

.

.

.

(1, n) - final a.c term position.

n = The position after which zero bits are assigned to remaining coefficients (in Figure 4.4.3.1.1, n is shown as 11, which is the value obtained when coding at an average bit-rate of 1.Ø bits/pixel).

The d.c. term is the first term considered. Errors are forced into each position of the word corresponding to the d.c. term (in turn so that only one bit is ever in error). The effects on various sub-images are observed :-

Term (1, 1) - the d.c. term.(when zonal coding)

The d.c. term affects the overall brightness of a sub-image. It is thus not surprising that the introduction of an error, particularly in the more significant bits, of the d.c. code word, results in a reconstruction that is displeasing to the eye. To generalize, it is found that an error in any but the last one or two bits of the nine-bit word, results in the sub-image noticeably in error, completely out of character with it's neighbours. Often the features of the sub-image, although still present, are less distinguishable due to the altered brightness. A typical result, obtained from a particular sub-image, is now given :-

A rough sketch of the particular sub-image is shown in Figure 4.4.3.1.2

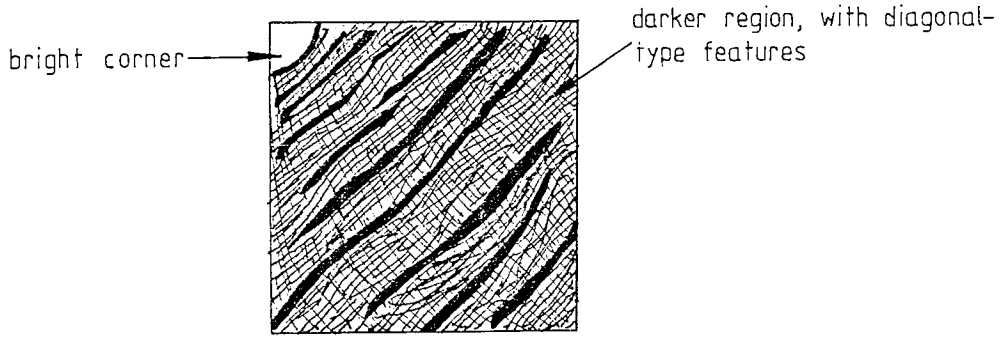
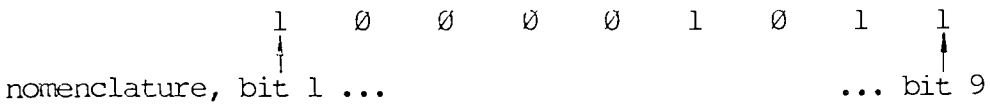


Figure 4.4.3.1.2 A Typical Sub-image

The d.c. word, after source coding to an average rate of 1.Ø bits/pixel is found to be :-



Bit 1 is the most significant bit. The results are :-

D(i) Error in bit 1 only

The brightness of the entire sub-image is affected - it is too bright compared to it's neighbours. The features of the sub-image are, apparently, unaffected.

D(ii) Error in bit 2 only

The reconstruction is even brighter than in the previous result. The features are less distinguishable, due to the brightness.

D(iii) Error in bit 3 only

The brightness of the sub-image is almost the same as it's neighbours, but is sufficiently different to make sub-image boundaries noticeable. Features are apparently unaffected.

D(iv) Error in bit 4 only

Reconstruction is very dark, hence the features are not very clear, although they can be seen upon close scrutiny.

D(v) Error in bit 5 only

A similar result to D(iv), although the reconstruction is not as dark.

D(vi) Error in bit 6 only

Reconstruction still too dark, but becoming brighter. Features apparently unaffected.

D(vii) Error in bit 7 only

Sub-image boundaries are noticeable, though error is no longer so displeasing to the eye.

D(viii) Error in bit 8 only

Sub-image reconstruction is slightly darker than it should be, but this is only noticeable upon close scrutiny.

D(ix) Error in bit 9 only

No noticeable effect.

A similar procedure is carried out with various a.c. terms:-

a.c. terms.(when zonal coding)

In general, it can be said that an error in any position of the words corresponding to the first few a.c. terms, i.e. those with more bits assigned to them, produces undesireable results which range from :-

(a) Reconstructions that are totally unrecogniseable as the original.

(b) Reconstructions that produce boundary effects, similar to blocking, due to alterations either, in the features of the sub-image itself, or, in the brightness of various areas of the sub-image.

Also, it can be said that, an error in any position of the words corresponding to the last few a.c. terms, i.e. those with the least bits assigned to them, produce less disastrous results, ranging from,

(a) Slight blurring of features, giving rise to boundary effects.

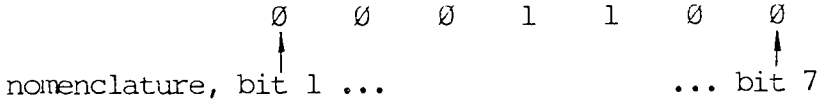
(b) No apparent difference between the reconstruction achieved with the error and that achieved without it.

The last result (in (b) above) is ideal, but it is noted that the a.c. terms under consideration have code words of only, say, 4 bits or less and, also, this result tends to apply only to the least significant bit positions of the word under consideration.

A particular example, taken at the time of testing, to help demonstrate how the importance of the a.c. term varies with it's position in the sub-image, and which also records the effects of the error occuring in a certain position of the word assigned to the particular a.c. term, is now given. The sub-image selected is that sketched previously in Figure 4.4.3.1.2 and the a.c. terms considered are the first, second, fourth, and ninth (for nomenclature refer to Figure 4.4.3.1.1)

First a.c. term.(when zonal coding)

The first a.c. term's word, after source coding to an average of 1.Ø bits/pixel is found to be ¦-



Bit 1 is the most significant bit. The results are ¦-

Al(i) Error in bit 1 only

The sub-image, on reconstruction, is light on one half (the right-half) and seems to be the correct shade on the other. Sub-image boundaries are clearly visible.

Al(ii) Error in bit 2 only

The reconstructed features resemble the "reconstruction without errors" only slightly. The effect is one of vertical bands, with varying degrees of brightness - going from light (on the left-hand side) to dark (on the right-hand side).

Al(iii) Error in bit 3 only

Sub-image boundaries are still visible as the overall shading of the sub-image is slightly incorrect. Features are of a reasonable reconstruction.

Al(iv) Error in bit 4 only, bit 5 only, bit 6 only, and bit 7 only

Each error produces slightly different features that are

often compatible with that obtained when no errors exist in the word. The slight differences can be seen when writing over the original, reconstructed, sub-image.

Second a.c. term.(when zonal coding)

The second a.c. term's word, after source coding to an average rate of 1.Ø bits/pixel is found to be :-

```
                    Ø   Ø   Ø   1   Ø   1   1
                    ↑                       ↑
nomenclature, bit 1 ...                 ... bit 7
```

Bit 1 is the most significant bit. The results are :-

A2(i) Error in bit 1 only

The original features of the sub-image can still be noticed although they have been changed slightly. The brightness of the sub-image is affected, the central region containing a bright vertical band (almost of the correct brightness) with the left and right-hand boundaries much darker than they should be.

A2(ii) Error in bit 2 only

The sub-image is totally unrecogniseable, with the features no longer present. There is a central, vertical band of darkness in the middle of the sub-image with bright bands to either side of it.

A2(iii) Error in bit 3 only

A better reproduction of the features is obtained on the left and right-hand side of the sub-image. There is still a vertical band of darkness in the central region and boundaries are visible.

A2(iv) Error in bit 4 only

Sub-image boundaries are still visible and the reconstruction appears slightly darker than it should be. Features appear unaffected.

A2(vi) Error in bit 5 only, bit 6 only, and bit 7 only

The same result as that obtained in A1(iv) is achieved.

Fourth a.c. term.(when zonal coding)

The fourth a.c. term's word, after source coding to an average of 1.Ø bits/pixel is found to be :-

```
                    Ø   Ø   Ø   1   Ø   1
                    ↑                   ↑
nomenclature, bit 1 ...             ... bit 6
```

Bit 1 is the most significant bit. The results are :-

A4(i) Error in bit 1 only

The features of the sub-image are altered, although, the result still 'fits-in' with neighbouring sub-images. Sub-image boundaries are visible due to the difference intro-

duced.

A4(ii) Error in bit 2 only

The sub-image reconstruction is totally unrecogniseable. It consists of vertical bands of alternating bright and dark shading.

A4(iii) Error in bit 3 only

Sub-image features are different, but again 'fits-in' with the surrounding sub-images. The overall brightness of the sub-image is incorrect but the boundaries are only just visible.

A4(iv) Error in bit 4 only, bit 5 only, and bit 6 only

The same result as that obtained in A1(iv) is achieved.

Ninth a.c. term.(when zonal coding)

The ninth a.c. term's word, after source coding to an average rate of 1.Ø bits/pixel is found to be :-

```
                    Ø   Ø   1   1
                    ↑           ↑
nomenclature, bit 1 ...   ... bit 4
```

Bit 1 is the most significant bit. The results are :-

A9(i) Error in bit 1 only

Features are changed, though not drastically. They are more blurred and boundary effects are only slight.

A9(ii) Error in bit 2 only

A similar result to that obtained in A9(i) is achieved.

A9(iii) Error in bit 3 only and bit 4 only

No noticeable effects result from single errors introduced in these positions.

4.4.3.2 Single Errors in Threshold Coded Sub-images.

The most important word of the threshold scheme, in it's present form is the end-of-sub-image (E.O.S.) word. To recall, this word indicates when all the bits of one sub-image have been processed, thus defining where the next sub-image starts. Preliminary testing shows that single errors in the bit-stream could cause the "apparent" deletion or insertion of an E.O.S. word. Either effect produces disastrous results :-

E.O.S. "Apparent" Deletion, Due to a Single Channel Error.

This effect can take place when a single error appears in an E.O.S. word itself. A simple example shows what may take place :-

Taking the Huffman code words to be x_1, x_2, x_3, and x_4 (which is also the E.O.S. word) as shown in the following table.

Message	Code Word	
x_1	Ø	Huffman code table (A)
x_2	1Ø	
x_3	11Ø	
x_4 (E.O.S.)	111	

and, considering an image consisting of four sub-images, coded into the bit stream :-

coded, sub-image: 1 2 3 4

Ø 111	1Ø Ø 111	Ø 1Ø 1Ø 111	Ø 1Ø 111
x_1 EOS	x_2 x_1 EOS	x_1 x_2 x_2 EOS	x_1 x_2 EOS

If the first E.O.S. word (111) has a single error in the second bit, the bit-stream becomes :-

Ø1Ø11ØØ111Ø1Ø1Ø111Ø1Ø111

which is thus decoded, according to the Huffman code table (A) as,

decoded as

sub-image: 1 2 3

Ø 1Ø 11Ø Ø 111	Ø 1Ø 1Ø 111	Ø 1Ø 111
x_1 x_2 x_3 x_1 EOS	x_1 x_2 x_2 EOS	x_1 x_2 EOS

Hence the error causes :-

(a) The first sub-image to contain too many bits, resulting in additional coefficients decoded as belonging to the first sub-image. Often the original features of this sub-image are still distinguishable, although an additional pattern is superimposed onto it.

(b) Subsequent sub-images are decoded into the wrong positions, i.e. sub-image 3 is now in position 2, and sub-image 4 is now in position 3. This causes a "slicing" of the image.

(c) A reduction in the number of sub-images. In this case a sub-image number 4 is not found, hence the terminology "apparent" E.O.S. deletion.

E.O.S "Apparent" Insertion, Due to a Single Channel Error.

This effect can take place when a single error occurs, corrupting the bit-stream in such a way that an E.O.S. word occurs where originally one did not exist. Again, what may take place is better seen using a simple example. The Huffman code words are

kept the same as in the last example, conforming to the Huffman code table (A).

Considering an image to consist of four sub-images, coded into the following bit-stream :-

coded sub-image	1	2	3	4
	Ø 1Ø 11Ø 1Ø 111	Ø 1Ø 111	Ø 111	1Ø Ø 111
	x x x x EOS	x x EOS	x EOS	x x EOS
	1 2 3 2	1 2	1	2 1

If the second message (x_2) of the first sub-image has it's second bit corrupted, the bit-stream becomes :-

Ø1111Ø1Ø1111Ø1Ø1111Ø1111ØØ111

which is thus decoded, according to the Huffman code table (A) as,

decoded as

sub-image	1	2	3	4	5
	Ø 111	1Ø 1Ø 111	Ø 1Ø 111	Ø 111	1Ø Ø 111
	x EOS	x x EOS	x x EOS	x EOS	x x EOS
	1	2 2	1 2	1	2 1

Hence the error causes :-

(a) A shortening of the number of bits in the first sub-image. The effect on the reconstruction of this sub-image depends on how many coefficients are lost. The more that are lost, the worse the reconstruction is.

(b) The "apparent" insertion of a sub-image, formed from the trailing coefficients lost from the previous sub-image. This "inserted" sub-image is most likely to be totally out of character with it's neighbours due to the way it was formed - hence the terminology "apparent" E.O.S. insertion.

(c) Subsequent sub-images decoded into the wrong positions, only this time one position later i.e. sub-image 2 is now in position 3, sub-image 3 is now in position 4 and sub-image 4 is assigned a position 5 - which often means this last sub-image is ignored. Again "slicing" of the image results, causing a disastrous reconstruction.

The two effects are shown in Figures 4.4.3.2.1 and 4.4.3.2.2

It may be thought that, the word distinguishing zero-run-lengths from coefficients (the z.r.l. 'call' word) also carries a high degree of importance. However the preliminary testing results suggest that, due to the nature of the coding technique, the reconstruction errors caused by this one word being corrupted are often similar to those resulting from the corruption of any other word, apart from the E.O.S. effects described earlier. The explanation for this can be seen later.

Figure 4.4.3.2.1 'Apparent' E.O.S. Deletion

Figure 4.4.3.2.2 'Apparent' E.O.S. Insertion

Because of this, the preliminary testing on the threshold coder, when considering single errors only, now consists of observing the effect of 'forcing' a single error into a particular word, the word being identified simply by it's position in the bit-stream (i.e. 1st, 2nd, 3rd word etc.). Z.r.l. 'call' words are treated in an identical fashion to coefficient code words. The results indicate that only the general case need be presented, with an explanation as to why this is, given after.

Single Error Effects - Other Than E.O.S. Error Effects.

A single error in a word of a particular sub-image can cause effects ranging from :-

(a) Severe alterations leading to sub-image reconstructions which are totally unrecogniseable as to what they should be.

(b) Only slight alterations in features that can be unnoticeable.

The explanation for this is now given.

The problems are caused by the nature of Huffman coding. To recall, part of the threshold coding process involves scanning the data in a zig-zag fashion and Huffman, along with zero-run-length, coding involves identifying the location of a particular coefficient. A single error, even in only one word, can cause

severe reconstruction errors if it results in the loss of the correct locations of correctly decoded coefficients. A simple example shows the effect ¦-

Using the Huffman code table (B) below ¦-

messages	code words	
x 1	Ø	Huffman Code Table (B)
x 2	1Ø	
x 3	11Ø	
x 4	111Ø	
x 5	1111	

Taking the bit-stream to consist of the words shown below

x	x	x	x	x	x
5	5	3	2	1	5

1111 1111 11Ø 1Ø Ø 1111

A single error in the second bit of the first word in the bit-stream causes the following decoding

x	x	x	x	x	x	x
2	5	5	1	2	1	5

1Ø 1111 1111 Ø 1Ø Ø 1111

When reading the words back to the identical positions the result would be ¦-

(a) Incorrect words

(b) Correct words in incorrect locations

(c) An increase in the overall number of words. In other examples, a reduction in the overall number of words can

result.

It is seen that, although the decoding 'settles down', in relation to bit synchronization after a few erroneous coefficients are decoded, the word synchronization is still incorrect. The word synchronization is very important and hence disastrous results can occur if it is lost.

From this, the two types of problems that can result from a single error are now put forward and the general answers found are given.

(i) If word synchronization is not lost, how do errors affect the sub-image reconstruction ?

Here, the results are similar to those obtained with the zonal coder in-as-much-as an error in the first few words seem to cause the more highly undesireable reconstruction effects. However, an error in the last few words of a sub-image can still produce unacceptable results e.g. boundary effects.

(ii) Is loss of word synchronization always fatal ?

The results suggest that in most cases it is. However, the effects are less disturbing if word synchronization is lost only

towards the tail-end of the sub-image, provided sub-image synchronization itself is not lost (see previous sections dealing with errors in E.O.S. word)

It is also the case that, in general, word synchronization is likely to be lost if an error occurs in the bit-stream.

Having gleaned results from single error "forcing", the next stage is to subject the two systems to random, uniform, simulated channel errors, when coding at the lowest achievable rate of 1.Ø bits/pixel and to observe the effects.

4.4.4 Unprotected Source Coded Data Subjected to Uniform Random Errors.

Having determined what single errors can do, the two coding schemes are now subjected to simulated channel errors, at various error rates, and the effects on unprotected code observed. Sub-image sizes of 16x16 and 8x8 pixels square are used, and for comparison the average bit-rate is kept constant at 1.Ø bits/pixel. The error rate is varied from 1 in $1Ø^{6}$ bits in error .

to 1 in $1Ø^{2}$ bits in error. The results, in the form of photographs and error spectra speak for themselves.

4.4.4.1 Zonal Coding - Unprotected Code Subject to Various Channel Error Rates.

Results for error rates at $1Ø^{-6}$ and $1Ø^{-5}$ are not recorded. This is because the channel errror rate is so low that all the bits tend to be received without any errors occurring in the bit-stream. The resulting reconstructions are thus the same as that, without errors, shown in Figures 4.3.1.2(b) (sub-image size 16x16 pixels square) and 4.3.1.3(a) (sub-image size 8x8 pixels square). These error rates could thus indicate the type of rates required if the zonal source coded data is to be transmitted without error protection.

For a fixed sub-image size of 16x16 pixels square, and a fixed average source coding rate of 1.Ø bits/pixel, the results of transmitting zonal source coded data over a channel with an error rate of $1Ø^{-4}$ or greater are indicated in Figure 4.4.4.1.1. As expected, the higher this error rate is, the worse the reconstruction of the image is. It is seen that, a channel error

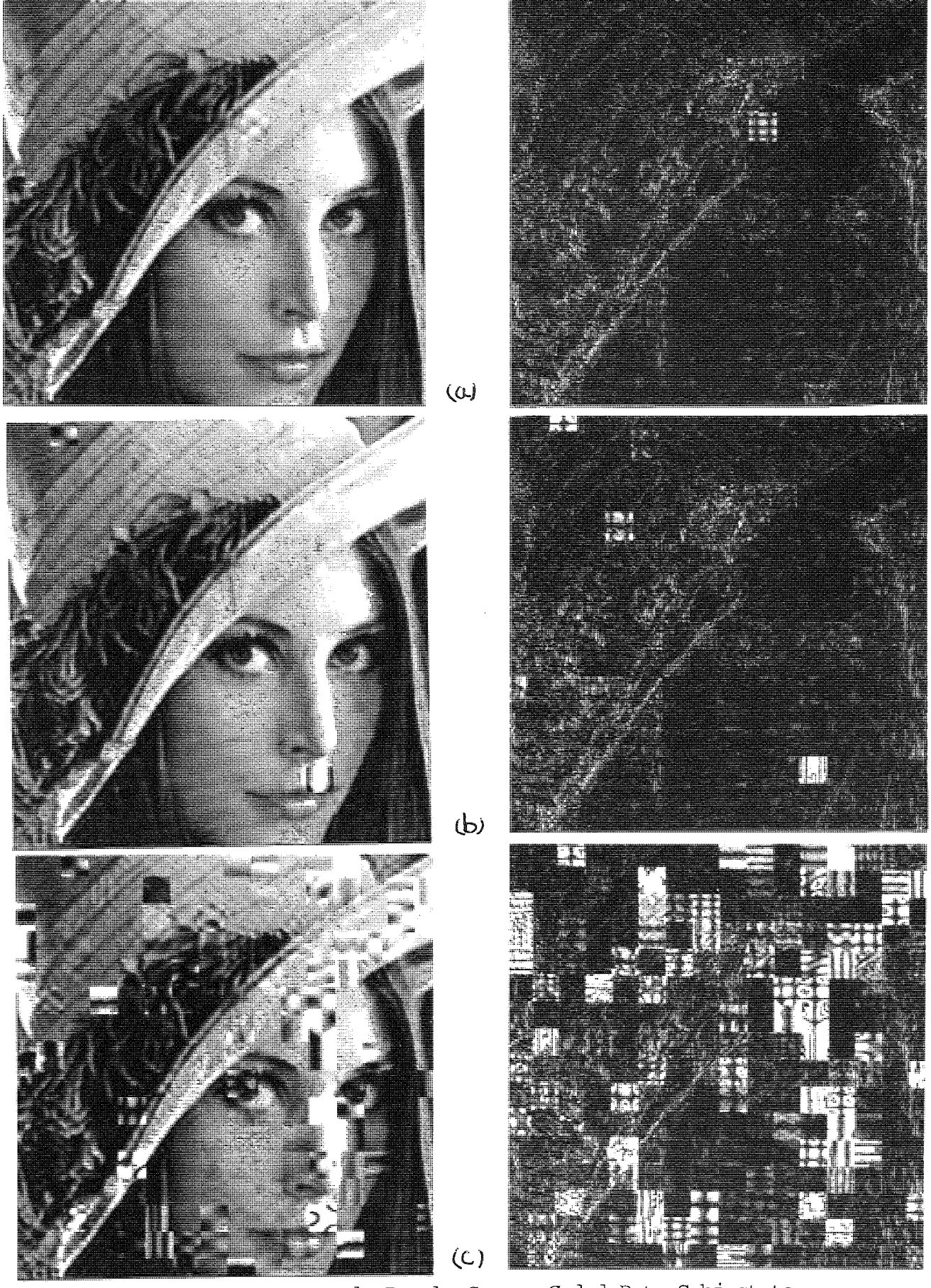
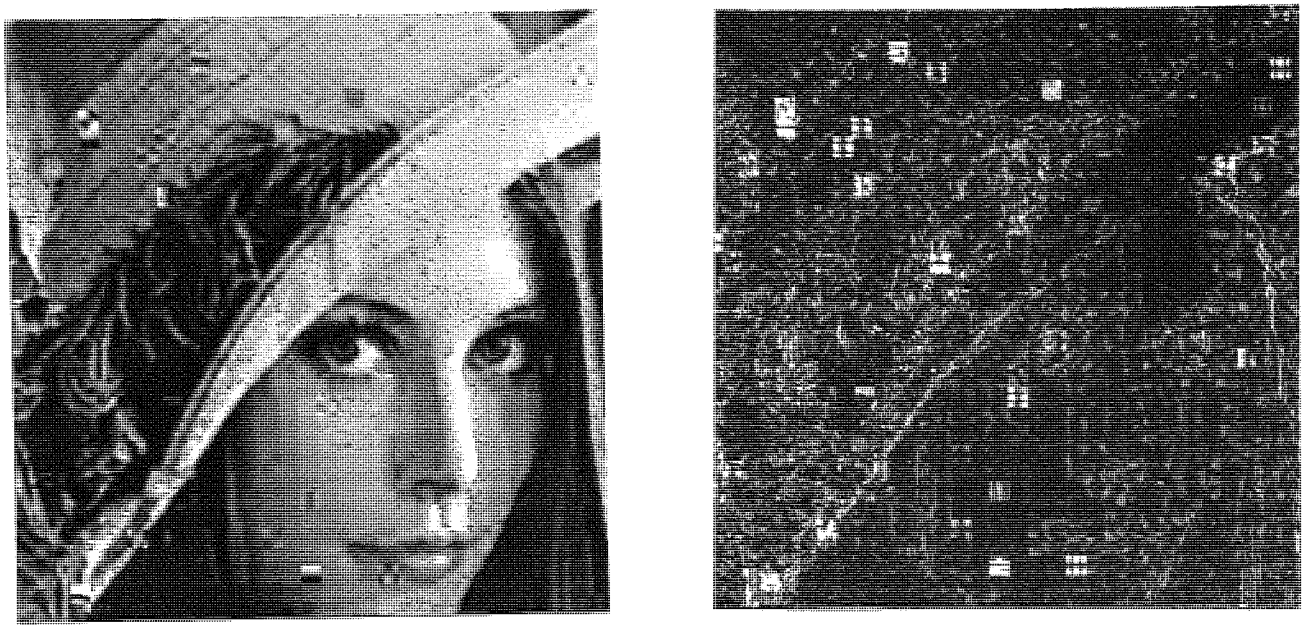


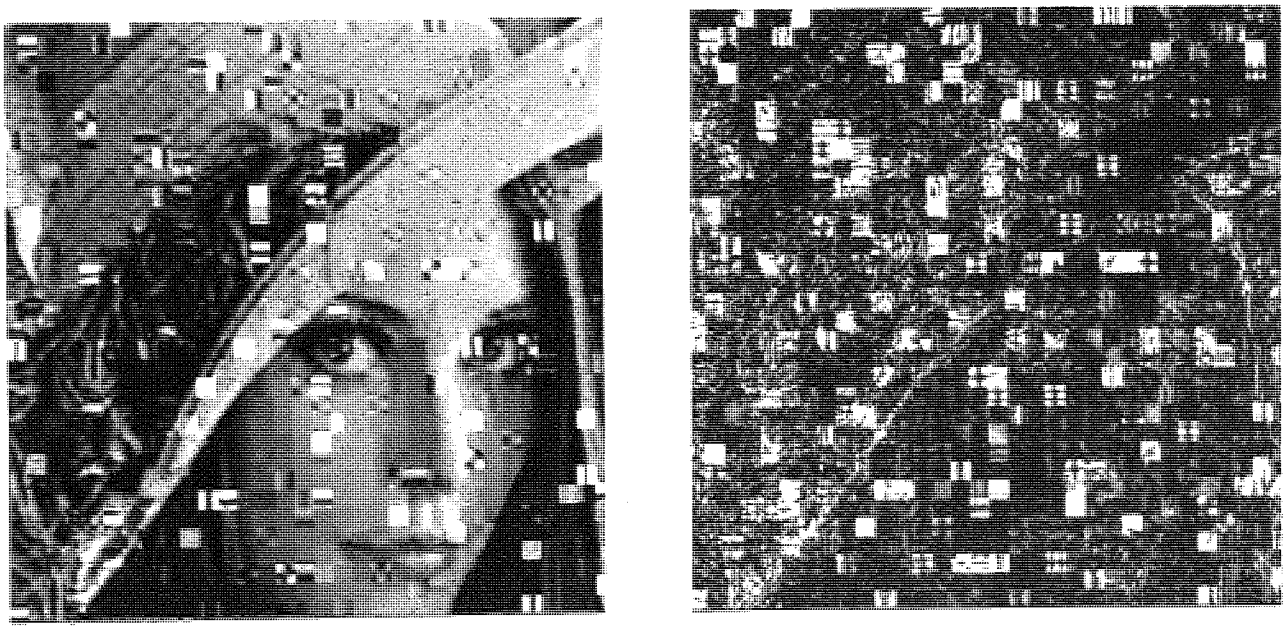
Figure 4.4.4.1.1 Unprotected, Zonal, Source Coded Data Subject to Channel Error Rates of (a) 1 in 10000 (b) 1 in 1000 (c) 1 in 100, for Sub-image Size 16x16

rate of 10^{-4} results in only one or two sub-images noticeably in error, when observing the reconstructed image (a) only - viewing the error spectrum highlights the fact that there are actually more sub-images affected due to channel errors. With a channel error rate of 10^{-3} the reconstruction, (b), shows effects that are very disturbing to the eye - more so when sub-images are wrongly reconstructed in regions of the image to which the viewer would be automatically drawn. With channel error rates of 10^{-2} or greater, (c), the image becomes so severely affected that the original features, even those most prominent, begin to fall into the background, or are obscured completely, and the reconstruction takes on a "patchwork quilt" type outlook. This effect is also obvious when viewing the corresponding error spectrum.

Figure 4.4.4.1.2 indicates the results of transmitting zonal source coded data over a channel with an error rate of 10^{-3} or greater for a fixed sub-image size and average coding rate of 8x8 pixels square and 1.0 bits/pixel respectively. Results, when using a channel error rate of 10^{-4}, were found to produce reconstructed images which did not differ noticeably from that obtained in the absence of channel errors. This may, however, be due to the fact that for the zonal scheme, 1.0 bits/pixel is actually too low and the source coding errors themselves are quite



(a)



(b)

Figure 4.4.4.1.2 Unprotected, Zonal, Source Coded Data Subject to Channel Error Rates of (a) 1 in 1000 (b) 1 in 100, for Sub-image Size 8x8.

substantial. Ignoring the source coding errors, the results are presented for comparison. With a channel error rate of 10^{-3} many sub-images are seen to be wrongly reconstructed, quite obvious from either the reconstructed image or it's error spectrum. Very severe distortion is seen to occur at a channel error rate of 10^{-2} or greater.

Comparison of Figure 4.4.4.1.1 and 4.4.4.1.2 at corresponding channel error rates show that using the smaller sub-images restricts the errors in reconstruction to smaller regions and hence the error effects are, relatively, more sparsely distributed over the entire image. However, on the whole, a greater number of sub-images 'appear' affected when using the smaller sub-images indicating that each particular sub-image is more sensitive to errors than each particular sub-image of a larger size. Both Figures indicate a need for channel error protection.

4.4.4.2 Threshold Coding - Unprotected Code Subject to Various Channel Error Rates.

As with zonal coding, when using channel error rates of 10^{-6} and 10^{-5} all bits in the bit-stream tend to be received without

corruption of any of the bits. However, at any rate above this E.O.S. error effects, described earlier (Section 4.4.3.2), sooner-or-later take place. These effects result in corruption and "slicing" of the reconstructed image making the average sample mean square error exceedingly high and also resulting in very dense/heavy error spectra. Because of this, it is already envisaged that, at a later stage, some form of protection will be provided to ensure that these E.O.S. error effects have a low probability of occurrence. Results are thus obtained ensuring E.O.S. error effects do not occur, thus providing further comparison with the zonal coding scheme.

Figure 4.4.4.2.1 shows typical results obtained when subjecting unprotected, source coded data of the threshold scheme to channels with error rates of 10^{-4} and greater - once again keeping the average bit-rate and sub-image size constant at 1.0 bits/pixel and 16x16 pixels square respectively. With a channel error rate of 10^{-4}, (a), only one or two sub-images are noticeably in error in the reconstructed image, although the error spectrum indicates that there are more. Disturbing effects are achieved when the data is subject to a channel error rate of 10^{-3}, (b),

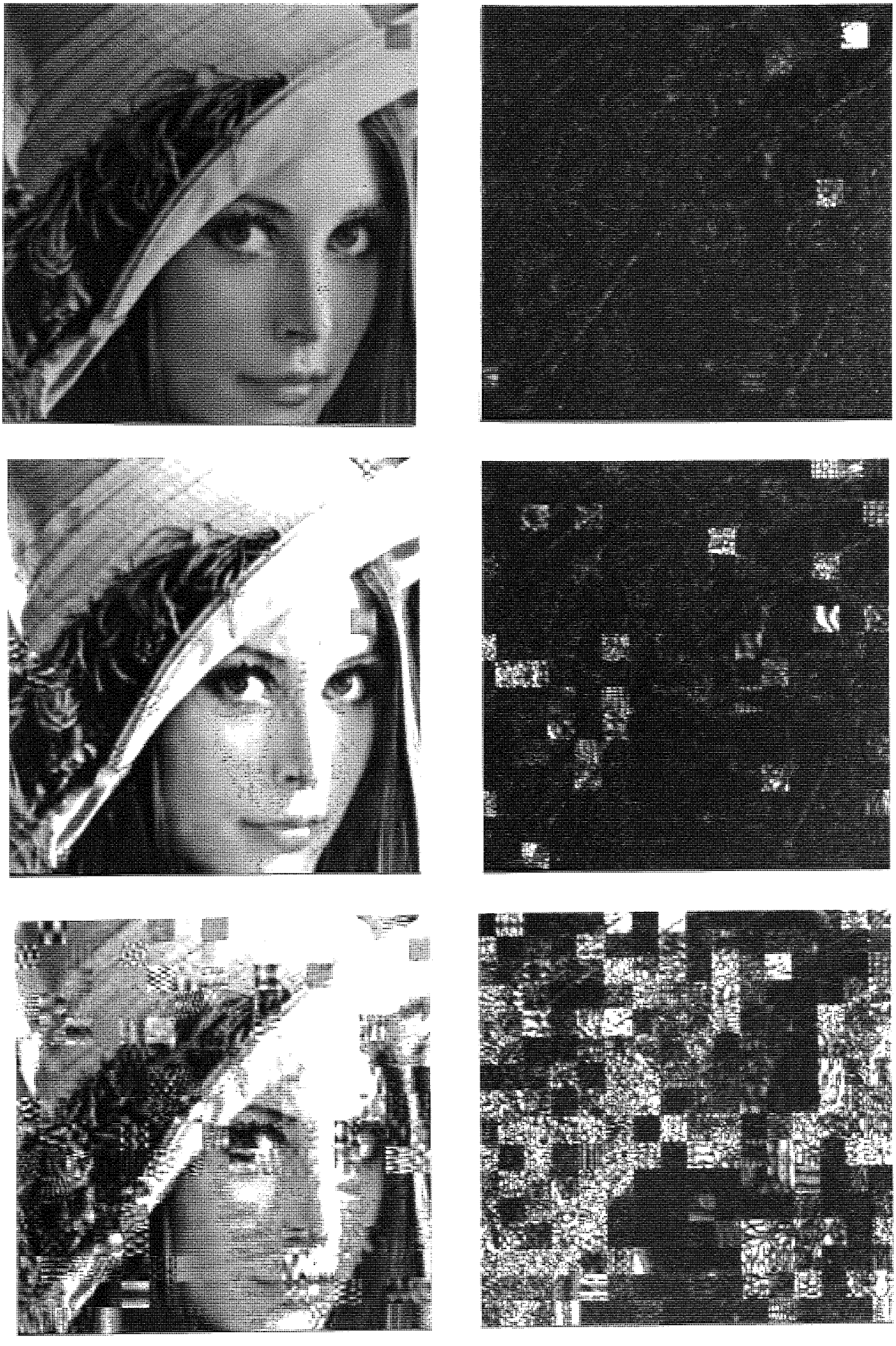
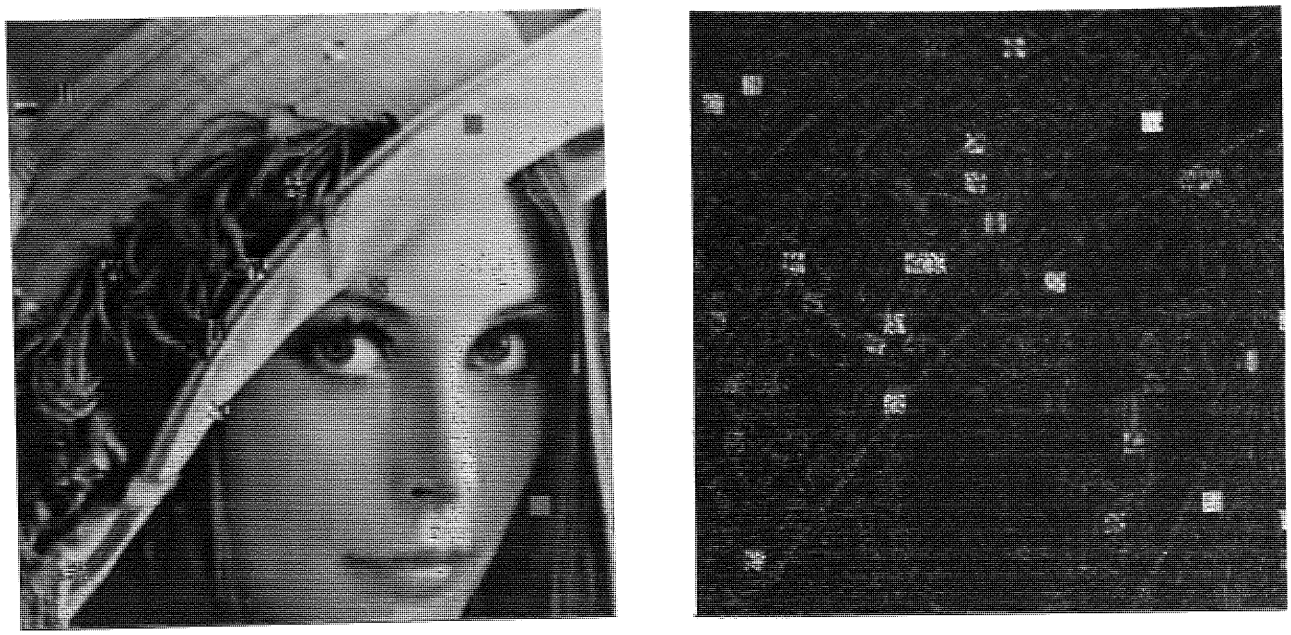


Figure 4.4.4.2.1 Unprotected, Threshold, Source Coded Data Subject to Channel Error Rates of (a) 1 in 10000 (b) 1 in 1000 (c) 1 in 100, for Sub-image Size 16x16

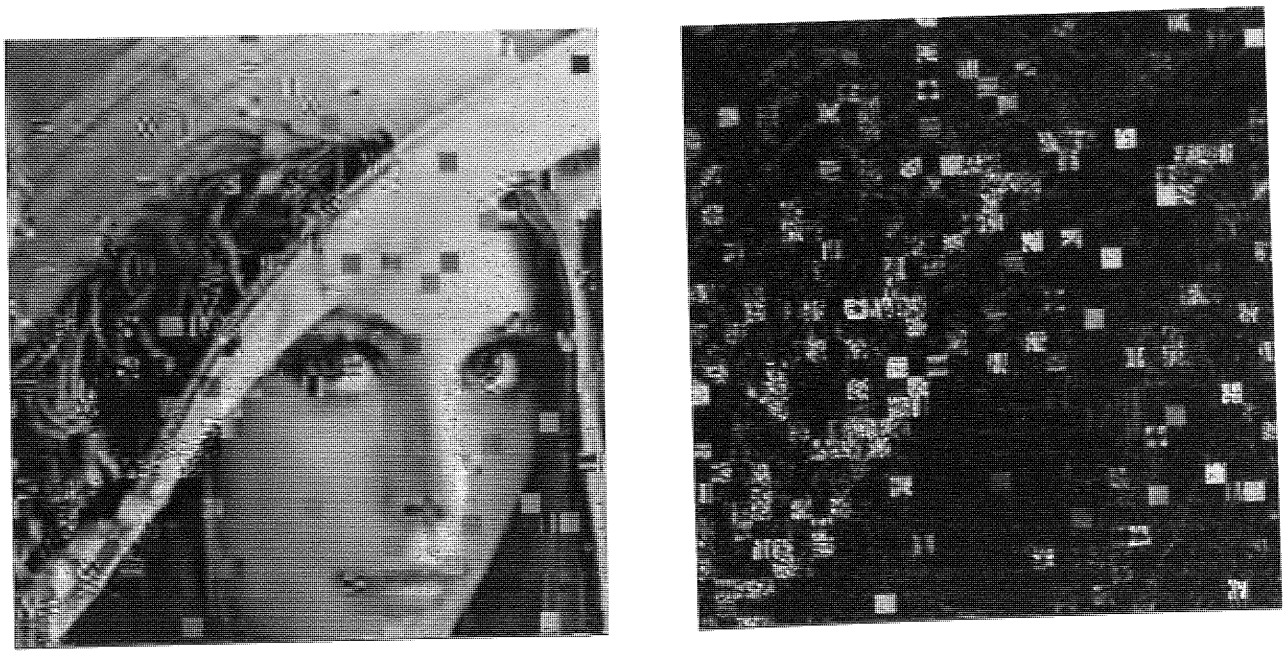
and once again channel error rates of 10^{-2} or greater, (c), force even the most prominent features into the background and the "patchwork quilt" effect appears. In comparison with the zonal coder (Figure 4.4.4.1.1) the effects are as bad, or worse, indicating that what the threshold scheme achieves over the zonal scheme, in terms of lower distortion, appears to wane when channel errors are considered.

The results of transmitting source coded data over a channel with an error rate of 10^{-3} or greater for an average source coding rate of 1.0 bits/pixel and fixed sub-image size of 8x8 pixels square are indicated in Figure 4.4.4.2.2. Like the zonal coder, results when using a channel error rate of 10^{-4} were found to produce reconstructed images with negligible differences to those obtained in the absence of channel errors. However, unlike the zonal coder, 1.0 bits/pixel is more satisfactory in the case of the threshold coder and so this error rate could, perhaps, form the boundary of acceptability when transmitting in the absence of error protection and fixed sub-image size of 8x8 pixels square (giving a slight advantage over the larger sub-image sizes). Channel error rates of 10^{-3}, (a), and 10^{-2}, (b), produce increasingly disturbing reconstructions.

Figure 4.4.4.2.2 Unprotecte
Subject to
(b) 1 in 1
Square.



(a)



(b)

Figure 4.4.4.2.2 Unprotected, Threshold, Source Coded Data Subject to Channel Error Rates of (a) 1 in 1ØØØ (b) 1 in 1ØØ, for Sub-image Size 8x8 Pixels Square.

Like the zonal scheme, the smaller sub-image size restricts the errors in reconstruction to smaller regions resulting in error effects having a relatively, more sparse appearance to that obtained when using the larger sub-image size. However, unlike the zonal scheme, the error effects are also restricted to regions of the image where they are less noticeable, i.e. regions of high energy/activity such as the feathers of the hat and the hair. This is a more pleasing result as these are the regions of the image which the eye does not automatically scrutinize. The result is thought to be due to the adaptivity of the threshold scheme, that is to say, there are more bits assigned to the regions of higher energy/activity hence if errors occur in the bit-stream with equal probability, more are likely to occur in these regions. This effect is also the case with the 16x16 sub-image size although the larger size obscures the effect to a certain extent. When considered in conjunction with the smaller sub-image size, the effect tends this time to favour the threshold scheme.

However, with both the zonal and the threshold scheme, the results indicate the need for channel error protection.

4.5 Summary of Overall Performance including Channel Errors.

The summary/comparison of the overall performance is, again, given in tabular form :-

Zonal Scheme(non-adaptive)	Threshold Scheme(adaptive by nature)
(a)In the absence of channel errors, generally, has a poorer performance(in terms of compression and resulting distortion) than the threshold coder.	In the absence of channel errors, generally, has a better performance than the zonal coder.
(b)Though more robust than the threshold coder, is never-the-less sensitive to channel errors, particularly at error rates higher than $1 \text{ in } 10^5$.	Very sensitive to channel errors, due to the coding processes involved, particularly at error rates higher than $1 \text{ in } 10^5$.
(c)Channel errors in certain coefficients are more likely to give more displeasing	In general, all bits in a data word tend to carry the same amount of importance. Channel errors cannot

Zonal Scheme (non-adaptive)	Threshold Scheme (adaptive by nature)
reconstructions than in others- in general, it would seem that the more sensitive coefficients occur in a radial fashion around the top-left corner of the transformed sub-image. Channel errors in the least significant positions of data words, generally, still give noticeable visual error although these are less disturbing than those resulting from the most significant bits being in error.	be tolerated in any bit position of the sub-image data, unless they occur towards the 'tail-end' of a sub-image. Even then the E.O.S. error effects can occur. The E.O.S. word, in particular, causes the scheme to be very vulnerable to channel errors.
(d)An error in one word of the coded data does not affect the decoding of neighbouring words.	An error in one word can cause incorrect decoding of several following words, hence the increase in sensitivity.
(e)The visual effect of an error, or errors, in a sub-image is distributed over	As zonal scheme, though (due to the inter-frame nature) an error in one sub-image can affect subsequent sub-

Zonal Scheme (non-adaptive)	Threshold Scheme (adaptive by nature)
the entire sub-image. Each sub-image is independent from another.	images, thus the error can affect more of the image than just the local sub-image in which it occurs.
(f) Error effects tend to be evenly distributed, in a random fashion over the whole of the image.	Due to the adaptivity of this coder, error effects appear to be more restricted to regions of higher energy/activity. Thus there is more apparent 'order' to the error effects, particularly with the smaller sub-image size.

From this summary, and the foregoing results, protection schemes are devised to ensure that each system can cope with higher channel error rates. Various trade-offs are then considered.

CHAPTER 5

CHANNEL ERROR PROTECTION SCHEMES.

5.1 Introduction - Coding Theorems.

The source encoding schemes developed thus far have, implicitly, made use of the source coding theorem. It is beneficial to state this theorem explicitly:-

5.1.1 Source Coding Theorem.

Given :-

(i) A discrete memoryless source (d.m.s.) i.e. a source where the successive letters of each word, $\bar{x} = (x_1, x_2, x_3, \ldots x_n)$ are independent identically distributed (i.i.d.) thus conforming to the requirement:

$$P(\bar{x}) = \prod_{t=1}^{n} P(x_t) \qquad (5.1.1.1)$$

where,

$P(x_t)$ is the probability associated with the letter x_t.

The d.m.s. is thus denoted by the symbol $\{ X_t , P \}$.

(ii) A single letter fidelity criterion F_ρ, where F_ρ is a sequence of word distortion measures ¦-

$$F_\rho = \{ \rho_m (\bar{x}, \bar{y}), 1 \leq m < \infty \} \qquad (5.1.1.2)$$

and,

$$\rho_m (\bar{x}, \bar{y}) = \frac{1}{m} \sum_{t=1}^{m} \rho(x_t , y_t) \qquad (5.1.1.3)$$

Let ¦-

(i) $R(\cdot)$ denote the rate-distortion function of $\{ X_t , P \}$ with respect to F_ρ.

(ii) D denote the upper limit of average distortion.

(iii) ϵ be the arbitrarily small error.

Then ¦-

For any $\epsilon > 0$ and any $D \geq 0$, an integer n can be found such that there exists a $(D+\epsilon)$ - admissible code of blocklength n with rate $R < R(D) + \epsilon$. In other words, the inequality [8]

$$n^{-1} \log K(n, D+\epsilon) < R(D)+\epsilon \qquad (5.1.1.4)$$

holds for n sufficiently large,

where,

K is a constant, and,

$(D+\epsilon)$ - admissible implies the code can achieve an average distortion $\leq (D+\epsilon)$.

Noiseless coding is a direct consequence of the source coding theorem special case where the rate-distortion function for zero distortion, $R(\emptyset)$, is equal to the source entropy i.e.

$$R(\emptyset) = H(X) \qquad (5.1.1.5)$$

where,

$H(X)$ is the entropy of the source alphabet $\{ X \}$. With equation (5.1.1.5) holding, then for $\epsilon > \emptyset$ and $R > H(X)$, for sufficiently large n there exists a code of length n with at the most 2^{nR} code words and average distortion $< \epsilon$. In other words, the source can be essentially perfectly represented using only about $H(X)$ bits per sample. Noiseless coding is thus sometimes referred to as entropy coding of which Huffman coding is a typical example.

With a noisy channel, the data encoded through, either implicit or explicit, use of the source coding theorem or noiseless coding assumption can be severely affected, as already seen in the previous chapter. Attention is thus attracted to Shannon's noisy coding theorem or channel coding theorem |-

5.1.2 Noisy Coding Theorem.

Given :-

(i) A discrete memoryless channel (d.m.c.) with capacity, C.

(ii) A d.m.s.

Let :-

(i) The d.m.s. have entropy rate H.

Then :-

If $H \leq C$, where both are measured in the same units per source letter, the output can be encoded for transmission over the channel with an arbitrarily small frequency of errors (or an arbitrarily small equivocation). If $H > C$, then it is possible to encode the source so that the equivocation is less than $H-C+\epsilon$, where ϵ is arbitrarily small, but there is no way of encoding which gives an equivocation less than H-C.

The noisy coding theorem can prove unsatisfactory from a practical point of view if the codes that conform to it suffer from the following three defects :-

(a) They are hard to find - The proof that a certain code should exist, does not mean that it's structuring and method of implementation can be easily deduced.

(b) They are hard to analyze - Given a code one cannot be certain how efficient or effective it is unless it can be analyzed to a certain degree.

(c) They are hard to implement - Often it can be found that what is theoretically simple and logical for the human mind to deduce or set down in writing is not always so easy to put into practice, inparticular decoding procedures.

However, as might be expected, there are some very good codes which are free from these three defects. It is these codes on which attention is focused when error protection codes are to be incorporated into the image communication schemes.

Before looking at any particular combined source/channel error protection scheme the structures and methods of employment of the channel error protection codes and/or decoding algorithms are given :-

5.2 Source Coding Parameter Adjustments.

The first logical procedure, when attempting to make the image coding schemes less sensitive to channel errors, is to see whether adjustments made to the more obvious coding parameters

result in more pleasing image reconstructions, initially, without resorting to additional error protection coding. One parameter that has already been observed in this work, and which is to be extended further in later sections, is the difference between using a non-adaptive scheme to an adaptive one. With this borne in mind, adjustments in other parameters can be observed.

5.2.1 Zonal Coder Parameter Adjustment.

The non-adaptivity of the zonal coder keeps each sub-image independent from the next with regard to channel errors. Errors in one word of the source coded data have been found not to affect the decoding of neighbouring words. This property is desir-able and is a result of the bit-map coding procedure. Hence no modifications to the zonal source coder are made, thus retaining both the non-adaptivity, still required for comparative purposes, and the property of separation of words that are not in error from those that are.

This leaves only two main parameters of the source coding scheme to consider. These two parameters are :-

(i) The total average source coding bit-rate.

(ii) The sub-image size.

A start has already been made with the second of these two parameters in this work, but requires further observation when (i) is altered and not kept constant at 1.Ø bits/pixel, with channel errors present.

5.2.2 Threshold Coder Parameter Adjustment.

It has been seen that the adaptivity of the threshold source coding scheme can be a liability when channel errors are present. This was observed when errors occur in the E.O.S. word. A method is thus required which reduces or eliminates the E.O.S. error effects, hence restricting channel error effects to the local sub-image in which they occur. The following method provides a crude but simple solution.

First, the problems are seen to be caused by the fact that the very important piece of information, the E.O.S. word, is placed at the end of each sub-image. The proposal then, is to eliminate this parameter altogether and introduce sub-image headers, (S.H.), words at the beginning of each sub-image which indicate how many bits belong to each sub-image. The header is also protected by a simple, repetition, channel error protection

method. All headers are of a fixed number of bits, n_b, typically 8. The number of bits in the header limits the number of possible headers. For 8 bits the number of headers is 255 (the "all zeroes" header being excluded as a possibility). There thus appears to be a choice of either; adding extra digits or, to delete the trailing digits of the bits corresponding to a sub-image in order to make the number an acceptable one i.e. corresponding to an "allowed", integer, header value. Preliminary testing was in favour of deleting trailing digits.

Additional processing is required, but is simplistic in nature and requires minimal time. The pre-run is left unaltered and coding is carried out as before, only this time the number of bits assigned to each sub-image is recorded and the E.O.S. word omitted. Also, the minimum and maximum number of bits over all sub-images are stored as b_{min} and b_{max} respectively. Preliminary testing also indicated that a one-to-one, linear, mapping between number of bits and header for number of bits ≤ 17 ensures that coded sub-images consisting of only one word are not truncated, which would be fatal as the remaining bits would be ignored on decoding, thus denying the sub-image's existence. The header value is thus determined by :-

$$h_q = b_q \quad , \quad b_q \leq 16 \qquad (5.2.2.1)$$

$$h_q = \text{AINT}\left[\frac{(n_1 - 17.0).(b_q - b_{min})}{(b_{max} - b_{min})} + 17.0 \right], \qquad 16.0 < b_q \leq b_{max} \qquad (5.2.2.2)$$

where,

h_q = the, integer, header value of the q^{th} sub-image

b_q = the, stored, number of bits of the q^{th} sub-image,

obtained during the initial coding.

$n_1 = 2^{nb} - 1$, the number of "allowed" headers

and AINT implies truncation of the calculated value in the parentheses, [ ].

The number of "allowed" bits, a_q, in the sub-image is then obtained from equation (5.2.2.2), by isolating b_q. Ignoring the AINT, equation (5.2.2.2) would be,

$$h_q = (n_1 - 17.0)\frac{(b_q - b_{min})}{(b_{max} - b_{min})} + 17.0$$

hence,

$$b_q = \frac{(h_q - 17.0)(b_{max} - b_{min})}{(n_1 - 17.0)} + b_{min}$$

thus,

$$a_q = \text{AINT}\left[\frac{(h_q - 17.0)(b_{max} - b_{min})}{(n_1 - 17.0)} + b_{min} \right] \qquad (5.2.2.3)$$

It is found that using the above method

$$a_q \leqslant b_q \qquad (5.2.2.4)$$

Thus the number of bits to delete from the q^{th} sub-image is

$$b_q - a_q \qquad (5.2.2.5)$$

After initial coding, equations (5.2.2.1) and (5.2.2.2) are used to determine the header value and equations (5.2.2.3) and (5.2.2.5) are used to determine the number of bits to delete in the q^{th} sub-image which is then carried out. The header is converted to a binary number and inserted at the beginning of the q^{th} sub-image. To decode, equations (5.2.2.3) and (5.2.2.1) are used. Figure 5.2.2.1 indicates the linear mapping procedure outlined above.

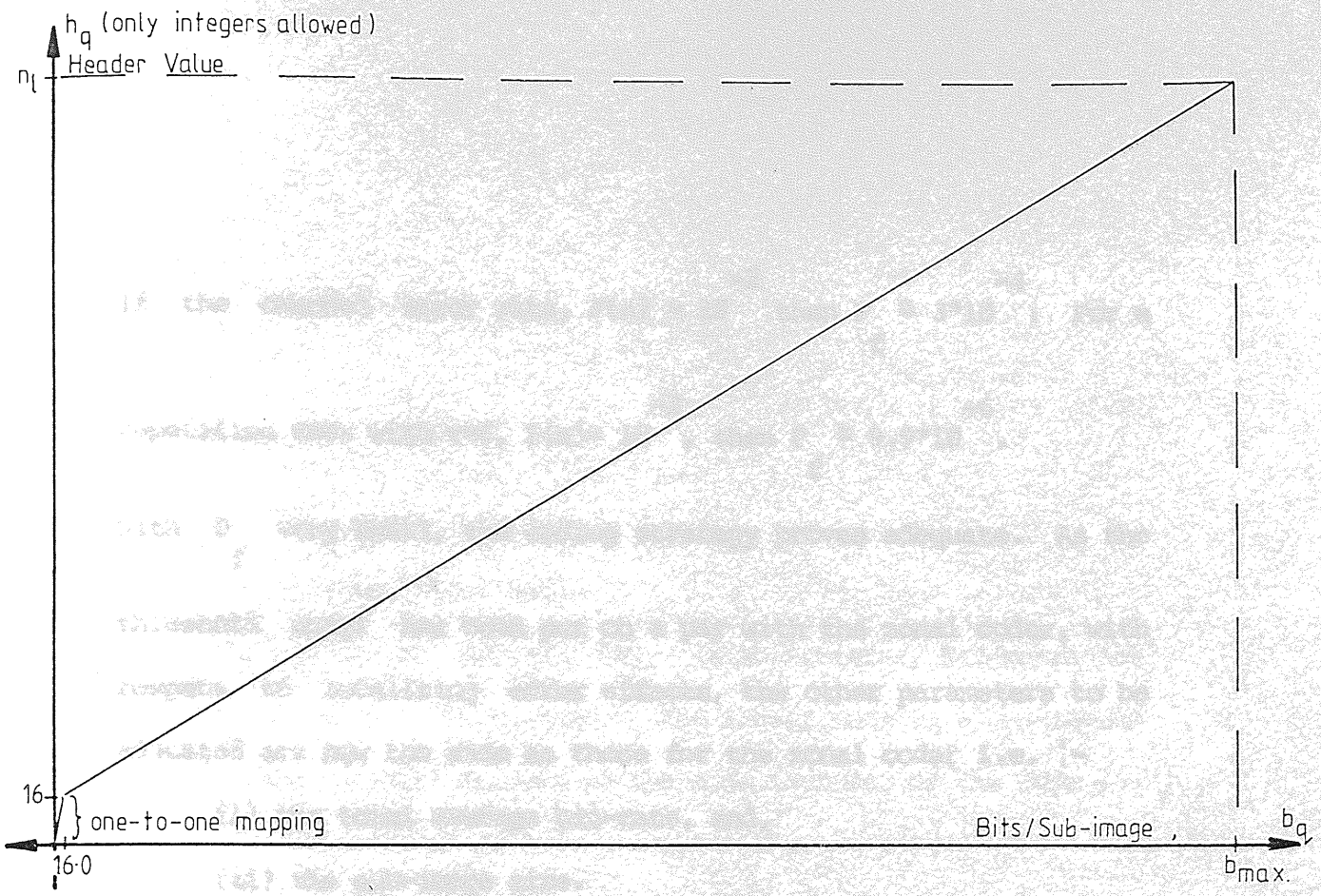


Figure 5.2.2.1 Linear Mapping Bits per Sub-image to Header Value.

Repetition of each header bit, r times, provides heavy protection of the header, ensuring that there is a low probability that it will be received in error. For example, if r=3 the probability of code failure, P_f, is, if 2 or 3 errors occur in any block of 3 bits !

$$P_f = P_2 + P_3$$

$$= {}_3C_2\, P(e)^2 \left(1-P(e)\right) + {}_3C_3\, P(e)^3$$

where,

P(e) is the raw bit-error probability i.e. the error rate

$$\therefore P_f = 3P(e)^2 - 2P(e)^3 \qquad (5.2.2.6)$$

If the channel error rate, $P(e) = 10^{-2}$ then $P_f \simeq 3*10^{-4}$. For a repetition code with r=5, $P(e) = 10^{-2}$, then $P_f \simeq 9.9*10^{-6}$.

With P_f very small, the coding strategy proves adequate. As the threshold coder has been put on a par with the zonal coder, with respect to localizing error effects, the other parameters to be adjusted are now the same as those for the zonal coder i.e. :-

(i) the total average bit-rate, and,

(ii) the sub-image size.

5.3 Block Codes with Linear Properties.

5.3.1 General Linear Code Theory.

In very general terms, there are two classes of error protection codes and block codes are the first of these classes to be considered here. Moreover, the block codes to be considered are to have, desireable, linear properties. The linear mapping,

$$\underline{v}_m = \underline{u}_m [\, G \,] \qquad (5.3.1)$$

where,

If the channel error rate, $P(e) = 10^{-2}$ then $P_f \simeq 3*10^{-4}$. For a repetition code with r=5, $P(e) = 10^{-2}$, then $P_f \simeq 9.9*10^{-6}$.

With P_f very small, the coding strategy proves adequate. As the threshold coder has been put on a par with the zonal coder, with respect to localizing error effects, the other parameters to be adjusted are now the same as those for the zonal coder i.e. :-

(i) the total average bit-rate, and,

(ii) the sub-image size.

5.3 Block Codes with Linear Properties.

5.3.1 General Linear Code Theory.

In very general terms, there are two classes of error protection codes and block codes are the first of these classes to be considered here. Moreover, the block codes to be considered are to have, desireable, linear properties. The linear mapping,

$$\underline{v}_m = \underline{u}_m [\, G \,] \qquad (5.3.1)$$

where,

where -

$\underline{v}_m$ = code vector

$\underline{u}_m$ = data vector

[G] = generator matrix,

defines a general linear code.

The generator matrix, [G], is an arbitrary KxL matrix of zeroes and ones. To be useful, the linear code is a one-to-one mapping from the data vectors to the code vectors, of the form :-

$$[G] = \begin{bmatrix} & g_{1,K+1} & \cdots & \cdots & g_{1,L} \\ & g_{2,K+1} & \cdots & \cdots & g_{2,L} \\ & \vdots & \ddots & & \vdots \\ [I] & \vdots & & \ddots & \vdots \\ & \vdots & & & \vdots \\ & g_{K,K+1} & \cdots & \cdots & g_{K,L} \end{bmatrix} \qquad (5.3.2)$$

where,

[I] is the KxK identity matrix.

It can be seen that a linear code (5.3.1), generated by the matrix (5.3.2), has it's first K symbols identical to the data symbols :-

$$v_{mn} = u_{mn} \qquad n = 1,2,\ldots \ldots K \qquad (5.3.3)$$

and the remainder are given by :-

$$v_{mn} = \sum_{k=1}^{K} u_{mk}\, g_{kn} \qquad n = K+1, K+2, \ldots \ldots L \qquad (5.3.4)$$

where,

$\underline{v}_m$ = code vector

$\underline{u}_m$ = data vector

[G] = generator matrix,

defines a general linear code.

The generator matrix, [G], is an arbitrary KxL matrix of zeroes and ones. To be useful, the linear code is a one-to-one mapping from the data vectors to the code vectors, of the form :-

$$[G] = \begin{bmatrix} & g_{1,K+1} & \cdots & \cdots & g_{1,L} \\ & g_{2,K+1} & \cdots & \cdots & g_{2,L} \\ & \vdots & \ddots & & \vdots \\ [I] & \vdots & & \ddots & \vdots \\ & \vdots & & & \vdots \\ & g_{K,K+1} & \cdots & \cdots & g_{K,L} \end{bmatrix} \qquad (5.3.2)$$

where,

[I] is the KxK identity matrix.

It can be seen that a linear code (5.3.1), generated by the matrix (5.3.2), has it's first K symbols identical to the data symbols :-

$$v_{mn} = u_{mn} \qquad n = 1,2,\ldots \ldots K \qquad (5.3.3)$$

and the remainder are given by :-

$$v_{mn} = \sum_{k=1}^{K} u_{mk}\, g_{kn} \qquad n = K+1,\ K+2,\ \ldots \ldots L \qquad (5.3.4)$$

where,

$\sum$ indicates modulo-2 summation.

An example of equations (5.3.3) and (5.3.4) in relation to the matrix (5.3.2) is given below :-

$$\begin{pmatrix} u_{m1} & u_{m2} \end{pmatrix} \begin{bmatrix} 1 & 0 & g_{13} & g_{14} \\ 0 & 1 & g_{23} & g_{24} \end{bmatrix} = \begin{pmatrix} u_{m1} & u_{m2} & u_{m1}g_{13} + u_{m2}g_{23} & u_{m1}g_{14} + u_{m2}g_{24} \end{pmatrix}$$

$$= \begin{pmatrix} \underbrace{v_{m1} \quad v_{m2}}_{\text{equation (5.3.3)}} & \underbrace{v_{m3} \quad v_{m4}}_{\text{equation (5.3.4)}} \end{pmatrix}$$

Such a code, which transmits the original K data symbols unchanged with L-K parity check symbols is called a systematic code. L is the total number of bits in a coded word.

A simple table look-up technique can be used for decoding linear codes.

From equation (5.3.3),

$$v_{mn} = u_{mn} \qquad n = 1,2,\ldots \ldots K$$

$$\therefore v_{mk} = u_{mk}$$

Substituting into equation (5.3.4) gives

$$v_{mn} = \sum_{k=1}^{K} v_{mk} g_{kn} \qquad n = K+1, K+2, \ldots \ldots, L$$

Adding v_{mn} - modulo-2 to both sides gives

$$v_{mn} \oplus v_{mn} = \emptyset = \bigotimes_{k=1}^{K} v_{mk}\, g_{kn} \oplus v_{mn}$$

which can be written in vector form as:-

$$\underline{\bar{\emptyset} = \bar{v}_m [H]^T} \qquad (5.3.5)$$

where,

$[H]^T$ is the Lx(L-K) matrix :-

$$[H]^T = \begin{bmatrix} g_{1,K+1} & \cdots & g_{1,L} \\ g_{2,K+1} & \cdots & \vdots \\ \vdots & \ddots & \vdots \\ g_{K,K+1} & \cdots & g_{K,L} \\ 1 & & [\emptyset] \\ \ddots & & \\ [\emptyset] & & 1 \end{bmatrix} \qquad (5.3.6)$$

It's transpose, the matrix [H] is the parity check matrix. Equation (5.3.5) indicates that any code vector multiplied by $[H]^T$ gives the $\bar{\emptyset}$ vector; thus the code vectors constitute the null-space of the parity check matrix. If the received vector $\bar{y}$ is post-multiplied by $[H]^T$, the resulting (L-K) dimensional binary vector is called the syndrome of the received vector and is given by :-

$$\bar{s} = \bar{y}.[H]^T \qquad (5.3.7)$$

If no errors are made $\overline{y} = \overline{v}_m$ and consequently the syndrome is zero. The syndrome can also be written in terms of the error vector :-

If the arbitrary sequence of errors is written as :-

$$\overline{e} = (e_1, e_2, \ldots \quad \ldots e_L)$$

and the convention is adopted that

$$e_n = \begin{cases} 1 & \text{if an error occurs in the } n^{th} \text{ transmitted symbol} \\ \emptyset & \text{" no " " " " " " "} \end{cases}$$

Then, if $\overline{v}_m$ is transmitted

$$\overline{y} = \overline{v}_m \oplus \overline{e} \qquad (5.3.8)$$

and,

$$\overline{v}_m \oplus \overline{y} = \overline{e} \qquad (5.3.9)$$

Hence, from equation (5.3.7)

$$\overline{s} = \overline{y}.[H]^T = (\overline{v}_m \oplus \overline{e}).[H]^T$$

$$= \underline{\overline{e}.[H]^T} \qquad (5.3.1\emptyset)$$

Hence, in words, the syndrome is an (L-K) bit number that results from writing a "Ø" for each of the (L-K) parity checks which is correct and a "1" for each failure. It can represent, at the most, $2^{(L-K)}$ parity checks.

5.3.2 Q-ary Symmetric Channels with Syndrome Decoding.

At the receiver of the image communication system, one would not be directly interested in $\overline{e}$, but rather $\overline{v}_m$, the transmitted vector. However, equation (5.3.9) shows that as $\overline{y}$, the received vector is known then if $\overline{e}$ is found $\overline{v}_m$ can also be found. Thus attention can be focused on the problem of finding $\overline{e}$, when decoding.

The syndrome representing $2^{(L-K)}$ parity checks has already been defined. Equation (5.3.10) shows how the syndrome can first be calculated knowing the received vector $\overline{y}$ and the transpose of the parity check matrix, $[H]^T$. The same equation can be used to set up simultaneous equations in 'e', e.g.

$$(s_1\ s_2\ s_3) = (e_1\ e_2\ e_3\ e_4\ e_5\ e_6\ e_7) \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & \cdot & \cdot \\ \cdot & \cdot & \\ & & \\ \cdot & & \\ h_{71} & & h_{73} \end{bmatrix} \qquad (5.3.2.1)$$

Equations (5.3.2.1) indicate that to find $\overline{e}$, in the above case, requires solving three equations in seven unknowns. Progress has been made as initially there were $2^7 = 128$ a priori possibilities for $\overline{e}$ and this number has been reduced to $2^4 = 16$ possibilities. In general there will always be (L-K) equations and L unknowns with a reduction in possibilities from 2^L to 2^K. It is noted then, that the syndrome provides some information about $\overline{e}$ but not enough. Further information is required to reduce the possibilities to only one. This is acquired by knowing something about the channel. The basic assumption is thus made that the channel is q-ary symmetric (qSC), the simplest example of which is the Binary Symmetric Channel (BSC). This assumption means that if $\overline{X}$ is a random vector modelling the channel output, then $\overline{Y} = \overline{X}+\overline{E}$, where $\overline{E} = (E_1, E_2, \ldots \ldots, E_L)$, is a random vector whose components are independent, identically distributed, random variables with common distribution

$$\left.\begin{aligned} P\{ E = \emptyset \} &= 1 - (q - 1).E_{ps} \\ P\{ E = e \} &= E_{ps} \quad \text{if } e \neq \emptyset \end{aligned}\right\} \qquad (5.3.2.2)$$

where E_{ps} is a constant, the probability that 'E' takes the value 'e'.

For the BSC, q = 2 and e = 1, equations (5.3.2.2) become :

$$\left.\begin{aligned} P\{ E = \emptyset\} &= 1 - (2 - 1).E_{ps} = 1 - E_{ps} \\ P\{ E=1 \} &= E_{ps} \end{aligned}\right\} \qquad (5.3.2.3)$$

which is intuitively correct.

For the qSC the vector probability $P\{\overline{E} = \overline{e}\}$ is given by the product of the probability of the occurrence of the components of $\overline{e}$ i.e.

$$P\{\overline{E} = \overline{e}\} = [1 - (q - 1).E_{ps}]^{L-w_H(\overline{e})} . E_{ps}^{w_H(\overline{e})} \qquad (5.3.2.4)$$

where, $w_H(\overline{e})$, the Hamming weight of $\overline{e}$, is defined to be the number of nonzero components of $\overline{e}$. Alternatively, it is the number of errors occuring in $\overline{e}$. If $E_{ps} \leq 1/q$, the right-hand side of equation (5.3.2.4) is a decreasing function of $w_H(\overline{e})$ and so the most probable e is the one of the smallest weight. The syndrome decoding algorithm for a qSC can be written as :-

1. Compute the syndrome $\overline{s} = \overline{y}.[H]^T$
2. Find a minimum-weight vector in the co-set corresponding to $\overline{s}$. Call it $z_\emptyset$.

3. Output the code-word $\hat{x} = \bar{y} - \bar{z}_{\emptyset}$

Provided K and L-K are both relatively small, it is practically economical to implement step 2 via a table look-up procedure.

5.3.3 Hamming Codes

Section 2.2.5 has already stated that the most powerful error correcting block codes known are the Bose-Chaudhuri-Hocquenghem (BCH) codes. The Hamming code can be regarded as a special case of the BCH codes. In order to develope the coder(s) required, it is beneficial to set down the general specification obtained from the major assumption that errors occur due to white noise. The problem thus becomes :-

To find the best encoding scheme for single error correction for white noise.

The Hamming weight has already been introduced in the previous section. The concept of the Hamming Distance is now defined.

Between any two vectors $\bar{x}$ and $\bar{y}$ of the same length, the Hamming distance is defined as :-

$$d_H (\bar{x}, \bar{y}) = w_H (\bar{y} - \bar{x}) \qquad (5.3.3.1)$$

The Hamming codes are based on the concept of the Hamming Distance. In order to correct 'c' errors in a block code a minimum (Hamming) distance of

$$D_{min} = 2c + 1 \qquad (5.3.3.2)$$

is required. As D_{min} is the minimum value for all possible code words, the general requirement is :-

$$D \geq 2c + 1 \qquad (5.3.3.3)$$

The construction and implementation of the Hamming codes used in this research, i.e. the (7, 4) and (15, 11) codes, make use of the syndrome decoding strategy outlined in the previous section but in a simpler way.

First it is assumed that the channel is the BSC, and that the fewer 1's (errors) in an error pattern, the more likely it is to have been the actual error pattern. If this is the case then for the (7, 4), (15, 11) codes there is always a unique solution to equation (5.3.10) i.e.

$$\bar{s} = \bar{e}.[H]^T$$

of weight "0" or "1".

The unique solution provides the last step towards developing the Hamming decoding algorithm :-

1. Compute the syndrome $\bar{s} = \bar{y}.[H]^T$

2. If $\bar{s}$ = zero vector, set $\hat{e} = 0$; go to 4.

3. Locate the unique column of $[H]^T$ which is equal to $\bar{s}$;

call it column i; set $\hat{e}$ = 0's except for a single 1 in

th
the i coordinate.

4. Set $\hat{x} = \bar{y} + \hat{e}$ (this is the decoder's estimate of the transmitted code word).

By careful positioning of the parity checks in each transmitted word it is possible to make the parity checks themselves indicate the position of one digit in error. This is done by considering the binary representation of each digit position. Letting the parity check digits be "a_{n-1}", then it is found that they should be placed according to :-

Parity check digit a_{n-1} in position 2^{n-1} of the code word.

e.g.

a_1 in position $2^0 = 1$, checking digits 1,3,5,7,...

a_2 " " $2^1 = 2$, " " 2,3,6,7,...

a_4 " " $2^2 = 4$, " " 4,5,6,7,...

etc.

By dividing the bit-stream into blocks which are relatively small, the assumption of only having 1 digit in error for any one of these blocks can be accepted and thus the decoding algorithm can be written in the following form :-

Considering the (7, 4) code

1. Select the parity check from the received word $\bar{y}$; call

it $a_4 a_2 a_1$.

2. Deduce the parity check from the data digits of $\bar{y}$; call it $b_4 b_2 b_1$.

3. Form $a_4 \oplus b_4 \oplus 1 = \hat{e}_4$, $a_2 \oplus b_2 \oplus 1 = \hat{e}_2$, $a_1 \oplus b_1 \oplus 1 = \hat{e}_1$.

4. $\hat{e}_4 \hat{e}_2 \hat{e}_1$ gives the location of the digit in error; call it i; change digit i in the received word $\bar{y}$; call it $\dot{x}$.

5. $\hat{x}$ is the estimate of the word sent; deduce the required data.

Figure 5.3.3.1 indicates the table for the Hamming error correcting code for the (7, 4) code. The (15, 11) code follows the same procedures. The immediately preceding method and algorithm are those implemented when using the Hamming codes for the work in this thesis.

Character	Digit Number							
	1	2	3	4	5	6	7	
	a_1	a_2		a_4				- Parity Checks
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	
1	1	1	Ø	1	Ø	Ø	1	
2	Ø	1	Ø	1	Ø	1	Ø	
3	1	Ø	Ø	Ø	Ø	1	1	
4	1	Ø	Ø	1	1	Ø	Ø	
5	Ø	1	Ø	Ø	1	Ø	1	
6	1	1	Ø	Ø	1	1	Ø	
7	Ø	Ø	Ø	1	1	1	1	
8	1	1	1	Ø	Ø	Ø	Ø	
9	Ø	Ø	1	1	Ø	Ø	1	
10	1	Ø	1	1	Ø	1	Ø	
11	Ø	1	1	Ø	Ø	1	1	
12	Ø	1	1	1	1	Ø	Ø	
13	1	Ø	1	Ø	1	Ø	1	
14	Ø	Ø	1	Ø	1	1	Ø	
15	1	1	1	1	1	1	1	

Figure 5.3.3.1 Hamming (7, 4) Single Error Correction.

5.3.4 Hamming Code Rate and Probable Failure.

When considering the performance of an error protection scheme, in general, two parameters of interest are the code rate, R, as this will affect the overall average bit-rate, and the block error probability, as this will indicate how well the code will perform compared to the channel error rate.

The rate, R, of a block code can be obtained by considering how fast the channel can transmit bits compared to how fast the source produces them. For example, with the (7, 4) Hamming code the assumption is being made that the channel can transmit 7 bits in the time it takes the source to produce 4 bits. Hence the code rate, R, is 4/7 = Ø.571. For the (15, 11) Hamming code the rate, R, is 11/15 = Ø.733. In general the rate of an (n, k) block code is given by :-

$$R = k/n \qquad (5.3.3.1)$$

If $\hat{x}$ and $\bar{x}$ are the estimated and actual transmission code words respectively, and P_E denotes the block error probability $P(\hat{x}=\bar{x})$, then :

$$P_E = \sum_{k=2}^{n_b} C^{n_b}_{k} . p^{k} (1-p)^{n_b - k} \qquad (5.3.3.2)$$

where,

n_b = the total number of bits in one transmitted code word.

k = the number of bits in error.

Thus with the Hamming (7, 4) code

$$P_E = \sum_{k=2}^{7} C_{7\,k} . p^k (1-p)^{7-k}$$

$$= 21p^2 - 70p^3 + \text{terms in higher powers of } p$$

and, if the error rate, $p=10^{-2}$

$$P_E \simeq 21.7 * 10^{-4}$$

With the Hamming (15, 11) code and $p = 10^{-2}$

$$P_E \simeq 1.05 * 10^{-2}$$

From the above, the higher protection the (7, 4) code provides over the (15, 11) can be seen although this is at the expense of an increase in the number of transmitted bits. The values of P_E can be compared to p, the channel error rate. If this is done for error rates up to that indicated, i.e. 10^{-2}, then the increased protection can be seen since P_E is very much lower than p. For an error rate of 10^{-2} it can be seen that the probability of code failure can encroach upon this error rate, implying the code fails

at roughly the same rate with which errors occur in the bit-stream. This type of comparison gives a guide to the practical results to be obtained.

5.4 Convolutional Codes.

5.4.1 General Convolutional Code Theory.

The second class of error protection codes come under the heading of Convolutional Codes (CC). It is assumed, for simplicity, that the codes are for use on a binary input channel. Three approaches to CCs are mentioned; the polynomial matrix approach, the scalar matrix approach and the shift register approach, contributing a more thorough understanding to the general theory. The last approach is then adopted for use in this thesis.

The Polynomial Matrix Approach.

An (n, k) CC is characterized by a kxn generator matrix [G], where the entries g_{ij} are polynomials. For example,

$$[G] = [\ x^2 + 1,\ x^2 + x + 1\ ] \qquad (5.4.1)$$

is the generator matrix for a (2, 1) CC, and

$$[G] = \begin{bmatrix} 1 & \emptyset & x+1 \\ \emptyset & 1 & x \end{bmatrix} \qquad (5.4.2)$$

is the generator matrix for a (3, 2) CC. Three important parameters for a CC are :-

(i) The memory

$$M = \max_{ij}[\deg(g_{i,j})] \qquad (5.4.3)$$

(ii) The constraint length

$$K = M + 1 \qquad (5.4.4)$$

(iii) The rate

$$R = k/n \qquad (5.4.5)$$

In order to use [G] to encode scalar information the information bits must be mapped into the coefficients of a k-tuple of polynomials $\overline{I} = (I_{\emptyset}(x),\ldots\ldots, I_{k-1}(x))$. The codeword $\overline{C} = (C_{\emptyset}(x),\ldots\ldots, C_{n-1}(x))$ is an n-tuple of polynomials defined by :-

$$\overline{C} = \overline{I} \cdot [G] \qquad (5.4.6)$$

where the dot denotes vector-matrix multiplication. For example, the polynomial information $\overline{I} = (x^3 + x + 1)$ would be encoded into the polynomial codeword $\overline{C} = (x^5 + x^2 + x + 1,\ x^5 + x^4 + 1)$ when using the (2, 1) CC of equation (5.4.1) with equation (5.4.6).

Specification of the exact correspondence between k- and n-tuples of polynomials and bit patterns leads naturally to the scalar matrix approach to CCs.

The Scalar Matrix Approach.

The scalar (in bits) representation of a polynomial codeword $\overline{C} = (C_{\emptyset}(x), \ldots, C_{n-1}(x))$ is obtained by interleaving the polynomials' coefficients. If the j^{th} polynomial is $C_j(x) = C_{j\emptyset} + C_{j1}x + \ldots$, the scalar representation of $\overline{C}$ is

$$\overline{C} = (C_{\emptyset\emptyset}, C_{1\emptyset}, \ldots, C_{n-1,\emptyset}, C_{\emptyset 1}, \ldots, C_{n-1,1}, \ldots) \qquad (5.4.7)$$

A scalar version of [G] can be constructed by letting:

$$[G] = \sum_{v=\emptyset}^{M} G_v x^v \qquad (5.4.8)$$

For example, the polynomial generator matrix of (5.4.1), expanded by equation (5.4.8) is:

$$[G] = [1, 1] + [\emptyset, 1].x + [1, 1].x^2 \qquad (5.4\ 9)$$

The scalar generator matrix for equation (5.4.1) is:

$$G = \begin{matrix} 1\ 1\ \emptyset\ 1\ 1\ 1 & \rightarrow \\ 1\ 1\ \emptyset\ 1\ 1\ 1 & \\ 1\ 1\ \emptyset\ 1\ 1\ 1 & \searrow \\ \downarrow & \end{matrix}$$

The scalar information corresponding to the polynomial information $\overline{I} = (x^3 + x + 1)$ is (1 1 Ø 1) and the scalar

codeword corresponding to the polynomial codeword $\overline{C} = (x^5 + x^2 + x^5 + x^4 + 1, x^5 + x^4 + 1)$ is (1 1 1 0 1 0 0 0 0 1 1 1).

In a practical application there may be a maximum allowable degree of the information polynomial, which leads to the definition of the L^{th} truncation of a CC :-

First, it is required that $\deg[\overline{I}_i(x)] \leq L-1$, $i = 0,1,\ldots,k-1$. Then by equations (5.4.3) and (5.4.6) each component, in the corresponding codeword $\overline{C} = (C_0(x),\ldots,C_{n-1}(x))$ has degree $\leq M + L - 1$ and thus $\overline{I} = (I_0(x),\ldots,I_{k-1}(x))$ can be represented by kL bits and the codeword $\overline{C}$ by n(M+L) bits. Mapping from $\overline{I}$ to $\overline{C}$ can then be represented by $\overline{C} = \overline{I}.G_L$, where G_L is the L^{th} truncation of G. The L^{th} truncation of an (n, k) CC can thus be viewed as an (n(M+L), kL) linear block code, and in this sense a CC is a special kind of block code. The rate of the truncated code is given by:

$$R_L = \frac{kL}{n(M+L)} \qquad (5.4.10)$$

In most practical situations L >> M hence

$$R_L \simeq k/n = R \qquad (5.4.11)$$

and this is one reason why the rate of the code is taken as R - another reason is given later.

The Shift Register Approach.

This approach is used for the simulation of the CCs in this work, as it is simple and lends itself readily to implementation. To implement an encoder to perform the operations of the generator matrix given in equation (5.4.1) a device would be required which accepts an input stream of information bits $\overline{I} = (\overline{I}_{\emptyset}, \overline{I}_{1}, \ldots)$ and produces an encoded stream $\overline{C} = (C_{\emptyset\emptyset}, C_{1\emptyset}, C_{\emptyset 1}, C_{11}, \ldots)$ as output, where $\overline{I}$ and $\overline{C}$ are related by:

$$C_{\emptyset}(x) = C_{\emptyset\emptyset} + C_{\emptyset 1}\, x + \ldots$$

$$= (x^2 + 1).(I_{\emptyset} + I_{1}\, x + \ldots)$$

$$= (x^2 + 1)\, \overline{I}(x)$$

and,

$$C_{1}(x) = C_{1\emptyset} + C_{11}\, x + \ldots$$

$$= (x^2 + x + 1)\, \overline{I}(x)$$

The encoder must be capable of multiplying the input bit-stream by $x^2 + 1$ and $x^2 + x + 1$. Figure 5.4.1 depicts the circuit which performs the required operations.

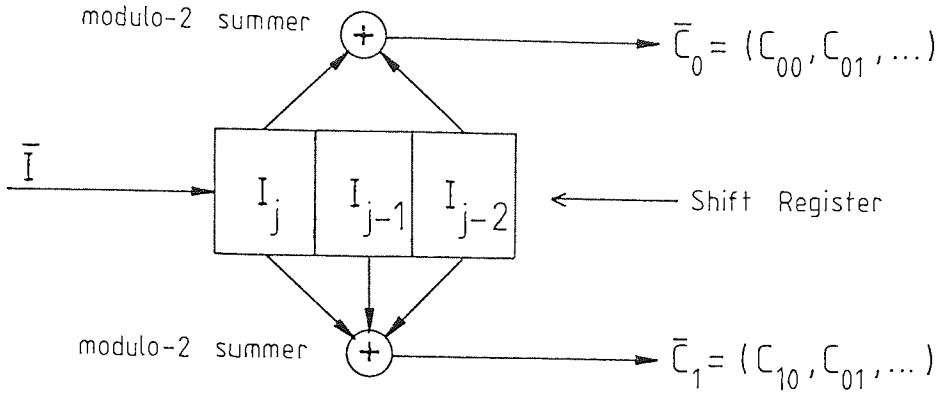


Figure 5.4.1 A Shift Register Encoder.

The input and output streams are related by:

$$C_{\emptyset j} = I_j + I_{j-2} \quad , j = \emptyset,1,2,... \qquad (5.4.12)$$

$$C_{1j} = I_j + I_{j-1} + I_{j-2}$$

Showing that the j^{th} output bits depend, not only on the j^{th} input bit I_j, but also on the preceding bits I_{j-1} and I_{j-2}. Hence the code must " remember " these two bits i.e. the memory of the code, M=2. Also, it can be seen from Figure 5.4.1 that there are 2 output bits for every 1 input bit; thus the code's rate is 1/2 (=k/n).

5.4.2 Hagelbarger Convolutional Coding.

Hagelbarger's code uses the preceding, shift register, approach for the formation of CCs. The codes to be described were designed for correcting errors which occur in relatively small

bursts, b, and which have a relatively high rate, R. The principle is to insert the parity check bits associated with the message bits spread out in time so that error bursts are not likely to include more than one bit of the group whose parity is checked. The codes are simple to implement and are to be used as an example of simple CCs in comparison with Hamming codes, in the presence of uniform, random, independent errors. The performance between block and convolutional coding is compared, bearing in mind that in the case of bursts of errors of more than 1 bit, the CC would be expected to have a much superior performance to the corresponding Hamming code.

Both CCs used in this work have a rate, R=1/2, and are thus (2, 1) CCs. The codes have constraint lengths, K=7 and 11 and are thus identified as CC(7) and CC(11) respectively, for future reference. The encoder for CC(7) is illustrated in Figure 5.4.2.1. Before being transmitted the data digits are shifted

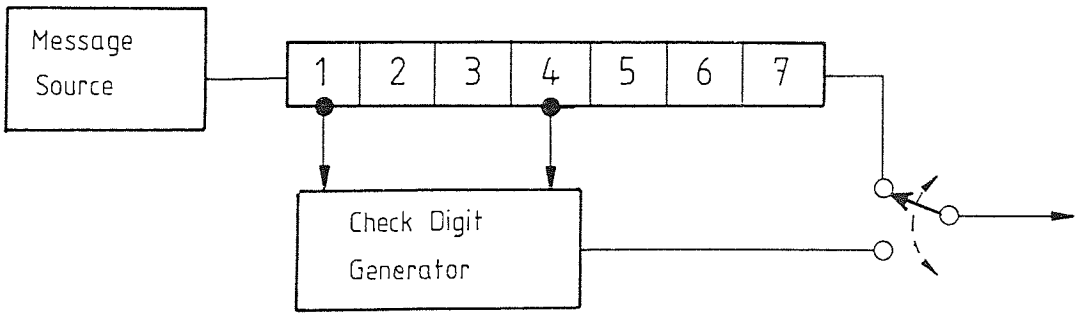


Figure 5.4.2.1 Error Protection Coder for CC(7).

through the shift register after entering from the left (position 1). A parity check digit is generated for each shift, forming an even parity between the check digit and the data digits in the first and the fourth positions (a modulo-2 summation). The check digit is transmitted before the data digit in the seventh position. Upon making a shift, the data digit which was in the seventh position is transmitted and a new check digit is calculated. Two coded digits are thus transmitted during the same time it takes for the acceptance of one new data digit by the encoder, hence the rate R=1/2. The decoder for CC(7) is shown in Figure 5.4.2.2. The received code enters an alternating switch

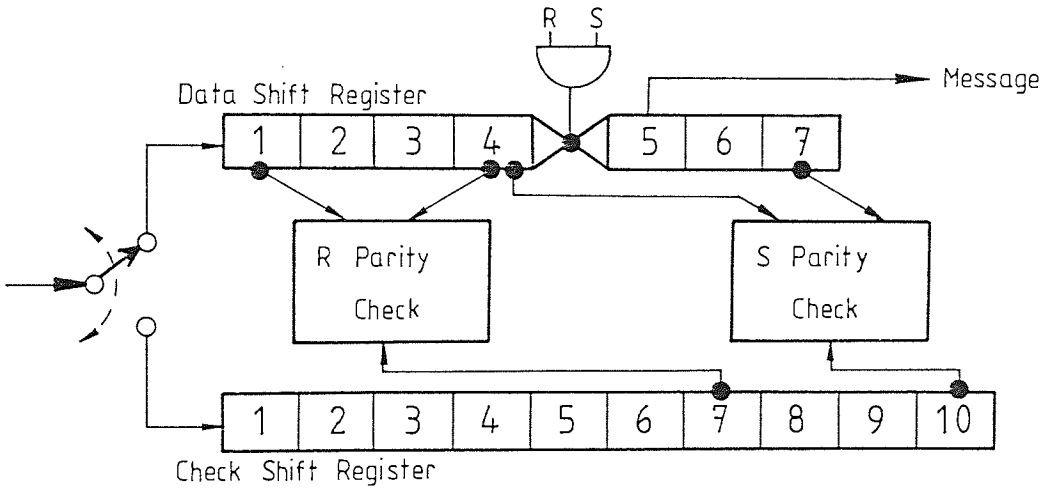


Figure 5.4.2.2 Decoder for CC(7).

The received code enters an alternating switch which directs the message and check bits into separate shift registers as shown in the figure. Two parity checks designated R and S are made

simultaneously, one on stages 1 and 4 and the other on stages 4 and 7 of the message register. The decoding rule is:

Change the data digit (Ø→1, 1→Ø) while shifting it to position 5 whenever both R and S fail. No change is to be made if only one parity check digit fails.

When R holds and S fails, there is an error in the check digit in the tenth state of the check shift register. If the message was to be re-transmitted this digit would be reversed. The corrected data digits are available at position 5 of the data shift register. After any burst of length 6 or less, a 2Ø-digit error-less message is enough to completely refill the decoder shift registers. A further burst can thus be corrected without interference from the previous one provided there is a "clean" message of 19 or more digits between bursts. Figure 5.4.2.3 shows the particular parameters of CC(7) and CC(11) in terms of correction of bursts of any even length 2B.

(1/2)xBURST LENGTH (B)	BURST LENGTH (2B)	DATA SHIFT REGISTER LENGTH (CONSTRAINT LENGTH) (2B + 1)	CHECK SHIFT REGISTER LENGTH (3B + 1)	GUARD SPACE BETWEEN BURSTS. (6B + 1)
3	6	7	1Ø	19
5	1Ø	11	16	31

Figure 5.4.2.3 Code Parameters of CC(7) and CC(11).

5.4.3 Rate and Probable Hagelbarger Code Failure.

The L^{th} truncation rate of an (n, k) CC with memory M and truncation of the information vector $\overline{I} = (I_{\emptyset}(x),...I_{k-1}(x))$ so that $\deg[\overline{I}_i (x)] \leq L-1$, $i = \emptyset,1,...,k-1$ has been seen to be given by equation (5.4.10) ¦-

$$R_L = \frac{kL}{n(M+L)}$$

The rate R, of the non-truncated code, has been seen to be given by equation (5.4.11) ¦-

$$R = k/n$$

Hence,

$$R_L = R\frac{L}{M+L} = R.\frac{1}{1+(M/L)} \qquad (5.4.3.1)$$

As with the block codes, for a given CC and a given channel, interest lies in the resulting " error probability ". However, there are several possible definitions of error probability , depending on what is meant by an "error". Decoding of CCs can be undertaken algebraically or by sequential or maximum likelihood decoding. This work uses the algebraic decoding method (Section 5.4.2). Richters [92] uses the same method, but shows the encoder and decoder in a slightly different form, which simplifies the derivation of an expression giving the probability of code failure. The basis of his ideas are adopted here, although he

considered " Pareto Error Statistics " whereas the following derivation is somewhat simpler as random, uniform, independent error statistics are assumed. The following form summarizes the problem that is under consideration| given that an information digit error is made by the channel, what is the probability that the decoder will fail to correct it? Richters shows that for Hagelbarger codes of 5Ø% redundancy, there are eight patterns of errors that cause correction of an information digit. It is assumed, for simplicity, that all zeros are transmitted, so that a digit may be represented by a 1 if it is in error and by a Ø if it is correct. With no transmission errors, the syndrome becomes the sum of the channel output with information and no errors and the error pattern with all zeros in place of the information. Since the syndrome is zero with no errors, it thus depends only on the error pattern, justifying the assumption.

Following the nomenclature of Figure 5.4.2.3, the burst length is given by 2B (B is thus one half of the burst length) and the eight error patterns causing correction of the information digit, number "Ø", which is in error are found by Richters to be as shown in Table 5.4.3.1 ¦-

TABLE 5.4.3.1

DIGIT NUMBER	3(2B+1)		2(2B+1)	2(2B)		(2B)+1	(2B)		Ø		-(2B)
Pattern Number,i											
1	Ø		Ø	Ø		Ø	Ø		1		Ø
2	1	B	Ø	1	B	Ø	Ø	B	1	B	Ø
3	Ø	L	Ø	Ø	L	1	Ø	L	1	L	1
4	1	A	Ø	1	A	1	Ø	A	1	A	1
5	1	N	1	Ø	N	Ø	1	N	1	N	Ø
6	Ø	K	1	1	K	Ø	1	K	1	K	Ø
7	1	S	1	Ø	S	1	1	S	1	S	1
8	Ø		1	1		1	1		1		1

A blank in the table represents any combination of ones and zeros (don't cares).

P_i is defined as the probability that pattern i appears at the decoder input given that digit number 'Ø' is in error. Letting P_e be the probability that digit number 'Ø' is not corrected, gives P_e as :-

P_e = (Prob. that digit number 'Ø' is in error) x (prob. that digit 'Ø' is not corrected).

= (Prob. that digit number 'Ø' is in Error) x (1-{prob. that digit 'Ø' is corrected}).

For random, uniform, independent errors :-

Prob. that digit number 'Ø' is in error = p,

where, p is the digit error probability.

So,

$$P_e = p.\left(1- \sum_{i=1}^{8} P_i \right) \qquad (5.4.3.1)$$

where,

$\sum_{i=1}^{8} P_i$, is the sum of the probabilities with which the patterns P_i occur.

Calculating the P_i in turn - the fact that digit number 'Ø' is always in error in the correcting patterns fixes each pattern, giving no other digit combination for those digits shown in TABLE 5.4.3.1. Considering the remaining digits of the error pattern probabilities $P_1, P_2, \ldots \ldots P_8$ gives :-

P_1 = Prob. that all remaining digits in pattern 1 are zero.

$$= (1-p)^6$$

P_2 = Prob. that digit numbers '3(2B)+1', '2(2B)' are 1 and the remaining digits are zero

$$= p^2 (1-p)^4$$

Similarly,

$$P_3 = p^2 (1-p)^4 , \quad P_4 = p^4 (1-p)^2 , \quad P_5 = p^3 (1-p)^3 ,$$

$$P_6 = p^3 (1-p)^3 , \quad P_7 = p^5 (1-p) , \quad P_8 = p^5 (1-p).$$

$$\therefore \sum_{i=1}^{8} P_i = (1-p)^6 + 2p^2 (1-p)^4 + 2p^3 (1-p)^3 + p^4 (1-p)^2 + 2p^5 (1-p)$$

$$= 1 - 6p + 17p^2 - 26p^3 + \text{terms in highers orders of } p.$$

(5.4.3.2)

Substituting (5.4.3.2) into (5.4.3.1) gives :-

$$P_e = p(1 - (1 - 6p + 17p^2 - 26p^3 + \ldots))$$

$$\therefore P_e \simeq 6p^2 - 17p^3 + 26p^4 \qquad (5.4.3.3)$$

With the higher channel error rate of 10^{-2}, equation (5.4.3.3) thus gives P_e approximately as, $P_e = 6 \times 10^{-4}$, which is the theoretical code failure for both CC(7) and CC(11).

The above is only a guide to code failure since, for instance, the consideration of falsely correcting an information digit has not been undertaken. P_e can also be interpreted as the expected number of output errors per data digit error and thus can be taken as an indication of the performance of the code.

5.5 Maximum Likelihood Decoding of Huffman Coded Image Data.

Maximum likelihood decoding has it's normal application in the decoding of convolutional codes. This section, however, deals

with examining the possibility of using such a decoder on source coded data, without error-protection coding. This idea deals only with data encoded via the Huffman coding algorithm, outlined in Section 3.6.2, and would thus be applicable only to the threshold coding scheme and not the zonal coding scheme. It's basis lies in the fact that when an error occurs in a stream of bits that have been source encoded incorporating the Huffman coding algorithm, the resulting reconstructed image is often very obviously in error. If only one bit-error in a sub-image causes such a drastic result then there may be a way of locating and correcting it. A maximum likelihood decoder could, perhaps, provide the solution. The basic ingredients for a maximum likelihood decoder can be determined :-

(i) The a priori probability of a given source output stream - this is known, as the a priori probabilities of the source output words are already calculated when Huffman coding.

(ii) The channel error probability - this is assumed known.

(iii) The a posteriori probability that a given source word sequence was sent - this is calculated based on the received word sequence.

(iv) The number of words and bits in the given sequence - this is also assumed known.

The above imply that for each received sequence the a posteriori probability that a particular source output word sequence was sent can be calculated. From this, the maximum likelihood decoder can then select the most likely sequence as the one that was sent.

To determine these a posteriori probabilities, the reverse (backward) probabilities $P(\overline{X}_j / \overline{Y}_i)$ are required, where, $\overline{X}_j$ and $\overline{Y}_i$ are the transmitted vector (word sequence), and the received vector (word sequence) respectively. $P(\overline{X}_j / \overline{Y}_i)$ denotes the probability that $\overline{X}_j$ was transmitted given that $\overline{Y}_i$ is received. Through the use of Baye's theorem, the reverse probabilities are given by :-

$$P(\overline{X}_j / \overline{Y}_i) = [P(\overline{Y}_i / \overline{X}_j).P(\overline{X}_j)] / [P(\overline{Y}_i)] \qquad (5.5.1)$$

where,

$P(\overline{X}_j)$, $P(\overline{Y}_i)$ are the probabilities with which $\overline{X}_j$, $\overline{Y}_i$ occur respectively,

$P(\overline{Y}_i / \overline{X}_j)$ are the forward probabilities defined as; the probability that $\overline{Y}_i$ will be received given that $\overline{X}_j$ was transmitted.

$P(\overline{Y}_i/\overline{X}_j)$ can be found by considering a string of binary data with N bits :-

$$P_N(\overline{Y}_i/\overline{X}_{mj}) = \prod_{n=1}^{N} P(y_n/x_{mn})$$

the m being inserted as there are assumed to be m possible combination of bits that may have been sent i.e.

$$\text{Received vector, } \overline{Y}_i = (y_1, y_2, \ldots, y_N)$$

$$\text{Transmitted vector, } \overline{X}_j = (x_{m1}, x_{m2}, \ldots, x_{mn})$$

Thus for the BSC

$$P_N(\overline{Y}_i/\overline{X}_{mj}) = P(e)^{d_H} (1-P(e))^{N-d_H} \qquad (5.5.2)$$

where,

d_H = the Hamming distance between the two, N length vectors

$\overline{X}$ and $\overline{Y}$, as defined in Section 5.3.3

$P(e)$ = the probability of an error in one symbol of the N

length received vector $\overline{Y}$.

$P(\overline{Y}_i)$ can be found by considering the joint probability distribution $P(\overline{Y}_i, \overline{X}_j)$ where,

$$P(\overline{Y}_i, \overline{X}_j) = P(\overline{Y}_i/\overline{X}_j).P(\overline{X}_j)$$

Thus,

$$P(\bar{Y}_i) = \sum_{j=1}^{m} P(\bar{Y}_i / \bar{X}_j).P(\bar{X}_j) \qquad (5.5.3)$$

In a practical situation $P(\bar{X}_j)$ will be known as it can be calculated from the probabilities associated with each source code word, and these are required initially to determine the Huffman coding tables. Hence as $P(\bar{X}_j)$ is known the reverse, a posteriori, probability that a certain word sequence was sent can be calculated by using equations (5.5.3) and (5.5.2) and substituting them into equation (5.5.1). It is noted that, since the number of words and bits in the sequence is assumed known, some sequences that do not contain an exact number of words can be neglected i.e. the probability of an incomplete word is zero and hence the probability of such a sequence is also zero.

The information outlined above can be used to implement a maximum likelihood decoder. The method of implementation and use of the above information is outlined in a later section, Section 5.6.6, and forms, what is later termed as, "Protection Scheme 6".

5.6 Channel Error Protection Schemes.

With the knowledge of the acquired intermediate results and having selected the type of error-protection codes to be used when guarding against channel errors, the stage has now been

reached where combined source/channel coding schemes can be presented. Also, based on the acquired results, a decision is made to use sub-image sizes of 16x16 or 8x8 pixels square only throughout the remainder of the work.

5.6.1 Scheme 1 - Parameter Adjustments Only.

As the heading suggests, this scheme uses the propositions put forward in Sections 5.2.1 and 5.2.2, making the required modifications (threshold scheme only) and simply observing the effects on the sensitivity of the reconstructions to channel errors when varying :-

(a) The total average bit-rate.

(b) The sub-image size.

(c) The type of transform coder.

(d) The channel error rate.

The results from this scheme can be seen as an extension to the results obtained when source coding at the lowest, fixed, achievable bit-rate without error protection that were obtained in Chapter 4, thus preserving continuity of the work.

5.6.2 Scheme 2 - Equal Importance Protection through Hamming Block Coding.

This scheme uses the result that, in some cases, an error in any position of the bit-stream of one sub-image produced some change in that sub-image. The scheme ignores the result that some words appear to carry more importance than others (Schemes 4 and 5), but in doing so reduces the amount of complexity required of the channel coder. By assuming that every bit of the source coded bit-stream of a sub-image carries the same amount of importance, individual sub-image bit-streams need not be considered. Instead they can be viewed as one, continuous image bit-stream. This continuous bit-stream is broken into blocks of 7 or 11 bits. Using the Hamming block coding technique, given in Section 5.3.3, protection against a single error in any one block is provided through the use of the (7, 4) or (15, 11) Hamming code. In this way it is hoped to eliminate as many changes as possible that would occur in a sub-image due to errors in any position. Provided the block is made small enough the possibility that two or more errors occur, in any one block, is reduced. Since the channel error protection decoder is set for single error correction it is kept in mind that, if failure occurs, this is quite

likely to result in the injection of a further error into the bit-stream. This, in turn, could lead to further, unwanted, reconstruction errors.

5.6.3 Scheme 3 - Equal Importance Protection through Hagelbarger Convolutional Coding.

As with Scheme 2, this scheme uses the idea of reducing any error effect that might arise through channel errors, by assuming equal importance of every bit of the continuous image bit-stream. This time the bit-stream need not be divided into blocks. Instead, the complete stream can be taken as one, long, continuous message which is coded using encoders similar to that shown in Figure 5.4.2.1. With parameters set accordingly, this type of encoder, which employs the Hagelbarger convolutional coding technique given in Section 5.4.2, produces the codes previously identified as CC(7) or CC(11). The scheme uses both codes for comparison with each other, as well as for comparison with the other schemes.

5.6.4 Scheme 4 - Percentage Sub-image Protection, using Hamming Block Codes.

This scheme takes into account the result that the initial transform words (coefficients) appear to carry more importance than the trailing words (coefficients) as well as the result that it is best to protect every bit of these words. Thus only a certain percentage of the bits of any sub-image are protected through the use of the (7, 4) or (15, 11) Hamming code. With the zonal coder, this implies the addition of a further zonal barrier on the bit-map which divides the sub-image into a section of words to be protected and a section to be left unprotected. Figure 5.6.4.1 shows examples of this further zonal protection

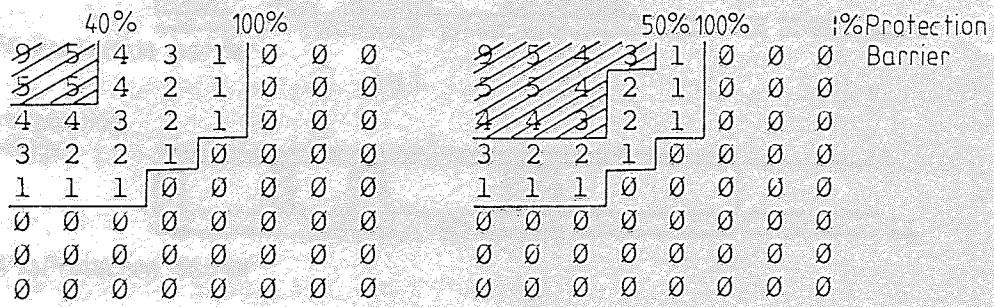


Figure 5.6.4.1 Zonal Bit-map Indicating Percentage Protection Barriers.

barrier, the shaded area indicating the region which is protected by the Hamming block coding technique.

With the threshold technique, there is no set number of bits to be protected for each sub-image. However, this protection scheme can still be applied and results in a variable number of "bits to be protected" for each sub-image. The percentage is simply set and the corresponding, leading, bits of each sub-image protected using the Hamming block coding technique. The trailing bits are again left unprotected. Figure 5.6.4.2 shows this more clearly, the protected bits indicated by the shaded areas of this figure. If there is only one word in the sub-image the percentage protection barrier is ignored and the whole of the word is protected.

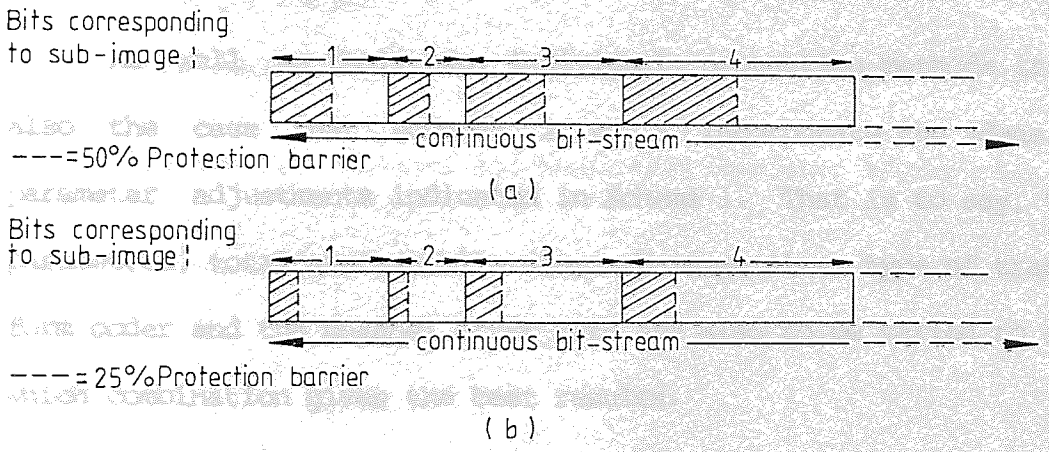


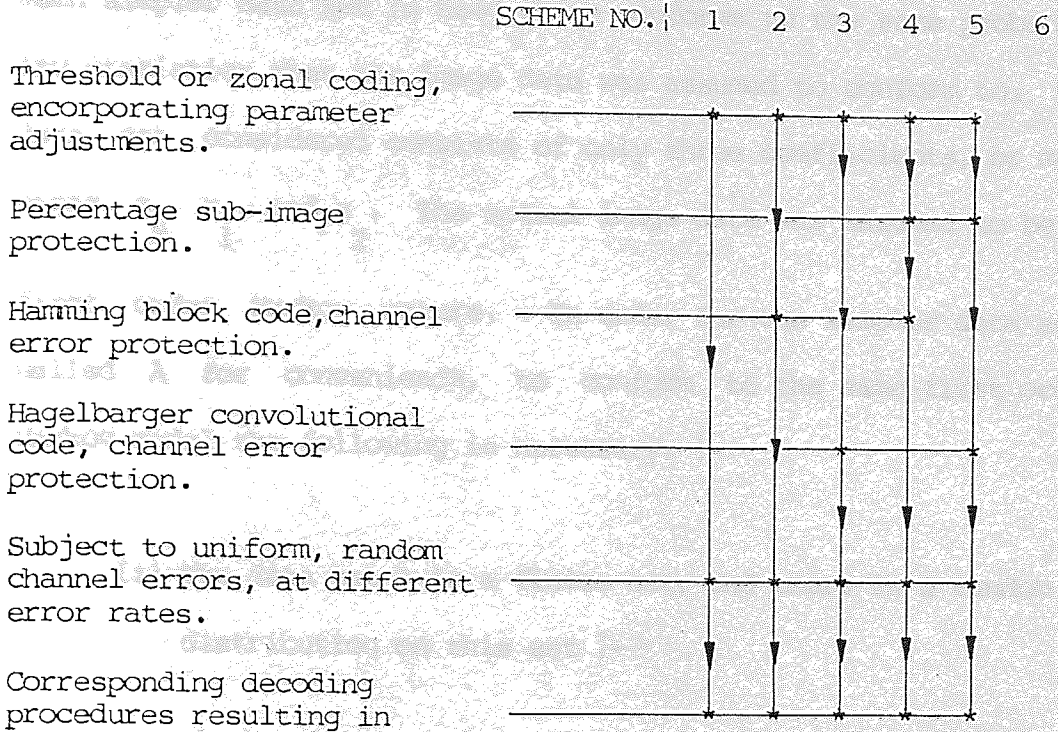
Figure 5.6.4.2 Percentage Protection Barriers on Threshold Coded Data.

5.6.5 Scheme 5 - Percentage Sub-image Protection, using Hagelbarger Convolutional Codes.

This scheme is identical to Scheme 4 except that the bits to be protected are, this time, to be protected through the use of the Hagelbarger convolutional codes CC(7) and CC(11), defined earlier. The percentage barrier implementation still applies i.e.Figures 5.6.4.1 and 5.6.4.2 are still relevant to this section.

As well as their own, individual, protection methods it is also the case that schemes 2 to 5 incorporate the ideas of parameter adjustments indicated in Scheme 1. That is to say, the parameters, total average bit-rate, sub-image size, type of transform coder and the channel error rate are also to be varied to see which combination gives the best results.

The overall processes involved, and to which schemes they apply, are shown in Figure 5.6.5.5 which provides a comparative view of all schemes. Scheme 6 is included for completeness, but all the processes in Figure 5.6.5.5 are not applicable to it and the reason why becomes apparent from the next section.



* indicates that this scheme incorporates the corresponding step.

Figure 5.6.5.5 Schemes and their Corresponding Processes.

5.6.6 Scheme 6 - The Theoretical Performance, Maximum Likelihood Decoding of a Huffman Coded Image Data Bit-stream.

This scheme makes use of the ideas and theory given in Section 5.5. However, although the idea appears simple, there are practical difficulties that limit it's application. It was found not feasible to consider the actual image source coded data formed from the Huffman code tables of the threshold coder. Instead, a

much simpler data set is used which conforms to the same probability statistics that the image data was assumed to conform to. The data set considered consists of only three coefficients, or messages, x_0, x_1, and x_2. The actual image data was assumed to be of first order Markov nature. In order for the simpler data set, called A for convenience, to conform to the same first order Markov model the following is necessary. ¦-

(i) The data set A is a finite set, and there is a random distribution on this set ¦-

$$p(x) \geq 0, \quad \sum_{x \in A} P(x) = 1 \qquad (5.6.6.1)$$

(ii) There is a stochastic transition matrix on A ¦-

$$p(x'/x) \geq 0, \quad \sum_{x' \in A} P(x'/x) = 1, \qquad (5.6.6.2)$$

for every $x \in A$, such that for any $x' \in A$,

$$\sum_{x \in A} p(x'/x).p(x) = p(x') \qquad (5.6.6.3)$$

The information source $\{ A, \mu \}$ is then called a Markov information source if the probability of any n-dimensional block is given by the equality ¦

$$\mu([x_0, x_1, \ldots, x_n]) = p(x_0).p(x_1/x_0).p(x_2/x_1)\ldots p(x_n/x_{n-1})$$

(5.6.6.4)

The data set A is taken as :-

Message	Codeword	
x_0	0	Huffman code table (C)
x_1	10	
x_2	11	

To conform to the required conditions the choices made are :-

$$p(x_0/x_0) = p(x_1/x_1) = p(x_2/x_2) = 0.75,$$

$$p(x_1/x_0) = p(x_2/x_1) = p(x_0/x_2) =$$

$$p(x_0/x_1) = p(x_1/x_2) = p(x_2/x_0) = 0.125,$$

and $p(x_i/x_j x_k) = p(x_i/x_j)$, implying that decisions about the element x_i under consideration are only dependent on the previous element x_j.

When using maximum likelihood decoding of the Huffman coded bit-stream, the first result of interest is the theoretical probability of failure of such a scheme. The total average probability with which an incorrect decoding occurs, for each code word sequence has therefore to be calculated.

Let, $d(a_j)$ be the average probability of an incorrect decoding of the j^{th} transmitted sequence a_j.

Then,

$$d(a_j) = \sum_{i=1}^{m} P(b_i).P(\hat{a}_j/b_i) \quad , \hat{a}_j \neq a_j \qquad (5.6.6.6)$$

where,

b_i = the i^{th} received sequence.

$\hat{a}_j$ = the maximum likelihood estimate of the transmitted sequence, a_j.

$P(b_i)$ = the probability with which sequence b_i occurs.

$P(\hat{a}_j/b_i)$ = the reverse probability, that $\hat{a}_j$ was sent having received b_i.

m = the number of possible sequences, b_i.

The total average probability of an incorrect decoding is found by taking the average of all $d(a_j)$ i.e. :-

$$\Delta_a = (1/n) \sum_{j=1}^{n} d(a_j) \qquad (5.6.6.7)$$

---

where,

n = the number of possible sequences, a_j.

The number of words and bits in the sequence is varied from 4 words in 4 bits to 4 words in 8 bits.

The schemes outlined in this section, Section 5.6, are thus implemented and results gathered. These results are presented in the next chapter, Chapter 6, and pave the way for the ultimate conclusions.

CHAPTER 6

RESULTS/PERFORMANCE FOR OVERALL SOURCE CODED AND CHANNEL ERROR PROTECTED IMAGE COMMUNICATION SCHEMES.

6.1 Results/Performance with Source Coding Parameter Adjustments Only (Scheme 1).

The average bit-rate of both the zonal and the threshold scheme was varied to obtain average rates from approximately 1.00 bits/pixel to approximately 1.85 bits/pixel and the effect of uniform, random, independent channel errors, occuring at different rates, observed. Sub-image size was also varied, being either 8x8 or 16x16 pixels square.

The results indicate that it is only necessary to present those obtained at the extremes of the bit-rate range, i.e. at 1.00 bits/pixel and 1.85 bits/pixel. At rates between these extremes the results show similar effects which lie in the vicinity of those presented here, achieving a performance between the extremes. The channel error rate is increased in magnitude from

1 in 10^{6} to 1 in 10^{2}, by a factor of ten for each result. It is helpful to plot the average, normalized sample mean squared error against the log of the channel error rate, rather than against the channel error rate itself. These results are given in the graphs shown in Figures 6.1.1 and 6.1.2.

Figure 6.1.1 and 6.1.2 thus indicate results obtained for the different schemes when the sub-image size is fixed at 8x8 or 16x16 pixels square respectively. Table A1.6.1.1 in the Appendix gives the numerical results of values for the graphs. Both graphs show similar trends. Looking at either one, it is seen that the higher average bit-rate results in lower overall distortion, provided the channel error rate is relatively low, i.e. specifically < 1 in 10^{3}, with the threshold coding technique predominating over the zonal technique. However, at relatively high channel error rates, particularly > 1 in 10^{3}, all schemes show a definite inability to keep the distortion low, and at an acceptable level. A comparison of the two graphs shows that the curves for a set sub-image size of 16x16 pixel square have a steeper incline as they approach the channel error rate of 1 in 10^{2}, thus indicating that at increasingly higher channel error rates a smaller sub-image size helps to reduce the escalating increase in distortion.

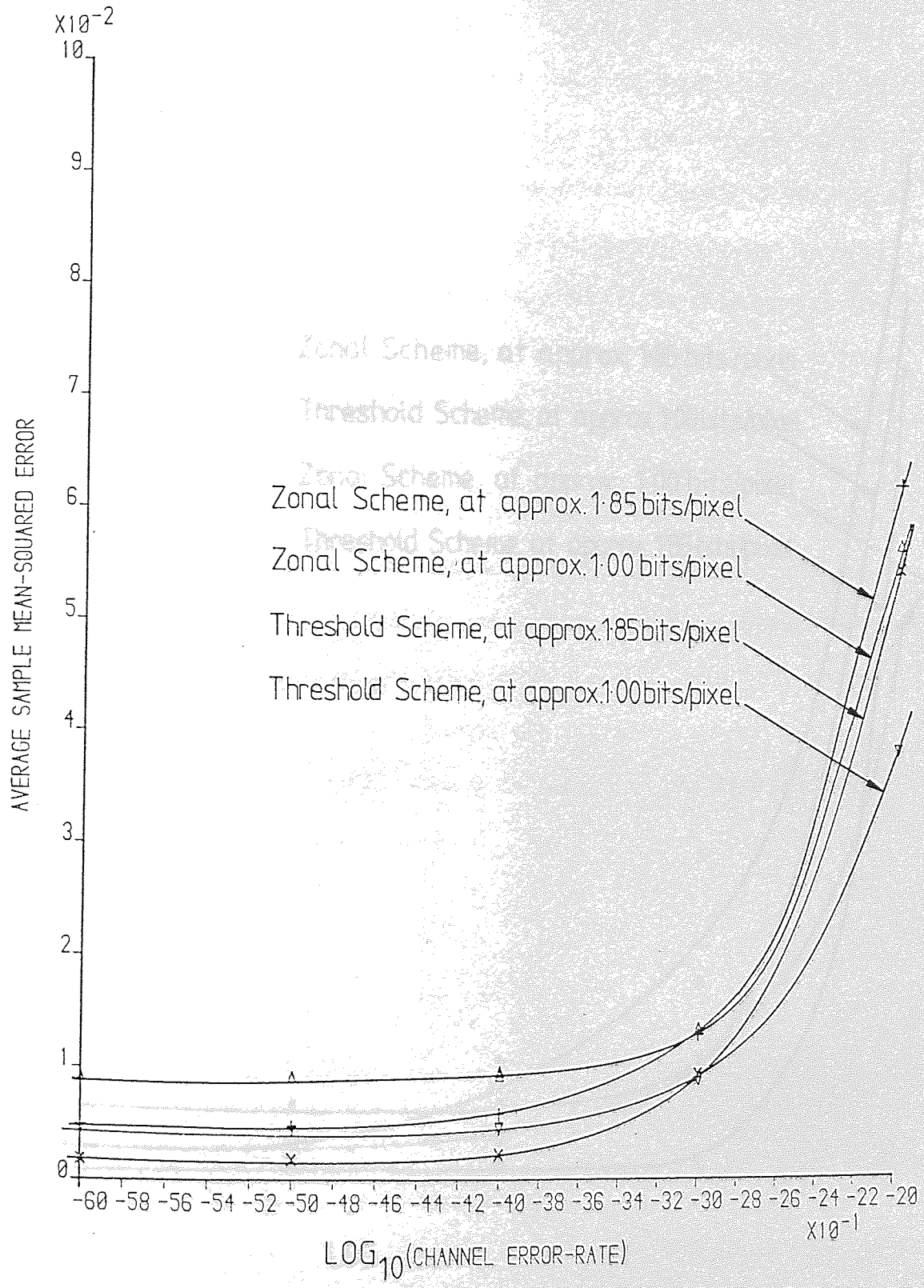


Figure 6.1.1 Distortion versus LOG_{10}(Channel Error-rate) for Sub-image Size 8 × 8 Pixels Square

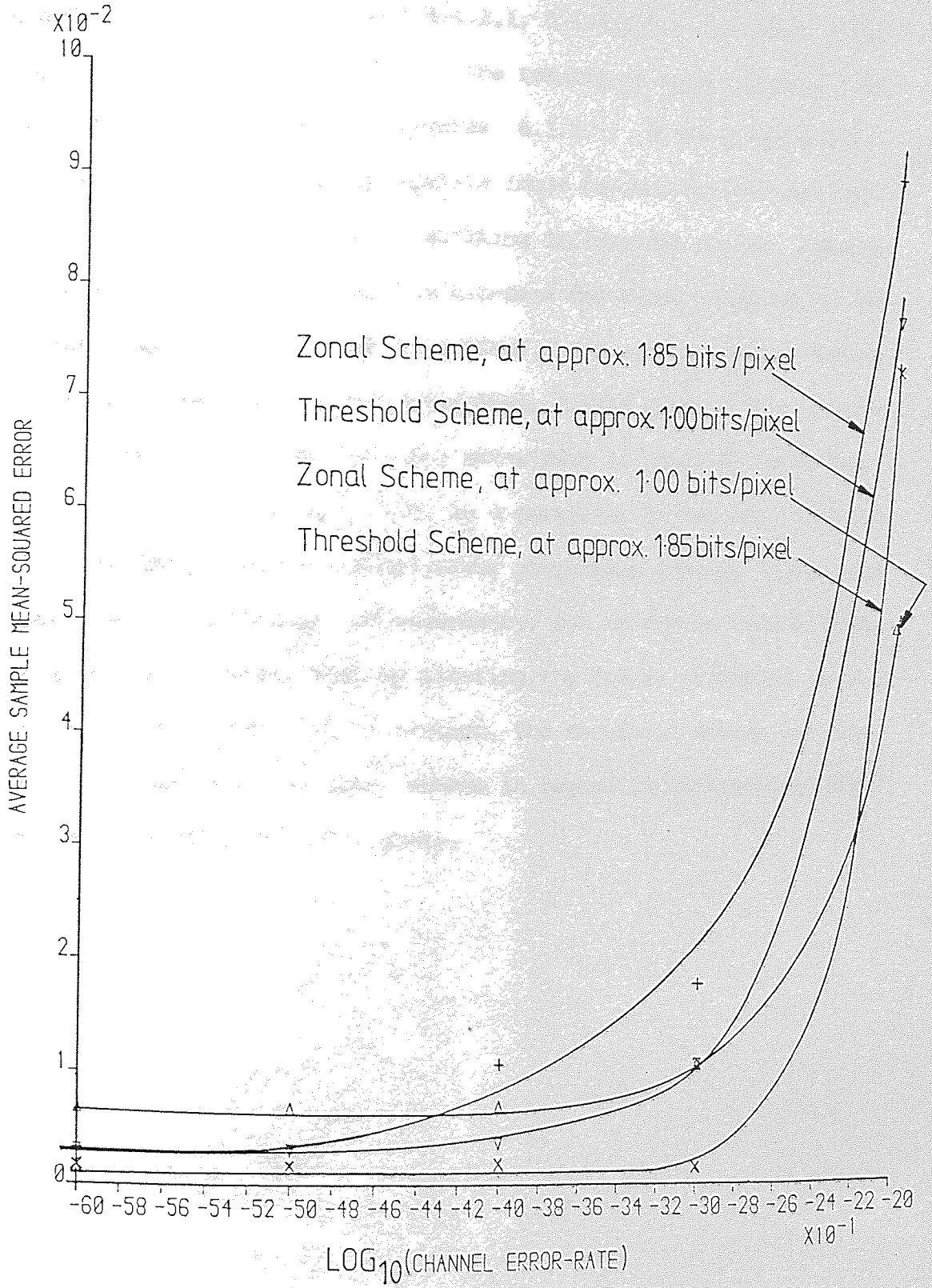


Figure 6.1.2 Distortion versus LOG_{10}(Channel Error-rate) for Sub-image Size 16×16 Pixels Square

The more practical result can be seen from the photographs of the reconstructed images. Refer to Figures 4.4.4.1.1, 4.4.4.1.2 (zonal) and 4.4.4.2.1, 4.4.4.2.2 (threshold) for results at 1.ØØ bits/pixel. The results at approximately 1.85 bits/pixel are shown in Figures 6.1.3 (zonal) and 6.1.4 (threshold). If the appropriate image reconstructions are compared, there appears to be no striking differences between results obtained at the lower average bit-rate and those obtained at the higher average rate of 1.85 bits/pixel. The average source coding bit-rate is approaching 2.Ø bits/pixel, double that of the acceptable level, and the resulting bit-stream is then 5Ø% redundant. Further increase in bit-rate, as a parameter adjustment, is thus not considered, as the channel error protection schemes themselves introduce this amount of redundancy, but in a more useful form. It is noted, however, that by altering the E.O.S. parameter to the SH parameter with high protection, the threshold scheme has been put on a par with the zonal scheme in regard to restricting channel error effects to local areas.

(a) (b)

Figure 6.1.3 Unprotected, Zonal, Source Coded Data (1.85 Bits/pixel) Subject to Channel Error Rates of 1 in 10000, 1 in 1000, 1 in 100, for Sub-image Sizes of (a) 16x16 (b) 8x8 Pixels Square.

(a) (b)

Figure 6.1.4 Unprotected, Threshold, Source Coded Data (1.85 Bits/pixel) Subject to Channel Error Rates of 1 in 10000, 1 in 1000, 1 in 100 for Sub-image Sizes of (a) 16x16 (b) 8x8 Pixels Square.

6.2 Results/Performance of Image Coding Systems with Equal Importance Block Coding (Scheme 2).

Every bit of the source coded bit-stream is protected to a certain degree by use of either the (7, 4) or the (15, 11) Hamming block code. The expanded bit-stream is then subject to uniform, random, independent channel errors that occur at rates ranging from 1 in 10^{6} to 1 in 10^{2}.

The main result is that for all error rates up to and including 1 in 10^{3}, upon decoding, very little or no further distortion is introduced into the reconstructed image with either the zonal or the threshold schemes. This is ideal, but has been obtained at the expense of increased bit-rate. In the case of the (15, 11) Hamming code, approximately 27% of the bits are redundant and, in the case of the (7, 4) Hamming code, the redundancy is approximately 43% of the total bit-stream.

With an increase of a factor of ten in the channel error rate, bringing it to 1 in 10^{2}, the protection tends to break down and error effects can again be seen in the reconstructed image. Even the increase in protection afforded by the (7, 4) Hamming code, over the (15, 11), is not sufficient to eliminate all obvious error effects. When plotting graphs of distortion against total average bit-rate, the effect on the distortion at this error rate is seen quite clearly, i.e. in Figures 6.2.1 and 6.2.2.

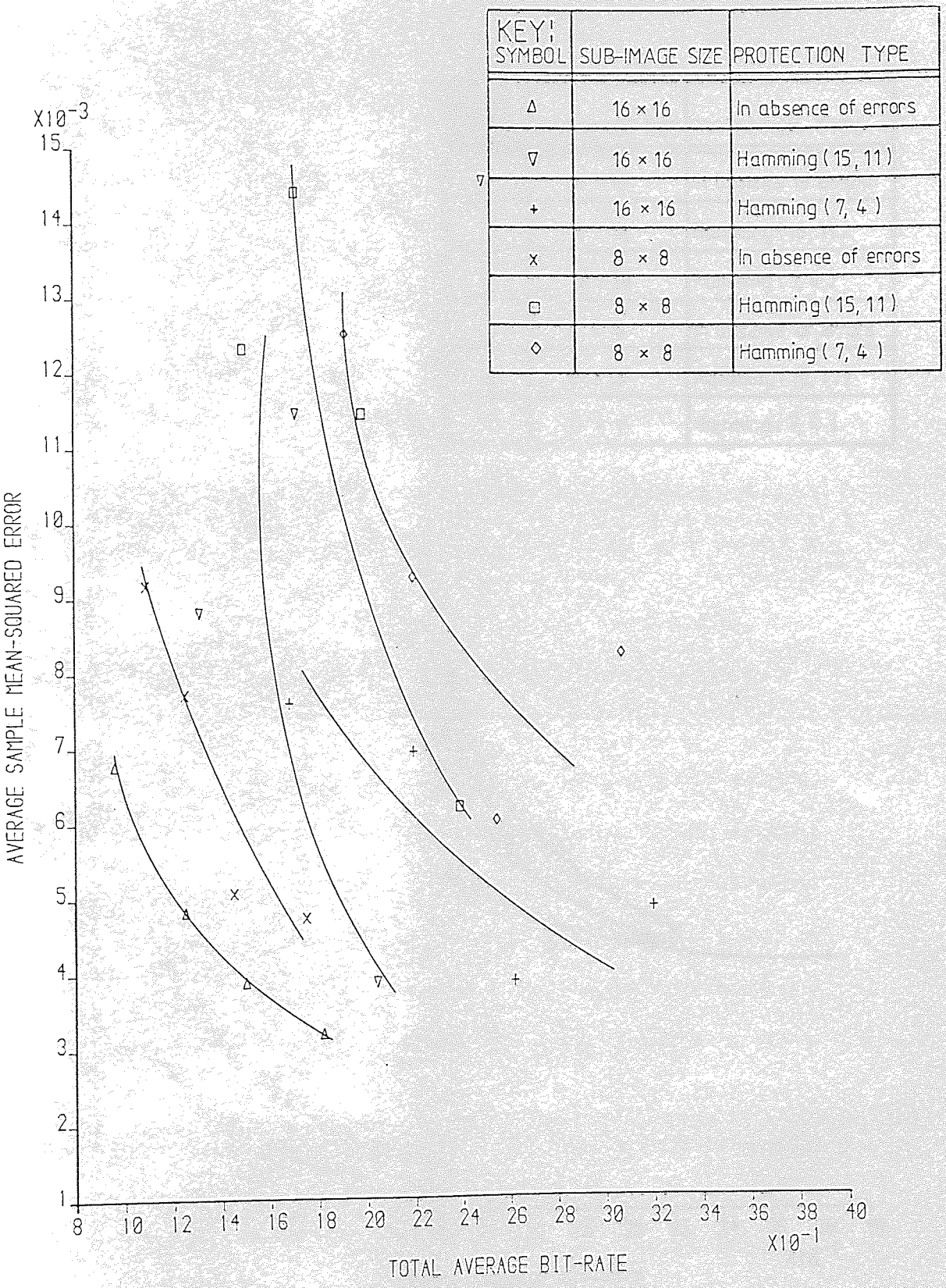


Figure 6.2.1 Zonal Scheme, Distortion versus Total Average Bit-rate for Channel Error Rate 1 in 100

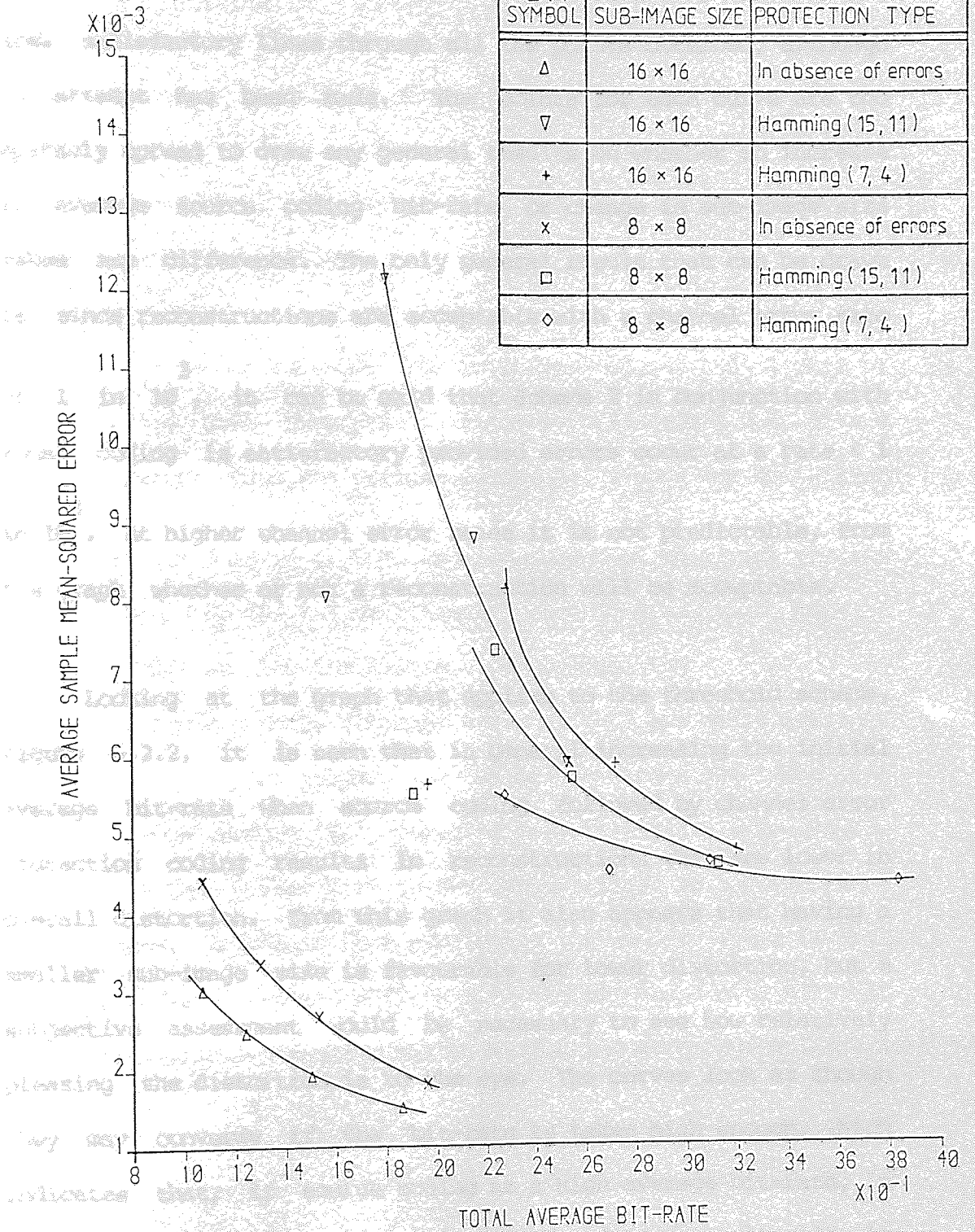


Figure 6.2.2 Threshold Scheme, Distortion versus Total Average Bit-rate for Channel Error Rate 1 in 100.

With the zonal scheme, Figure 6.2.1, it is very difficult to draw satisfactory lines through all the points obtained, although an attempt has been made. The points for each curve are too sparsely spread to draw any general results on whether an increase in average source coding bit-rate or change in sub-image size makes any difference. The only general result that can be drawn is, since reconstructions are acceptable with a channel error rate of 1 in 10^3, it can be said that Scheme 2 in conjunction with zonal coding is satisfactory provided errors occur at a rate < 1 in 10^3. At higher channel error rates it is not predictable, from the graph, whether or not a reconstruction will be acceptable.

Looking at the graph that applies to the threshold scheme, Figure 6.2.2, it is seen that in general increasing the initial average bit-rate when source coding, followed by channel error protection coding results in reconstructions that are lower in overall distortion. From this graph it also appears that having a smaller sub-image size is favourable for lower distortion, but a subjective assessment would be necessary to see how relatively pleasing the distortion is to the eye. The curves look as though they may converge if the bit-rate is taken high enough, which indicates that, if source coding at a high average bit-rate, in conjunction with equal importance block code error protection is carried out, other parameter adjustments tend to make little

difference. It is noted that the total average bit-rate, with the threshold scheme, includes taking into account the headers (SH words) and their repetitive error protection method.

The numerical results for the two graphs are found in the Appendix as Tables A1.6.2.1.

Image reconstructions for an initial source coding rate of approximately 1.85 bits/pixel, Figures 6.2.3 and 6.2.4, give a better insight into the effects of channel errors for the higher error rate of 1 in 10^2. Both figures show that the extra protection afforded by the (7, 4) Hamming code does eliminate more of the error effects than the alternative of using the (15, 11) Hamming code, although error effects are still evident. In contrast to the graphs, the use of the larger sub-image size produces more subjectively pleasing results, possibly due to the fact that the error effect for one sub-image is spread over a larger area, lessening it's subjective influence. Comparison of the appropriate reconstructions of Figure 6.2.3 with those of Figure 6.2.4 indicate that, although, for the given channel error rate of 1 in 10^2, both schemes result in approximately the same number of sub-images in error, for a given sub-image size, those of the threshold scheme are not as disturbing as those of the

Sub-image Size = 16x16 | Sub-image Size = 8x8

Equal Importance Hamming(7,4) Protection

Sub-image Size = 16x16 | Sub-image Size = 8x8

Equal Importance Hamming(15,11) Protection

Figure 6.2.3 Zonal Image Coding with Equal Importance Block Coding, subject to Channel Errors at a Rate of 1 in 100.

Sub-image Size = 16x16

Sub-image Size 8x8

Equal Importance Hamming(7,4) Protection

Sub-image Size 16x16

Sub-image Size 8x8

Equal Importance Hamming(15,11) Protection

Figure 6.2.4 Threshold Image Coding with Equal Importance Block Coding Subject to Channel Errors at a Rate 1 in 100.

zonal scheme. This appears to be due to the fact that the sub-images in error with the threshold scheme often manage to retain some of the original patterns, or at least some of the original shading, whereas those of the zonal scheme are mostly altered much more drastically to give reconstructed sub-images totally incompatible with neighbouring sub-images.

Although both Figure 6.2.3 and 6.2.4 indicate that at the high channel error rate of 1 in 10^2 error effects will often occur, a comparison with unprotected source coded data subject to errors at the same error rate, i.e. Figures 6.1.3 and 6.1.4, show that a very significant improvement in reconstructed image quality has been acheived.

6.3 Results/Performance of Image Coding Systems with Equal Importance Convolutional Coding (Scheme 3).

The convolutional codes CC(7) and CC(11), described in Section 5.4.2, are used to protect every bit of the source coded bit-stream. Again the expanded bit-stream is subject to uniform, random, independent channel errors that occur at rates ranging from 1 in 10^6 to 1 in 10^2. The initial source coding rate is

varied from approximately 1.Ø to 2.Ø bits/pixel, and the effect of different sub-image sizes once again observed.

With the zonal or the threshold schemes, very little or no further distortion is introduced into the reconstructed image for all error rates up to, and including 1 in $1Ø^3$. Both codes, CC(7) and CC(11), introduce 5Ø% redundancy into the total bit-stream and this is the price paid for achieving protection against channel errors for this scheme.

At error rates around 1 in $1Ø^2$, the protection afforded by this method of convolutional coding also breaks down with error effects noticeable in the reconstructed images. Graphs of distortion against total average bit-rate, for a channel error rate of 1 in $1Ø^2$, show the effect of having different sub-image sizes and varying the initial average source coding bit-rate.

With the zonal scheme, Figure 6.3.1, it is easier to estimate curves through the points obtained than it was in the case of the corresponding graph when block coding (Figure 6.2.1) indicating perhaps, more conformity to a smooth curve relationship. The curves tend to indicate that lower distortion can be achieved by increasing the initial source coding bit-rate

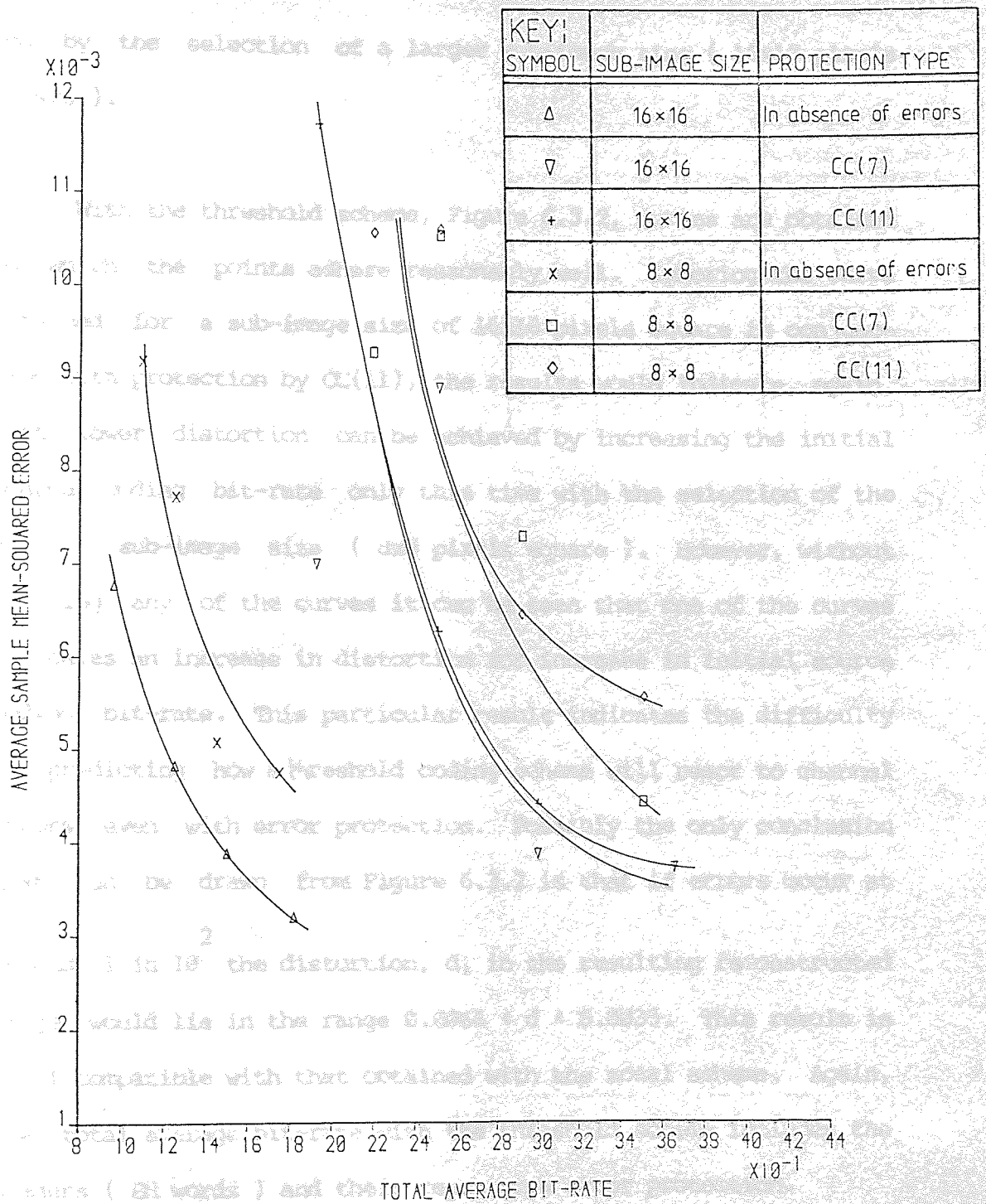


Figure 6.3.1 Zonal Scheme, Distortion versus Total Average Bit-rate for Channel Error Rate 1 in 100.

and by the selection of a larger sub-image size (16x16 pixels square).

With the threshold scheme, Figure 6.3.2, curves are obtained to which the points adhere reasonably well. Ignoring the curve obtained for a sub-image size of 16x16 pixels square in conjunction with protection by CC(11), the results would indicate, again, that lower distortion can be achieved by increasing the initial source coding bit-rate only this time with the selection of the smaller sub-image size (8x8 pixels square). However, without ignoring any of the curves it can be seen that one of the curves indicates an increase in distortion for increase in initial source coding bit-rate. This particular result indicates the difficulty in predicting how a threshold coding scheme will react to channel errors even with error protection. Possibly the only conclusion that can be drawn from Figure 6.3.2 is that if errors occur at around 1 in 10^2 the distortion, d, in the resulting reconstructed image would lie in the range $0.0066 < d < 0.0035$. This result is still compatible with that obtained with the zonal scheme. Again, the total average bit-rate with the threshold scheme includes the headers (SH words) and their repetitive error protection.

The numerical results for the two graphs are found in the Appendix as Tables A1.6.3.1.

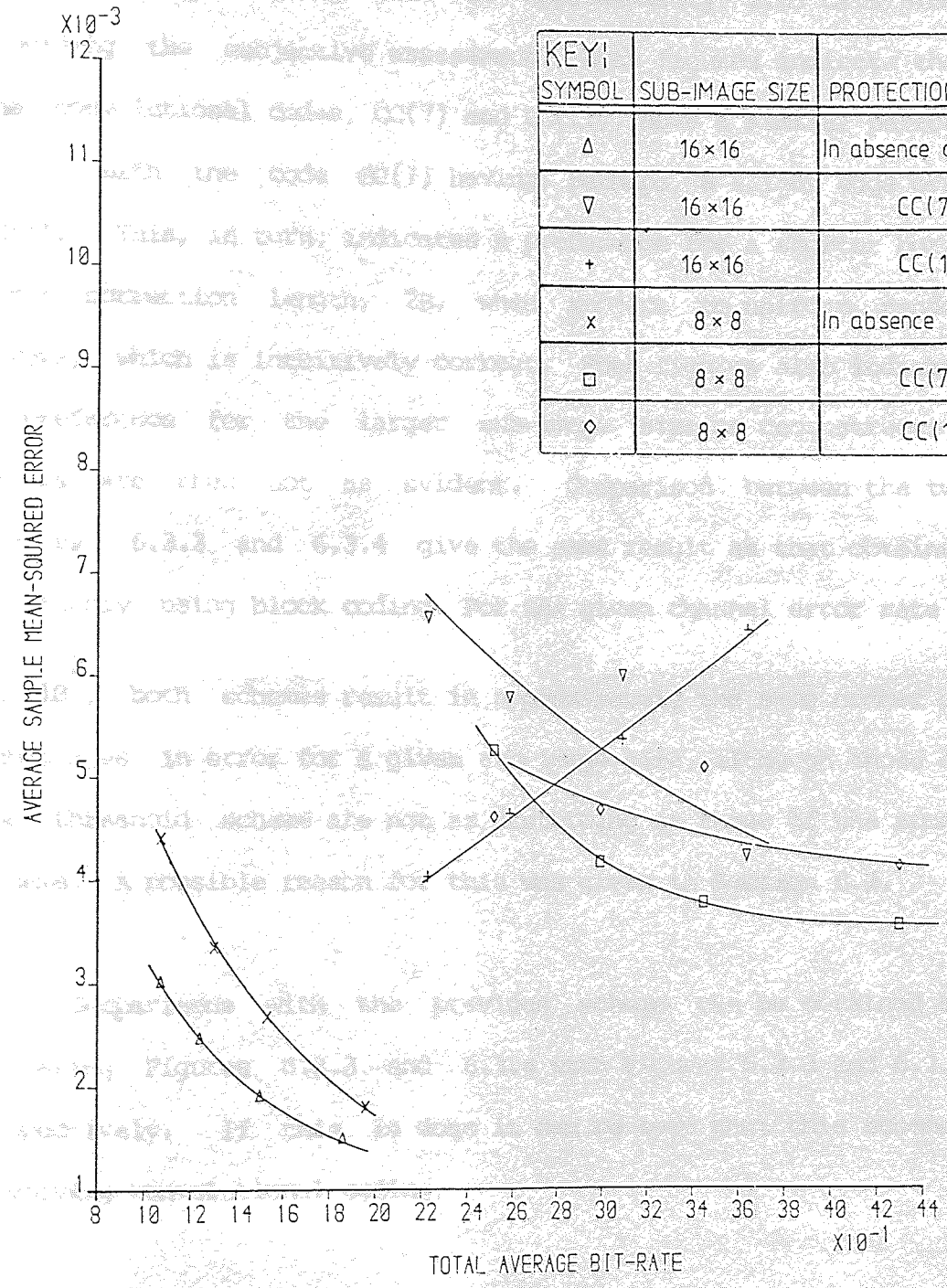


Figure 6.3.2 Threshold Scheme, Distortion versus Total Average Bit-rate for Channel Error Rate 1 in 100.

Figures 6.3.3 and 6.3.4 show typical image reconstructions for an initial coding rate of approximately 1.85 bits/pixel providing the subjective assessment. Both figures indicate that the convolutional codes, CC(7) and CC(11), have a similar performance, with the code CC(7) having, perhaps, a slight edge over CC(11). This, in turn, indicates a preference for a shorter burst error correction length, 2B, when subject to uniform, random errors, which is intuitively correct. Both figures also indicate a preference for the larger sub-image size as reconstruction errors are then not as evident. Comparison between the two Figures, 6.3.3 and 6.3.4 give the same result as that obtained previously using block coding. For the given channel error rate 1 in 10^{2}, both schemes result in approximately the same number of sub-images in error for a given sub-image size, although those of the threshold scheme are not as disturbing as those of the zonal scheme. A possible reason for this was given in Section 6.2.

Comparisons with the previous scheme can be obtained by comparing Figures 6.3.3 and 6.3.4 with Figures 6.2.3 and 6.2.4 respectively. If this is done it can be seen that this scheme, involving convolutional coding,

Sub-image Size = 16x16 | Sub-image Size 8x8

Equal Importance Convolution, CC(7), Protection

Sub-image Size = 16x16 | Sub-image Size = 8x8

Equal Importance Convolution, CC(11), Protection

Figure 6.3.3 Zonal Image Coding with Equal Importance Convolution Coding Subject to Channel Errors at a Rate 1 in 100.

Sub-image Size = 16x16

Sub-image Size = 8x8

Equal Importance Convolution, CC(7), Protection

"AS ABOVE"

Sub-image Size = 16x16

Sub-image Size = 8x8

Equal Importance Convolution, CC(11), Protection.

Figure 6.3.4 Threshold Image Coding with Equal Importance Convolutional Coding Subject to Channel Errors at a Rate 1 in 100.

has a better performance than either of the block codes used, in terms of image reconstruction. The order of error protection code preference, based on these results is :-

Code	Amount of Redundancy Introduced.
Hagelbarger Convolutional code, CC(7)	50%
" " " , CC(11)	50%
Hamming Block Code, (7, 4)	43%
" " " , (15, 11)	27%

with the performance between CC(7) and CC(11) very similar. The above verifies that planned increased redundancy does result in increased reconstructed image fidelity.

Finally, when comparing Figures 6.3.3 and 6.3.4, with the relevant results for unprotected code given in Figures 6.1.3 and 6.1.4, the very significant improvement in reconstructed image quality is, once again, clearly seen.

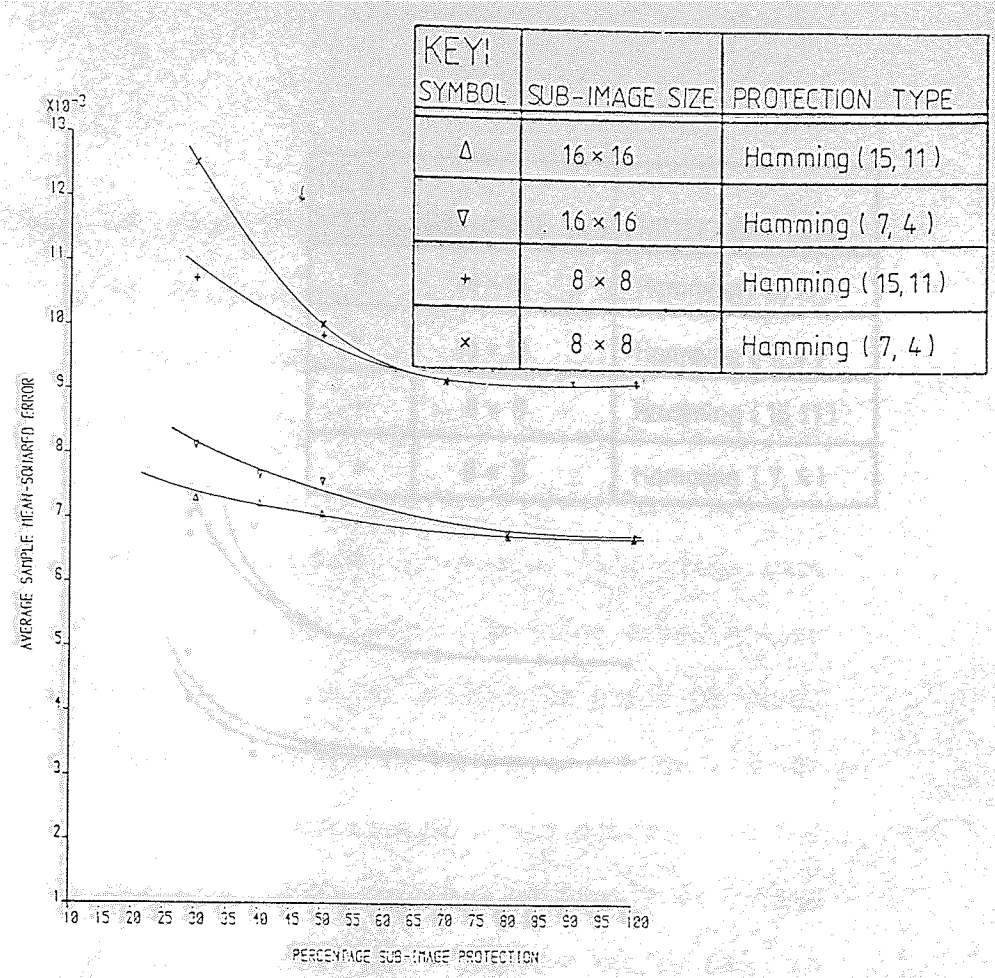
6.4 Results/Performance of Image Coding Systems with Percentage Sub-image Block Coding (Scheme 4).

As outlined in Section 5.6.4, Scheme 4, only a certain percentage of the bits of any sub-image are protected through the use of the (7, 4) or (15, 11) Hamming Code. The percentage of

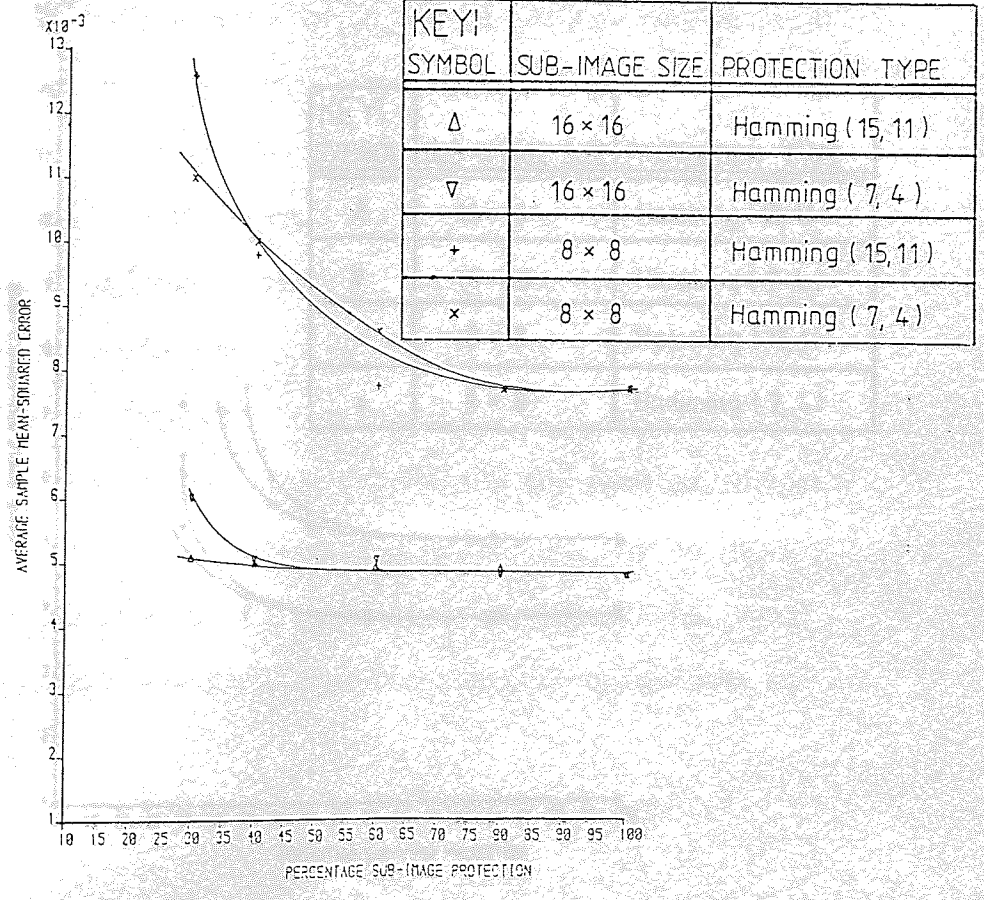
bits to be protected is varied from 100%, i.e. protection of all bits, to around 30% protection of the bits of any one sub-image. All bits, protected and unprotected, are then subject to uniform, random, independent channel errors that occur at a rate of either 1 in 10^3 or 1 in 10^2. The initial source coding rate is varied from approximately 1.00 to 1.85 bits/pixel and the sub-image size is set at either 16x16 or 8x8 pixels square.

With the zonal scheme, results for a channel error rate of 1 in 10^3 are shown in Figures 6.4.1 and 6.4.2.

The graphs given in these figures show how the distortion in the reconstructed image varies with the percentage sub-image protection. In general, the trend shown is that expected, that is, as the percentage protection is decreased the distortion increases in a non-linear fashion. All four graphs show a general trend of a very gradual rise in the distortion when varying the applied protection from 100% of the sub-images up to around 40 to 50% protection. Here-after the rise in distortion is, generally, much steeper. Later results (subjective photographs, Figure 6.4.4) indicate that, at the 40 to 50% protection mark, image

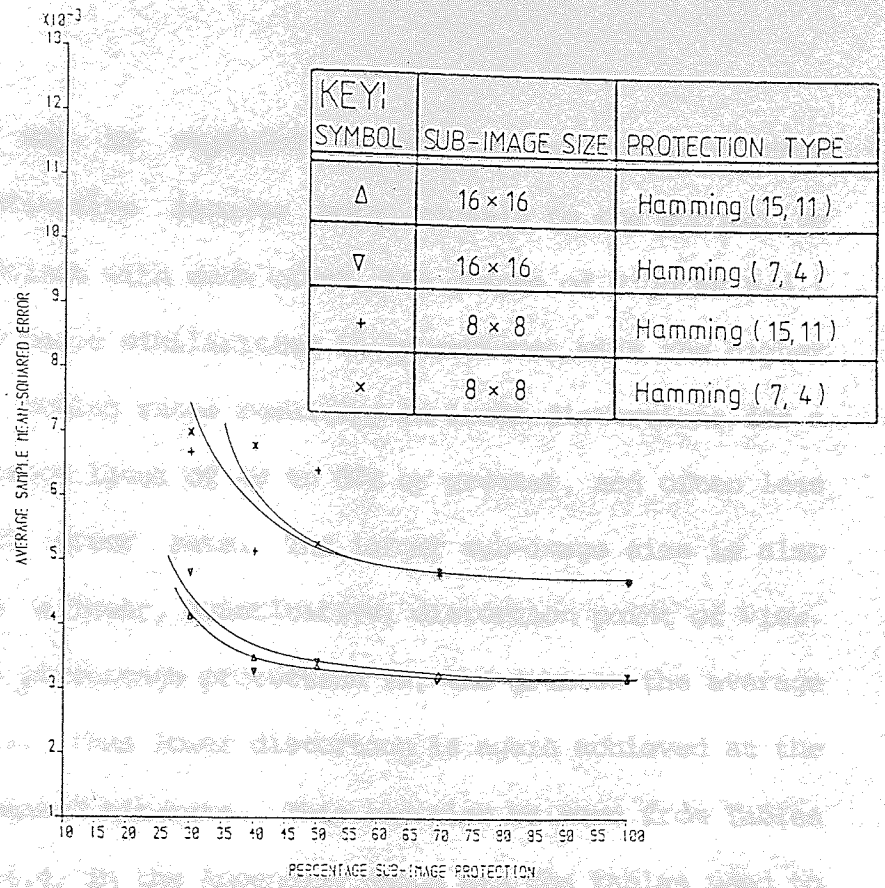


(a) Initial, Source Coding, Rate = 1·00 Bits/pixel

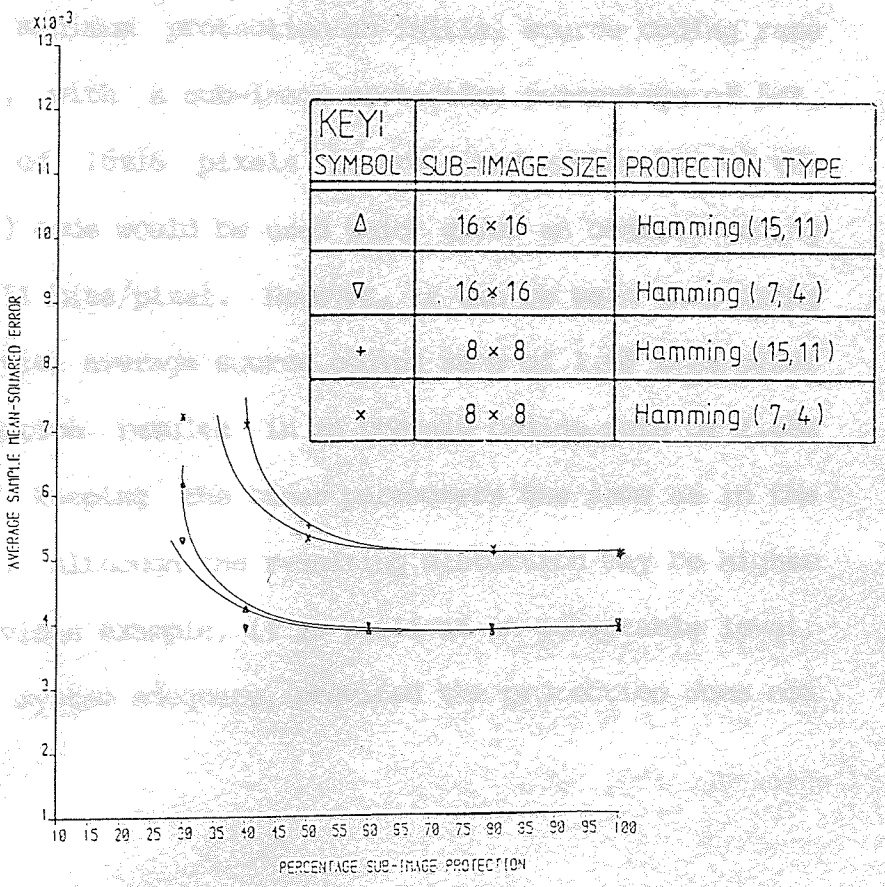


(b) Initial, Source Coding, Rate = 1·25 Bits/pixel

Figure 6.4.1 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1000.



(a) Initial, Source Coding, Rate = 1·85 Bits/pixel



(b) Initial, Source Coding, Rate = 1·50 Bits/pixel

Figure 6.4.2 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1000.

reconstructions may be regarded as becoming unacceptable, thus tying the quantitative results more closely to the subjective ones. In comparison with each other, the graphs of Figures 6.4.1 and 6.4.2 show basic similarities in appearance, with the higher initial source coding rates resulting in lower distortions for a sub-image protection level of 40 to 50% or greater, and often less at this channel error rate. The larger sub-image size is also favourable from a lower, quantitative, distortion point of view. The higher the percentage protection is, the greater the average bit-rate becomes, thus lower distortion is again achieved at the expense of increased bit-rate. This can also be seen from Tables A1.6.4.1 - A1.6.4.4, in the Appendix, which are the tables used to plot the graphs of Figures 6.4.1 and 6.4.2. For the lowest distortion and minimum protection an initial source coding rate 1.85 bits/pixel, with a sub-image protection percentage of 50%, sub-image size of 16x16 pixels square, and employment of the Hamming (7, 4) code would be used which gives an overall coding rate of about 2.51 bits/pixel. However, it can be seen that using the lowest initial average source coding rate of 1.00 bits/pixel and 100% protection results in an overall coding rate of 1.682 bits/pixel when keeping the other parameters the same as in the previous example. Although the resulting distortion may be higher than in the previous example, it is still at an acceptable level, thus making this system adequate, provided the protection does not

break down, as this tends to push the distortion immediately to an unacceptable level.

Figure 6.4.3 shows the subjective results obtained for the zonal scheme with 40% sub-image protection, channel error rate of $1 \text{ in } 10^{3}$, and an initial source coding rate of approximately 1.85 bits/pixel. The advantage of using the larger sub-image size can be seen from this figure. With the smaller sub-image size there are fewer coefficients per sub-image, which appears to make each one relatively more important to the sub-image than is the case with the more plentiful coefficients of the larger sub-image. If this is true then it follows that with the smaller sub-image size a greater percentage of that sub-image's bits would require protection to prevent noticeable error effects. Hence the noticeable differences between the reconstructions. With the reconstructed images utilizing the smaller sub-image size, the increased protection afforded by the Hamming (7, 4) code over the Hamming (15, 11) code is more obvious.

The quantitative results for the zonal scheme, in conjuction with the channel error rate of $1 \text{ in } 10^{2}$, are shown in Figures 6.4.4 and 6.4.5. It has already been seen (Scheme 2) that, at this relatively high channel error rate, even 100% sub-image protection

Using Hamming(7,4)

Using Hamming(15,11)

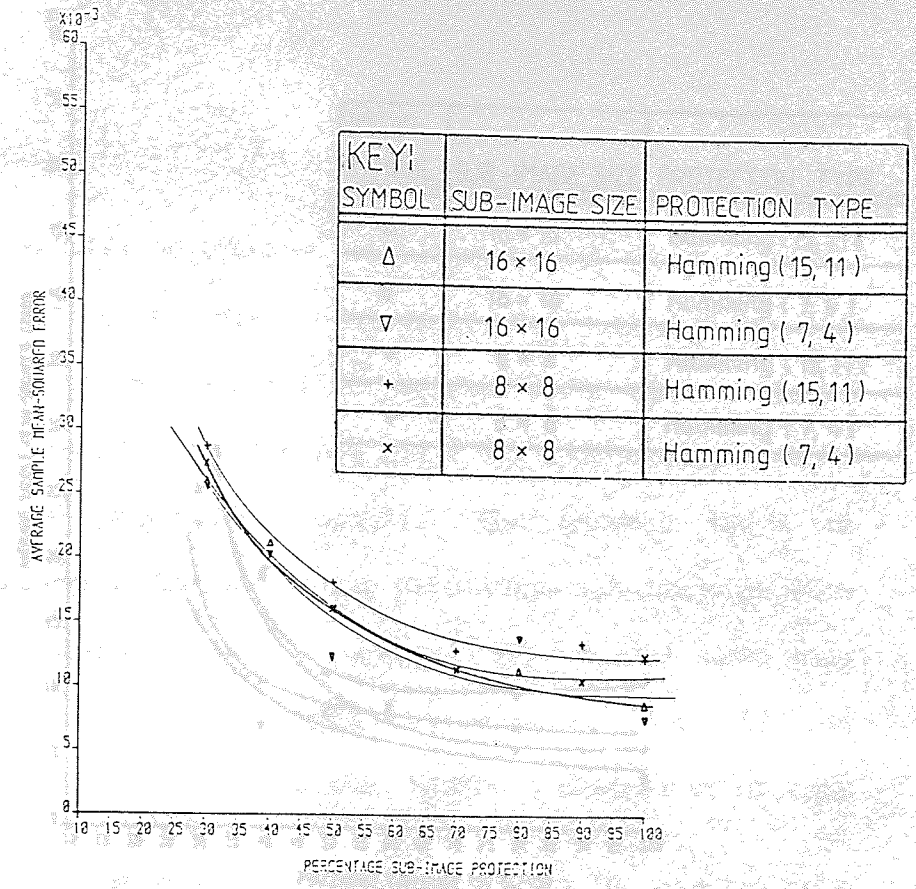
Sub-image Size = 16x16 Pixels Square

Using Hamming(7,4)

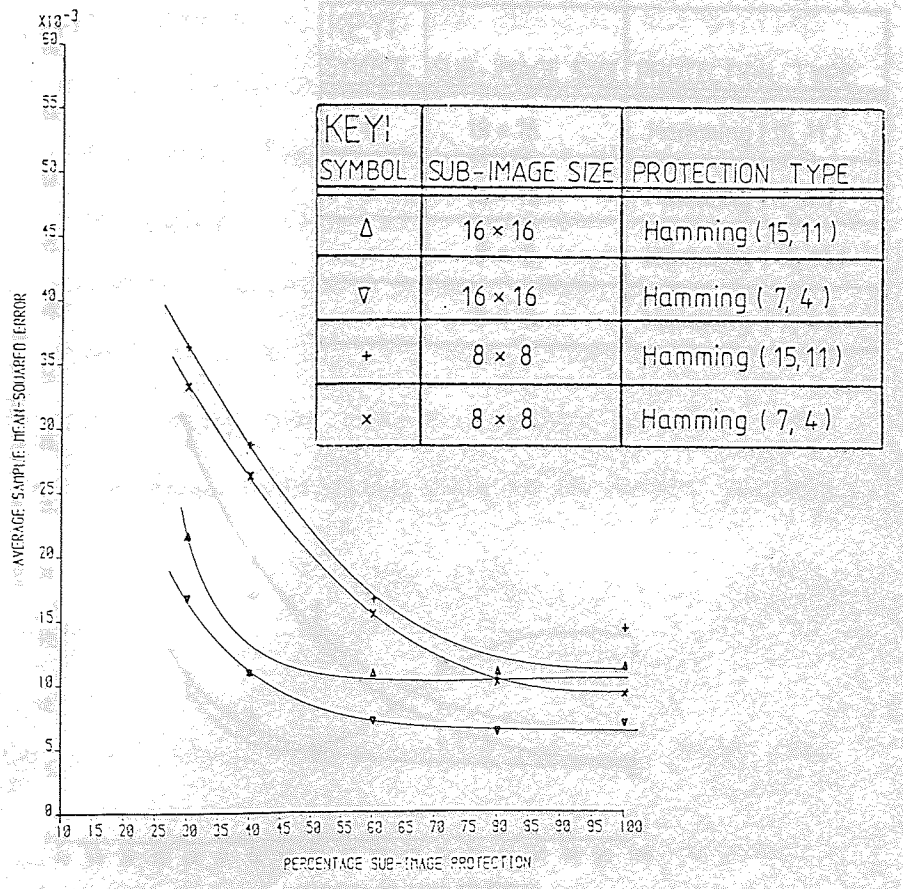
Using Hamming(15,11)

Sub-image Size = 8x8 Pixels Square

Figure 6.4.3 Zonal, Image Reconstructions at 40% S.I.P., Channel Error Rate = 1 in 1000, Initial Source Coding Rate = 1.85 Bits/pixel

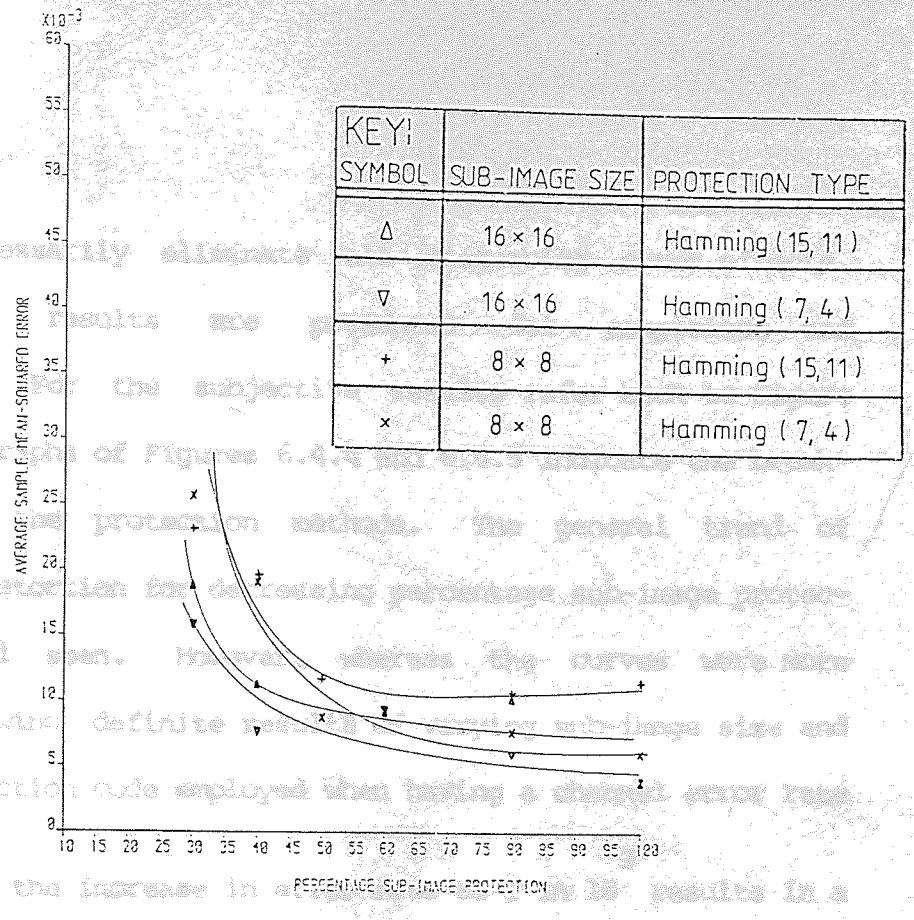


(a) Initial, Source Coding, Rate = 1·00 Bits/pixel

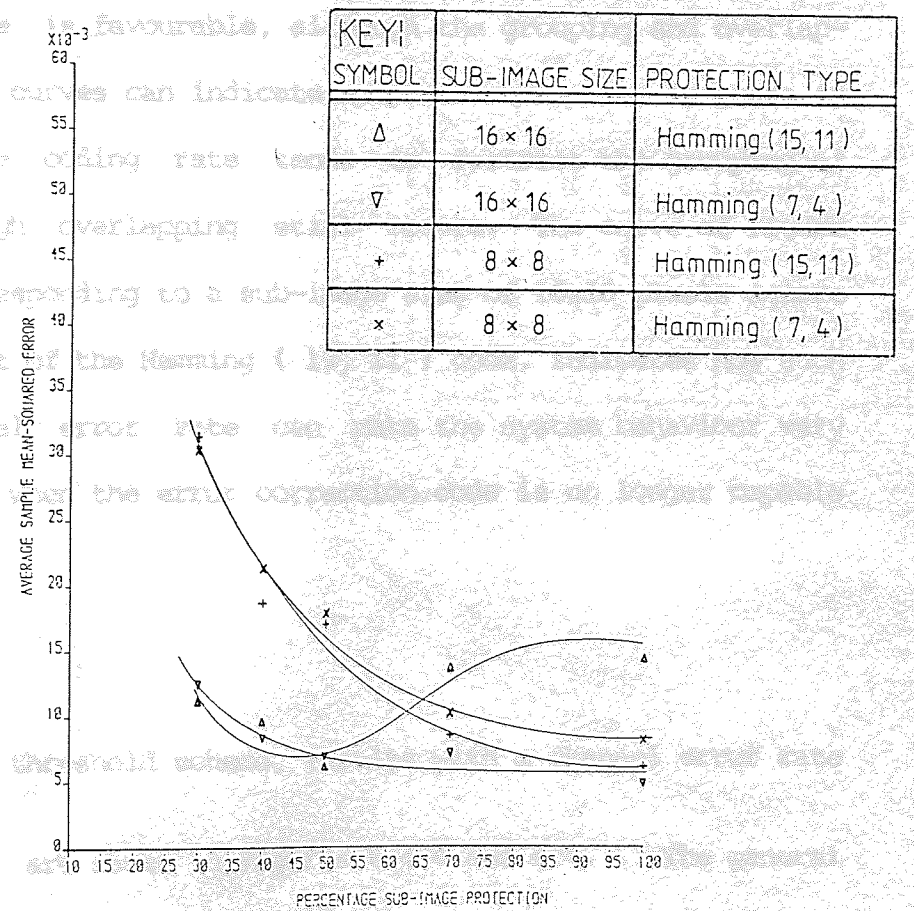


(b) Initial, Source Coding, Rate = 1·25 Bits/pixel

Figure 6.4.4 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 100



(a) Initial, Source Coding, Rate = 1·50 Bits/pixel

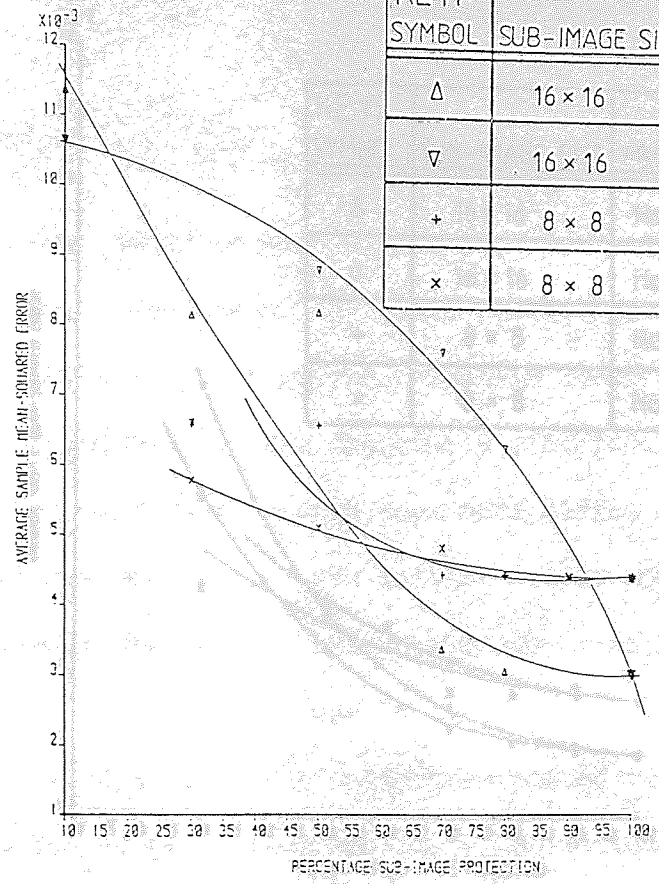


(b) Initial, Source Coding, Rate = 1·85 Bits/pixel

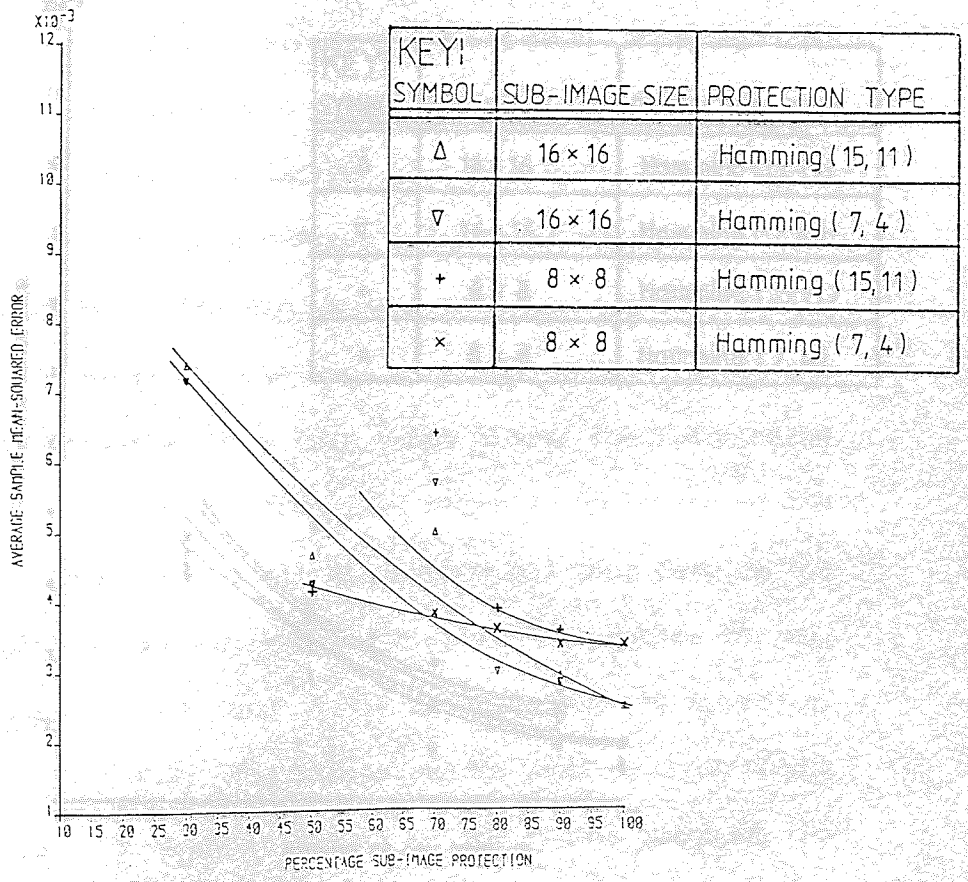
Figure 6.4.5 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 100

does not necessarily eliminate all disturbing error effects. However, the results are presented for comparison and completeness. For the subjective results refer back to Figure 6.2.3. The graphs of Figures 6.4.4 and 6.4.5 indicate the breaking down of the protection methods. The general trend of increasing distortion for decreasing percentage sub-image protection is still seen. However, whereas the curves were more distinct, showing definite results of varying sub-image size and type of protection code employed when having a channel error rate of 1 in 10^3, the increase in error rate to 1 in 10^2 results in a grouping and often overlapping of the curves obtained. The larger sub-image size is favourable, although the grouping and overlapping of the curves can indicate no preference. The increase in initial source coding rate tends to restrict the grouping of curves although overlapping still occurs. The curve of Figure 6.4.5(b), corresponding to a sub-image size of 16x16 pixels square and employment of the Hamming (15, 11) code, indicates how such a high channel error rate can make the system behaviour very unpredictable when the error correction code is no longer capable of coping.

With the threshold scheme, results with a channel error rate of 1 in 10^3 are shown in Figures 6.4.6 and 6.4.7. The general

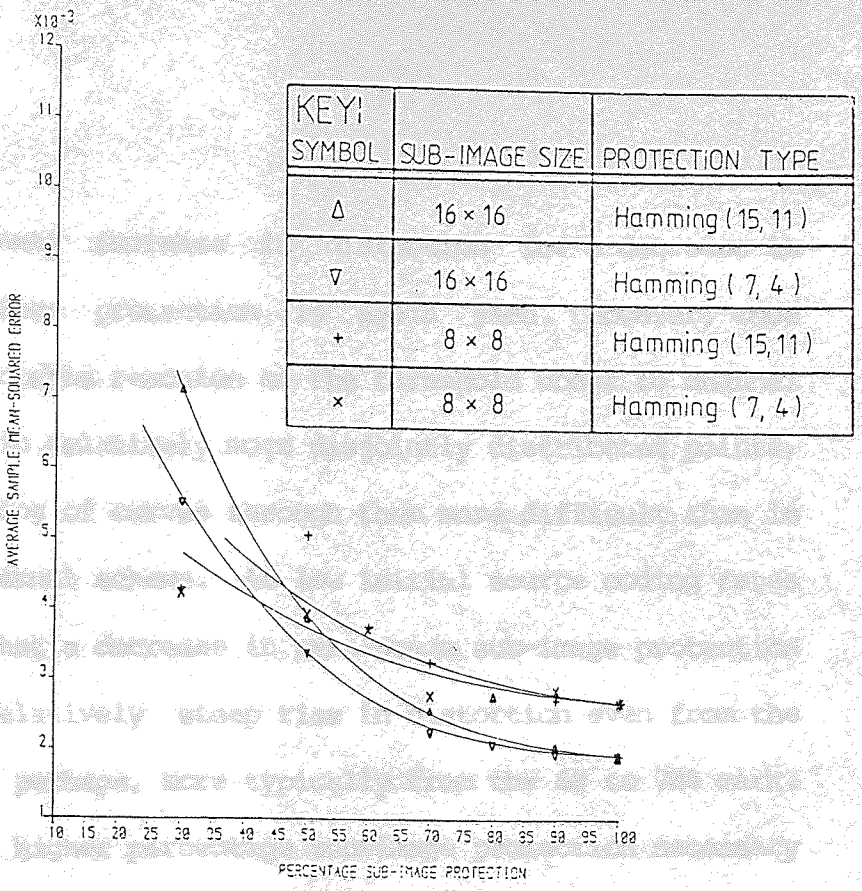


(a) Initial, Source Coding, Rate = 1·00 Bits/pixel

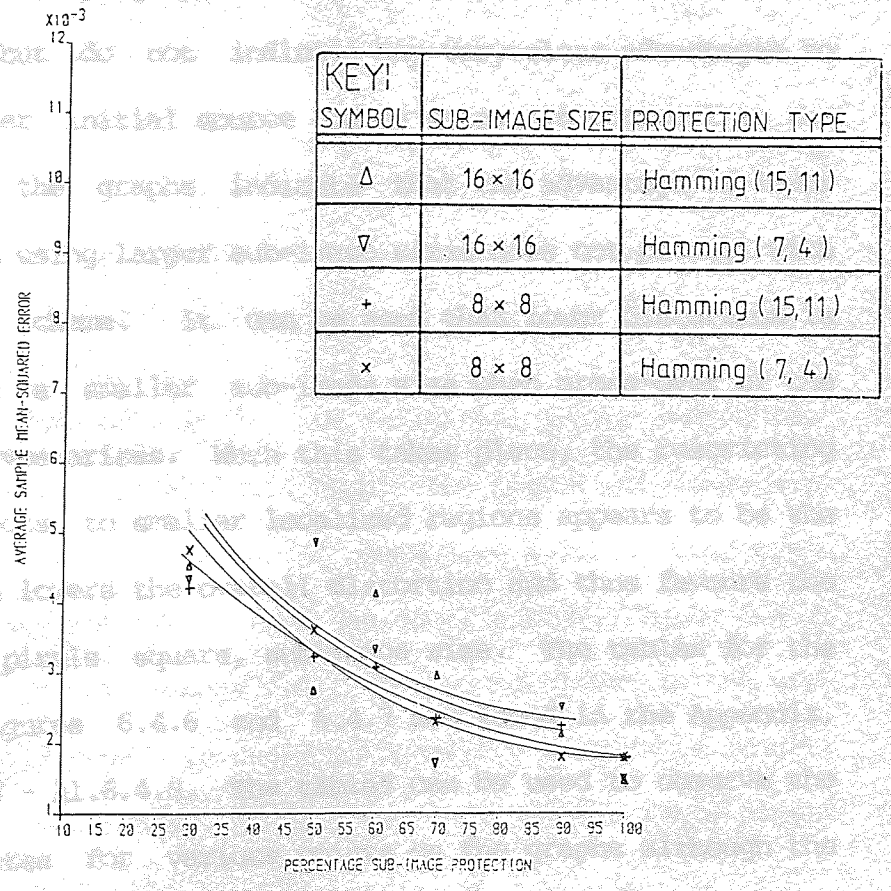


(b) Initial, Source Coding, Rate = 1·25 Bits/pixel

Figure 6.4.6 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1000



(a) Initial, Source Coding, Rate = 1·50 Bits/pixel



(b) Initial, Source Coding, Rate = 1·85 Bits/pixel

Figure 6.4.7 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1000

trend of non-linear increase in distortion for a decrease in percentage sub-image protection is again seen. However, once again the unpredictable reaction of the threshold coder to channel errors results in relatively more disjointly distributed points, making the drawing of curves through them more difficult than in the case of the zonal scheme. At low initial source coding rates it can be seen that a decrease in percentage sub-image protection results in a relatively steep rise in distortion even from the 100% mark though, perhaps, more typically from the 60 to 70% mark. This indicates a higher percentage sub-image protection necessary to keep resulting reconstructed images acceptable. In comparison with each other, the graphs of Figures 6.4.6 and 6.4.7 show some similarities, but do not indicate any very clear advantages to having a higher initial source coding rate. In contrast to the zonal scheme, the graphs indicate that the advantage of lower distortion when using larger sub-image sizes does not prevail with the threshold scheme. It can be seen that lower distortion is achieved using a smaller sub-image size when cross-over of the appropriate curves arises. When this takes place, the restricting of error effects to smaller localized regions appears to be the property which lowers the overall distortion and thus favours the smaller, 8x8 pixels square, sub-image size. The tables for the results of Figures 6.4.6 and 6.4.7 are found in the Appendix, tables A1.6.4.5 - A1.6.4.8. The tables can be used to observe the overall bit-rates for various points on the graphs although the

percentage sub-image protection is already an indication of the overall bit-rate. As an example, once again, for the lowest distortion and minimum protection an initial source coding rate of 1.85 bits/pixel, with a sub-image protection of 60% (below this unacceptable reconstructions result), sub-image size of 16x16 pixels square, and employment of the Hamming (7, 4) code would be used which gives an overall coding rate of about 2.674 bits/pixel. Again it is found that using a higher percentage sub-image protection (100%) with a lower initial coding rate of 1.00 bits/pixel gives an overall coding rate of 1.965 bits/pixel when keeping the other parameters the same as in the previous example. The higher protection ensures less error effects can occur and, though the overall distortion is higher than in the previous example, it is still acceptable.

Figure 6.4.8 shows the subjective results obtained for the threshold scheme with 60 or 70% sub-image protection (S.I.P.), a channel error rate of 1 in 10^3 and an initial source coding rate of approximately 1.85 bits/pixel. Again the larger sub-image size appears favourable for the quoted percentage S.I.P.. Comparison with the results obtained for the zonal scheme, Figure 6.4.3, indicates that the percentage S.I.P. has to be higher in the case of the threshold scheme in order to acheive a comparable result. Even so, more sub-images (in the case of the 16x16 pixels

Using Hamming(7,4), 60% S.I.P.

Using Hamming(15,11), 70% S.I.P.

Sub-image Size = 16x16 Pixels Square

Using Hamming(7,4), 60% S.I.P.

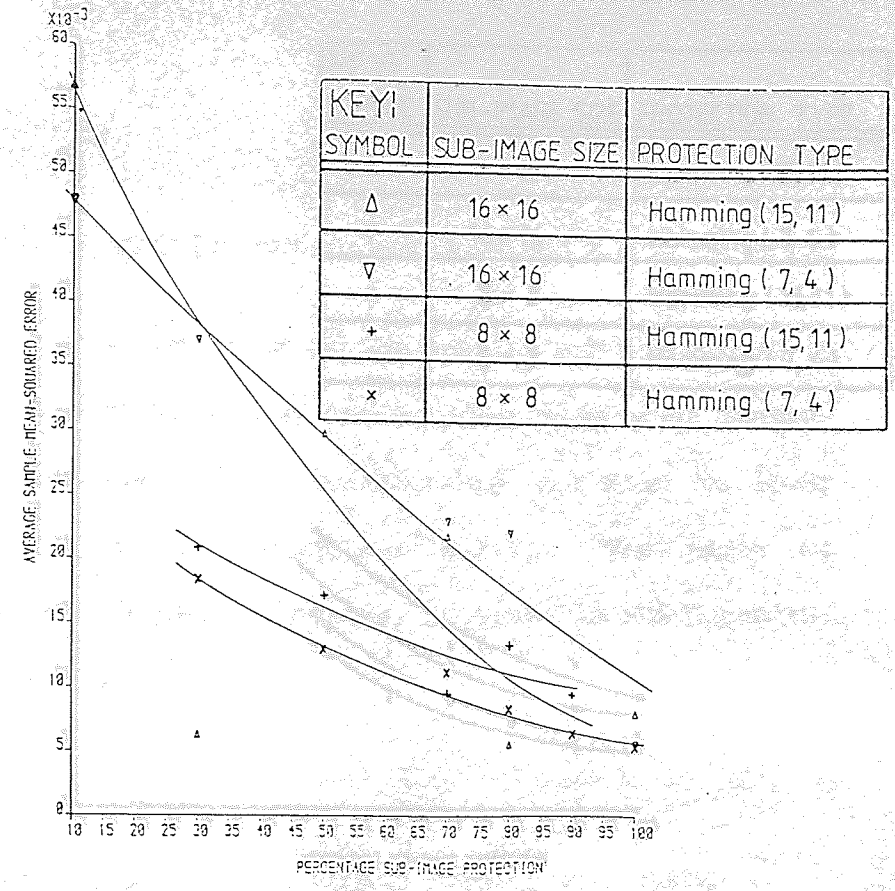
Using Hamming(15,11), 60% S.I.P.

Sub-image Size = 8x8 Pixels Square

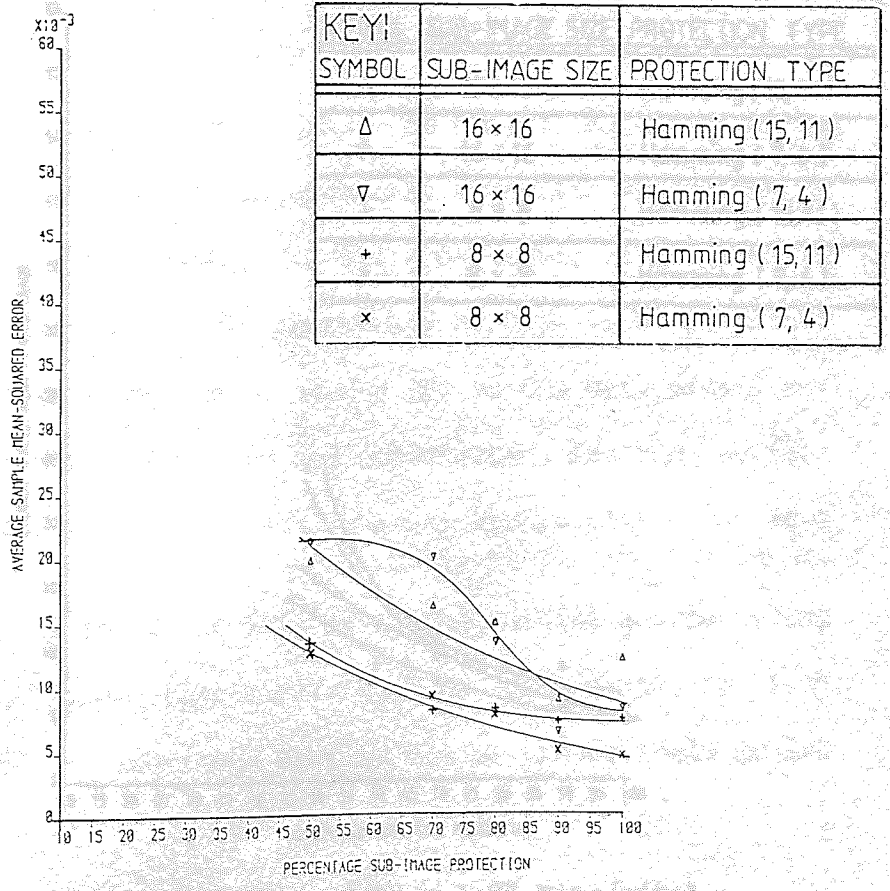
Figure 6.4.8 Threshold, Image Reconstructions at Indicated Percentage S.I.P., Channel Error Rate = 1 in 1000, Initial Source Coding Rate = 1.85 Bits/pixel.

square) are noticeably in error. As the percentage S.I.P. is higher, this in turn indicates that the overall coding rate is pushed higher. The increased protection afforded by the Hamming (7, 4) code over the (15, 11) code is also noticeable.

The quantitative results for the threshold scheme, in conjunction with the channel error rate of 1 in 10^2, are shown in Figures 6.4.9 and 6.4.10. Once again, as not all error effects are eliminated even at 100% S.I.P., Figure 6.2.4 is referred back to for the subjective results. The graphs of Figures 6.4.9 and 6.4.10 indicate that, for relatively low initial source coding bit-rates and a sub-image size of 16x16 pixels square, the points obtained are very disjointly distributed making the drawing of adequate curves difficult. The increase in initial source coding bit-rate tends to make the points less disjointly distributed. This characteristic is not displayed as much when using the smaller sub-image size of 8x8 pixels square. With this sub-image size, the points obtained tend to lie closer to estimated curves indicating, once more, the favourability of this sub-image size to restrict distortion to smaller localized regions thus giving a smoother distortion versus percentage S.I.P. relationship. This points to the selection of a small sub-image size, if relatively high distortion is expected, in order to make the system more predictable. All four graphs indicate that, with the threshold

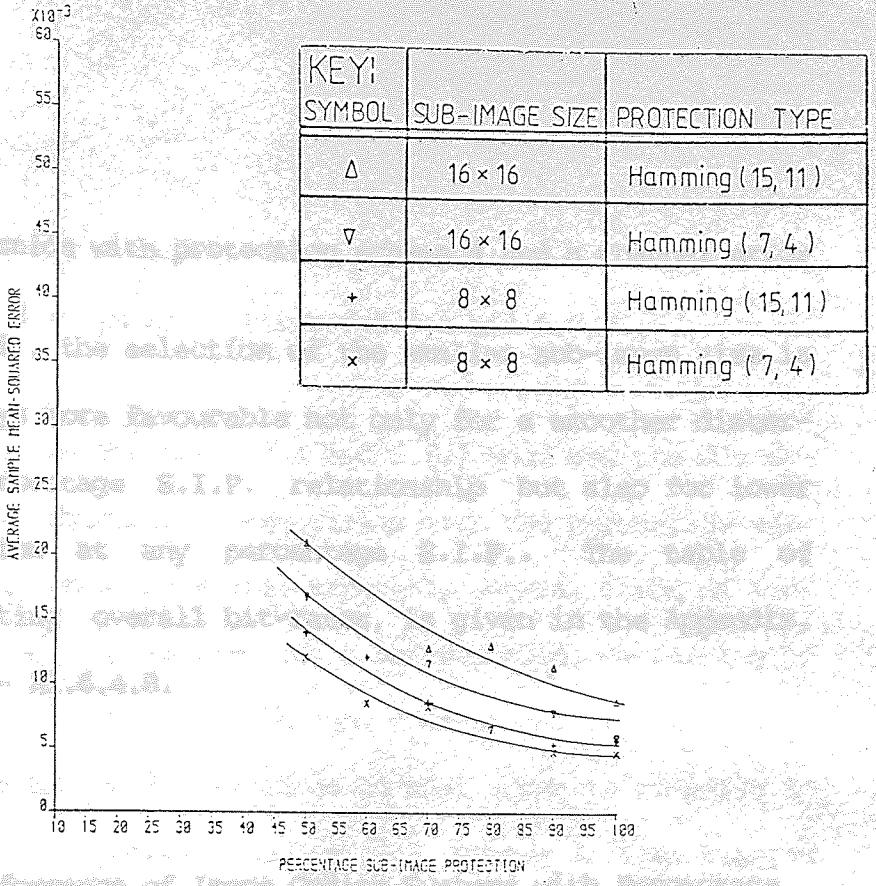


(a) Initial, Source Coding, Rate = 1·00 Bits/pixel

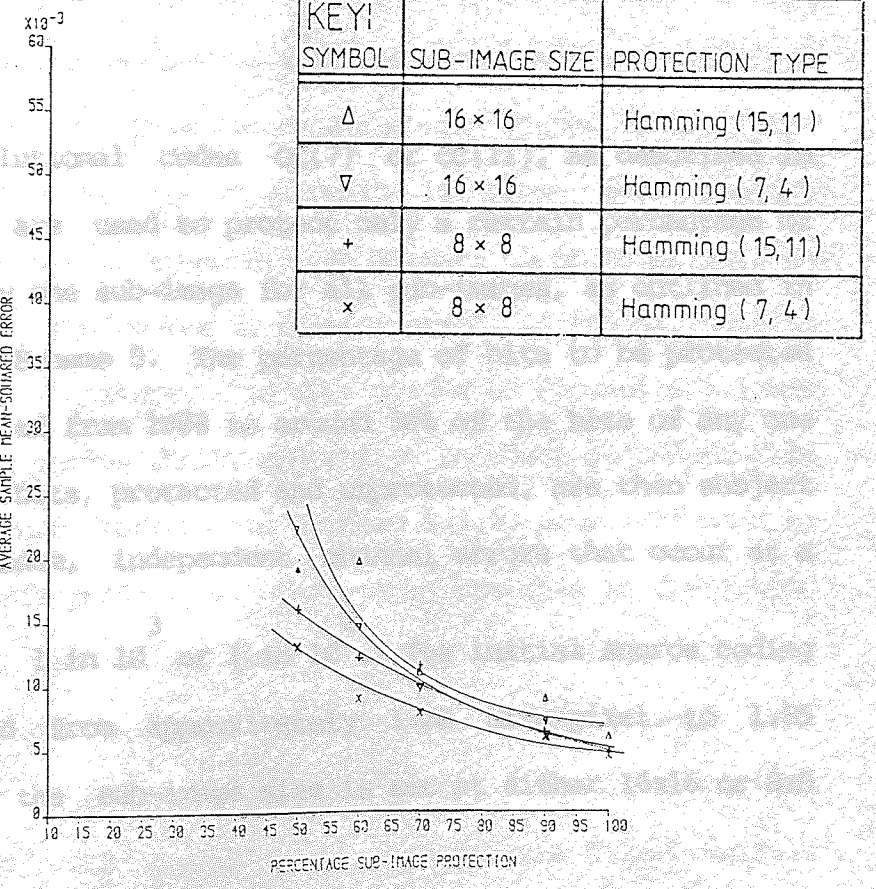


(b) Initial, Source Coding, Rate = 1·25 Bits/pixel

Figure 64.9 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 100



(a) Initial, Source Coding, Rate = 1.5Ø Bits/pixel.



(b) Initial, Source Coding, Rate = 1.85 Bits/pixel.

Figure 6.4.1Ø Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1ØØ.

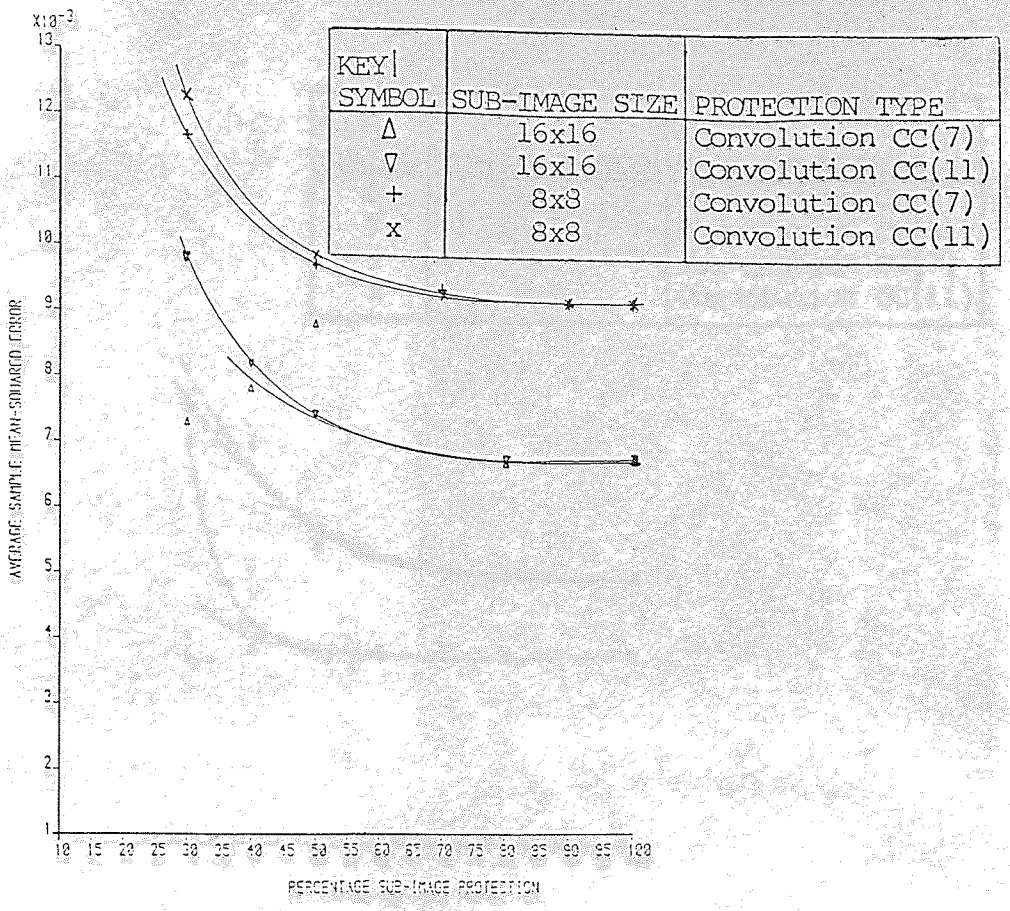
scheme in conjunction with protection scheme 4 and a channel error rate of 1 in 10^2, the selection of the smaller sub-image size is practically always more favourable not only for a smoother distortion versus percentage S.I.P. relationship but also for lower overall distortion at any percentage S.I.P.. The table of results, indicating overall bit-rates, is given in the Appendix, tables A1.6.4.5 - A1.6.4.8.

6.5 Results/Performance of Image Coding Systems with Percentage Sub-image Convolutional Coding (Scheme 5).

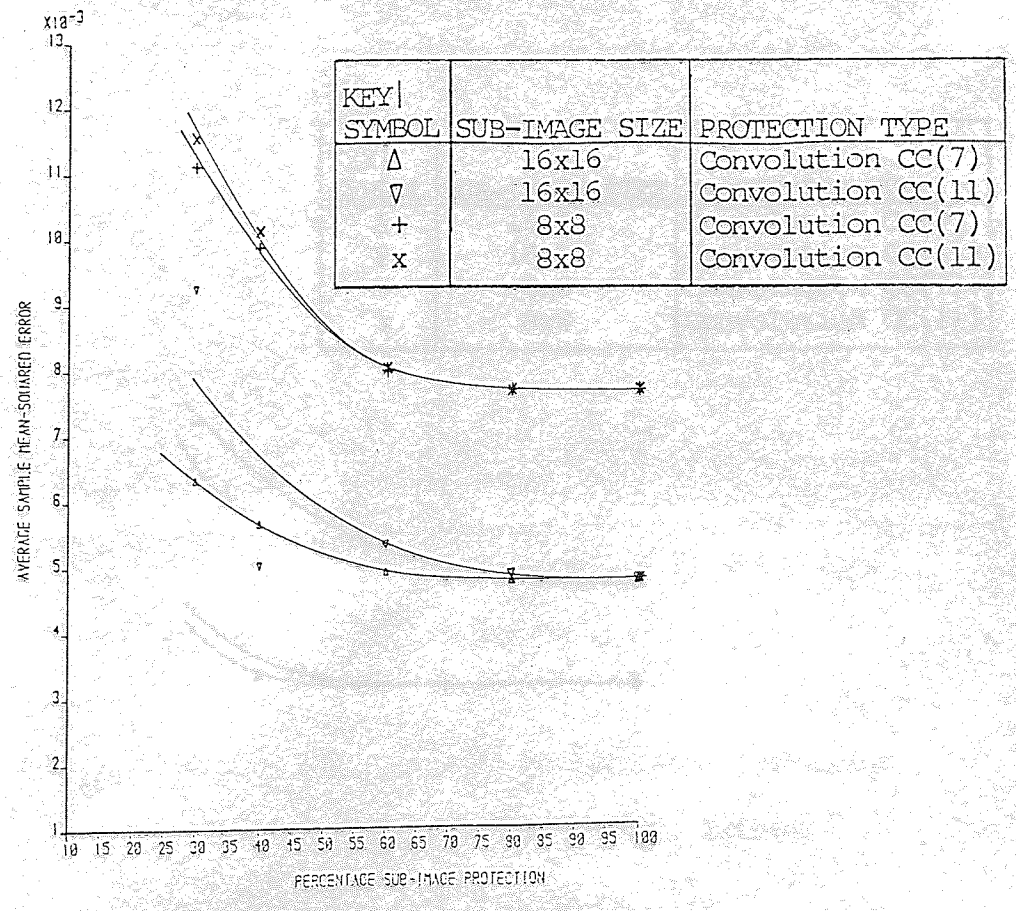
The convolutional codes CC(7) or CC(11), as described in Section 5.4.2, are used to protect only a certain percentage of the bits of any one sub-image for all sub-images, as outlined in Section 5.6.5, Scheme 5. The percentage of bits to be protected is, again, varied from 100% to around 30% of the bits of any one sub-image. All bits, protected and unprotected, are then subject to uniform, random, independent channel errors that occur at a rate of either 1 in 10^3 or 1 in 10^2. The initial source coding rate is varied from approximately 1.00 bits/pixel to 1.85 bits/pixel and the sub-image size is set at either 16x16 or 8x8 pixels square.

The results for a channel error rate of 1 in 10^3, when using the zonal scheme, are shown in Figures 6.5.1 and 6.5.2.

The graphs of Figures 6.5.1 and 6.5.2 show how the distortion in the reconstructed image varies with the percentage sub-image protection. Once again the, expected, general trend of non-linear, increasing, distortion for a corresponding decreasing of percentage S.I.P. is found. Although, occassionally, points of a particular curve do not lie as close to that curve as compared to the corresponding previous results for Scheme 4, i.e. Figures 6.4.1 and 6.4.2, the resulting curves of both schemes are very similar. Recalling that the difference between the two schemes is that, where previously a Hamming code was used now a convolutional code is implemented, this suggests that the convolutional technique is as good as the block coding technique, even when the method of decoding is algebraic, as apposed to probabilistic in nature and when the source of channel errors is random, uniform and independent in nature. The four graphs of Figures 6.5.1 and 6.5.2 thus show curves which, generally, indicate a gradual rise in distortion when varying the applied S.I.P. from 100% down to around 40 to 50% protection. Here-after the rise in distortion is, once again, generally much steeper, indicating that the reconstructions are becoming unacceptable. This agrees with later, subjective, results shown in Figure 6.5.3. In comparison with each other, the graphs indicate that the higher initial

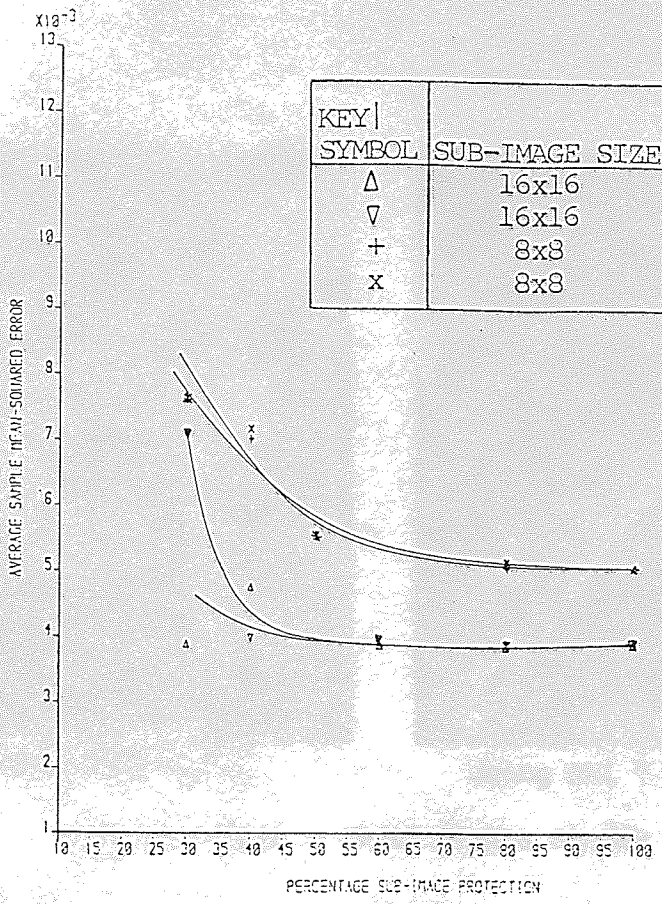


(a) Initial, Source Coding, Rate = 1.00 Bits/pixel.

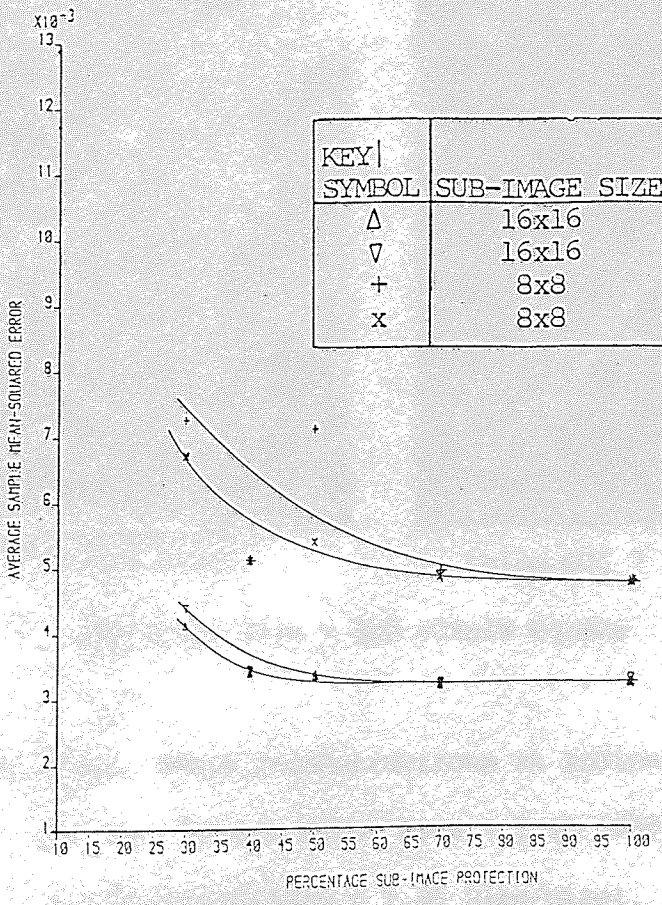


(b) Initial, Source Coding, Rate = 1.25 Bits/pixel.

Figure 6.5.1 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1000.



(a) Initial, Source Coding, Rate = 1.5Ø Bits/pixel.



(b) Initial, Source Coding, Rate = 1.85 Bits/pixel.

Figure 6.5.2 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1ØØØ.

Using CC(11), 40% S.I.P.

Using CC(7), 40% S.I.P.

Sub-image Size = 16x16 Pixels Square

Using CC(11), 40% S.I.P.

Using CC(7), 40% S.I.P.

Sub-image Size = 8x8 Pixels Square

Figure 6.5.3 Zonal, Image Reconstructions at Indicated Percentage S.I.P., Channel Error Rate = 1 in 1000, Initial Source Coding Rate = 1.85 Bits/pixel.

Using CC(11), 40% S.I.P.

Using CC(7), 40% S.I.P.

Sub-image Size = 16x16 Pixels Square

Using CC(11), 40% S.I.P.

Using CC(7), 40% S.I.P.

Sub-image Size = 8x8 Pixels Square

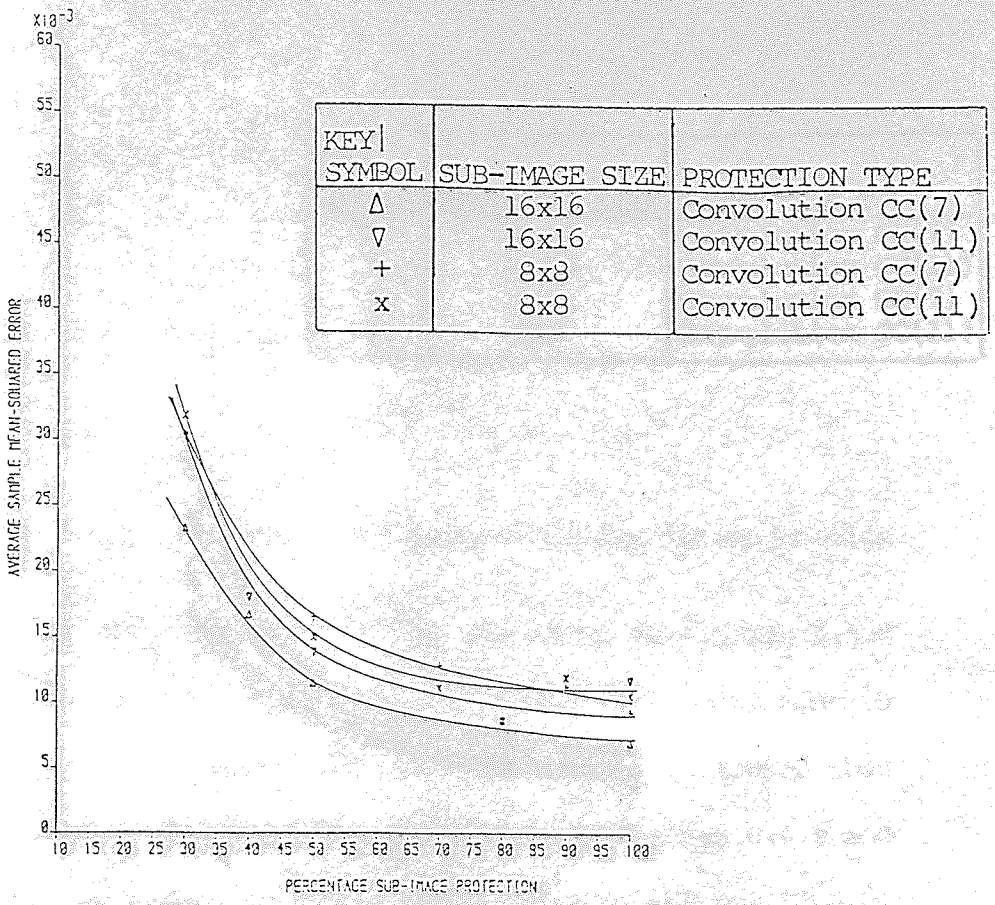
Figure 6.5.3 Zonal, Image Reconstructions at Indicated Percentage S.I.P., Channel Error Rate = 1 in 1000, Initial Source Coding Rate = 1.85 Bits/pixel.

source coding rates result in lower distortions for a sub-image protection level of 40 to 50% or greater (and sometimes less). The larger sub-image size is more favourable. The difference between the results obtained when using CC(7) as apposed to CC(11) are seen to be fairly marginal. This might be expected since both error protection codes introduce the same amount of redundancy and afford the same amount of protection, for independent random errors, to the source coded bit stream. Tables A1.6.4.1 to A1.6.4.4, in the Appendix, are the tables containing the results used to plot the graphs of Figures 6.5.1 and 6.5.2. They also indicate the increase in overall average bit-rate that results from using a certain percentage S.I.P., from which the cost of lowering the distortion on reconstruction of the image can be seen. This is to be remarked on again in Section 6.7.

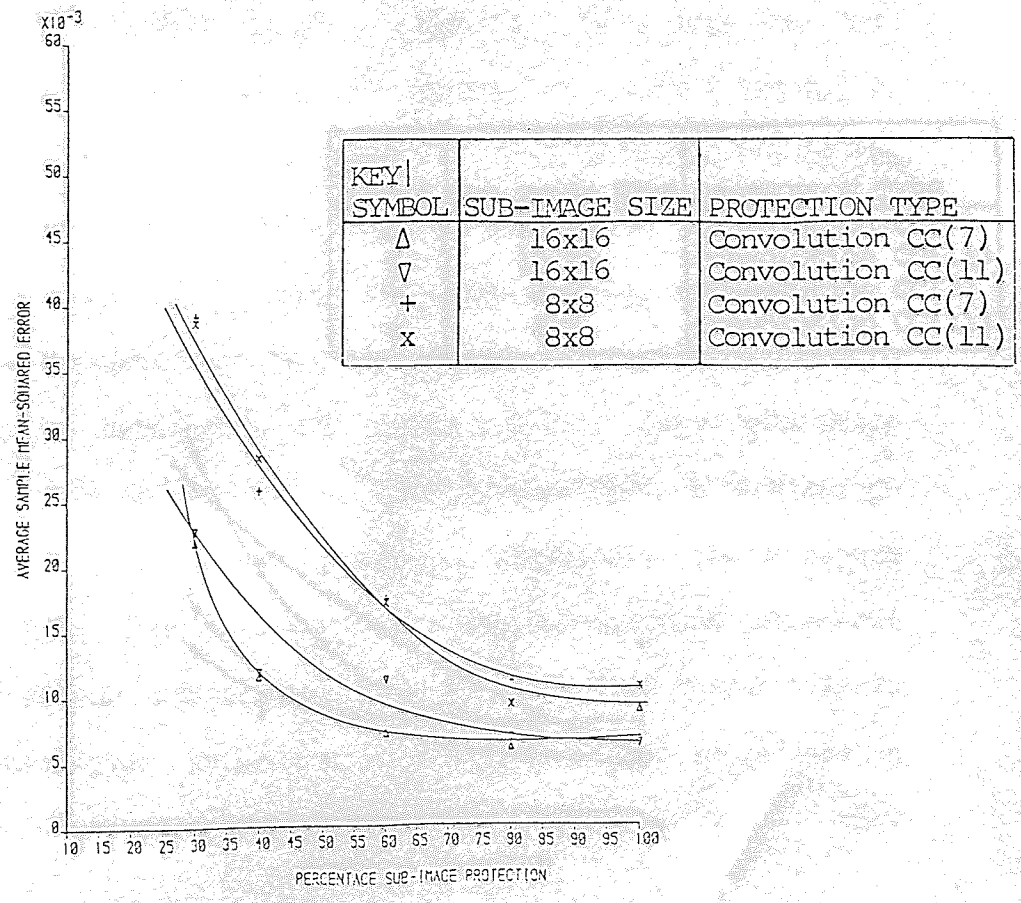
Figure 6.5.3 shows the subjective results obtained for the zonal scheme with 40% S.I.P., channel error rate of 1 in 10^3, and an initial source coding rate of approximately 1.85 bits/pixel. As was the case with the graphs, the subjective results obtained in Figure 6.5.3 are comparable with the results of the previous

scheme, i.e. Figure 6.4.3, thus the advantage of having a larger sub-image size, for the channel error rate of 1 in 10^3, applies to these results in the same way as in the previous scheme. No particularly advantage is seen using CC(7) over CC(11) or vice versa.

Figure 6.5.4 and 6.5.5 show the quantitative results for the zonal scheme in conjunction with the channel error rate of 1 in 10^2. 100% S.I.P. has already been seen to be required (Scheme 3) to eliminate as many disturbing error effects as possible and so, for corresponding subjective results refer back to Figure 6.3.3. However, once again, the quantitative results in the form of distortion versus percentage S.I.P. show the very close similarity to the results obtained in the previous scheme, i.e. Figures 6.4.4 and 6.4.5. This suggests that this is probably the case for the zonal scheme at most channel error rates. The curves of the graphs in Figures 6.5.4 and 6.5.5 thus display grouping and over-lapping characteristics, with the advantage of using a larger sub-image becoming less as the percentage S.I.P. is decreased, in a similar manner to that obtained in the previous results. In this way a breaking down of the protection methods is indicated. The similarity in results obtained thus far with the zonal scheme leads to an ultimate conclusion to be discussed in Chapter 7. The table of values for the graphs is found in the Appendix, Tables

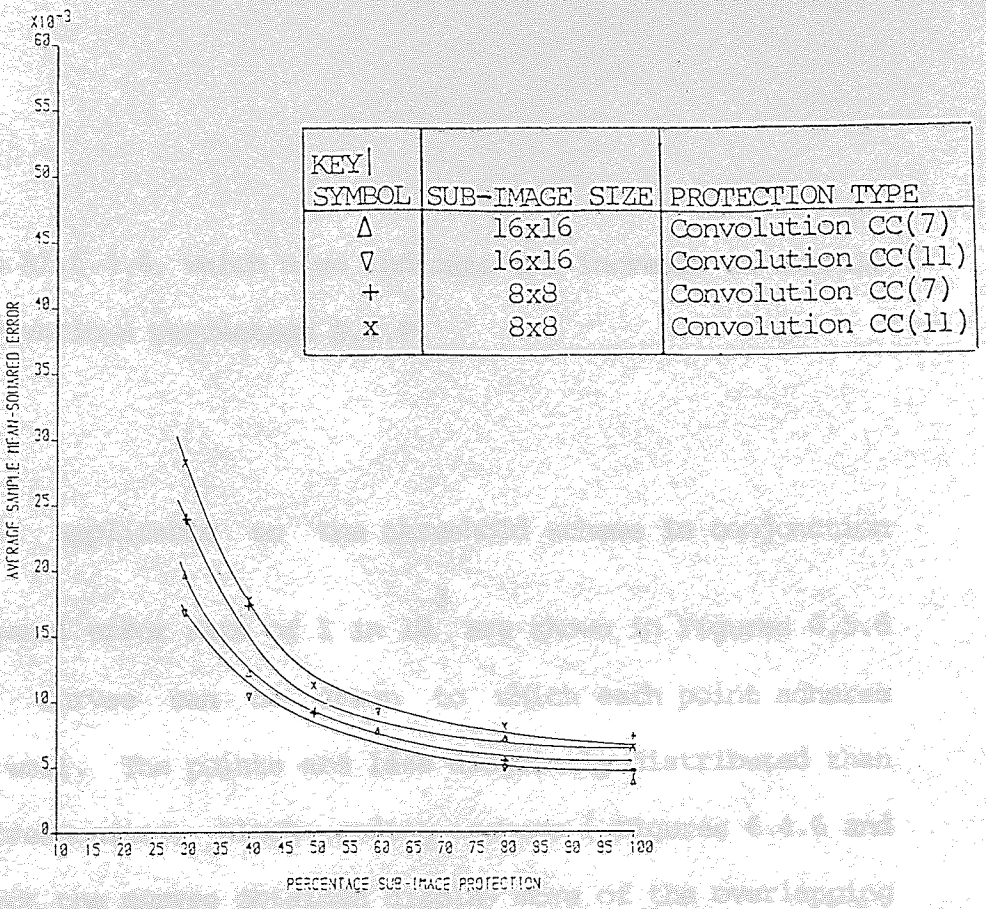


(a) Initial, Source Coding, Rate = 1.ØØ Bits/pixel.

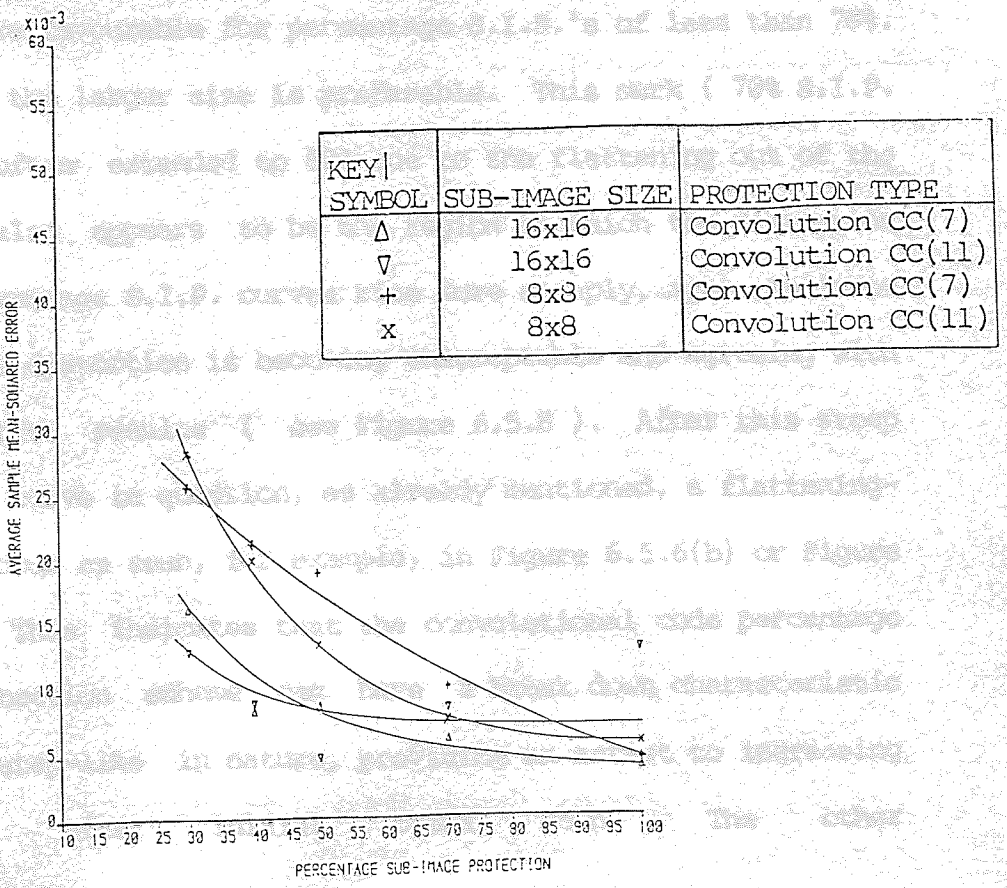


(b) Initial, Source Coding, Rate = 1.25 Bits/pixel.

Figure 6.5.4 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1ØØ.



(a) Initial, Source Coding, Rate = 1.50 Bits/pixel.

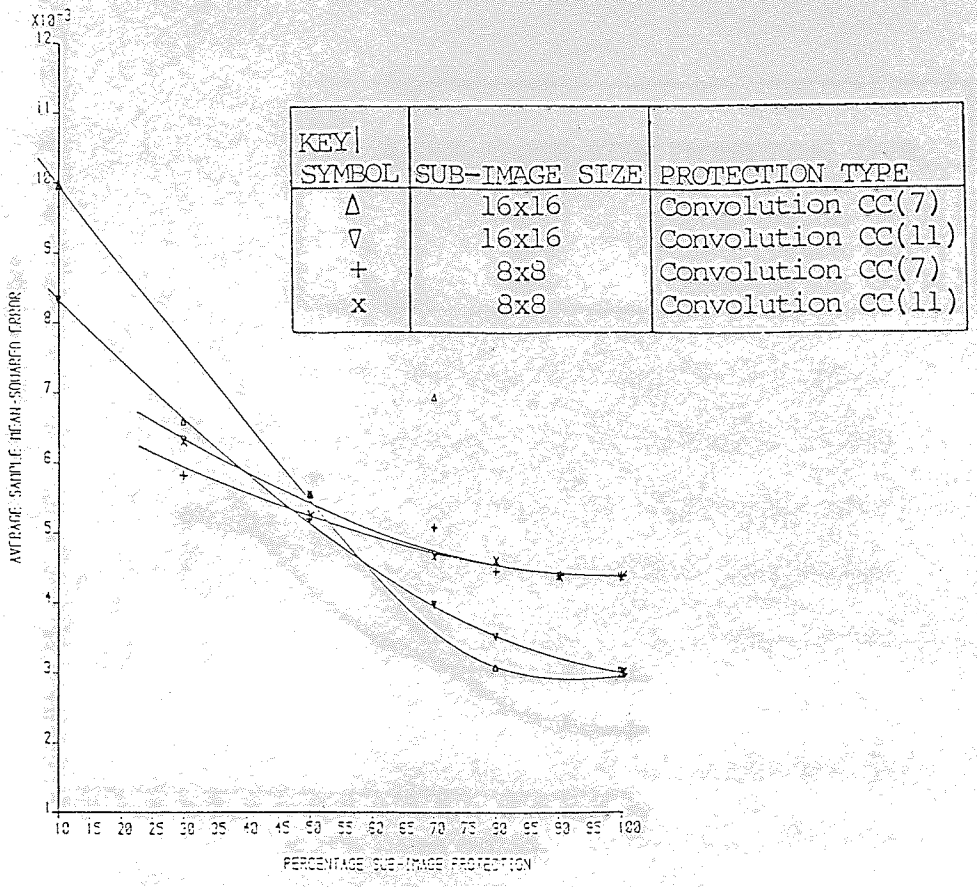


(b) Initial, Source Coding, Rate = 1.85 Bits/pixel.

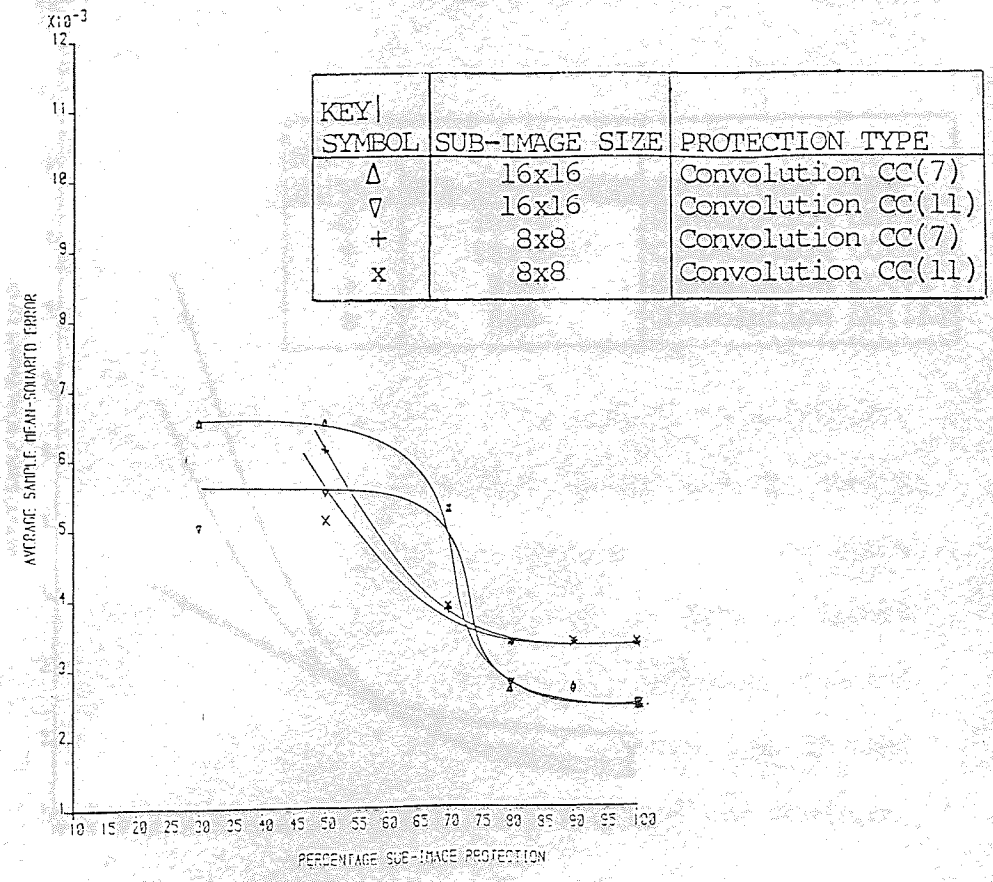
Figure 6.5.5 Zonal Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 100.

Al.6.4.1 to Al.6.4.4, which also indicate the increase in overall bit-rate for various percentage S.I.P.

Results applicable to the threshold scheme in conjunction with a channel error rate of 1 in 10^3 are shown in Figures 6.5.6 and 6.5.7. Curves can be drawn to which each point adheres reasonably well. The points are less disjointly distributed than in the corresponding, block coding scheme (Figures 6.4.6 and 6.4.7), but the curves obtained display more of the overlapping characteristic making it more difficult to assess which sub-image size is more favourable for percentage S.I.P.'s of less than 70%. Above this the larger size is preferable. This mark (70% S.I.P. which is often extended to 50% due to the flattening out of the curves) also appears to be the region at which the distortion versus percentage S.I.P. curves rise more steeply, again indicating that the distortion is becoming unacceptable and agreeing with the subjective results (see Figure 6.5.8). After this steep rise of the curve in question, as already mentioned, a flattening-out can occur as seen, for example, in Figure 6.5.6(b) or Figure 6.5.7(a). This indicates that the convolutional code percentage S.I.P. protection scheme can have a break down characteristic which is step-like in nature, providing an arrest to increasing distortion after initial break down. The other

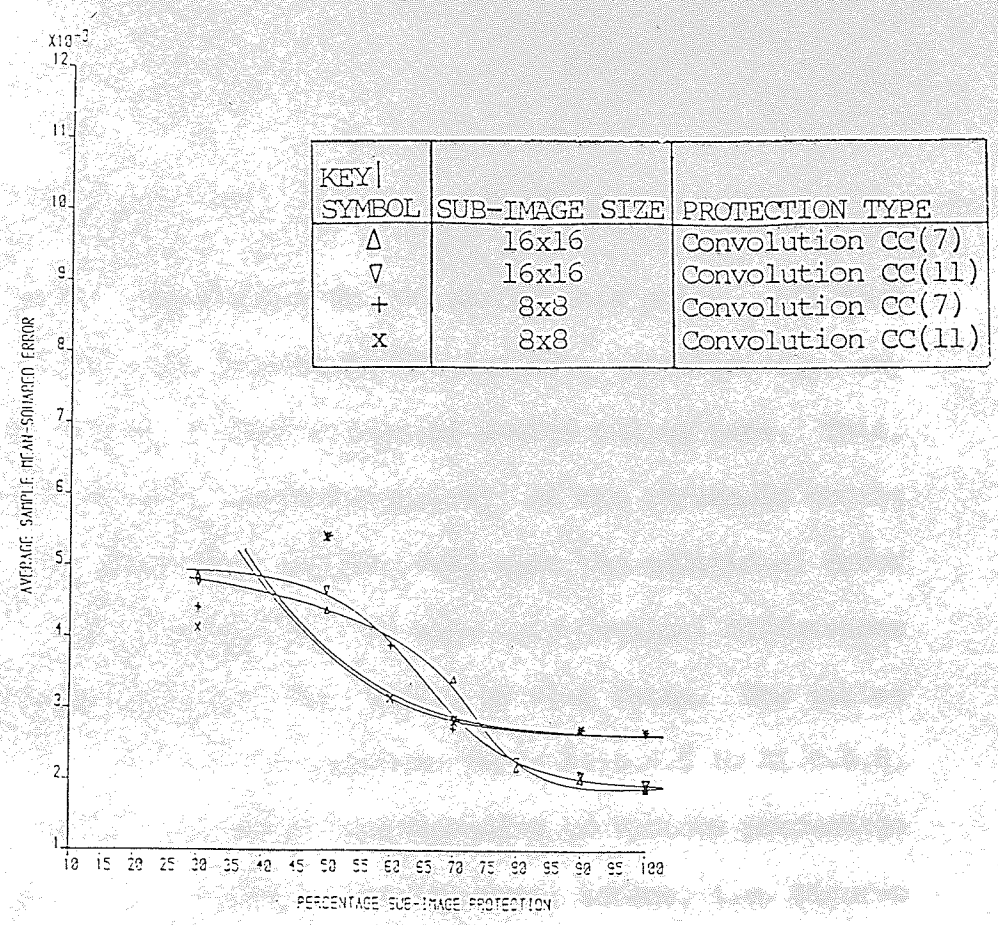


(a) Initial, Source Coding, Rate = 1.00 Bits/pixel.

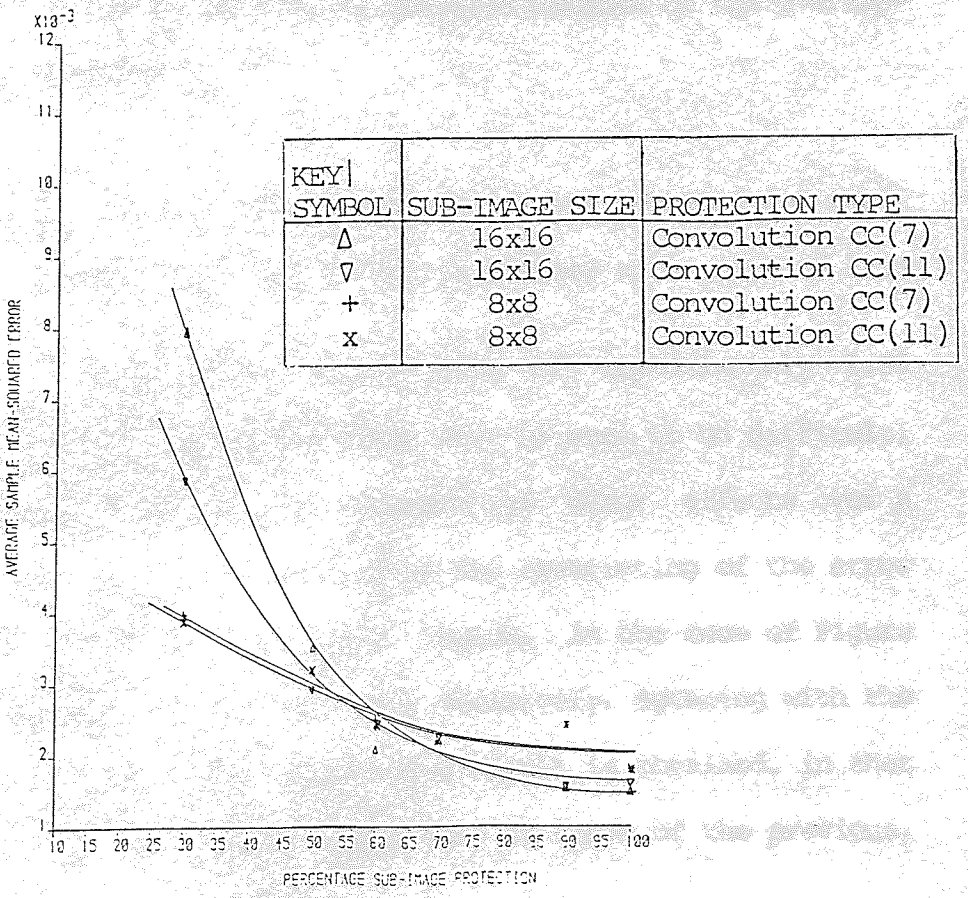


(b) Initial, Source Coding, Rate = 1.25 Bits/pixel.

Figure 6.5.6 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1000.



(a) Initial, Source Coding, Rate = 1.50 Bits/pixel.



(b) Initial, Source Coding, Rate = 1.85 Bits/pixel.

Figure 6.5.7 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1000.

curves of these two figures do not verify this point. All four graphs of these two figures do not indicate, conclusively, any advantage to using a higher initial source coding rate. This, again, points to the unpredictability of the threshold coding scheme in the presence of errors, even with the additional error protection. The four graphs also show only marginal differences in the employment of CC(7) over CC(11) or vice versa. The tables of results are found in the Appendix, table A1.6.4.5 to A1.6.4.8, indicating overall bit-rates corresponding to various percentage S.I.P. levels. In contrast to the zonal scheme, i.e. Figures 6.5.1 and 6.5.2, the threshold scheme is seen to have resulting curves which cannot be as clearly isolated because of the overlapping of one with others.

Figure 6.5.8 shows the subjective results obtained for the threshold scheme with 50% S.I.P., a channel error rate of 1 in 10^3 and an initial source coding rate of approximately 1.85 bits/pixel. The choice of sub-image size is seen to be difficult, even subjectively. The distribution of error effects over a larger region has to be compared to the restricting of the error effects to a smaller localized region. In the case of Figure 6.5.8 the choice cannot be made decisively, agreeing with the quantitative assessments. A pleasing result is obtained, in that these subjective results are comparable to those of the previous,

Using CC(11), 50% S.I.P.

Using CC(7), 50% S.I.P.

Sub-image Size = 16x16 Pixels Square

Using CC(11), 50% S.I.P.

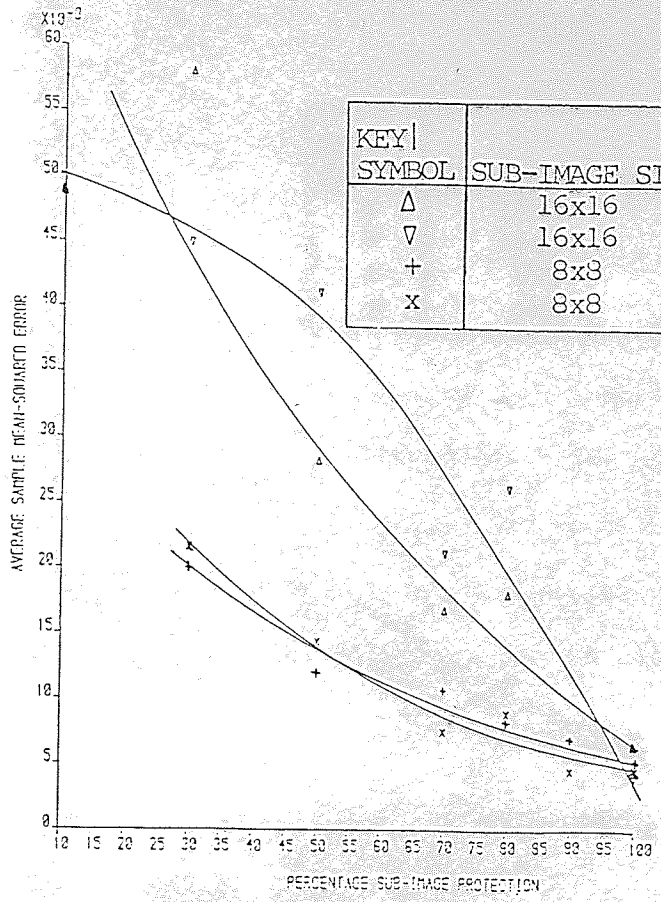
Using CC(7), 50% S.I.P.

Sub-image Size = 8x8 Pixels Square

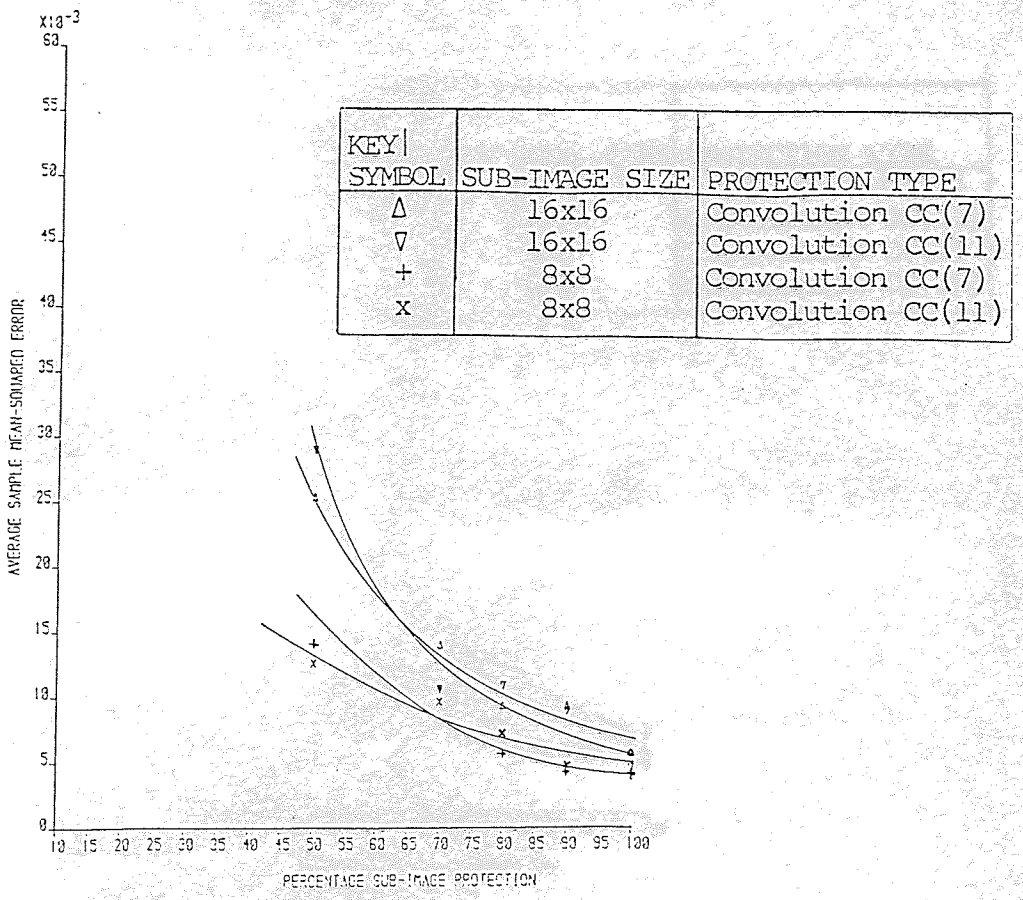
Figure 6.5.8 Threshold, Image Reconstructions at Indicated Percentage S.I.P., Channel Error Rate = 1 in 1000 Initial Source Coding Rate = 1.85 Bits/pixel.

block coding, scheme (Figure 6.4.8). In fact the results suggest that the convolutional code percentage S.I.P. scheme can result in a lower necessary percentage S.I.P., which will have the advantage of lower overall average bit-rate than the corresponding block coding percentage S.I.P. approach would have. In comparison with the zonal scheme, Figure 6.5.3, the percentage S.I.P. level is seen to be 10% higher for comparable results as apposed to the 20 - 30% difference noticeable in the previous scheme (Scheme 4). In agreement with the quantitative results, no distinct advantage of CC(7) over CC(11) or vice versa is observed.

Figures 6.5.9 and 6.5.10 show the quantitative results for the threshold scheme in conjunction with the channel error rate of 1 in 10^2. Only at the relatively low, initial, source coding, rate of 1.00 bits/pixel are the points relatively disjointed in their distribution, although this result is still less pronounced than those of the corresponding previous scheme (Scheme 4, Figures 6.4.9 and 6.4.10). The implementation of a convolution code appears to restrict this disjointed effect. Once again, at this relatively high error rate, the restriction of error effects to smaller localized regions through the use of the smaller sub-image proves more advantages in terms of lower distortion in the reconstructed image. The increase in initial source coding rate tends to make the points less disjointly

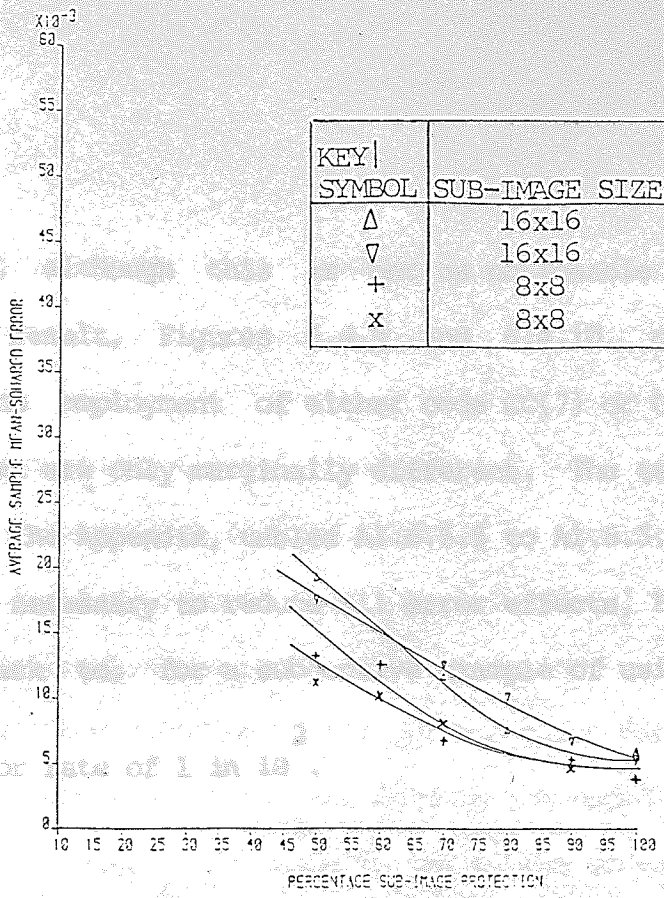


(a) Initial, Source Coding, Rate = 1.00 Bits/pixel.



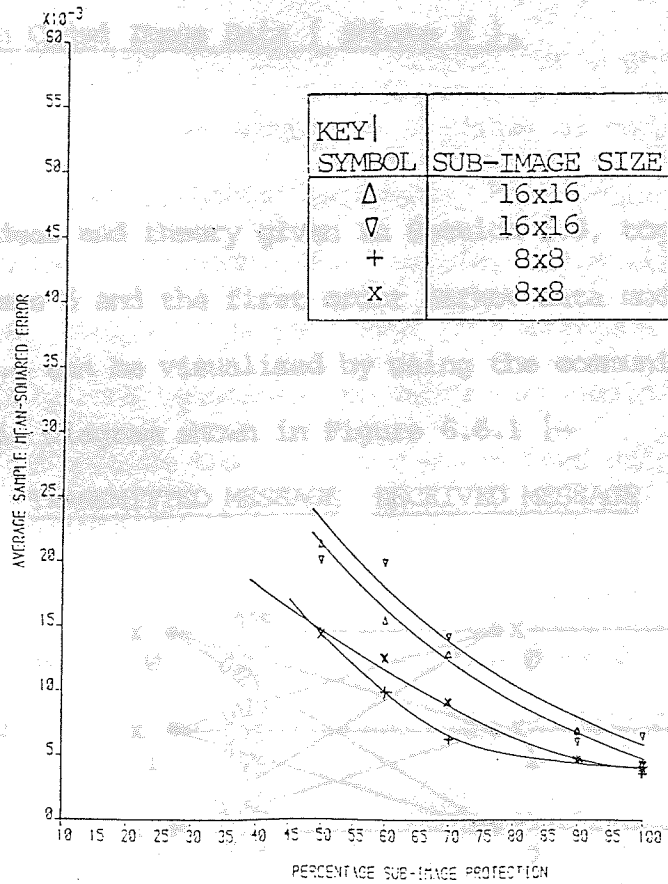
(b) Initial, Source Coding, Rate = 1.25 Bits/pixel.

Figure 6.5.9 Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 100.



KEY SYMBOL	SUB-IMAGE SIZE	PROTECTION TYPE
Δ	16x16	Convolution CC(7)
∇	16x16	Convolution CC(11)
+	8x8	Convolution CC(7)
x	8x8	Convolution CC(11)

(a) Initial, Source Coding, Rate = 1.5Ø Bits/pixel.



KEY SYMBOL	SUB-IMAGE SIZE	PROTECTION TYPE
Δ	16x16	Convolution CC(7)
∇	16x16	Convolution CC(11)
+	8x8	Convolution CC(7)
x	8x8	Convolution CC(11)

(b) Initial, Source Coding, Rate = 1.85 Bits/pixel.

Figure 6.5.1Ø Threshold Scheme, Percentage Sub-image Protection with Channel Error Rate = 1 in 1ØØ.

distributed, although this is not as noticeable as in the corresponding result, Figures 6.4.9 and 6.4.10, of the previous scheme. The employment of either code CC(7) or CC(11) produces results that are only marginally different. The table of results is found in the Appendix, tables A1.6.5.5 to A1.6.5.8. Since 100% S.I.P. is necessary to reduce all error effects, Figure 6.3.4 is referred back to, for a subjective example of using this scheme with an error rate of 1 in 10^2.

6.6 Theoretical Performance of a Maximum Likelihood Decoder on Huffman Coded Image Data (Scheme 6).

The ideas and theory given in Section 5.5, together with the devised Scheme 6 and the first order Markov data model outlined in Section 5.6.6 can be visualized by using the communication process probabilistic diagram shown in Figure 6.6.1 :-

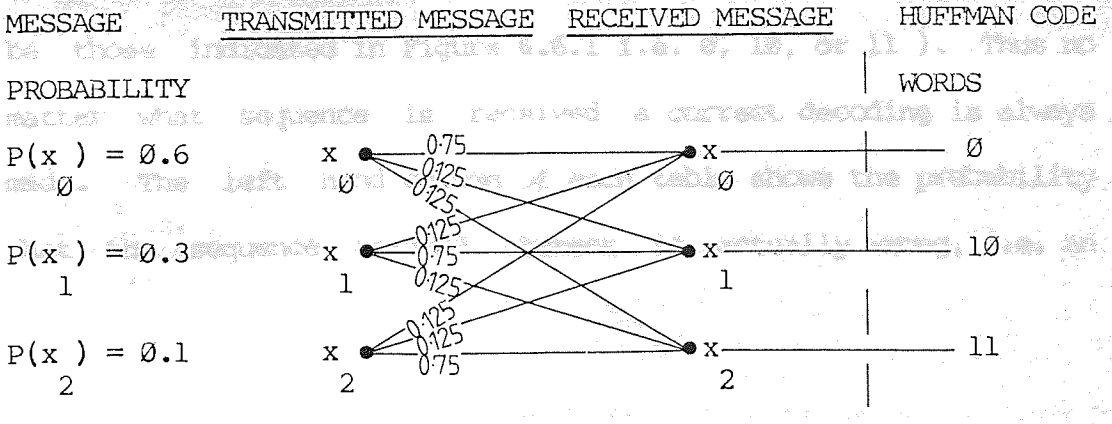


Figure 6.6.1 A Communication Process Probabilistic Diagram.

The transmitted and received messages are selections from the same alphabet.

Using equations (5.5.1) through (5.5.3) and ensuring the data is in the correct form by adhering to equations (5.6.6.1) through (5.6.6.4), then equations (5.6.6.6) and (5.6.6.7) are used to estimate the probability of failure associated with a maximum likelihood decoding of the Huffman coded data when knowing the number of words and bits contained in the coded word sequence. The resulting output is shown in the tables contained in Figure 6.6.2.

As the number of words and bits for a given sequence is assumed known, so the right hand column of each table in this figure shows the possible sequences that could have been sent based on this information. For example, table (1) shows that if it is known that 4 words were sent in a sequence of 4 bits then there is only one possible sequence that could have been sent (N.B. this is because the only code words that exist are known to be those indicated in Figure 6.6.1 i.e. 0, 10, or 11). Thus no matter what sequence is received a correct decoding is always made. The left hand column of each table shows the probability that the sequence assumed correct is actually wrong, i.e. an

TABLE 1 - 4 WORDS/4 BITS

Probability of choosing incorrect code sequence	
AVERAGE PROB.	SEQ. ASSUMED CORRECT
0.00000E+00	0000

Δ_a=TOTAL AVE. = 0.00000E+00

TABLE 2 - 4 WORDS/5 BITS

Probability of choosing incorrect code sequence	
AVERAGE PROB.	SEQ. ASSUMED CORRECT
0.60060E-03	00010
0.30090E-03	00011
0.19970E-03	00100
0.19970E-03	00110
0.19970E-03	01000
0.14990E-03	01100
0.44955E-03	10000
0.10000E-03	11000

Δ_a=TOTAL AVE. = 0.27501E-03

TABLE 3 - 4 WORDS/6 BITS

Probability of choosing incorrect code sequence	
AVERAGE PROB.	SEQ. ASSUMED CORRECT
0.77486E-03	001010
0.12914E-03	001011
0.12919E-03	001110
0.51752E-03	001111
0.42955E-04	010010
0.35813E-04	010011
0.12919E-03	010100
0.35813E-04	010110
0.42955E-04	011010
0.35813E-04	011011
0.35813E-04	011100
0.25773E-03	011110
0.12886E-03	100010
0.86017E-04	100011
0.17907E-04	100100
0.21477E-04	100110
0.51610E-03	101000
0.10744E-03	101100
0.42955E-04	110010
0.21531E-04	110011
0.59689E-05	110100
0.71591E-05	110110
0.14390E-04	111000
0.17203E-03	111100

Δ_a=TOTAL AVE. = 0.13786E-03

TABLE 4 - 4 WORDS/7 BITS

Probability of choosing incorrect code sequence	
AVERAGE PROB.	SEQ. ASSUMED CORRECT
0.96284E-03	0101010
0.16047E-03	0101011
0.26759E-04	0101110
0.16047E-03	0101111
0.16051E-03	0111010
0.26752E-04	0111011
0.16059E-03	0111110
0.72321E-03	0111111
0.12023E-03	1001010
0.20039E-04	1001011
0.23367E-04	1001110
0.10027E-03	1001111
0.10027E-03	1010010
0.80257E-04	1010011
0.36137E-03	1010100
0.80277E-04	1010110
0.23367E-04	1011010
0.20039E-04	1011011
0.10055E-04	1011100
0.12023E-03	1011110
0.33422E-04	1101010
0.66797E-05	1101011
0.77891E-05	1101110
0.33422E-04	1101111
0.77891E-05	1110010
0.55692E-05	1110011
0.20089E-04	1110100
0.55703E-05	1110110
0.46735E-04	1111010
0.40078E-04	1111011
0.26766E-04	1111100
0.28041E-03	1111110

Δ_a=TOTAL AVE. = 0.12362E-03

TABLE 5 - 4 WORDS/8 BITS

Probability of choosing incorrect code sequence	
AVERAGE PROB.	SEQ. ASSUMED CORRECT
0.18864E-02	10101010
0.31440E-03	10101011
0.52400E-04	10101110
0.31440E-03	10101111
0.52400E-04	10111010
0.87332E-05	10111011
0.52400E-04	10111110
0.31440E-03	10111111
0.10480E-03	11101010
0.17466E-04	11101011
0.29111E-05	11101110
0.17466E-04	11101111
0.10480E-03	11111010
0.17467E-04	11111011
0.10480E-03	11111110
0.62879E-03	11111111

Δ_a=TOTAL AVE. = 0.24963E-03

Figure 6.6.2 Average Probability of an Incorrect Decoding of Huffman Coded Bit-stream using Maximum Likelihood Decoding

incorrect result obtained by maximum likelihood decoding. For example, in table (2), if a sequence is decoded by maximum likelihood as 0010 then the probability that the sequence was actually 00011, 00100, 00110,... ...,11000 is 0.60060×10^{-3}. In the same way, in table (1), if it is known that 4 words existed in a sequence of 4 bits the fact that there is only one possible sequence to fit this information gives the probability of an incorrect decoding of the 4 word sequence as zero. When keeping the number of words in the sequence constant at 4, and varying the number of bits from 4 to 8, tables (1) to (5) show that the total average probability of failure ranges from 0.0 to 0.27501×10^{-3}, a relatively small range. This indicates that a decoding scheme based on the maximum likelihood ideas presented here would have a maximum percentage failure rate of approximately 0.027% when used with the appropriate data model.

6.7 Discussion on Results of Combined Source/Channel Error Protection Schemes.

With Schemes 1 to 5, each one improves the performance of the image encoding systems, in the presence of channel errors, to a certain degree. This degree varies from a mainly

theoretical/quantitative improvement, seen in Scheme 1, to a very substantial improvement where subjective and quantitative assessment show the ability of the overall schemes to cope with channel error rates which are becoming relatively high.

Ideally, a scheme such as Scheme 6, would provide a unique solution to the problem of channel errors since no further redundancy need be added to the source coded data. Rather, use is made of any residual redundancy. Such a scheme in effect converts otherwise useless redundancy. The fact that remaining redundancy, after source coding, is mostly in a useless form is borne out in the results presented with all Schemes 1 to 5 since increasing the initial source coding bit-rate did not result in very significant improvement in coder performance.

Attention has been focused on " state-of-the-art " transform coders, although the problem of channel errors may take on a different appearance when considering source encoders which function in a slightly different manner. An example of such a coder is the texture source encoder presented by Spann, Wilson, and Dodgson [99]. Here the basic idea is to pull out various textures from an image, leaving a smoother image which is suited more to encoding by the conventional methods that have been examined in this thesis. At the receiving end, the texture is put back in. The cost in extra bits is minimal. Such a scheme may be able to

aid channel error recovery since, when the texture is put back in it may write-over channel error effects that have occured during transmission. It is noted that, as with Scheme 6, the idea moves away from the more conventional, but never-the-less as important, methods of protection.

It remains to draw the final conclusions on the combined source/channel error protection coding schemes based on the results presented in this chapter and those obtained in previous chapters. These are now discussed in the concluding chapter.

CHAPTER 7

CONCLUSION AND DISCUSSION

The communication problem, in relation to the transmission of image data using transform techniques, has been extensively examined. Various channel error effects have been isolated and highlighted, forming a more solid foundation on which to base methods of error protection, rather than guessing at the type of error effects that might occur, and assuming a hypothetical order of coefficient and/or digit importance. Through use of this foundation, various error protection schemes have been devised which improve coder performance to varying degrees, each scheme chosen to provide continuity from one to the next. Which scheme to use has been seen to depend on :-

(i) the permissible form, and amount, of distortion,

(ii) the required overall average coding rate,

and (iii) the channel error rate.

Inherent in the results obtained, when considering image data transmission, is the noticeable, comparable, performance of convolutional codes, whose main employment is for burst errors, to

cope with uniform, random, independent channel errors, with block codes which are designed specifically for the latter type of errors.

The conclusions reached are best presented in a question and answer form. These questions were outlined in Section 1.3, which set out the aims of this research, and are now to be given in an explicit form. They culminate in the answer to the main objective which, as a reminder, was to find " the best system for transmission of image data over a channel which has a specified error rate ".

7.1 Does Source Coder Complexity Make any Difference ?

This question asks, essentially, for the differences in results obtained when using the " adaptive-by-nature " threshold scheme as apposed to the non-adaptive zonal scheme.

Both source coding techniques were seen to answer the first of the component problems relating to the fundamental communication problem i.e. they make a choice in deciding from all of the available source data what should be transmitted and what should not. The reduction in the number of bits left representing the

image, a compression typically from 8 bits/pixel to 1 bit/pixel, whilst maintaining acceptable image accuracy or quality indicates that both approaches, when tackling this component problem are justified.

In terms of complexity, the zonal scheme is less complex to implement than the threshold scheme. The added complexity is certainly worthwhile in the absence of channel errors, as the results obtained indicate that, in general, coding can be accomplished at a lower average bit-rate and with much lower distortion on reconstruction of the image. The distortion obtained with the zonal scheme was seen to be related to the bit-map or variable quantizer and is "snow-like" in nature, whereas that with the threshold scheme has been seen to be related to the set threshold and the accuracy of the "fixed" quantizer and took the form of a slight smoothing of the image. However, the distortion for both types of source coder is distributed in the same manner, i.e. distributed over the entire sub-image, giving the overall error-spectrum of the image itself of relatively uniform distribution for acceptable compression ratios.

In the presence of channel errors, the added complexity of the threshold source coder was seen, initially, to be more of a liability, particularly at error rates greater than 1 in 10^5. In

the absence of error protection, both schemes were seen to be sensitive to channel errors, with the threshold scheme showing extreme sensitivity. The zonal scheme proves more robust, with sub-image and word areas of less sensitivity noticeable, whereas, with the threshold scheme errors cannot be tolerated in practically any bit position of the source coded data bit stream. The liability of the added complexity when errors are present was seen from the result that, with the zonal scheme, an error rate of 1 in 10^{2} and no error protection results in a terrible reconstruction but never-the-less one in which the original image could still be visibly detected. This was not the case with the threshold scheme where the added complexity prevented obtaining any comparable result, even at lower error rates. Modification of the threshold scheme was required before comparison could be undertaken. This modification took the form of a relatively highly redundant error protection method on the most sensitive parameters of the threshold scheme. Once this had been done the added complexity was again favourable in terms of lower overall distortion and even overall bit-rate, since the " highly redundant error protection " still only counts for a few extra bits provided the number of sub-images is not too great.

The added complexity of the threshold source coding scheme thus resulted in added complexity to the corresponding channel

error protection schemes for this coder. After the modification, the adaptive nature of the threshold scheme tended to restrict error effects more to regions of higher activity/energy as these are the regions which are assigned the larger number of bits. This proved to be a favourable result, as these were the regions where errors tend to be less noticeable, and this also gave more apparent 'order' to the error effects. The non-adaptive nature of the zonal scheme resulted in error effects that tended to be evenly distributed, in a random fashion over the whole of the image.

In general, the answer to the question on added source coder complexity being worthwhile is that, it is provided the resulting added complexity in the error protection of the most sensitive parameters does not push the overall bit-rate too high, and provided the error protection on the remaining bits is adequate to prevent practically any error from going uncorrected. This can be accomplished most of the time since the initial reduction in bits, when using the adaptive source coding threshold scheme, allows relatively more leniency when increasing redundancy for error protection.

7.2 Are Errors in Certain Coefficients More Likely to Produce More Undesireable Effects than in Others?

With the zonal coding scheme, the answer to the above question was found to be that channel errors in certain coefficients are indeed likely to give more undesireable reconstructions than in others. In general, it appears that the more sensitive coefficients occur in a radial fashion around the top-left hand corner of the transformed sub-image, with sensitivity decreasing with movement out from this corner. This was backed up with the protection Schemes 4 and 5, i.e. percentage S.I.P. schemes. For example, it was found that the percentage protection barrier, which moves in the same radial fashion out from the top-left hand corner for increasing protection, could be reduced to provide protection for only 40 to 50% of the bits of each sub-image, for a channel error rate of 1 in 10^3. This was found to be less than that of the corresponding protection scheme when used in conjunction with the threshold scheme.

Due to the nature of the Huffman coding involved with the threshold scheme, the relative importance of each coefficient becomes dependent on neighbouring coefficients. Thus an error in one coefficient code word can cause incorrect decoding of several

following words. This tends to increase the sensitivity of the overall schemes when used together with the threshold coding method. In general then, all coefficients tend to carry an equal amount of importance and channel errors cannot be tolerated in any coefficient unless they occur towards the "tail-end" of a sub-image. Even then, disastrous E.O.S. error effects which cause the error effect to propagate to further sub-images, see Section 4.4.3.2, can occur unless this parameter is replaced by the sub-image header (SH) parameter, as in Section 5.2.2, and high redundancy protection provided on all SH words.

The conclusion is thus reached that, with the zonal scheme, due to coefficient word independence and the ability to define some coefficients as more sensitive than others, protection schemes can be used which concentrate and give more consideration to those more sensitive coefficient areas. With the threshold scheme this is true to a lesser degree and provided the most sensitive parameters, which usually contribute to give the scheme it's adaptivity, are highly protected in order to avoid disastrous reconstruction effects.

7.3 Are all Bits of a Word, Correspon-ding to a Certain Coefficient, as Important as Each Other?

Results with the zonal scheme tend to suggest that there is a varying degree of importance when considering the position of any bit in a defined coefficient code word. However, although this is the case it tends to be more noticeable with the trailing coefficient words, which tend to be short, consisting of typically 4 bits or less. In general, errors in the least significant positions of coefficient code words still give noticeable visual error even though the effects are less disturbing than those resulting from the most significant bits being in error. To avoid as much visual error as possible, particularly at high channel error rates, it is better to ignore the varying degree of bit importance of a coefficient code word. This will also cut down on channel error protection complexity, resulting in simpler communication schemes.

The dependency in importance of coefficient code words, with the threshold scheme, came about due to the nature of the Huffman coding process involved. This same factor leads to a similar result in that, all the bits of any particular coefficient code word tend to carry the same amount of importance. Channel errors cannot be tolerated in practically any position of a coefficient

code word, since false decoding usually results in a shortening or lengthening of that particular code word and the following ones causing a loss of word synchronization. Word synchronization is usually regained after a few words have been incorrectly decoded, however, the corresponding, subsequent, decoded coefficients will almost certainly be placed in incorrect address positions.

The conclusion reached here then is that both zonal and threshold schemes can be protected through methods which ignore any relative importance of bit position of any word. With the zonal scheme, the varying degree of importance, often, does not include unnoticeable error effects, and with the threshold scheme, the nature of the Huffman coding process results in all bits carrying roughly the same amount of importance.

7.4 Does Adjustment of Source Coding Parameters Make the Coded Data Less Sensitive to Channel Errors?

The effect of varying the initial source coding bit-rate and the sub-image size, in conjunction with the zonal scheme, and also in conjunction with various error protection schemes has been observed and general conclusions can be drawn. It was seen that, in the absence of error protection, an increase in the initial

source coding rate serves to lower the overall distortion and image reconstructions are acceptable when subject to low channel error rates, specifically less than 1 in 10^3. However, this increase in performance is slight and does not seem to justify the resulting increase in overall bit-rate. For all initial source coding rates, in the absence of error protection, the smaller sub-image size helps to reduce the escalating increase in distortion that results as the channel error rate is increased past 1 in 10^3. However, once again, despite this restraining property, at relatively high error rates reconstructions are as bad, subjectively, for either the 8x8 or 16x16 pixels square sub-image size. With error protection Schemes 2 to 5, an increase in initial source coding rate generally results in a lowering of the overall distortion, although once again the corresponding increase in overall bit-rate is not always justified, particularly at high error rates e.g. > 1 in 10^3. Also, the selection of the larger sub-image size is preferable when lower distortion is required and this tends to hold generally for the zonal scheme. Should the channel error rate become very high, and protection start to break down, e.g. with an error rate of 1 in 10^2, then no preference is found when considering sub-image size.

With the threshold scheme, the E.O.S. parameter requires modification to the SH parameter with relatively high error

protection before the scheme can even hope to cope with any practical channel error rate. Hereafter observation of other parameters are the same as those for the zonal scheme. It was found that, in the absence of any error protection scheme, results were similar to those of the zonal scheme giving no clear advantage in having a higher initial coding rate or larger/smaller sub-image size. With equal importance Hamming or Convolutional coding, Schemes 2 and 3, an increase in the initial source coding bit-rate can result in lower averall distortion although the resulting systems can be unpredictable. With the percentage S.I.P. schemes, Schemes 4 and 5, there is no distinct advantage in increasing the initial source coding rate, in terms of lower distortion, however it does tend to give a smoother ditortion versus percentage S.I.P. relationship, thus making the system more predictable. With all schemes, Schemes 2 to 5, the confining of error effects to smaller localized regions through use of the smaller sub-image size is preferable, not only in terms of lower distortion but also to make the system response more predictable, specifically at high error rates, approaching 1 in 10^{2}.

To summarize, the general conclusions reached are :-

For the Zonal Scheme.

Lower distortion, in the presence of errors, will be acheived through :

(a) Increasing the initial source coding rate,

(b) Using the larger Sub-image size,

(c) Use of an error protection scheme,

but (a) will result in a significant increase in bit-rate, especially when used with (c), which cannot always be justified when compared to using a lower initial source coding rate if this leads to higher, but equally, acceptable distortion.

For the Threshold Scheme.

Protection and possibly modification of the most sensitive parameter, the E.O.S. word, is imperative if channel errors are present. Increasing the initial source coding rate is, in general, not justified in terms of lower distortion, but can serve to make system behaviour more predictable, especially when an error protection scheme is used. A smaller sub-image size tends to result in lower overall distortion when errors are present and which cannot all be corrected by the protection scheme, i.e. for high error rates approaching 1 in 10^2.

7.5 What are the Best Combined Source/Error Protection Methods? And Hence what would be the Best System for Transmission of Image Data over a Channel which has a Specified Error Rate.

The error protection schemes provide a solution to the second of the component problems related to the fundamental communication problem, i.e. they make a decision as to how to transmit the necessary image data accurately. As expected, in most cases there is a resulting expansion, an increase in redundancy, in the data necessary for transmission.

It is unfortunate that, in the case of the threshold scheme, the error protection scheme (Scheme 6) which promised much, in terms of keeping the overall bit-rate the same as the initial source coding rate for lowest, acceptable, distortion did not prove to be implementable due to the amount of data storage required and the length of time that would be involved when searching through this data upon decoding. Should this difficulty be overcome, the theoretical result indicates that such a protection scheme would be extremely useful. In fact any such scheme, trying to make use of any remaining redundancy in the source coded bit-stream, would be useful particularly as they do not add to the overall average bit-rate of the source coded data.

At very high channel error rates, approaching 1 in 10^2, it has been seen that it is advis-able to protect every bit of the source coded data when using the Hamming or Convolutional error correcting schemes described in this thesis. This is true no matter what type of source coder is used. Thus, at such error rates, equal importance protection schemes would be used.

At lower channel error rates, less than or equal to 1 in 10^3, a percentage S.I.P. scheme, such as those described in this thesis can be used. The lower the channel error rate, the less the percentage S.I.P. need be for adequate protection. The non-adaptive zonal scheme lends itself better to such a protection method and it is thought that such a method would also be suited to an adaptive zonal scheme. The increased sensitivity of the threshold scheme due to the nature of coding restricts reducing the percentage S.I.P. barrier to far.

At very low channel error rates, typicaly 1 in 10^6, an increase in initial source coding bit-rate and use of the larger sub-image size (Scheme 1) can suffice for protection, more for the zonal scheme than the threshold one.

In general, the Hagelbarger convolutional codes described in this thesis would not be used to protect data from random,

uniform, independent channel errors. Rather, use would be made of the Hamming codes which are specifically designed for this " white noise " type of error. However, it has been seen that the algebraic implementation of the convolutional codes with image data results in a performance which is comparable with the Hamming codes. This suggests that convolutional codes of this kind are a superior form of error protection to use, for the protection of image data, as they are also able to cope with burst errors, whereas the Hamming codes cannot. For adequate protection at high error rates, if a Hamming code is used then it would be the Hamming (7, 4) code, which does not achieve much over the 50% redundant Hagelbarger convolutional codes in terms of the redundancy re-introduced into the source coded data stream. This would be true, particularly when the source coding rate is very low. This, again, points to the conclusion of using convolutional codes as apposed to block codes.

To summarize these conclusions, it has been found that the best methods of protection are dependent on the requirements of the system i.e. which type of coder is to be used and the channel error rate with which the system has to cope :-

For the best results in terms of lowest bit-rate and acceptable distortion when :

Zonal Coding

In general, source code to the lowest possible/acceptable bit-rate and then use any of Schemes 1,3,5 depending on what the known channel error rate is.

Threshold Coding

In general, source code to the lowest possible/acceptable bit-rate and then protect every bit of the remaining data through use of the Hagelbarger convolutional codes, Scheme 3, ensuring first that the sub-image headers (SH) words have adequate protection. Alternatively, if a scheme such as the maximum likelihood decoding of Huffman coded image data can be implemented this would be the one to try first.

From the above a further, general, conclusion can be drawn and that is that the zonal coding scheme lends itself more readily to different methods of error protection than the more sensitive threshold scheme, but despite this, provided the right protection is used, the threshold scheme can still out-perform the zonal scheme and thus would be the ultimate choice.

7.6 Suggestions for Further Work.

The work in this thesis provides a foundation from which various extensions are possible. The work covers the basic error effects, and simple, but effective, schemes to provide protection against them, hitherto these have been guessed at or glossed over or often ignored in the design of an image coding scheme. To this extent the work is complete in it's present form. The study of the schemes could now be extended by observing the effect of using probabilistic convolutional decoding methods, or by turning attention to double error correcting codes etc. Examination of the behaviour of image coding schemes where the channel error rate starts to exceed 1 in 10^{2} would certainly require turning attention to error protection codes with a higher correcting capability.

Further work could also be carried out to observe the behaviour of an adaptive zonal scheme, which would probably have a performance lying somewhere between the non-adaptive zonal and the threshold schemes.

Finally, it would be of great benefit, in regard to the study of transmission of source coded image data, if an error correction scheme, such as the maximum likelihood decoding of

Huffman coded image data, could be implemented on actual image data.

APPENDIX

APPENDIX 1

A.1 A DCT Algorithm from a FFT algorithm.

The main formula defining the forward, one-dimensional DCT was seen to be given by equation (3.4.1) :-

$$C(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x).\cos\frac{(2x+1)u\pi}{2N} \qquad (A1)$$

from De Moivres theorem :

$$e^{j\emptyset} = \cos\emptyset + j\sin\emptyset \qquad (A2)$$

enabling equation (A1) to be written as :

$$C(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x).\ \mathrm{Re}\left[\ \exp\frac{-j(2x+1)u\pi}{2N}\ \right],$$

where,

Re denotes the real part of the expression in parenthesis. Thus,

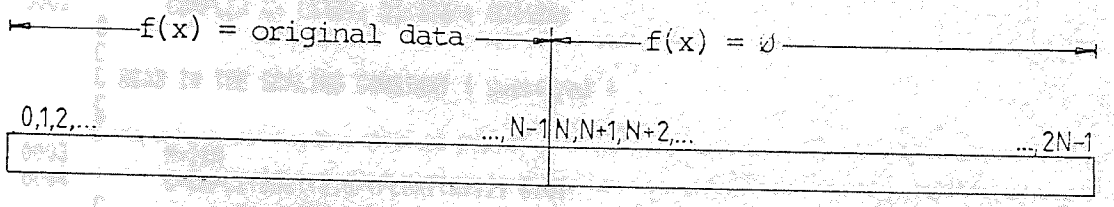
$$C(u) = \sqrt{\frac{2}{N}}\ .\ \mathrm{Re}\left[\ \exp\left(\frac{-ju\pi}{2N}\right)\sum_{x=0}^{N-1} f(x).\exp\left(\frac{-j2\pi\, ux}{2N}\right)\ \right] \qquad (A3)$$

The one-dimensional, forward, FFT is given by :

$$F(u) = \frac{1}{N} \sum_{x=\emptyset}^{N-1} f(x).\exp\left(\frac{-j2\pi ux}{N}\right) \qquad (A4)$$

The similarity between equations A3 and A4 can be seen. In order to get A3 into the exact form one can consider additional terms for which f(x) is set to zero (i.e. N, N+1, ..., 2N-1, for which f(x) = ∅).

i.e. :



Hence,

$$\sum_{x=\emptyset}^{N-1} f(x).\exp\left(\frac{-j2\pi ux}{2N}\right) \Longrightarrow \sum_{x=\emptyset}^{M-1} f(x).\exp\left(\frac{-j2\pi ux}{M}\right),$$

$$f(x) = \emptyset, \quad N \leqslant x \leqslant M-1$$

$$M = 2N$$

Thus A3 becomes :

$$C(u) = \sqrt{\frac{2}{N}}\, \mathrm{Re}\left[\exp\left(\frac{-ju\pi}{2N}\right) \sum_{x=\emptyset}^{M-1} f(x).\exp\left(\frac{-j2\pi ux}{M}\right)\right], \qquad (A5)$$

$$f(x) = \emptyset \quad N \leqslant x \leqslant M-1$$

$$M = 2N$$

N = No. of spatial data coefficients

The forward DCT algorithm is thus as shown in the following program entitled " FORCOS.* "

```
FORTRAN IV      V02.5-3     Thu 13-Jun-85 20:01:20

C                                  ' FORCOS.S '
C                                  ============
C                            (SUBROUTINE)
C
C                    ********************************************
C TO PERFORM A * 2*N POINT FORWARD, DISCRETE COSINE TRANSFORM *
C                    ********************************************
C                     USING THE F.F.T. ALGORITHM.
C
C
C CALLING FORMAT;-        U-INPUT/OUTPUT DATA
C                         N-NO. OF SAMPLES( THE BLOCK WIDTH
C                                           BEING PROCESSED)
C                          e.g. Block size=32*32
C                          Therefore     N=32
C
0001        SUBROUTINE FORCOS(U, N)
C
C
C DECLARE VARIABLES/DIMENSIONS
C
C
0002        COMPLEX C, U(32), U1(128), U2(128)
C
C
C READ IN THE SCALING CONSTANT ( sqrt(2/n) )
C
C
0003        M=2*N
0004        C=CMPLX(SQRT(2.0/(FLOAT(N))), 0.0)
C
C
C READ IN 'U' TO U1 AND THE CORRESPONDING SCALING
C VALUES, IN (r, @) FORM, TO U2
C
C
0005              DO 30 J=1, N
0006                 U1(J)=U(J)
0007                 U2(J)=CMPLX( 1.0, -(FLOAT(J-1))*3.141593/
     1                                                 (FLOAT(2*N)) )
0008  30          CONTINUE
C
C
C FILL IN REST OF U1 WITH ZEROS ( TO SATISFY; f(n)=0,N-1<n<M-1) )
C
C
0009              DO 35 J=N+1, M
0010                 U1(J)=CMPLX(0.0, 0.0)
0011  35          CONTINUE
C
C
C TRANSFER U1, U2, C TO DBF#1, 2, & 7
C
C
0012                CALL KHFAB(U1,1,2*M)
0013              CALL KHFAB(U2,2,2*M)
0014              CALL KHFAB(C,7,2)
C
C
C PERFORM F.F.T. ON U1 (IN DBF#1)
C
C
0015              CALL KFTC1(2*M,1,1)
C
C
C BIT REVERSE RESULT INTO DBF#9
C
C
0016              CALL KBRVC(9,1)
C
C
C CONVERT VALS. IN DBF#2 FROM (r,@) FORM TO (x,y) FORM;
C RESULT IN DBF#3
C
```

```
C
0017            CALL KPOCH(3,2)
C
C
C MULTIPLY DBF#9 WITH DBF#3 TO FORM:-
C                                  M-1
C  EXP[-(JU#)/(2N)] * {   f(x).EXP[-(J2#ux)/(M)]
C                                  x=0
C
C RESULT IN DBF#4
C
C
0018            CALL KMULC(4,9,3)
C
C
C TAKE THE REAL PART BY DEMULTIPLEXING DBF#4(REQ. RESULT IN DBF#5)
C
C
0019            CALL KDMX1(5,6,4)
C
C
C PUT IMAGINARY PART = 0 BY ZEROING DBF#6, ALSO ZERO DBF#4
C
C
0020            CALL KZRDB(6)
0021            CALL KZRDB(4)
C
C
C MULTIPLEX DBF# 5 & 6 BACK TO DBF#4
C
C
0022            CALL KMUX1(4,5,6)
C
C
C MULTIPLY REAL PART BY C i.e.SQRT(2/N). RESULT IN DBF#8
C
C
0023            CALL KMLSC(8,4,7)
C
C
C TRANSFER RESULT TO HOST, REPEAT
C
C
0024            CALL KABHF(U1,8,2*N)
C
C
C READ FIRST N VALUES OF U1 BACK INTO U
C
C
0025            DO 40 J=1, N
0026               U(J)=U1(J)
0027  40        CONTINUE
0028            U(1)=U(1)*0.70711
0029  500   CONTINUE
0030        CALL KINIT
0031        RETURN
0032        END
```

The inverse DCT, by a similar process, can be implemented using a 4N point FFT algorithm defined by :

$$f(x) = \frac{1}{\sqrt{N}}.C(\emptyset) + \frac{4N.\sqrt{2}}{\sqrt{N}}.\ \mathrm{Re}\left[\frac{1}{4N}\sum_{u=\emptyset}^{4N-1} C'(u).\exp\left(\frac{j2\Pi uz}{4N}\right)\right] \quad (A5)$$

where,

$$z = 2x+1$$

$$C'(\emptyset) = \emptyset$$

$$C'(u) = C(u) \qquad 1 \leqslant u \leqslant N-1$$

$$C'(u) = \emptyset \qquad N \leqslant u \leqslant 4N-1$$

the other parameters being as defined under equation (3.4.1)

Using equation (A5) the algorithm developed to implement the inverse DCT is then as shown in the following program entitled " INVCOS.* ".

```
      C                                  ' INVCOS.* '
      C                                  ============
      C                           (SUBROUTINE)
      C
      C                 *********************************************
      C TO PERFORM A  * 2*N POINT INVERSE, DISCRETE COSINE TRANSFORM *
      C                 *********************************************
      C                   USING THE INVERSE F.F.T. ALGORITHM.
      C
      C
      C CALLING FORMAT:-        U-INPUT/OUTPUT DATA
      C                         N-NO. OF SAMPLES( THE BLOCK WIDTH
      C                                         BEING PROCESSED)
      C                           e.g. Block size=32*32
      C                           Therefore     N=32
      C                           (MAX. N=32)
      C
 0001         SUBROUTINE INVCOS(U, N)
      C
      C
      C DECLARE VARIABLES/DIMENSIONS
      C
      C
 0002         COMPLEX C1, U(32), U1(128), U2(128)
      C
      C
      C READ IN THE MULTIPLYING CONSTANT ( 4*N*sqrt(2/n) )
      C
      C
 0003         C1=CMPLX(4.0*FLOAT(N)*SQRT(2.0/(FLOAT(N))), 0.0)
      C
      C
      C READ IN U TO U1 (1 TO N) AND FILL N+1 TO 2N,
      C 2N+1 TO 3N, AND 3N+1 TO 4N WITH ZERO.
      C          ( TO ENSURE C(u)=0, n-1< u <4n-2 )
      C
      C
 0004               DO 30 J=1, N
 0005                 U1(J)=U(J)
 0006                 U1(N+J)=CMPLX(0.0, 0.0)
 0007                 U1(2*N+J)=CMPLX(0.0, 0.0)
 0008                 U1(3*N+J)=CMPLX(0.0, 0.0)
 0009 30            CONTINUE
      C
      C
      C TAKE & STORE VALUE CORRESPONDING TO C = C(0)/sqrt(n)
      C
      C
 0010             CC=REAL(U1(1))/SQRT(FLOAT(N))
      C
      C
      C SET U1(1)=ZERO ( C(0)=0 ), SO THAT SUMMATIONS CAN BE
      C STARTED FROM ZERO:-
      C
      C
 0011             U1(1)=CMPLX(0.0, 0.0)
      C
      C
      C
      C TRANSFER U1, C1, CC TO DBF#1, 2, & 3
      C
      C
 0012                 CALL KHFAB(U1,1,8*N)
 0013             CALL KHFAB(C1,2,2)
 0014             CALL KHFAB(CC,3,1)
      C
      C
      C BIT REVERSE INPUT DATA OF DBF#1, BEFORE TAKING INVERSE FFT,
      C RESULT IN DBF#4
      C
      C
 0015             CALL KBRVC(4, 1)
      C
      C PERFORM INVERSE F.F.T. ON U1 (IN DBF#4), IN PLACE
      C
      C
 0016             CALL KITC1(8*N,4,4)
      C
```

```
C
C MULTIPLY ALL VALUES BY C1(DBF#2),THUS FORMING
C
C
C                         4n-1               J2#uy/4n
C         SQRT(2/N) {    C'(u).e
C                          u=0
C
C
C I.E. DBF#4 * DBF#2 RESULT IN DBF#5
C
C
        CALL KMLSC(5,4,2)
C
C
C TAKE THE REAL PART BY DEMULTIPLEXING DBF#5(REQ. RESULT IN DBF#6)
C
C
        CALL KDMX1(6,7,5)
C
C
C DEMULTIPLEX AGAIN, THIS TIME KEEPING THE 'IMAGINARY' PART
C
C
        CALL KDMX1(8,9,6)
C
C
C DBF#9 SHOULD NOW HOLD THE REAL VALUES OF:-
C
C                         4n-1         J2#u(2x+1)/4n
C         sqrt(2/n) {  C'(u).e
C                         u=0
C
C PUT 'IMAGINARY' PART = 0 BY ZEROING DBF#8, ALSO ZERO DBF#6
C
C
        CALL KZRDB(8)
        CALL KZRDB(6)
C
C
C ADD CC ( C(0)/SQRT(N) ) I.E. DBF#3 TO ALL VALUES OF DBF#9,
C LEAVE RESULT IN DBF#9
C
C
      IF ( ABS(CC).LT.0.00001 ) GO TO 50
          CALL KADDS(9,9,3)
 50   CONTINUE
C
C
C ALTHOUGH REAL RESULTS ARE OBTAINED, THE DATA IS ASKED FOR IN
C COMPLEX FORM SO MULTIPLEX DBF#9 WITH THE ZERO DBF#8, PUTTING
C THE RESULT IN DBF#6
C
C
        CALL KMUX1(6,9,8)
C
C
C DBF#6 SHOULD NOW CONTAIN THE REQUIRED ANSWER BUT WILL HOLD
C 4N VALUES, ONLY THE FIRST 2N VALUES ARE REQUIRED, SO TRANSFER
C ONLY THESE BACK TO U1 IN HOST
C
C
        CALL KABHF(U1,6,2*N)
C
C
C READ THESE N COMPLEX VALUES OF U1 BACK INTO ROW OF ARRAY
C
C
        DO 40 J=1, N
           U(J)=U1(J)
 40     CONTINUE
 500  CONTINUE
      CALL KINIT
      RETURN
      END
```

A.2 Two-dimensional DCT from One-dimensional Counter-parts.

Due to the separable property of the two-dimensional DCT, an algorithm for it's implementation can be derived by successive application of the one-dimensional case. This can be illustrated by Figure A.2.1 :

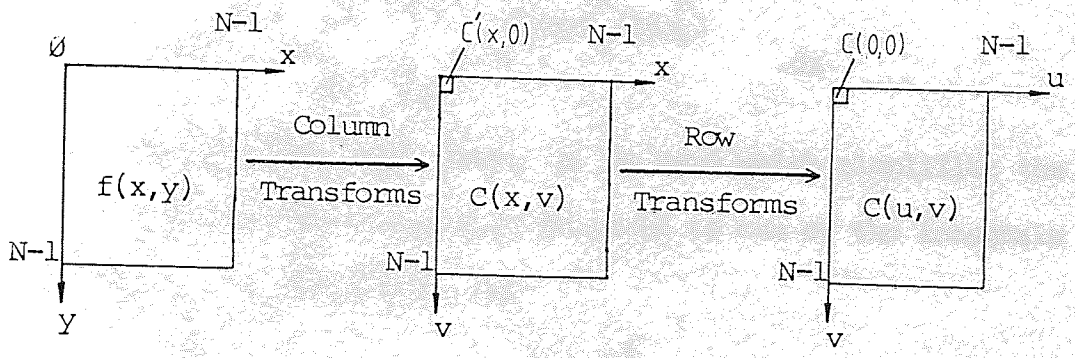


Figure A.2.1 Two-dimensional, Forward, DCT Implementation.

The inverse algorithm can be found in the same manner.

The method outlined above is adopted in this thesis.

A.3 An Integral Evaluation (of Equation (3.7.3.2) to give Equation (3.7.3.3)).

Equation (3.7.3.2) was seen to be :

$$y = \frac{(2\pi)^{1/6}.\sigma_a^{1/3}\int_{\emptyset}^{z} e^{(z-\mu)^2/(6\sigma_a^2)}.dz}{(2\pi)^{1/6}.\sigma_a^{1/3}\int_{\emptyset}^{1} e^{(z-\mu)^2/(6\sigma_a^2)}.dz} \qquad (A6)$$

It is insured that the mean, μ, is zero which simplifies the above. Consideration need only be given to one of the integrals in the above equation and so, set

$$I = \int_{\emptyset}^{z} (2\pi)^{1/6}.\sigma_a^{1/3}.e^{(z)^2/(6\sigma_a^2)}.dz \qquad (A7)$$

Equation (A7) can be solved using a series technique, since

$$e^{x} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

Substituting $x = z^2/(6\sigma_a^2)$, the following can be written :

$$e^{(z)^2/(6\sigma_a^2)} = 1 + \frac{z^2}{6\sigma_a^2} + \left(\frac{z^2}{6\sigma_a^2}\right)^2.\frac{1}{2!} + \left(\frac{z^2}{6\sigma_a^2}\right)^3.\frac{1}{3!} + \left(\frac{z^2}{6\sigma_a^2}\right)^4.\frac{1}{4!} + \ldots$$

$$= 1 + \frac{z^2}{6\sigma_a^2} + \frac{z^4}{6^2\sigma_a^4}.\frac{1}{2!} + \frac{z^6}{6^3\sigma_a^6}.\frac{1}{3!} + \frac{z^8}{6^4\sigma_a^8}.\frac{1}{4!} + \ldots$$

Substituting this expression into equation (A7) and solving the resulting integral gives:

$$I = (2\pi)^{1/6}\,\sigma_a^{1/3}\left[\, z + \frac{z^3}{\sigma_a^2 3.6} + \frac{z^5}{\sigma_a^4 5.6^2.2!} + \frac{z^7}{\sigma_a^6 7.6^3.3!} + \ldots \,\right]_{\emptyset}^{z} \qquad (A8)$$

The above equation (A8) is seen to be the numerator of the required result (equation (3.7.3.3)). The required denominator can be obtained in the same way and results in the same expression with the limits of 1 and Ø in place of z and Ø. Hence, the integral evaluation is:

$$y = \frac{\left[\, z + \frac{z^3}{\sigma_a^2 3.6}\cdot\frac{1}{} + \frac{z^5}{\sigma_a^4 5.6^2.2!}\cdot\frac{1}{} + \frac{z^7}{\sigma_a^6 7.6^3.3!}\cdot\frac{1}{} + \ldots \,\right]_0^{z}}{\left[\, z + \frac{z^3}{\sigma_a^2 3.6}\cdot\frac{1}{} + \frac{z^5}{\sigma_a^4 5.6^2.2!}\cdot\frac{1}{} + \frac{z^7}{\sigma_a^6 7.6^3.3!}\cdot\frac{1}{} + \ldots \,\right]_0^{1}} \qquad (A9)$$

which is seen to be the required equation (3.7.3.3)

APPENDIX 2

This Appendix contains the numerical tables of results obtained in this work. The following key indicates the section in the thesis where the figure occurs, the figure number itself, and the corresponding table(s) of results that can be found, in order, in this Appendix.

RELEVANT SECTION	FIGURE NUMBER	TABLE NUMBER
4.3.1	4.3.1.4(a)	A1.4.3.1.1 to A1.4.3.1.4
	4.3.1.4(b)	A1.4.3.1.5 to A1.4.3.1.8
4.3.2	4.3.2.5 to 4.3.2.8	A1.4.3.2.1 to A1.4.3.2.8
6.1	6.1.1 6.1.2	A1.6.1.1
6.2	6.2.1 6.2.2	A1.6.2.1
6.3	6.3.1 6.3.2	A1.6.3.1
6.4 & 6.5	6.4.1 & 6.5.1 6.4.2 & 6.5.2 6.4.4 & 6.5.4 6.4.5 & 6.5.5	A1.6.4.1 A1.6.4.2 A1.6.4.3 A1.6.4.4
6.4 & 6.5	6.4.6 & 6.5.6 6.4.7 & 6.5.7 6.4.9 & 6.5.9 6.4.10 & 6.5.10	A1.6.4.5 A1.6.4.6 A1.6.4.7 A1.6.4.8

TABLE A1.4.3.1.1
Zonal Coder
Sub-image Size= 32 X 32

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)
1·410	0·0039301	31·067
1·187	0·0044447	30·533
1·065	0·0048792	30·128
0·986	0·0052976	29·771
0·917	0·0059321	29·279
0·876	0·0063070	29·013
0·811	0·0072080	28·433
0·751	0·0083541	27·792

TABLE A1.4.3.1.2
Zonal Coder
Sub-image Size= 16 X 16

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)
1·699	0·0033777	31·727
1·449	0·0040727	30·913
1·340	0·0042944	30·548
1·230	0·0048741	30·132
1·156	0·0055385	29·578
1·102	0·0060184	29·217
0·961	0·0067784	28·700
0·867	0·0080390	27·959

TABLE A1.4.3.1.3
Zonal Coder
Sub-image Size= 8 X 8

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)
1·984	0·0036931	31·338
1·750	0·0047614	30·234
1·422	0·0065612	28·841
1·281	0·0081612	27·894
1·094	0·0091810	27·383
0·953	0·0123971	26·078
0·734	0·0199233	24·018

TABLE A1.4.3.1.4
Zonal Coder
Sub-image Size= 4 X 4

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)
1·938	0·0068183	28·675
1·563	0·0090434	27·448
1·375	0·0110340	26·584
1·125	0·0167380	24·774
1·063	0·0172133	24·653
0·813	0·0235042	23·300
0·750	0·0235201	23·297

TABLE A1.4.3.1.5
Zonal Coder
Sub-image Size= 32 X 32

Average Bit-rate (bits/pixel)	Normalized, Theoretical Distortion
1.410	0.0002132
1.187	0.0004016
1.065	0.0005661
0.986	0.0007540
0.917	0.0008840
0.876	0.0010593
0.811	0.0014103
0.751	0.0016484

TABLE A1.4.3.1.6
Zonal Coder
Sub-image Size= 16 X 16

Average Bit-rate (bits/pixel)	Normalized, Theoretical Distortion
1.699	0.0001641
1.449	0.0003275
1.340	0.0004909
1.230	0.0005775
1.156	0.0007205
1.102	0.0008634
1.016	0.0011494
0.961	0.0014354
0.867	0.0020073

TABLE A1.4.3.1.7
Zonal Coder
Sub-image Size= 8 X 8

Average Bit-rate (bits/pixel)	Normalized, Theoretical Distortion
1.984	0.0003191
1.750	0.0006378
1.422	0.0012180
1.281	0.0019352
1.094	0.0028914
0.953	0.0059991
0.891	0.0071944
0.734	0.0119755

TABLE A1.4.3.1.8
Zonal Coder
Sub-image Size= 4 X 4

Average Bit-rate (bits/pixel)	Normalized, Theoretical Distortion
1.938	0.0143317
1.563	0.0148986
1.375	0.0435618
1.125	0.0614763
1.063	0.0711150
0.813	0.2334560
0.750	0.3588570

TABLE A1.4.3.2.1
Threshold Coder
Sub-image Size = 32 X 32

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)	Threshold
1.896	0.001344	35.729	10.0
1.623	0.001696	34.718	12.0
1.419	0.002052	33.891	14.0
1.267	0.002384	33.238	16.0
1.138	0.002751	32.617	18.0
1.031	0.003109	32.085	20.0
0.945	0.003471	31.607	22.0
0.863	0.003851	31.124	24.0

TABLE A1.4.3.2.2
Threshold Coder
Sub-image Size = 16 X 16

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)	Threshold
2.016	0.001265	35.991	10.0
1.711	0.001624	34.906	12.0
1.504	0.001943	34.128	14.0
1.334	0.002286	33.421	16.0
1.207	0.002611	32.844	18.0
1.092	0.002972	32.280	20.0
0.997	0.003331	31.786	22.0
0.920	0.003655	31.383	24.0

TABLE A1.4.3.2.3
Threshold Coder
Sub-image Size= 8 X 8

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio(SNR), (dB)	Threshold
1.519	0.002212	33.563	16.0
1.370	0.002547	32.952	18.0
1.249	0.002893	32.399	20.0
1.144	0.003247	31.897	22.0
1.063	0.003569	31.486	24.0
0.991	0.003895	31.106	26.0
0.870	0.004576	30.407	30.0
0.774	0.005278	29.787	34.0

TABLE A1.4.3.2.4
Threshold Coder
Sub-image Size= 4 X 4

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)	Threshold
1.919	0.002141	33.706	16.0
1.525	0.003941	31.056	24.0
1.107	0.005267	29.796	34.0
1.002	0.006224	29.070	40.0
0.887	0.007757	28.114	50.0

TABLE A1.4.3.2.5
Threshold Coder
Sub-image Size = 32 X 32

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio(SNR), (dB)	Threshold
1.605	0.001561	35.079	10.0
1.389	0.001911	34.199	12.0
1.226	0.002254	33.481	14.0
1.098	0.002609	32.847	16.0
0.994	0.002958	32.301	18.0
0.902	0.003329	31.788	20.0
0.829	0.003694	31.336	22.0
0.763	0.004057	30.929	24.0

TABLE A1.4.3.2.6
Threshold Coder
Sub-image Size= 16 X 16

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)	Threshold
1.742	0.001496	35.263	10.0
1.496	0.001838	34.367	12.0
1.319	0.002166	33.655	14.0
1.175	0.002507	33.020	16.0
1.068	0.002829	32.494	18.0
0.966	0.003204	31.955	20.0
0.884	0.003550	31.509	22.0
0.821	0.003878	31.125	24.0

TABLE A14.3.2.7
Threshold Coder
Sub-image Size = 8 X 8

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio(SNR), (dB)	Threshold
1.358	0.002410	33.192	16.0
1.231	0.002738	32.638	18.0
1.121	0.003086	32.118	20.0
1.033	0.003434	31.654	22.0
0.961	0.003752	31.269	24.0
0.894	0.004089	30.895	26.0
0.788	0.004764	30.231	30.0
0.702	0.005473	29.630	34.0

TABLE A14.3.2.8
Threshold Coder
Sub-image Size= 4 X 4

Average Bit-rate (bits/pixel)	Normalized, Average Sample Mean Square Error	Signal-to-Noise Ratio (SNR), (dB)	Threshold
1.734	0.002295	33.403	16.0
1.272	0.003713	31.314	24.0
0.998	0.005415	29.675	34.0
0.905	0.006376	28.966	40.0
0.801	0.007912	28.028	50.0

TABLE A1.6.1.1

Channel Error-rate	Sub-image Size	Bit-rate (approx.)	Zonal Scheme Distortion	Threshold Scheme Distortion
0.000001	8 * 8	1.00	0.009181	0.004417
0.000010			0.009181	0.004417
0.000100			0.009340	0.004537
0.001000			0.013507	0.008990
0.010000			0.057412	0.038708
0.000001	8 * 8	1.85	0.004761	0.001805
0.000010			0.004761	0.001805
0.000100			0.005894	0.002283
0.001000			0.012902	0.008786
0.010000			0.062902	0.055223

Channel Error-rate	Sub-image Size	Bit-rate (approx.)	Zonal Scheme Distortion	Threshold Scheme Distortion
0.000001	16 * 16	1.00	0.006779	0.003029
0.000010			0.006779	0.003029
0.000100			0.007032	0.003690
0.001000			0.010678	0.010712
0.010000			0.055080	0.077842
0.000001	16 * 16	1.85	0.003213	0.001628
0.000010			0.003213	0.001628
0.000100			0.010732	0.001694
0.001000			0.017881	0.014576
0.010000			0.090808	0.073351

TABLES A1.6.2.1

THRESHOLD SCHEME

Sub-image Size (pixels square)	Initial Average Bit-rate (bits/pixel)	Initial Normalized Distortion	Total Average Bit-rate for: Hamming code (15, 11)	Total Average Bit-rate for: Hamming code (7, 4)	Normalized Average Sample Mean Square Error for: Hamming code (15, 11)	Normalized Average Sample Mean Square Error for: Hamming code (7, 4)
16 * 16	1.070	0.003029	1.567	1.965	0.008067	0.005652
	1.240	0.002480	1.809	2.278	0.012155	0.008162
	1.500	0.001920	2.153	2.718	0.008812	0.005900
	1.860	0.001503	2.524	3.194	0.005914	0.004779
8 * 8	1.070	0.004417	1.909	2.273	0.005529	0.005503
	1.300	0.003359	2.236	2.693	0.007374	0.004520
	1.530	0.002683	2.546	3.090	0.005723	0.004639
	1.960	0.001805	3.123	3.831	0.004611	0.004371

ZONAL SCHEME

Sub-image Size (pixels square)	Initial Average Bit-rate (bits/pixel)	Initial Normalized Distortion	Total Average Bit-rate for: Hamming code (15, 11)	Total Average Bit-rate for: Hamming code (7, 4)	Normalized Average Sample Mean Square Error for: Hamming code (15, 11)	Normalized Average Sample Mean Square Error for: Hamming code (7, 4)
16 * 16	0.960	0.006779	1.311	1.682	0.008824	0.007617
	1.250	0.004836	1.710	2.190	0.011494	0.006969
	1.500	0.003899	2.040	2.618	0.003899	0.003899
	1.820	0.003213	2.488	3.192	0.014627	0.004899
8 * 8	1.090	0.009182	1.492	1.914	0.012353	0.012560
	1.250	0.007729	1.710	2.190	0.014454	0.009304
	1.450	0.005084	1.982	2.543	0.011493	0.006056
	1.750	0.004761	2.387	3.063	0.006235	0.008291

TABLES A1.6.3.1

THRESHOLD SCHEME

Sub-image Size (pixels square)	Initial Average Bit-rate (bits/pixel)	Initial Normalized Distortion	Total Average Bit-rate for: Hagelbarger code CC(7)	Total Average Bit-rate for: Hagelbarger code CC(11)	Normalized Average Sample Mean Square Error for: Hagelbarger code CC(7)	Normalized Average Sample Mean Square Error for: Hagelbarger code CC(11)
16 * 16	1.070	0.003029	2.225	2.225	0.006596	0.004052
	1.240	0.002490	2.581	2.581	0.005812	0.004669
	1.500	0.001920	3.084	3.084	0.006037	0.005409
	1.860	0.001503	3.629	3.629	0.004268	0.006493
8 * 8	1.070	0.004417	2.513	2.513	0.005291	0.004638
	1.300	0.003359	2.988	2.988	0.004203	0.004713
	1.530	0.002683	3.443	3.443	0.003810	0.005136
	1.960	0.001805	4.290	4.290	0.003598	0.004176

ZONAL SCHEME

Sub-image Size (pixels square)	Initial Average Bit-rate (bits/pixel)	Initial Normalized Distortion	Total Average Bit-rate for: Hagelbarger code CC(7)	Total Average Bit-rate for: Hagelbarger code CC(11)	Normalized Average Sample Mean Square Error for: Hagelbarger code CC(7)	Normalized Average Sample Mean Square Error for: Hagelbarger code CC(11)
16 * 16	0.960	0.006779	1.922	1.922	0.007025	0.011775
	1.250	0.004806	2.500	2.500	0.008942	0.006302
	1.500	0.003899	2.992	2.992	0.003899	0.004431
	1.820	0.003213	3.649	3.649	0.003760	0.003760
8 * 8	1.090	0.009182	2.188	2.188	0.009319	0.010612
	1.250	0.007729	2.500	2.500	0.010589	0.010656
	1.450	0.005084	2.907	2.907	0.007341	0.006494
	1.750	0.004761	3.500	3.500	0.004465	0.005603

TABLES A1.6.4.1

Zonal Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.311	1.682	1.922	1.922	0.006779	0.006779	0.006779	0.006779
80.0	1.240	1.538	1.731	1.731	0.006785	0.006833	0.006797	0.006785
50.0	1.136	1.321	1.442	1.442	0.007096	0.007595	0.008833	0.007436
40.0	1.101	1.248	1.344	1.344	0.007253	0.007695	0.007835	0.008210
30.0	1.066	1.178	1.250	1.250	0.007334	0.008157	0.007318	0.009828
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL								
100.0	1.492	1.914	2.188	2.186	0.009182	0.009182	0.009182	0.009182
90.0	1.452	1.832	2.078	2.078	0.009182	0.009182	0.009187	0.009182
70.0	1.372	1.668	1.860	1.860	0.009199	0.009209	0.009385	0.009299
50.0	1.293	1.504	1.641	1.641	0.009882	0.010058	0.009739	0.009885
30.0	1.213	1.340	1.422	1.422	0.010754	0.012558	0.011709	0.012301

Zonal Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.710	2.194	2.508	2.508	0.004836	0.004836	0.004836	0.004836
80.0	1.619	2.007	2.258	2.258	0.004925	0.004849	0.004836	0.004836
60.0	1.528	1.819	2.008	2.008	0.004932	0.005071	0.004963	0.005358
40.0	1.436	1.629	1.754	1.754	0.005015	0.005085	0.005683	0.005037
30.0	1.391	1.535	1.629	1.629	0.005105	0.006037	0.006341	0.009258
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL								
100.0	1.705	2.188	2.500	2.500	0.007729	0.007729	0.007729	0.007729
80.0	1.614	2.000	2.250	2.250	0.007735	0.007740	0.007734	0.007729
60.0	1.523	1.813	2.000	2.000	0.007788	0.008640	0.008056	0.008056
40.0	1.432	1.625	1.750	1.750	0.009825	0.010038	0.009906	0.010147
30.0	1.387	1.531	1.625	1.625	0.012596	0.011016	0.011132	0.011563

TABLES A1.64.2

Zonal Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.040	2.618	2.992	2.992	0.003899	0.003899	0.003899	0.003899
80.0	1.931	2.393	2.692	2.692	0.003899	0.003901	0.003899	0.003901
60.0	1.823	2.170	2.395	2.395	0.003922	0.003939	0.003923	0.003947
40.0	1.713	1.944	2.094	2.094	0.004252	0.003933	0.004782	0.003996
30.0	1.660	1.833	1.946	1.946	0.006199	0.005303	0.003899	0.007100
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL								
100.0	1.982	2.543	2.907	2.907	0.005084	0.005084	0.005084	0.005084
80.0	1.874	2.320	2.610	2.610	0.005099	0.005138	0.005111	0.005168
50.0	1.720	2.004	2.188	2.188	0.005528	0.005342	0.005563	0.005563
40.0	1.664	1.889	2.035	2.035	0.007151	0.007090	0.007050	0.007202
30.0	1.612	1.781	1.891	1.891	0.007223	0.007227	0.007643	0.007643

Zonal Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.488	3.192	3.649	3.649	0.003213	0.003213	0.003213	0.003213
70.0	2.289	2.782	3.102	3.102	0.003262	0.003218	0.003225	0.003225
50.0	2.157	2.510	2.739	2.739	0.003395	0.003442	0.003346	0.003315
40.0	2.057	2.372	2.555	2.555	0.003505	0.003281	0.003408	0.003408
30.0	2.023	2.234	2.371	2.371	0.004147	0.004821	0.004131	0.004391
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL								
100.0	2.387	3.063	3.500	3.500	0.004761	0.004761	0.004761	0.004761
70.0	2.193	2.664	2.969	2.969	0.004882	0.004874	0.004949	0.004861
50.0	2.068	2.406	2.625	2.625	0.006430	0.005282	0.007126	0.005394
40.0	2.006	2.278	2.454	2.454	0.005161	0.006812	0.005115	0.005115
30.0	1.943	2.149	2.282	2.282	0.006695	0.007009	0.007268	0.006726

TABLES A1.64.3

Zonal Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.311	1.682	1.922	1.922	0.008824	0.007617	0.007025	0.011775
80.0	1.240	1.538	1.731	1.731	0.011451	0.013902	0.008768	0.008679
50.0	1.136	1.321	1.442	1.442	0.016268	0.012378	0.011561	0.013928
40.0	1.101	1.249	1.344	1.344	0.021337	0.020350	0.016820	0.018141
30.0	1.066	1.178	1.250	1.250	0.026144	0.025751	0.023393	0.030367
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL								
100.0	1.492	1.914	2.188	2.188	0.012353	0.012560	0.009319	0.010612
90.0	1.452	1.832	2.078	2.078	0.013593	0.010662	0.011320	0.012013
70.0	1.372	1.668	1.860	1.860	0.012986	0.011451	0.012930	0.011193
50.0	1.293	1.504	1.641	1.641	0.018215	0.016054	0.016620	0.015077
30.0	1.213	1.340	1.422	1.422	0.028815	0.027455	0.030494	0.031993

Zonal Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.710	2.194	2.508	2.508	0.011494	0.006969	0.008942	0.006302
80.0	1.619	2.007	2.258	2.258	0.011132	0.006376	0.006032	0.006741
60.0	1.528	1.819	2.008	2.008	0.010990	0.007172	0.007123	0.011213
40.0	1.436	1.629	1.754	1.754	0.011061	0.010960	0.011650	0.011942
30.0	1.391	1.535	1.629	1.629	0.021596	0.016759	0.021884	0.022710
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL								
100.0	1.705	2.188	2.500	2.500	0.014454	0.009304	0.010589	0.010656
80.0	1.614	2.000	2.250	2.250	0.010421	0.010239	0.011081	0.009309
60.0	1.523	1.813	2.000	2.000	0.016708	0.015574	0.017456	0.017112
40.0	1.432	1.625	1.750	1.750	0.028792	0.026377	0.025885	0.028368
30.0	1.387	1.531	1.625	1.625	0.036374	0.033249	0.039190	0.038665

TABLES A1.64.4

Zonal Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.040	2.618	2.992	2.992	0.003899	0.003899	0.003899	0.004431
80.0	1.931	2.393	2.692	2.692	0.010314	0.006012	0.007206	0.004902
60.0	1.823	2.170	2.395	2.395	0.009283	0.009395	0.007670	0.009196
40.0	1.713	1.944	2.094	2.094	0.011376	0.007653	0.012286	0.010433
30.0	1.660	1.833	1.946	1.946	0.018926	0.015933	0.019778	0.016813
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL								
100.0	1.982	2.543	2.907	2.907	0.011493	0.006056	0.007341	0.006494
80.0	1.874	2.320	2.610	2.610	0.010792	0.007727	0.005424	0.008095
50.0	1.720	2.004	2.188	2.188	0.011732	0.008745	0.009076	0.011280
40.0	1.664	1.889	2.035	2.035	0.019747	0.019149	0.017382	0.017833
30.0	1.612	1.781	1.891	1.891	0.023236	0.025744	0.024016	0.028308

Zonal Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.488	3.192	3.649	3.649	0.014627	0.004899	0.003760	0.012864
70.0	2.289	2.782	3.102	3.102	0.013821	0.007222	0.006002	0.008392
50.0	2.157	2.510	2.739	2.739	0.006295	0.007022	0.008689	0.004687
40.0	2.057	2.372	2.555	2.555	0.009710	0.008398	0.008324	0.008908
30.0	2.023	2.234	2.371	2.371	0.011289	0.012547	0.016212	0.012900
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL								
100.0	2.387	3.063	3.500	3.500	0.006235	0.008291	0.004465	0.005603
70.0	2.193	2.664	2.969	2.969	0.008595	0.010266	0.010038	0.007496
50.0	2.068	2.406	2.625	2.625	0.017184	0.017990	0.018942	0.013355
40.0	2.006	2.278	2.454	2.454	0.018744	0.021383	0.021170	0.019965
30.0	1.943	2.149	2.282	2.282	0.031466	0.030479	0.025632	0.028261

TABLES A1.6.4.5

Threshold Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.567	1.965	2.225	2.225	0.003029	0.003029	0.003029	0.003029
80.0	1.491	1.811	2.018	2.018	0.003074	0.006266	0.003101	0.003542
70.0	1.454	1.734	1.915	1.915	0.003391	0.007648	0.007000	0.004002
50.0	1.374	1.579	1.709	1.709	0.008207	0.008813	0.005579	0.005234
30.0	1.303	1.423	1.501	1.501	0.008162	0.006617	0.006615	0.006359
10.0	1.228	1.268	1.294	1.294	0.011357	0.010620	0.009971	0.008334
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL								
100.0	1.909	2.273	2.513	2.513	0.004417	0.004417	0.004417	0.004417
90.0	1.875	2.203	2.415	2.415	0.004417	0.004422	0.004417	0.004420
80.0	1.841	2.132	2.324	2.324	0.004455	0.004459	0.004477	0.004638
70.0	1.810	2.065	2.231	2.231	0.004458	0.004844	0.005120	0.004693
50.0	1.742	1.926	2.045	2.045	0.006584	0.005120	0.005293	0.005589
30.0	1.673	1.782	1.853	1.853	0.006586	0.005794	0.005840	0.006331

Threshold Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.809	2.278	2.581	2.581	0.002480	0.002480	0.002480	0.002480
90.0	1.765	2.190	2.460	2.460	0.002938	0.002812	0.002739	0.002682
80.0	1.721	2.096	2.339	2.339	0.003603	0.002987	0.002712	0.002754
70.0	1.677	2.005	2.061	2.061	0.005005	0.005698	0.005293	0.005293
50.0	1.590	1.824	1.976	1.976	0.004671	0.004238	0.006538	0.005538
30.0	1.501	1.642	1.733	1.733	0.007399	0.007167	0.006549	0.005035
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL								
100.0	2.236	2.693	2.988	2.988	0.003359	0.003359	0.003359	0.003359
90.0	2.194	2.605	2.871	2.871	0.003520	0.003381	0.003359	0.003370
80.0	2.150	2.515	2.752	2.752	0.003893	0.003590	0.003370	0.003381
70.0	2.108	2.428	2.635	2.635	0.006417	0.003837	0.003851	0.003899
50.0	2.023	2.252	2.401	2.401	0.004161	0.004195	0.006154	0.005146

TABLES A1.6.4.6

Threshold Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.153	2.718	3.084	3.084	0.001920	0.001920	0.001920	0.001920
90.0	2.099	2.608	2.938	2.938	0.002023	0.001956	0.002030	0.002068
80.0	2.046	2.500	2.792	2.792	0.002790	0.002077	0.002184	0.002279
70.0	1.993	2.389	2.646	2.646	0.002565	0.002259	0.003441	0.002865
50.0	1.887	2.170	2.353	2.353	0.003893	0.003382	0.004396	0.004679
30.0	1.780	1.950	2.060	2.060	0.007151	0.005520	0.004849	0.004787
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL								
100.0	2.546	3.090	3.443	3.443	0.002683	0.002683	0.002683	0.002683
90.0	2.495	2.985	3.303	3.303	0.002733	0.002863	0.002722	0.002721
70.0	2.393	2.774	3.021	3.021	0.003275	0.002784	0.002732	0.002842
60.0	2.341	2.668	2.803	2.803	0.003773	0.003721	0.003898	0.003150
50.0	2.291	2.565	2.742	2.742	0.005075	0.003938	0.005434	0.005434
30.0	2.188	2.352	2.458	2.458	0.004288	0.004251	0.004431	0.004141

Threshold Scheme

CHANNEL ERROR RATE = 0.001

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.524	3.194	3.629	3.629	0.001503	0.001503	0.001503	0.001628
90.0	2.461	3.065	3.456	3.456	0.002146	0.002529	0.001557	0.001557
70.0	2.335	2.804	3.108	3.108	0.002978	0.001725	0.002257	0.002257
60.0	2.271	2.674	2.935	2.935	0.004160	0.003364	0.002086	0.002402
50.0	2.208	2.544	2.762	2.762	0.002777	0.004897	0.003522	0.002922
30.0	2.082	2.283	2.414	2.414	0.004541	0.004352	0.007962	0.005880
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL								
100.0	3.123	3.831	4.290	4.290	0.001805	0.001805	0.001805	0.001805
90.0	2.875	3.512	3.925	3.925	0.002262	0.001812	0.002431	0.002431
70.0	2.924	3.420	3.741	3.741	0.002374	0.002326	0.002179	0.002242
60.0	2.857	3.281	3.556	3.556	0.003109	0.003121	0.002483	0.002440
50.0	2.792	3.147	3.378	3.378	0.003246	0.003634	0.003129	0.003200
30.0	2.657	2.870	3.008	3.008	0.004226	0.004770	0.003986	0.003906

TABLES A1.6.4.7

Threshold Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.153	2.718	3.084	3.084	0.008812	0.005900	0.006037	0.005409
90.0	2.099	2.608	2.938	2.938	0.011537	0.007906	0.005303	0.006761
80.0	2.046	2.500	2.792	2.792	0.013219	0.006574	0.007693	0.010205
70.0	1.993	2.389	2.646	2.646	0.013011	0.011750	0.011770	0.012641
50.0	1.887	2.170	2.353	2.353	0.021155	0.016921	0.019418	0.017686
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.50 BITS/PIXEL								
100.0	2.546	3.090	3.443	3.443	0.005723	0.004639	0.003810	0.005136
90.0	2.495	2.985	3.303	3.303	0.005409	0.004846	0.005380	0.004709
70.0	2.393	2.774	3.021	3.021	0.008630	0.008372	0.006787	0.008022
60.0	2.341	2.668	2.803	2.803	0.012226	0.008615	0.012631	0.010241
50.0	2.291	2.565	2.742	2.742	0.014070	0.012191	0.013345	0.011279

Threshold Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	2.524	3.194	3.629	3.629	0.005914	0.004779	0.004268	0.006493
90.0	2.461	3.065	3.456	3.456	0.008946	0.007020	0.006858	0.006017
70.0	2.335	2.804	3.108	3.108	0.010975	0.009770	0.012879	0.014176
60.0	2.271	2.674	2.935	2.935	0.019742	0.014525	0.015437	0.019868
50.0	2.208	2.544	2.762	2.762	0.019210	0.022229	0.021373	0.020096
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.85 BITS/PIXEL								
100.0	3.123	3.831	4.290	4.290	0.004611	0.004371	0.003598	0.004178
90.0	2.875	3.512	3.925	3.925	0.006227	0.005829	0.004646	0.004423
70.0	2.924	3.420	3.741	3.741	0.011487	0.007784	0.006240	0.009081
60.0	2.857	3.281	3.556	3.556	0.012089	0.008929	0.009846	0.012515
50.0	2.792	3.147	3.378	3.378	0.015918	0.012991	0.014647	0.014338

TABLES A1.6.4.8

Threshold Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.567	1.965	2.225	2.225	0.005067	0.005652	0.006596	0.004052
80.0	1.491	1.811	2.018	2.018	0.005662	0.022206	0.018288	0.026422
70.0	1.454	1.734	1.915	1.915	0.021974	0.023119	0.017048	0.021383
50.0	1.374	1.579	1.709	1.709	0.029902	0.031602	0.028453	0.041372
30.0	1.303	1.423	1.501	1.501	0.006359	0.037116	0.058268	0.045097
10.0	1.228	1.268	1.294	1.294	0.056800	0.047893	0.048769	0.049758
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.00 BITS/PIXEL								
100.0	1.909	2.273	2.513	2.513	0.005529	0.005503	0.005291	0.004638
90.0	1.875	2.203	2.415	2.415	0.009586	0.006497	0.007090	0.004658
80.0	1.841	2.132	2.324	2.324	0.013426	0.008412	0.008331	0.009025
70.0	1.810	2.065	2.231	2.231	0.009663	0.011296	0.010831	0.007632
50.0	1.742	1.926	2.045	2.045	0.017265	0.013077	0.012054	0.014480
30.0	1.673	1.782	1.853	1.853	0.020886	0.018455	0.020061	0.021707

Threshold Scheme

CHANNEL ERROR RATE = 0.01

SUB-IMAGE SIZE = 16 * 16 PIXELS SQUARE

INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL

% Sub-image Protection	Bit-rate with Protection				Average Sample Mean Squared Error			
	Hamming		Convolution		Hamming		Convolution	
	(15,11)	(7, 4)	CC(7)	CC(11)	(15,11)	(7, 4)	CC(7)	CC(11)
100.0	1.809	2.278	2.581	2.581	0.012155	0.008162	0.005812	0.004669
90.0	1.765	2.190	2.460	2.460	0.009022	0.006417	0.009433	0.008980
80.0	1.721	2.096	2.339	2.339	0.015042	0.013462	0.009472	0.010975
70.0	1.677	2.005	2.061	2.061	0.016354	0.020139	0.014176	0.010722
50.0	1.590	1.824	1.976	1.976	0.019980	0.021361	0.025494	0.029096
SUB-IMAGE SIZE = 8 * 8 PIXELS SQUARE INITIAL BIT-RATE WITHOUT PROTECTION = 1.25 BITS/PIXEL								
100.0	2.236	2.693	2.988	2.988	0.007374	0.004521	0.004203	0.005522
90.0	2.194	2.605	2.871	2.871	0.007263	0.004948	0.004229	0.004713
80.0	2.150	2.515	2.752	2.752	0.008301	0.007805	0.005733	0.007313
70.0	2.108	2.428	2.635	2.635	0.008184	0.009238	0.010757	0.009760
50.0	2.023	2.252	2.401	2.401	0.013523	0.012682	0.014151	0.012657

REFERENCES

[1] Abate J.E. " Linear and Adaptive Delta Modulation " - Proc. of IEEE Vol.55, No.3 pp 298-308, March 1967.

[2] Ahmed N., Natarjen N.T., Rao K.R. " On Image Processing and a Discrete Cosine Transform " - IEEE Trans. Comput. Vol. C-23 pp 90-93, Jan. 1974.

[3] Ahmed N., Rao K.R. " Orthogonal Transforms " 1975.

[4] Anderson G.B., Huang T.S. " Picture Bandwidth Compression by Piecewise Fourier Transformation " - IEEE Trans. Comm. Techn. Vol.COM-19 pp 133-140, 1971.

[5] Arambepola B. " A New Architecture for a Fast DFT Processor " IEE Colloquium on Transform Techniques in IMage Processing Digest No.1983/50 pp 1/1-1/5 18 May 1983

[6] Arguello R.J., Sellner H.R., Stuller J.A. " The Effect of Channel Errors in the Differential Pulse-code Modulation Transmission of Sampled Imagery " - IEEE Trans. Comm. Techn. COM-19 pp 926-933, 1971.

[7] Bennett W.R., Davey J.R. " Data Transmission " - Bell
c
Telephone Laboratories, Pub. M Graw and Hill (c) 1965.

[8] Berger T. " Rate Distortion Theory ¦ A Mathematical Approach to Data Compression ". - Prentice Hall 1971

[9] Bose R.C., Ray-Chaudhuri D.K. " Further Results on Error-correcting Binary Group Codes " - Inf. Contr. Vol.3 pp 279-290, 1960.

[10] Bosworth R.S., Candy J.C. " A Companded One-bit Coder for PICTUREPHONE Transmission " - Bell Syst. Tech.J. Vol.48 pp 1459-1479, May 1969.

[11] Bowen E.G., Limb J.O. " Subjective Effect of Substituting Lines in a Video Telephone Signal " - IEEE Trans. on Comm. Vol.COM-24 pp 1208-1211, Oct. 1976.

[12] Budrikas Z.L. " Visual Fidelity Criterion and Modelling " - Proc. of the IEEE Vol.60 pp 771-779, July 1972.

[13] Burge R.E. " A Comparison Between DPCM and Transform Techniques in Image Compression " - IEC Colloquium on Transform Techniques in Image Processing Digest No. 1983/50 pp.2/1-2/4

[14] Campbell F.W., Kulikowski J.J. " Orientational Selectivity of the Visual System " - J.Physiol, Vol.187 pp 437-445, 1966

[15] Casasent D. " Pattern Recognition ¦ A Review " - IEEE Spectrum pp 28-31, March 1981.

[16] Chen W.H. " Scene Adaptive Coder " - Proc. ICC-81, 1-22.5.6 pp 22.5.

[17] Chen W.H., Pratt W.K. " Color Image Coding with the Slant Transform " - Proc. 1973 Symp. Appl. Walsh Functions pp 155-161, April 1973.

[18] Chen W.H., Smith C.H. " Adaptive Coding of Monochrome and Color Images " - IEEE Trans. Comm. Vol.COM-25, No.11 pp 1285-1292, Nov.1977.

[19] Chen W., Smith C.H., Fralick S. " A Fast Computational Algorithm for the Discrete Cosine Transform " - IEEE Trans. Comm Vol.COM-25 pp 1004-1009, 1977.

[20] Clarke R.J. " An Introduction to Transform Techniques for Image Processing " - IEE Colloquium on Transform Techniques in Image Processing, Digest No. 1983/50 pp 1/1 - 1/5, 18th May 1983.

[21] Cooley J.W., Tukey J.W. " An Algorithm for the Machine Computation of Complex Fourier Series " - Mathe. Computat. Vol. 19 pp 297-301, April 1965.

[22] Cornsweet T.N. " Visual Perception " - Academic Press 1970

[23] Correspondence " Signal-to-Quantizing-Noise Ratios for Differential PCM " - IEEE Trans. on Comm. Tech. Vol.COM.19 pp568-570, Aug. 1971.

[24] Cochran W.T., Cooley J.W., et al. " What is the Fast Fourier Transform ? " - Proc. of IEEE Vol.55 No.10 pp 1664-1674 Oct.1967.

[25] Culter C.C. " Differential Quantization of Communication Systems " U.S. Patent 2 605 361 July 29, 1952.

[26] Delp E.J., Mitchell O.R. " Image Compression Using Block Truncation Coding " - IEEE Trans. Comm. COM-27 pp 1335-1342, 1979.

[27] Elias P. " Coding for Noisy Channels " - IRE Conv. Rec., pt.4. pp 37-44, 1955.

[28] Elias P. " Error-free Coding " - IRE Trans. Inform. Theory IT-4 pp 29-37, 1954.

[29] Essman J.E., Wintz P.A. " The Effects of Channel Errors in DPCM Systems and Comparison with PCM Systems " - IEEE Trans. on Comm., Vol. COM-21 No.8 pp 867-877, Aug. 1973.

[30] Fano R.M. " A Heuristic Discussion of Probabilistic Decoding " - IEEE Trans. on Inf. Theo. Vol. IT-9 pp 64-74, 1963.

[31] Forney Jr G.D. " Generalized Minimum Distance Decoding " - IEEE Trans. on Inf. Theo. IT-12 pp 125-131, 1966.

[32] Golay M.J.E. " Binary Coding " - IRE Trans. Inf. Theo. Vol. PGIT-4 pp 23-28, 1954.

[33] Golay M.J.E. " Notes on Digital Coding " - Proc. IRE Vol.37 pp 657 , 1949.

[34] Gonzalez R.C. and Wintz P. " Digital Image Processing "

[35] Goyal S.K., O'Neal Jr. J.B. " Entropy Coded Differential Pulse Code Modulation for Television " - IEEE Trans. Comm. Vol. Com-23 pp 660-666, June 1975.

[36] Guiasu S. " Information Theory with Applications " - M Graw-Hill 1977.

[37] Habibi A. " Comparison of n^{th} Order DPCM Encoder with Linear Transformation and Block Quantization Techniques " - IEEE Trans. Comm. Techn. COM-19 pp 948-956, 1971.

[38] Habibi A. " Hybrid Coding of Pictorial Data " - IEEE Trans. Comm. COM-22 pp 614-621, 1974.

[39] Habibi A., Wintz P.A. " Image Coding by Linear Transformation and Block Quantization " - IEEE Trans. on Comm. Techn. COM-19 No.1 pp50-62, Feb. 1971.

[40] Hagelbarger D.W. " Recurrent Codes: Easily Mechanized, Burst Correcting Binary Codes " - Bell Syst. Techn. J. Vol.38 pp 969-984, 1959.

[41] Hale J.A.G., Saraga P. " Digital Image Processing " - Opto-electron (G.B) Vol.6 No.5 pp 333-348, Sept.1974.

[42] Hall E.L. " Computer Image Processing and Pattern Recognition " - Academic Press 1979.

[43] Hall C.F., Hall E.L. " A Nonlinear Model for the Spatial Characteristics of the Human Visual System " - IEEE Trans. on Systems, Man, and Cybernetics Vol.SMC-7 pp 161-17Ø, 1977.

[44] Hamming R.W. " Coding and Information Theory " - Prentice-Hall 198Ø

[45] Hamming R.W. " Error Detecting and Error Correcting Codes " - Bell Syst. Techn. J. Vol.29 pp 147-16Ø, 195Ø.

[46] Hotelling H. " Analysis of a Complex of Statistical Variables into Principal Components " - J. Educ. Psychology, Vol.24 pp 417-441 and 498-52Ø, 1933.

[47] Huang C.J., Rao K.R. " Channel Error Propagation in Adaptive DPCM Coders " - IEEE International Conf. on Communications ICC 82 Vol.2 pp 3G.2-1-5, June 1982.

[48] Huang J.J.Y, Schultheiss P.M. " Block Quantization of Correlated Gaussian Random Variables " - IEEE Trans. on Commun. Syt. pp 289-296, Sept.1963

[49] Hubel D.H., Wiesel T.S. " Brain Mechanisms of Vision " - Scientific American Vol. 241 No.3 pp 13Ø-146, Sept.1979.

[5Ø] Huffman D.A. " Method for the Construction of Minimum Redundancy Codes " - Proc. IRE Vol.4Ø pp 1Ø98-11Ø1, 1952.

[51] Jain A.K. " Advances in Mathematical Models for Image Processing " - Proc. of IEEE Vol.69 No.5 pp 5Ø2-528, May 1981.

[52] Jain A.K. " A Fast Karhunen-Loeve Transform for a Class of Random Processes " - IEEE Trans. Commun. COM-24 pp 1Ø23-1Ø29, 1976.

[53] Jain A.K. " Image Data Compression¦ A Review " - Proc. of IEEE Vol.69 No.3 pp349-391, March 1981.

[54] Jain A.K. " A Sinusiodal Family of Unitary Trqansforms " - IEEE Trans. Pattern Anal. Machine Intell. Vol.PAMI-1 pp 356-365, Oct.1979.

[55] Jayant N.S. " Adaptive Delta Modulation with one Bit Memory " - Bell Syst. Tech. J. Vol.49 pp 321-342, March 197Ø

[56] Kelly D.H. " Effects of Sharp Edges on the Visibility of

Sinusoidal Gratings " - J. Opt. Soc. Amer. Vol.6Ø pp 98-1Ø3, June 197Ø.

[57] Kelly D.H. " Spatial Frequency Selectivity in the Retina " - Vision Res. Vol.15 pp 565-572, 1975.

[58] Kincaid W.M., Kristofferson A.H. " Neural Formulation of the Effect of Target Size and Shape upon Visual Detection " - J. Opt. Soc. Amer. Vol.5Ø pp 143-148, 196Ø.

[59] Kitajima H. " A Symmetric Cosine Transform " - IEEE Trans. Comput. Vol.C-29 pp 317-323, 198Ø.

[6Ø] Kronander T. " Channel Errors and Their Influence on Transform Coded Images " - Proc. of the Third Scandinavian Conference on Image Analysis, Copenhagen, Denmark, July 12-14 Arranged by the Danish Pattern Recognition Society, 1983.

[61] Kurtenbach A.J., Wintz P.A. " Quantizing for Noisy Channels " - IEEE Trans. Commun. Tech. Vol. COM-17 pp 291-3Ø2, 1969.

[62] Legge G.E. " Space Domain Properties of a Spatial Frequency Channel in Human Vision " - Vision Res. Vol.18 pp 959-969, 1978.

[63] Lei T.R., Scheinberg N., Schilling D.L. " Adaptive Delta Modulation Systems for Video Encoding " - IEEE Trans. Commun. Vol.COM-25 pp 13Ø2-1314, Nov.1977.

[64] Lin S. " An Introduction to Error-Correcting Codes " - Prentice-Hall Englewood Cliffs N.J. 197Ø.

[65] Lippman R. " Influence of Channel Errors on DPCM Picture Coding " - Acta Electron Vol.19 No.4 pp 289-294, 1976.

[66] Lucky, Salz, Weldon " Principles of Data Communication " - Bell Telephone Laboratories Inc. Pub.- M Graw and Hill (c) 1968.

[67] MacWilliams F.J. " Error-Correcting Codes - a Historical Survey. In H.B. Mann(ed), Error-Correcting Codes pp 3-14 Wiley, New York, 1968.

[68] Mannos J.L., Sakrison D.J. " The Effects of a Visual Fidelity Criterion on the Encoding of Images " - IEEE Trans. on Info. Theo. Vol.IT-2Ø No.4 pp 525-536, July 1974.

[69] Massey J.L. " Variable-Length Codes and the Fano Metric " -

IEEE Trans. Info. Theo. Vol.IT-18 pp 196-198, 1972.

[70] Max J. " Quantizing for Minimum Distortion " - IEEE Trans. on Info. Theo. Vol.IT-6 No.1 pp 7-12, March 1960.

[71] Maxemchuk N.F., Stuller J.A. " Reduction of Transmission Error in Adaptively Predicted DPCM Encoded Picture " - Bell Syst. Tech. J. Vol.58 pp 1413-1425, July-Aug. 1979.

[72] M Eliece R.J. " The Theory of Information and Coding " - (Encyclopedia of Mathematics and it's Applications) Addison-Wesley 1977

[73] Mitchell O.R., Tabatabai A.J. " Channel Error Recovery for Transform Image Coding " - IEEE Trans. Comm. Vol.COM-29 No.12 pp 1754-1762, Dec.1981.

[74] Modestino J.W., Daut D.G. " Block Transform Image Coding in the Presence of Channel Errors " - SPIE Vol.249 Advances in Image Transmission 11 pp 112-122, 1980.

[75] Modestino J.W., Daut D.G. " Combined Source Channel Coding of Images " - IEEE Trans. Comm. Vol.COM-27 pp 1644-1659, Nov.1979.

[76] Modestino J.W., Daut D.G., Vickers A.L. " Combined Source-Channel Coding of Images Using the Block Cosine Transform " - IEEE Trans. on Commun. Vol.COM-29 No.9 pp 1261-1273, Sept. 1981.

[77] Mostafavi H., Sakrison D.J. " Structure and Properties of a Single Channel in the Human Visual System " - Vision Res. Vol.16 pp 957-968, 1976.

[78] Netravali A.N. " Quantizers for DPCM coding of Picture Signals " - IEEE Trans. Info. Theo. Vol.IT-23 pp 360-370, 1977.

[79] Ngan K.K. " Adaptive Tranform Coding of Video Signals " - IEE Proc. Vol.129 Pt F No.1 pp 28-40, Feb.1982.

[80] Ngan K.N., Steele R. " Enhancement of PCM and DPCM Images Corrupted by Transmission Errors " - IEEE Trans. Comm. Vol.30 No.1 Part 2 pp 257-265, Jan.1982.

[81] O'Neal J.B. " Differential Pulse-Code Modulation (PCM) with Entropy Coding " - IEEE Trans. on Info. Theo. Vol.IT-22 No.2 pp 169-174, March 1976.

[82] O'Neal J.B. Jr, Natarajan T.R. " Coding Isotropic Images " - IEEE Trans. Info. Theo. Vol.IT-23 pp 697-707, Nov.1977.

[83] O'Neal J.B. Jr " Delta Modulation Quantizing Noise - Analytic and Computer Simulation Results for Gaussian and Television Input Signals " - Bell Syst. Tech. J. Vol.45 pp 117-141, Jan.1966.

[84] Panter P.F., Dite W. " Quantization Distortion in Pulse-Count Modulation with Nonuniform Spacing of Levels " - Proc. IRE pp 44-48, Jan.1951.

[85] Patel A.S. " Spatial Resolution by Human Visual System - The Effect of Mean Retinal Illuminance " - J. Opt. Soc. Amer. Vol.506 pp 689-694, May 1966.

[86] Pearl J., Andrews H.C., Pratt W.K. " Performance Measures for Transform Data Coding " - IEEE Trans. Comm. Tech. Vol. COM-20 pp 411-445, June 1972.

[87] Pearlman W.A., Gray R.M. " Source Coding of the Discrete Fourier Transform " - IEEE Trans. on Info. Theo. Vol.IT-24 No.6, Nov.1978.

[88] Pearson D.E., Whybray M.W. " Transform Coding with Interleaving Blocks " - IEE Colloquium on Transform Techniques in Image Processing Digest No.1983/50 pp 1/1-1/5 18 May 1983

[89] Pratt W.K. " Digital Image Processing " - Wiley 1978

[90] Pratt W.K., Kane J., Andrews H.C. " Hadamard Transform Image Coding " - Proc. IEEE Vol.57 pp 58-68, 1969.

[91] Pratt W.K., Welch L.R., Chen W. " Slant Transform for Image Coding " - IEEE Trans. Comm. Vol.COM-22 pp 1075-1093, 1974.

[92] Richters J.S. " Application of Pareto Error Statistics to Hagelbarger Codes " - IEEE Trans. on Info. Theo. Vol.IT-11 No.4 pp 571-576, Oct.1965.

[93] Rosenfeld A., Kak A.C. " Digital Picture Processing " - Volume 1 Pub.- Academic Press 1982.

[94] Schindler H.R. " Delta Modulation " - IEEE Spectrum Vol.7 pp 69-78, Oct.1970.

[95] Segall A. " Bit Allocation and Encoding for Vector Sources " - IEEE Trans. on Info. Theo. Vol.IT-22 pp 162-169, March 1976.

[96] Shannon C.E. " Coding Theorems for a Discrete Source with a Fidelity Criterion " - IRE Nat. Conv. Record Pt.4 pp 142-163, 1959.

[97] Shannon C.E. " Mathematical Theory of Communication " - Bell Syst. Tech. J. Vol.27 pp 379-423 and pp 623-656, 1948.

[98] Song C.L., Garodnick J., Schilling D.L. " A Variable Step-Size Robust Delta Modulator " - IEEE Trans. Comm. Vol.COM-19 pp 1033-1044, Dec.1971.

[99] Spann M., Wilson R., Dodgson T.E " Coding Textures " - 2nd International Technical Symposium on Optical and Electro-Optical Applied Science and Engineering 2-6th December 1985

[100] Stockton T.G.Jr " Image Processing in the Context of a Visual Model " - Proc. IEEE Vol.60 pp 828-842, July 1972.

[101] Tasto M., Wintz P.A. " Image Coding by Adaptive Block Quantization " - IEEE Trans. Comm. Techn. Vol.COM-19 pp 50-60, 1971.

[102] Viterbi A.J. " Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm " - IEEE Info. Techn. Vol.IT-13 pp 260-269, 1967.

[103] Viterbi A.J., Omura J.K " Principles of Digital Communication " - Pub. M Graw and Hill, 1979.

[104] Westheimer G., Campbell F.W. " Light Distribution in the Image Formed by the Living Human Eye " - J. Opt. Soc. Amer. Vol.52 pp 1040-1045, 1962.

[105] Wilson R., Knutsson H.E., Granlund G.H. " Anisotropic Non-stationary Image Estimation and it's Applications Part I and II - Predictive Image Coding " - IEEE Trans. on Comm. Vol.COM-31 No.3 pp 388-406, March 1983.

[106] Wintz P.A. " Transform Picture Coding " - Proc IEEE Vol.60 pp 809-820, 1972.

[107] Yip P., Rao K.R. " A Fast Computational Algorithm for the Discrete Sine Transform " - IEEE Trans. Comm. Vol.COM-28 pp 304-307, 1980.